November 1981
US $2.95

80 microcomputing™

-the magazine for TRS-80* users

A WAYNE GREEN PUBLICATION

Get Down to BUSINESS

The Micro Connection

80 Microcomputing
11/81
New Instant Software Catalog
#23

DOSPLUS

FEATURES:

1) Radio Shack compatibility
2) Error free variable length records
3) Full lower case detection and support
4) Repeating keyboard with NO keybounce EVER
5) Shift [0] typewriter keyboard option
6) Execute only protection feature for BASIC programs
7) Automatic track support for 35 through 80 track drives (mixed)
8) Device I/O handling with FORCE command
9) Supports high speed clock modification (up to 4.0mhz)
10) Supports mixed mode (single & double density) automatically
11) Allows disable-enable of break key
12) Allows user to define step rate per drive and re-configure system disk
13) Allows for efficient use of double-headed drives
14) Built in screen printer (shift [CLEAR]) with [BREAK] key abort
15) Multiple command chaining with "DO"
16) Built in memory test with CLEAR command
17) New printer driver which allows complete forms control and paging
18) Automatic serial printer driver with optional auto linefeed
19) Execute any DOS command from BASIC and return to BASIC
20) Free space map of diskette with optional output to printer
21) Copy with variable length files
22) Complete RS232 control from keyboard with status check
23) Create and pre-allocate files from DOS
24) Display current date and time from DOS
25) More information from Directory with optional printer output
26) Enter DEBUG with shift [BREAK] to allow use of [BREAK] from BASIC
27) New DISKDUMP/CMD sector display/modify program (works with filespecs)
28) New DISKZAP/CMD single/double density disk editor
29) New BACKUP (more reliable, no more pack ID check)
30) New FORMAT (more reliable, no need to bulk erase disk first)
31) New MAP utility (maps out disk, showing where files are located)

New DOSPLUS Z80 Extended Disk BASIC

1) Faster loads and saves
2) BASIC Reference utility (lines, variables, keywords, printer option)
3) BASIC Renumber utility (renumber section of text, block text move)
4) Shorthand features for almost ANY direct command (LOAD, SAVE, etc.)
5) Shorthand features for editing (listing and editing with single key)
6) CMD"M" instantly displays currently set variables
7) Global search and replace in BASIC text
8) Line printer TAB to 255
9) OPEN"E" to end of sequential file (for output)
10) DI (delete and insert text line)
11) DU (duplicate text line)
12) ",R" & ",V" options after LOAD and RUN (files open & save variables)
13) OPEN"D" allowed (Model II compatible) equal to OPEN"R"
14) DOS commands from BASIC
15) Automatic, error-free variable length records
16) Single step execution with TRON (fabulous for debugging)
17) CRUNCH (BASIC program compressor)
18) New TBASIC (tiny BASIC) offers full BASIC commands
19) TBASIC and DOSPLUS together only use 8K of RAM (40K left in 48K TRS-80)

***** 7 MORE UTILITIES *****

1) Single drive copy
2) Restore (dead files)
3) Purge (unwanted files)
4) Clearfile (destroys data by writing zeros to file)
5) Transfer (moves all user files from one disk to another)
6) Spooler (allows printing of text while freeing up the CPU)
7) Crunch (Basic program compressor)

***** ALSO *****

- New I/O package 30% faster
- No BREAK key death from DOS
- No closing killed files and ruining diskettes

DOSPLUS gives you more of what you buy an operating system for. Speed and reliability without sacrificing simplicity and power. If you need extra power without extra wait, then you need DOSPLUS!

Single or double density systems available for Model I. Model III DOSPLUS ready for immediate delivery.

Perhaps the best investment you can make for your TRS-80! Listen to what others have had to say about DOSPLUS.

"Overall, DOSPLUS is the fastest operating system I have seen..."

Pete Carr in 80-US Journal.

"DOSPLUS...the better mousetrap."

Stewart Fason in 80-Microcomputing

"On a scale of 1 to 10, I give DOSPLUS a solid 9."

Reese Fowler in 80-Microcomputing
(Model III DOSPLUS review)

For the BASIC programmer, our features are unmatched. For the average businessman, our speed and simplicity cannot be beat.

So, join the satisfied users who have joined DOSPLUS. Experience excellence! Experience DOSPLUS!

DOSPLUS comes complete with full utilities, PLUS a FREE patch to enable Model I Scripsit/Super Script to run on Model III, UNLIMITED Backups!

Model I DOSPLUS — $99.95
Model III DOSPLUS — $99.95
Model I double density upgrade — $175.00
Master Directory 1.2 (double density) — $29.95

STEP ON THOSE DOS BUGS!! ORDER TODAY!!

VISA

master charge

MICRO SYSTEMS SOFTWARE INC ✓384

Specializing in the Tandy Line

(305) 983-3390

5846 Funston Street
Hollywood, FL 33023

CALL TOLL FREE FOR FAST SERVICE
(800) 824-7888, OPERATOR 193
FOR VISA/MASTERCHARGE/C.O.D. ORDERS
California dial (800) 852-7777, Operator 193
Alaska and Hawaii dial (800) 824-7919, Operator 193
TOLL FREE LINES WILL ACCEPT ORDERS ONLY!
For Applications and Technical information, call (305) 983-3390 or drop us a card.

Dealers inquiries invited

TRS-80* COMPUTING EDITION

©1981 Percom Data Co., Inc. **The Percom Peripheral** 35 cents

Percom's DOUBLER II™ tolerates wide variations in media, drives

GARLAND, TEXAS — May 22, 1981 — Harold Mauch, president of Percom Data Company, announced here today that an improved version of the Company's innovative DOUBLER™ adapter, a double-density plug-in module for TRS-80* Model I computers, is now available.

Reflecting design refinements based on both theoretical analyses and field testing, the DOUBLER II™, so named, permits even greater tolerance in variations among media and drives than the previous design.

Like the original DOUBLER, the DOUBLER II plugs into the drive controller IC socket of a TRS-80 Model I Expansion Interface and permits a user to run either single- or double-density diskettes on a Model I.

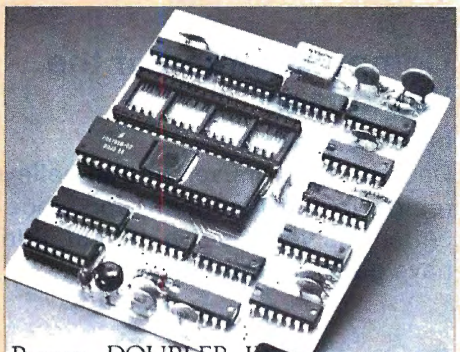
With a DOUBLER II installed, over four times more formatted data — as much as 364 Kbytes — can be stored on one side of a five-inch diskette than can be stored using a standard Tandy Model I drive system.

Moreover, a DOUBLER II equips a Model I with the hardware required to run Model III diskettes.

(*Ed. Note:* See "OS-80™: Bridging the TRS-80* software compatibility gap" elsewhere on this page.)

The critical clock-data separation circuitry of the DOUBLER II is a proprietary design called a ROM-programmed digital phase-lock loop data separator.

According to Mauch, this design is more tolerant of differences from diskette to diskette and drive to drive, and also provides immunity to performance degradation caused by circuit component aging.



Percom DOUBLER II

Mauch said "A DOUBLER II will operate just as reliably two years after it is installed as it will two days after installation."

The digital phase-lock loop also eliminates the need for trimmer adjustments typical of analog phase-lock loop circuits.

"You plug in a Percom DOUBLER II and then forget it," he said.

The DOUBLER II also features a refined Write Precompensation circuit that more effectively minimizes the phenomena of bit- and peak-shifting, a reliability-impairing characteristic of magnetic data recording.

The DOUBLER II, which is fully software compatible with the previous DOUBLER, is supplied with DBLDOS™, a TRSDOS*-compatible disk operating system.

The DOUBLER II sells for $2~~[illegible]~~95, including the DBLDOS diskette. Now $169.95!

✓1

Owners of original DOUBLERs may purchase a DOUBLER II upgrade kit, without the disk controller IC, for $30.00. Proof of purchase of an original DOUBLER is required, and each DOUBLER owner may purchase only one DOUBLER II at the $30.00 price.

The Percom DOUBLER II is available from authorized Percom retailers, or may be ordered direct from the factory. The factory toll-free order number is 1-800-527-1222.

Ed. note: Opening the TRS-80 Expansion Interface may void the Tandy limited 90-day warranty.

✓258

All that glitters is not gold

OS-80™ Bridging the TRS-80* software compatibility gap

Compatibility between TRS-80* Model I diskettes and the new Model III is about as genuine as a gold-plated lead Krugerrand.

True, Model I TRSDOS* diskettes can be *read* on a Model III. But first they must be converted and re-recorded for Model III operation.

And you *cannot write* to a Model I TRSDOS* diskette. Not with a Model III. You cannot add a file. Delete a file. Or in any way modify a Model I TRSDOS diskette with a Model III computer.

Furthermore, your converted TRSDOS diskettes cannot be converted *back* for Model I operation.

TRSDOS is a one-way street. And there's no retreating. A point to consider before switching the company's payroll to your new Model III.

Real software compatibility should allow the *direct, immediate* interchangeability of Model I and Model III diskettes. No read-only limitations, no conversion/re-recording steps and no chance to be left high and dry with Model III diskettes that can't be run on a Model I.

What's the answer? The answer is Percom's OS-80™ family of TRS-80 disk operating systems.

OS-80 programs allow *direct, immediate* interchangeability of Model I and Model III diskettes.

You can run Model I single-density diskettes on a Model III; install Percom's plug-in DOUBLER™ adapter in your Model I, and you can run double-density Model III diskettes on a Model I.

There's no conversion, no re-recording.

Slip an OS-80 diskette out of your Model I and insert it directly in a Model III.

And vice-versa.

Just have the correct OS-80 disk operating system — OS-80, OS-80D or OS-80/III — in each computer.

Moreover, with OS-80 systems, you can add, delete, and update files. You can read *and write* diskettes regardless of the system of origin.

OS-80 is the original Percom TRS-80 DOS for BASIC programmers.

Even OS-80 utilities are written in BASIC.

OS-80 is the Percom system about which a user wrote, in Creative Computing magazine, ". . . the best $30.00 you will ever spend."†

Requiring only seven Kbytes of memory, OS-80 disk operating systems reside completely in RAM. There's no need to dedicate a drive exclusively for a system diskette.

And, unlike TRSDOS, you *can* work at the track sector level, defining and controlling data formats — in BASIC — to create simple or complex data structures that execute more quickly than TRSDOS files.

The Percom OS-80 DOS supports single-density operation of the Model I computer — price is $29.95; the OS-80D supports double-density operation of Model I computers equipped with a DOUBLER or DOUBLER II; and, OS-80/III — for the Model III of course — supports both single- and double-density operation. OS-80D and OS-80/III each sell for $49.95.

✓429

Circuit misapplication causes diskette read, format problems.

High resolution key to reliable data separation

GARLAND, TEXAS — The Percom SEPARATOR™ does very well for the Radio Shack TRS-80* Model I computer what the Tandy disk controller does poorly at best: *reliably* separates clock and data signals during disk-read operations.

Unreliable data-clock separation causes format verification failures and repeated read retries.

CRC ERROR—TRACK LOCKED OUT

The problem is most severe on high-number (high-density) inner file tracks.

As reported earlier, the clock-data separation problem was traced by Percom to misapplication of the internal separator of the 1771 drive controller IC used in the Model I.

The Percom Separator substitutes a *high-resolution* digital data separator circuit, one which operates at 16 megahertz, for the low-resolution one-megahertz circuit of the Tandy design.

Separator circuits that operate at lower frequencies — for example, two- or four-megahertz — were found by Percom to provide only marginally improved performance over the original Tandy circuit.

The Percom solution is a simple adapter that plugs into the drive controller of the Expansion Interface (EI).

Not a kit — some vendors supply an untested separator kit of resistors, ICs and other paraphernalia that may be installed by modifying the computer — the Percom SEPARATOR is a fully assembled, fully tested plug-in module.

Installation involves merely plugging the SEPARATOR into the Model I EI disk controller chip socket, and plugging the controller chip into a socket on the SEPARATOR.

The SEPARATOR, which sells for only $29.95, may be purchased from authorized Percom retailers or ordered directly from the factory. The factory toll-free order number is 1-800-527-1222.

Ed. note: Opening the TRS-80 Expansion Interface may void the Tandy limited 90-day warranty.

✓508

PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE. PRICES DO NOT INCLUDE HANDLING AND SHIPPING.

PERCOM DATA COMPANY, INC. 11220 Pagemill Road Dallas, Texas 75243 (214) 340-7081

 †Creative Computing Magazine, June, 1980, page 26.

80 Contents

PUBLISHER/EDITOR
Wayne Green

EXECUTIVE VICE PRESIDENT
Sherry Smythe

ASSISTANT TO PRESIDENT
Matt Smith

ASSOCIATE PUBLISHER
Edward Ferman

ASSISTANT PUBLISHER
Jeff DeTray

DIRECTOR OF MARKETING
(603) 924-7296
Debra Boudrieau

BULK SALES MANAGER
Ginny Boudrieau

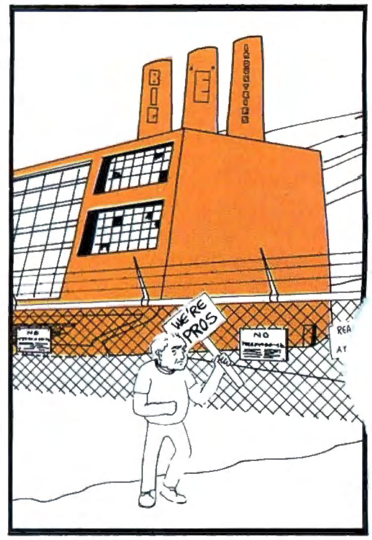
ADVERTISING SALES
(603) 924-7138
John Gancarz
Hal Stephens
Sales Coordinator: Penny Brooks

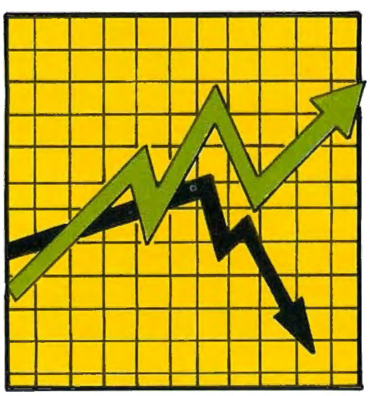
OFFICE MANAGER
Marcia Stone

DESIGN CONSULTANTS: Invisible Inc.
Elaine Cheever, Corporate Designer
Denzel Dyer, Howard Happ,
Laurie MacMillan, Joyce Pillarella,
Susan Stevens

Paid Audited Circulation

80 Microcomputing (ISSN -0199-6789) is published monthly by 1001001 Inc., 80 Pine St., Peterborough NH 03458. Phone: 603-924-3873. Second class postage paid at Peterborough, NH, and additional mailing offices. Subscription rates in U.S. are $25 for one year and $53 for three years. In Canada, $27—one year only, U.S. funds. Canadian distributor: Micro Distributing, 409 Queen St. West, Toronto, Ontario, Canada M5V 2A5. BC Canadian distributor: Graymar Data Services, Ltd., #4 258 E. 1st Ave., Vancouver, BC V5T 1A6. Foreign subscriptions (surface mail), $35—one year only, U.S. funds. Foreign subscriptions (air mail), please inquire. In Europe contact Monika Nedela, Markstr. 3, D-7778 Markdorf, W. Germany. In South Africa contact *80 Microcomputing*, P.O. Box 782815, Sandton, South Africa 2146. All U.S. subscription correspondence should be addressed to *80 Microcomputing*, Subscription Department, P.O. Box 981, Farmingdale, NY 11737. Please include your address label with any correspondence. Postmaster: Send form -3579 to *80 Microcomputing*, Subscription Services, P.O. Box 981, Farmingdale, NY 11737.





Cover by Alex Stevens; from an Instant Software program

80 REMARKS

by Wayne Green

". . . nothing like this is simple. Involved was. . . another unit which had been zapped by lightning. . ."

Bum Rap for Lynx

The modem review in the September issue ("Spanning the Electronic Nation," page 134) came down hard on the Lynx people, who responded with cries of anguish and foul play. After looking into their product I'm inclined to agree that they got a shafting this time.

Of course, nothing like this is simple. Involved was an old prototype system which had been kicking around our offices for months, plus another unit which had been totally zapped by a lightning strike nearby while it was in use. Add to that some severe phone company problems in merely getting the signals to the modem in useful shape.

On the positive side for Lynx, I don't recall any complaints from readers about the unit, and since I encourage this type of communication, I generally get quick feedback when there are product problems. And it has not escaped my notice that the Lynx unit has been used in most of the service net demonstrations at shows.

The Lynx people are moving rapidly in their development of the modem technology, working towards 9,600 baud communications. I visited their plant and watched some new products run through their paces. We may soon have an article on their new model, which is due soon and carries automation of the modem business much further.

I was not dismayed to discover two of their top people are hams. Most microcomputer firms seem to have hams involved, so I wasn't really surprised.

Their current model uses state-of-the-art design and construction and worked like a charm despite anything I could do to screw it up.■

The Office Revolution

Will the day come when we really don't need secretaries any more? I'm convinced it will and I think I see the way it will come about.

The first step is the normal executive desire to have the latest in gadgets. This will bring the computer to the elbow of the executive (as soon as some furniture firm wakes up to the need for an executive desk placing the computer where it is handy to use).

The computer will begin by providing instant interoffice communications, access to data for decision-making, E-Z Calc worksheet planning and so forth. The word processor will encourage sending notes in answer to business letters, getting us into a much less formal correspondence style. Once we can live with a business letter which does not have to reassure the recipient of his name and address, much of the secretarial role will have faded away.

"Will the day come when we really don't need secretaries anymore?"

Our computers will be able to remember addresses, file correspondence copies, and all those lovely things the secretary used to do.

Considering the problem of finding good secretaries, the path of least resistance for executives will be to depend more and more on that computer by their side.

Speaking of which, it's been some time since I've written about the need for smaller computers or terminals which are connected to the office computer system via a radio link instead of the usual umbilical cord. We have radio telephones now, so separating the computer from the system by a radio link is not a big step.

We're remote-controlling our television sets and lights, so why not our computers? My concept of the coming office computer is one that will be much like a hard bound book, with an LCD display inside the cover which can be read when you lift the lid. The keyboard will be like those on the hand-held computer systems, though in typewriter keyboard format.

With a portable terminal like this you would be able to use it as a computer by itself, or, if you were around a host system, you could dump letters for printing, access data, and communicate with others via it and the phone lines.

Recent calculator-sized television sets have an LCD screen which seems easily adaptable to our needs. Other than making things a bit smaller, there isn't a lot more to invent before we have this new type of micro-micro computer.■

TRS Typesetting?

Well, yes, to some degree it is coming to that.

More and more of the larger typesetting systems are able to accept material over phone lines. Instead of tying up machines costing $15,000 or so while setting manuscripts in type, it is possible to use the TRS with a good word processor program and dump the copy directly into the typesetting system.

Once you have a communications interface for a typesetting system, material can be accepted over the phone lines from authors, or run from disks or cassettes into a TRS and sent to the typesetter. I think we will be seeing more microcomputers used to write material for publication and to edit it.

We're nearly ready to handle columns and even articles submitted on disk or cassette. We'll edit them on a TRS and and then send the material to our typesetting system. There an operator will add the typesetting instructions. Since this takes a fraction of the time required to type the article, the flow of material through this end of the publishing cycle will be speeded up and costs cut substan-

Continued on page 48

TRS-80*
Model I
Model III
Products that set Precedents.
COUGAR
OMIKRON™
Word Processing
MAPPER CP/M Adaptor
CBASIC II*** CP/M** AND
THE BEST WORD PROCESSOR
$1000.00 VALUE ONLY $299
PHOENIX
UNPRECEDENTED SAVINGS
are available on selected CP/M programs through COUGAR, Omikron's official users group.
TRADEMARKS:
*TRS-80/Radio Shack/Tandy Corp.
**CP M/Digital Research
***CBASIC II/Compiler Systems
****PHOENIX/Palantir Inc.
OMIKRON
1127 Hearst St.
Berkeley, CA 94702
(415) 845-8013
367

"It will be satisfactory for virtually all personal computer communications."

Lynx Modem Reconsidered

September's review of modems (Spanning the Electronic Nation, page 134) drew loud cries of "foul" from Emtrol Systems, manufacturers of the Lynx modem (and from Lynx owners who wrote and called by the dozen to defend their machines). Yes, there were some problems with the review, and in fairness to Emtrol, I would like to explain briefly the events leading to the review and expand some comments about the modem. But first, two corrections: the Lynx board was manufactured on Mil. Spec. FR-4 epoxy glass, not phenolic material as I stated. Also, the current price is $299, not $249.

I spoke with John Bickel of Emtrol Systems, who took a great deal of time explaining Emtrol's philosophy, and in helping me track down exactly the unit I had used for review. He discovered that the Lynx machine I obtained through *80 Microcomputing* was a *prototype* intended for product news, not a production unit intended for review. Therefore, the terminal program and documentation were preliminary, and some of my conclusions about the unit's operation were consequently invalid. The current documentation provided by Emtrol is a 12-page, typeset booklet printed on heavy paper. Likewise, Mr. Bickel informs me that the dumb terminal program currently being shipped with the Lynx is reliable and exhibits no fragility.

Let me try to add some perspective to why I would put this prototype up against production units. The review process was begun over 15 months ago. Articles, especially long ones with photos or diagrams, go through an extensive in-house preparation process; this review was submitted at the end of 1980. During the interim, I made a few changes to the article, but my conclusions about the Lynx remained the same. Why no changes to the Lynx comments? Because, ironically, the two Lynx production machines I saw during that preparation period performed even less adequately than the copy I reviewed. This, Mr. Bickel tells me, is simply unlucky happenstance. One modem is Emtrol's "classic service problem," and not at all representative of their product. Furthermore, the present Lynx model is the third generation in production; I saw a mere prototype of the first generation.

I accept that information. I was *not* aware that I had a preliminary machine in my hands, nor were other *80* staff members. A prototype cannot be expected to perform with the finesse of a final product; in support of Mr. Bickel's contention, I understand software giants Lance Micklus and Scott Adams both endorse the Lynx, as does Charlie Butler of The Alternate Source, and it is used by many companies for demonstrations at computer expositions.

Mr. Bickel has also questioned my presentation of the Lynx's use of ports 232–235 (the same ones used by Radio Shack, and disallowing the use of the Shack's RS-232 board) as a deficiency in the unit. In fact, he says, Emtrol's design objective was to make the Lynx perform with the entire body of standard Radio Shack and independent communications software. This software is all written for the 232 area, thereby making the Lynx the most widely useful and compatible modem available. I do not withdraw my objection to this philosophy, because I must use my RS-232 board for computer-to-computer communications without using a modem. But I do agree with Mr. Bickel's observation that the majority of TRS-80 users will probably invest in only one communications device. In light of this, the Lynx unit is in fact easier to get up and running than the RS-232 board/acoustic modem combination provided by Radio Shack, and will be preferred by many TRS-80 owners. Very few users, Mr. Bickel suggests, will face the conflict I presented. Therefore, though the Lynx would not suffice for me, it will be satisfactory for virtually all personal computer communication applications. Furthermore, the Model III version (because of differences in the Model III port hardware) does not conflict with any internal communications devices inside the computer.

I mentioned that the Lynx's parts layout was not compact, and suggested that this might contribute to noise problems within the Lynx. Mr. Bickel disagrees, and suggests that any minor problems with the preliminary unit were unrelated to layout. Yet the electronic production industry has for some time suggested that compactness is in fact a very important consideration in digital circuitry (*Electronic Packaging and Production*, December 1980 and August 1981. As an aside, recent analysis shows that even flat PC board traces are more subject to noise than rounded ones!) Mr. Bickel points out that the Lynx board was professionally designed, and concludes that his vendors did not violate any standard design parameters. On this point, our theoretical disagreement stands.

The Lynx also has a few advantages I failed to point out: unlike the Microconnection, its power supply plugs in rather than needing attachment with a screwdriver. And no additional cable is needed to hook the unit up to the telephone lines (since a double jack is a part of the Lynx unit), whereas an additional hookup is not provided with the Microconnection. That makes the Lynx the only unit of the three "complete as shipped."

There was no intention on my part to (as one correspondent called it) "cold shoulder" the Lynx, but only to review the units in comparison with each other. In fact, I felt that my negative comments about the Lynx were quite mild, because I am not so naive as to believe manufacturers will not improve their product if they discover any limitations. Therefore, for the factual errors pointed out in the first paragraph above, I apologize to Emtrol and Lynx users. I look forward to receiving a new Lynx unit for review, and will report to readers as soon as possible. In the meantime, let me reiterate that the Lynx has been well received over the past year, and deserves the serious consideration of any reader about to enter the world of computer communication.

Dennis Bathory Kitsz
Roxbury, VT

Maxi Micro Manager

I wish to thank your magazine and Wynne Keller, the author, for the fair and impartial review that our program, "Maxi Micro Manager" received in your recent article, "Choosing a Data Base" (August 1981). Such an article has been long awaited and needed by the buying public to help in determining which of the many data base programs are best suited to individual needs.

However, as is the case in most reviews, errors and omissions occur. I would like to take this opportunity to correct the most obvious and set the record straight. First and foremost, the program reviewed by Mrs. Keller was one of our earlier releases. Both the name and the software has changed. The name was changed in January 1981 due to a conflict with another software package; the new name is Maxi Manager System. I hope that your readers take note so as not to purchase the same program twice.

I am pleased to say that we have just rewritten the MMS documentation and now include several simple, easy to follow examples covering all aspects of the program's functions. The new manual (over 120 pages) is now distributed in a three-ring binder to facilitate the addition of new pages as new features are added. Several sections of the previous manual have been retained but are now relegated to the technical reference section.

One of the major enhancements of the new A.3 version of MMS involves the initialization of a new data base. As stated in your review, "this takes time; go have a cup of coffee or lunch if you have several drives." With the A.3 version, the entire process takes less than two minutes. While the article mentions that MMS limits the length of any single field to a maximum of 40 characters, it doesn't mention that MMS allows a single record to contain a total of 800 characters, unlike both CCA and ADISIII, which limit record size to a maximum of 255 characters.

The article goes on to say that MMS rarely allows the user to exit from a routine once started. This is not true. Video screen prompts usually provide safe exits back to the main menu of the program. Input is also said to be slow due to the use of a flashing cursor. While it is true that MMS does use a high speed machine language INKEY$ routine, its speed of character acceptance (on Model 1 versions) is variable and can be set by the user to correct for Model I key bounce problems. Perhaps Mrs. Keller set the Key Debounce Factor to its slowest setting.

The Select features of MMS have also changed. A sixth search criteria, 'NOT INSTRING' has been added, logical And/Or relationships may now be establised between fields as well as within fields, the masked search feature has been corrected and now properly works with numeric fields. The most useful new feature of all is Maxi Manager's ability to locate records regardless of the upper/lowercase spelling of the selection criteria or data contained within the data base.

Mrs. Keller also fails to mention the Search and Delete feature, handy when maintaining subscription lists. While it is true that "MMS allows paging back or forward to the next record in the file," the software also allows paging forward to the next record in the search.

Perhaps the most welcome new feature of all is the Multi-Level Sort. We have to agree with Mrs. Keller when she states that if a data base cannot sort at least two fields at once, it is seriously flawed. Copies of the program that are delivered after August 15 will include this new feature. Also, registered owners will be notified and offered the upgrade.

Last but not least, we have listened to our user's problems regarding our sophisticated print module. The A.3 version of MMS includes a new utility entitled Docufile. This utility enables the most inexperienced MMS user to easily create the required print documents (labels, reports, letters, etc.).

We also see the need for different data base software for varied applications. In this regard, we share Mrs. Keller's sentiments regarding the AIDSIII package and now include a utility that allows AIDSIII files to be merged into MMS files. This option allows small data files to be created with AIDSIII; then when the files grow and become unmanageable, one simply merges them into MMS. This feature indirectly gives AIDSIII the ability to handshake with a word processor.

Dale Kubler
Exador, Inc.
Roswell, GA

Improved Version

I was unable to examine the new version of the Maxi Manager database program (formerly called Maxi Micro Manager, from Adventure International) in time to include comments on it in "Choosing a Data Base" (August 1981).

Since then I have used it extensively and have noticed considerable improvement over the earlier version reviewed in the article. A new printing utility program greatly simplifies the printout process. It is no longer necessary to use a word processor to create a file of report commands. The printing utility program may be requested instead. This utility allows all margin, paging, column and row options to be set by answering simple questions. I would now rate Maxi Manager easier to use than CCA for generating reports. A word processor such as Scripsit or Electric Pencil must still be used for form letter applications or for reports which need explanatory text.

Wynne Keller
Solon, ME

CCA Data Manager

As co-author of the program CCA Data Manager, sold by Personal Software, I was pleased to see it reviewed in the August issue of *80 Microcomputing*. The reviewer, Wynne Keller, did a fine job pointing out the strengths and differences of the three products.

There are two errors in the review regarding the CCA Data Manager, however. The arithmetic capabilities actually do provide exponentiation, including roots. Also, the report writer does allow a sorted file to be processed without exiting the program and re-naming with TRSDOS. Refer to the numbered message explanations at the back of the manual for instructions on these features.

Chet Floyd
Manhattan Beach, CA

Profile Left Out

I am an avid fan of *80 Microcomputing* and have been since its inception. I have noticed, though, that most of your authors tend to downgrade anything to do with Radio Shack. One article in particular concerned me enough to make me sit down with my super homebrew letter-writer for the Model III and write you this note.

The article by Wynne Keller, "Choosing a Data Base," spoke of three data base programs. The article was well written and did describe the three programs fairly well (I have two of the three), but the author omitted Radio Shack's Profile for two weak reasons: It can only sort one field at a time; and it cannot perform calculations. The author excused MMS for the same faults.

Now to Profile. Currently, we use two Mod IIIs and one Mod I to adjunct our main edp effort. Many of the programs we use are either on Profile or are homebrew. We use no other canned programs, because most business programs are either

Reader Service for facing page ✓20→

META TECHNOLOGIES

26111 Brush Avenue, Euclid Ohio 44132
CALL TOLL FREE 1-800-321-3552 TO ORDER
IN OHIO, call (216) 289-7500 (COLLECT)

MTC AIDS-III™

MODEL I . . .$69.95 **MODEL II . . .$99.95**

Introducing the latest addition to MTC's family of data management systems, AIDS-III. NO PROGRAMMING, easy to use. COMPLETE PACKAGE including demonstration application, documentation and MAPS-III (see below).

- Up to 20 USER-DEFINED FIELDS of either numeric- or character-type.
- CHARACTER-type fields may be any length (total: up to 254 characters).
- NUMERIC-type fields feature automatic formatting, rounding, decimal alignment and validation.
- Full feature EDITING when adding or changing records:
  - ENTER FIELD (can't type-in more characters than specified).
  - BACKSPACE (delete last character typed).
  - DELETE FIELD contents.
  - RESTORE FIELD contents.
  - RIGHT-JUSTIFY FIELD contents.
  - SKIP FIELD (to next or previous field).
  - SKIP RECORD (to next or previous record).
- SORTING of records is MACHINE CODE assisted.
  - 200 RECORDS (40 characters) in about 5 SECONDS.
  - ANY COMBINATION of fields (including numerics) with each field in ascending or descending order.
- SELECTION of records for Loading, Updating, Deleting, Printing and Saving is MACHINE CODE assisted.
  - Specify up to 4 CRITERIA, each using one of 6 RELATIONAL COMPARISONS.
  - LOAD or SAVE selected records using MULTIPLE FILES.
  - Example: Select records representing those people who live in the state of Colorado, but not in the city of Denver, whose last names begin with "F" and whose incomes exceed $9000.00.
  - Example: Select records representing those sales made to XYZ COMPANY that exceed $25.00, between the dates 03/15 and 04/10.

MAPS-III (MTC AIDS PRINT SUBSYSTEM), included at no charge, has the following features:

- Full AIDS-III SELECTION capabilities.
- Prints user-specified fields DOWN THE PAGE.
- Prints user-specified fields in titled, columnar REPORT FORMAT, automatically generating column headings, paging and (optionally) indentation.
- Can create a single report from MULTIPLE FILES.
- Prints user-defined formats for CUSTOM LABELS, custom forms, etc.

BELOW ARE TESTIMONIALS from owners of AIDS systems. These are absolutely authentic statements and are typical of the comments we receive.

"This program will do more for my business than all the other programs I have, combined."
David Wareham, Vice President (EDP), National Hospital and Health Care Services Inc.

"We have 32 different Data Base Management packages for the TRS-80. AIDS-III is easily the best. It also makes it easier for us to step up to our Model II since the package is available for both computers."
Jack Bilinski, President, 80 Microcomputer Services

"Your AIDS program is far and away the finest information management system that I've ever seen. I am currently using it to maintain a clear picture of the demographic data on all the kids in our residential treatment program and it is working for me superbly."
Frank Boehm, Director, Front Door Residential Treatment Program

MTC AIDS CALCULATION SUBSYSTEM-III™

MODEL I . . .$24.95 **MODEL II . . .$39.95**

MTC's most popular AIDS subsystem. Use for report generation involving basic manipulation of numeric data. Features are:

- User-specified page title
- Columnar Headings
- Optional Indentation
- Use for accounting, inventory, financial and other numeric-based information systems.
- Columnar subtotals generated when there is a change in a user-specified column.
- User-specified Columnar Totals
- Columnar values computed using constants and/or column values
- Balance forward calculations (Ex: Gross sales equals previous gross sales + sale amount + sales tax).

Compare AIDS-III™/CALCS-III™ with any other data management package under $100!

CALCS-III™ REQUIRES THE PURCHASE OF AIDS-III™

AIDS OWNERS! WE HAVE WHAT YOU'VE BEEN WAITING IV. . .

MTC CALCS-IV™, that is.

- More Computations
- Save Report Formats on Disk
- Faster, and more!

MTC CALCS-IV™. $39.95
For Model II. $59.95

MTC AIDS MERGE-III™

This subsystem will combine up to 14 AIDS-created data files into a single, large file. An optional purge capability removes duplicate entries while performing the merge operation (can even be used to eliminate duplicates in a single file). Machine-code assisted for high-speed performance, MERGE-III™ properly handles files sorted by any combination of fields, including numerics, with each field in ascending or descending order.

MTC AIDS MERGE-III™. $19.95
For Model II $29.95

THE COMPLETE MTC AIDS-III™ PACKAGE

SAVE $$$$

Includes MTC AIDS-III™ CALCS-III™ and MERGE-III™

A comprehensive system at a competitive price!

MODEL I & III $99.95
MODEL II $149.95

Add $25 for CALCS-IV™

MORE PRODUCTS

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

Products damaged in transit will be exchanged.

PRICES IN EFFECT Nov. 1, 1981 THRU November 30, 1981.
Prices, Specifications, and Offerings subject to change without notice.
8111

WE ACCEPT

- VISA
- MASTER CHARGE
- CHECKS
- MONEY ORDERS
- C.O.D.

- Add $3.00 for shipping & handling
- $3.00 EXTRA for C.O.D.
- Ohio residents add 5½% sales tax.

poor and don't really fit, or are so complicated that you must hire a programmer to operate the thing. Profile makes better use of disk space than any other data base program. It will accept up to four disk drives for one record. True, you wait awhile for the files to be formatted, and the sort is very slow, but the amount of files per disk is fantastic.

With MMS, you must call up another program in order to perform calculations. With Profile, you can write a basic program to call up Profile data direct from the disk and accomplish the same thing—automatically, if you want. I can even sort with a separate machine-language module, put Profile data on Visicalc, and handshake with Scripsit.

Profile is disk-based at all times. You will never lose the data during a power failure, except for the entry you didn't have time to save. Incidentally, the comparative number of files per disk (drive 0 with DOS), say of a file whose length is 132 characters including 16 spaces for printout, is AIDS = 187, MMS = 437, PROFILE = 1340.

A. R. Pedrick
Kissimmee, FL

Modifying Tiny Pascal

When Lt. Harrel's "Modifying Tiny Pascal for Disk" first appeared (July 1981) I didn't have disk drives. I received them late last month and one of the first things I did was code his PAS32K<PASMOD> fix. After several attempts I succeeded in getting the compiler on disk. I then entered the Filesave and Fileload programs on disk and saved them to disk. Using the short programs provided with Tiny Pascal I tested the new disk modification and was very pleased.

The real test came when I coded his version of Breakout using the Tiny Pascal tape. I attempted to save it as he had instructed and nine tenths of the way through, I got an overflow error in line 10.

Lo and behold, this program was approximately 3432 bytes (1991 codes). It began loading at &H73FO (29680) and continued through decimal 33112. As soon as AD = &H8000 (32768) an error is generated.

I found that by adding the lines below you will be able to continue to read beyond &H7FFF:

- Increase each line number by ten.
- Now J = PEEK (AD) is line 100.
- Insert: 105 if AD = 32767 then GOTO 170

```
160 FD = AD-65535:AD = FD: GOTO 100
170 A$ = A$ + CHR$(J): GOTO 160
```

Now when J = PEEK (32767) it will branch to 170, store the byte found there and return to 160 in order to convert to the negative numbers (−32767 through −1) in a 32 or 48K machine.

I thoroughly enjoyed his Z-Bug program and was successful in getting it up-high and on disk. However, in module 4, page 160 he defines the catalog and copy messages as MS and calls them as MSG1, MSG2, etc. . . could be a typo.

The Basic disassembler (August 1981) was a godsend. However, it won't PEEK beyond 7FFF and I tried a similar fix but to no avail. Any and all assistance would be greatly appreciated.

Lee Swaringen
Thomasville, GA

Symbolic Dump Printout

After I saw an article in the July issue of *80 Microcomputing* concerning patching RSM to allow it to make system tapes for Model III, I wrote patches to get RSM to print a symbolic dump out to my line printer.

The patches are as follows:

ADDRESS	FROM	TO
7070:	F5	C3
7071:	F1	81
7072:	E1	7F
7F81:	00	D3
7F82:	00	FB
7F83:	00	F5
7F84:	00	F1
7F85:	00	E1
7F86:	00	C9
7F87:	00	F1
7F88:	00	F5
7F89:	00	C3
7F8A:	00	76
7F8B:	00	70

These changes are for the 16K version of RSM-2. For a 32K version the addresses are 4000 (hex) larger, and begin with the prefix A. For a 48K version they begin with E.

Lester Beasley, Jr.
Dallas, TX

More RSM-2 Changes

In the July issue you published a short note from me giving patches to RSM2 or RSM2-D to allow it to read or write tapes for the TRS-80 Model III. This note gives further information that will allow RSM-2 to start and stop the tapes:

ADDRESS	CHANGE FROM	TO
6CDE	04	02
6CDF	FF	EC
6EC5	FF	EC

If you have a disk system, the RSM-2 or RSM-2D tape can be loaded by the system program Tape and the changes can be made with Patch.

Maynard B. Neher
Columbus, OH

WORD Update

Thank you for the excellent review on Word (August 1981, page 234). Mr. Cook had the program quite a long time ago. Here are the updates:

- WORD-4 has a compiled program included free of charge. The compiled version will print as fast as your printer. Many minor enhancements are also included.
- WORD-5 ($79) has also been available for almost a year. It can merge the data file maintained by our data base managers (IDM-4,5) and our mailing lists (MAIL-5). All versions are also available in Model-3.

Also, Program Listing 1 can be simplified by taking out the following lines:

```
10 .LL 65
20 .PL 50
80 .FI
```

Instead, use default values. Also delete: 200 .EN. It's automatic in 95 percent of the cases.

Tony Pow
Micro Architect Inc.
Arlington, MA

NEWDOS Messages

Owners of NEWDOS 80 V.2 who do not like the "NEWDOS 80 is ready" message can use Superzap to change that to something of equal or less length, such as "By your comand." Display sector 164, enter MOD C9 and input the hex code (see tables in the back of LII manual) for the letters of your preferred prompt.

You can likewise replace the three-line NEWDOS 80 banner with DOS instructions for your users, reminders, notes or whatever. Display sector 17. Enter MOD C4. Bytes C4 (which contains BF) through FE (which contains 2E) can be changed to modify the first line. (That is 59 bytes. Do not type over the "OA" linefeed at byte FF.)

The second line is tricky. The first character of the second line is byte 00 of sector 18. The rest of line two is in bytes 05 through 3D (57 bytes). Typing over bytes 01 through 04 will keep your disk from booting. Line 3 is bytes 3F through 75.

Nelson Ford
Houston, TX

Reader Service for facing page ✓20→

Hex Conversion

If you have a color computer with Extended Basic or a Model I or III and Disk Basic, you may not be aware of a capability you have. (This also applies to Microsoft's Level III Basic).

The two functions & and &H are used to convert an octal or Hex constant to decimal. This is a convenient feature, but if you want to use this function on a variable within a program, you will be irritated to find that it won't work! It is designed to work only with a constant.

Because the information following these statements is treated as a string constant without quotation marks the variable following these statements will be treated the same (eg: &HA$) and give you garbage.

The key to fooling the interpreter is the VAL function. Assuming that A$ = "FF", use this statement to convert A$ to a decimal number.

A = Val("&H" + A$)

After this statement is executed the variable A will equal 255. Use of the octal function is identical except you substitute & or &0 for $H. Assuming A$ = "7777"

A = Val("&" + A$)

After the above statement the variable A will equal 4095.

Bill Dickson
Computer Division Supervisor
Offshore Navigation Inc.
New Orleans, LA

Oddities

I've had my TRS-80 Model I for three years now and have never found an error in documentation or in hardware without reading about it first in the pages of *80*. Now all of a sudden these oddities are jumping out at me faster than I can take it!

- It is possible that we disk spinners are getting more for less. While formatting new disks, I answered the "Indicate the number of tracks on the disk" question with 41. Surprisingly, NEWDOS 80 formatted 41 working tracks! I own two Percom 40 track drives. A friend who owns the same equipment also formatted 41 valid tracks.
- On page 14 of the Daisy Wheel II manual a chart of control codes is printed. A top of form code is not listed. I LPRINTED CHR$(12) which was not listed in the chart. Instantly the printer went to the top of the next form. CHR$(11) or CHR$(12) will cause a TOF function. This is contrary to the paragraph above the chart which clearly states: "The Daisy Wheel recognizes the following control codes and ignores any others."

Faithful readers of *80* will remember Datagen in the August issue. Use Datagen to change your Z80 code into Basic data statements and use your terminal software's "Send Basic Program" function to send your buddy a copy.

Scott Kantner
Shoemakersville, PA

Graphics for LPVII

If you have purchased a TRS-80 Line Printer VII and have had difficulty in generating graphics this may help.

Since the line printer does not handshake in the dot graphics mode the program must allow the print head to return home before the next print command. A fixed delay would work, but reduce the overall throughput since the head return time is short for a short displacement and any delay used would have to be long enough to accommodate a full-scale displacement. To get around this problem I use a delay which is a function of the displacement as shown in line 220 of the program. The function was determined experimentally and might benefit through some fine tuning of the coefficients.

The magnitude of the displacement is also used to set the high (h) and low (l) field specifiers as required, line 160 (see note on page 11 of the manual).

Finally, a modulo-7 counter is used to suppress the line feed and generate the required graphics character, which provides an effective "line feed" or in this case a "row advance". After the current band is completed a full printer line feed is allowed in line 210 (if P = 6).

Henry E. Santana
Loveland, CO

```
100 DEFINT X,P,N,H,L,T
110 LP=14312 ' LINE PRINTER PORT ADDRESS
120 POKE LP,18 ' SET GRAPHICS MODE
130 PI=3.141593
140 FOR X=0 TO 360
150 N=200+200*(SIN(X*PI/90)+(SIN(X*PI/30)/3)+(SIN(X*PI/18)/5))
160 IF N>255 THEN H=1: L=N-255 ELSE H=0: L=N
170 POKE LP,27: POKE LP,16 ' SET DOT COLUMN ADDRESS MODE
180 POKE LP,H: POKE LP,L ' SET POSITION
190 P=X-7*INT(X/7) ' MODULO 7 COUNTER
200 POKE LP,128+2^P ' PRINT GRAPHICS CHARACTER
210 IF P=6 THEN POKE LP,10: ELSE POKE LP,26 ' CARRIAGE RETURN
220 FOR T=0 TO 200+3*N: NEXT T ' DELAY
230 NEXT X
240 POKE LP,30 ' RETURN TO CHARACTER MODE
250 END
```

Program Listing

Calculate Color Graphics Codes

The codes for the very low resolution color computer graphics may be calculated about as fast as they can be looked up. Consider the block:

8	4
2	1

For green, to light up 1 use 128 plus 1, for yellow 144 plus 1, for blue 160 plus 1, and so on. For green, to light the top two use 128 plus 8 plus 4 equals 140, for yellow 144 plus 8 plus 4 equals 156, and so on. Take the characteristic color code for unlighted pixels (128, 144, 160, 176, 192, 208, 224 and 240) and add to it the color number plus the number of each of the pixels you wish to light. Use CHR$(X) for rapid execution. Note that this is the opposite of Level II graphics where the blocks are numbered

1	2
4	8
16	32

and a constant 128 (all pixels off) is added to the sum of the blocks. You can do these operations very quickly and save time over referring to a graphics chart. (Note: you may do it, as for example, CHR$ (128 plus 8 plus 4) in a program, although it wastes memory.)

Franklyn D. Miller
Cincinnati, OH

Program Bugs

There are a couple of bugs running around loose out there in TRS-80 land. The first and most grievous is in Super-Utility by Kim Watt. Under certain circumstances SU will eat sector five of track 17.

Reader Service for facing page ✓542→

Right there in the middle of the directory there appears a great big 256-byte gap of nothing. I contacted Mr. Watt about this and he verified that two other persons had encountered and reported the error, and that he finally duplicated it. Apparently Mr. Watts' idea of a fix is to offer a properly operating version of Super-Utility at an exchange price of about $25. He has rewritten the program and is now marketing the new properly operating version.

The second bug I uncovered is in a fine word processing program known as Newscript. When used with the MX-80 printer, each time an underline is done the page creeps about 1/72nd of an inch. If much underlining is done your text can walk way up the page. This problem has now been fixed, and the newer versions do not have it. I would suggest holders of older versions of Newscript contact Prosoft and ask them about an update.

Finally, there's an error in NEWDOS 80, version 1.0. You can't assign a variable a name beginning with an A, D, or E with a second character (a numeric) without encountering a Syntax Error while in the command mode. Well, it is still there, but NEWDOS 80 version 2.0 has corrected the problem. I just got version 2.0 yesterday and checked it out. It looks great. Version 1.0 owners had better trade it in on the new one now.

Jerry L. Latham
Midwest City, OK

80 DEBUg

High Speed Data Tapes

The attached listings are submitted as required Debugs for the "High Speed Data Tapes" article (July, 1981).

In the final version of the subroutine, sufficient time was not allowed when writing the tape for strings longer than 63 bytes to be moved from the input buffer to the string storage location. Before the subroutine returns to the tape reading routines, the next string has already started to be input and the computer gets out of synchronization with the tape, causing the dreaded lock-up condition.

The changes submitted will calculate extra string move time needed for strings longer than 63 bytes and will delay the writing of the next string long enough for each string to be properly moved before the next string is input. The delay, although substantial in computer time, is not noticeable.

My thanks to Carl Jaquett for informing me about this problem. Any person who would like to use this subroutine in conjunction with Racet's GSF subroutines can send me a SASE and I will return a listing of the changes necessary for use of the subroutine as GSF-routine #39.

Jim Glosser
1425 Eden Road
York, PA 17402

```
00200          ORG    ENDMEM-527-13     ;START OF THIS PROGRAM
00201                 ;SET MEMORY SIZE TO 32227=16K
00202                 ;                 48611=32K
00203                 ;                 64995=48K
03211          LD     A,(IX)            ;(3) BYTES IN STRING
03212          SUB    64                ;(2) 63 DON'T REQ DELAY
03213          JR     C,NODEL           ;(2) SKIP DELAY IF < 64
03214          INC    A                 ;(1)ADD FOR 64-63=1
03215          LD     B,A               ;(1)DELAY CYCLES REQ'D
03216 DELAY    JR     DELCON            ;(2) 12 T STATES
03217 DELCON   DJNZ   DELAY             ;(2) 13 T STATES
03218          ;21 T STATES PER BYTE REQUIRED FOR BYTES OVER
03219          ;63 TO COMPENSATE FOR LDDR TIME IN LINE 2670.
03220 NODEL    DEC    DE                ;SUB 1 ELEMENTS TO WRITE
```

Listing 1.

```
100 CLEAR 3500:DEFINT A-Z:DIM T$(26)
110 REM DISK USERS MUST EXECUTE DEFUSR IN LINE 120 AS FOLLOWS
              16K=DEFUSR &H7DEE
              32K=DEFUSR &HBDEE
              48K=DEFUSR &HFDEE
180 FOR X=17TO26:T$(x)=STRING$(255,X+64):NEXT
540 REM IN LINE 550, POKE VALUE '0' AS FOLLOWS:
              16K=POKE 32227,0
              32K=POKE -16925,0
              48K=POKE -541,0
```

Listing 2.

Screen Printing for the MX-80

Here are a couple of zaps for NEWDOS PLUS and NEWDOS 80 to change the screen print routines for Epson MX-80 owners. These zaps allow screen printing of graphics characters using the JKL option. Both zaps work with the MX-80 set up in the standard mode (Dip switch 2-4 is off). This allows use of all print and character options with graphics characters which is not normally possible with either operating system. NEWDOS 80 does have a graphics capability, but without a change to the operating system, the printer must be in the TRS-80 mode which restricts print and character options. After applying the zaps to NEWDOS 80, the system option AK=N will give graphics capability with the printer in the standard mode. These two byte changes are easily made using the superzap utility.

First, run Superzap. Second, display NEWDOS PLUS—Track 0, relative sector 6 and NEWDOS 80—Track 0, relative sector 7. Third, change relative bytes NEWDOS PLUS—7F and 80 from 3E,2E to C6,20 and NEWDOS 80—B9 and BA from 3E,2E to C6,20.

The replaced code is in the screen print routine after the data is sorted for the normal ASCII character range. The original code changes all graphics codes to periods. The new code adds 20H to the graphics code to match the MX-80 standard coding table.

Herbert S. DuBois
New York, NY

✓See List of Advertisers on page 402

80 AID

T-Bug Tip

This letter is a response to Jonathan Yarden's plea for help in converting T-Bug for use on a Model III (*80 Microcomputing,* August, 1981).

Replacing the tape I/O subroutines with calls to the ROM is fairly straightforward, and requires changing about 20 bytes. However, since the Model III routines abort tape I/O when the break key is pressed, control will then be returned to Basic instead of T-Bug command level. To avoid this unpleasantness, the modifications must be more elaborate.

Load T-Bug into your Model III and use the M command to change the following bytes:

Start address:	Change to:
4643	21 03 42 36 C3 23 36 62 23 36 47 CD 96 02 00
46DD	CD 87 02
46EB	C3 FB 46
4762	CD F8 01 C3 A5 43
4782	C3 35 01
478C	C3 64 02
47C5	C3 F8 01

Then use the M command to set the desired baud rate, and write your Model III T-Bug tape (P 4380 4980 43A0 TBUG). Happy T-Bugging!

Herb Robinson
3814 Skyline Rd.
Carlsbad, CA 92008

Game Confusion Cleared

I have a solution for Dennis Banik's problem of reading data from different blocks within a Basic program ("Game Confusion" *80 Aid*, August 1981). Duplicate this routine, in each block which requires reading data, ahead of any Read statement and after any Clear statement.

```
W=PEEK(16627) : POKE 16640, PEEK(16628)+(W>239) : POKE 16639,W
```

If any Restore's are required in a particular block, make this a subroutine which the program skips over.

```
30010 REM BLOCK SIX
30020 GOTO 30040
30030 Q=PEEK(16627) : POKE 16640, PEEK(16628)+(Q7239) : POKE 16639, Q : RETURN
30040 CLEAR 5000 : REM 'CLEAR' SETS DATA POINTER TO BEGINNING OF BASIC PROGRAM
. . .
30160 GOSUB 30030 : REM ESTABLISHES DATA POINTER IN BLOCK SIX
. . .
30480 GOSUB 30030 : REM RESTORES DATA POINTER TO BEGINNING OF BLOCK SIX
```

The first block does not need this routine and can use Restore. In addition, this routine must be contiguous and on the same program line.

Vernon B. Hester
42403 Old Bridge Road
Canton, MI 48188

Software Problem?

I am truly delighted with our two disk, 48K Model III. TRSDOS 1.3 is more than I expected. Perhaps the neatest command in the system is CMD 0; it sorts 300 records of 100 byte length so quickly I am not sure how long it takes. The logic, versatility and documentation of the Model III exceeded my expectations.

But I still want more. I want a fast search program. With a disk file of 300 records, I want to be able to flash any one of the records onto the screen as soon as I get the record number. However, if I do an alphabetical search using "INSTR" it takes from 30–60 seconds. Is this too long? Do I have a software problem? To speed things up should I make a short file containing only the names and the record numbers, translate the name to a number and then get the number?

B. Jim Smith
St. John's High School
Shrewsbury, MA 01545

Missing Person

In response to Mr. Pfingstag's letter in the August 1981 issue I would like to apologize to any readers who might have been unable to contact me about my article "Selectric Hard Copy" (September 1980). I moved shortly after this article was published but I am indeed alive, well and living at the address below. I have discussed this project with hundreds of readers and would be more than happy to correspond with anyone interested in interfacing the Selectric. Avid readers of *80 Microcomputing* might have noticed that the missing parts list, other corrections, my new adress and telephone number have already been published. Where, you might ask? In the November *1980* issue of *80 Microcomputing*, where else? It is heartening to know that the editors of the best TRS-80 journal are too busy putting together another excellent issue to sit around all day reading old columns.

Michael W. Bickerton
248 Barren Hill Road
Conshhocken, PA 19428
215-825-7556

ROM RND Hangup

For some time, I have been dabbling in machine language programming on my Level II 16K Model I. I have run into difficulties, however, trying to access the ROM RND function for a generation of pseudo-random integers within a given range (RND(X)). James Farvour, in Microsoft Basic Decoded, suggests:

```
LD ,2
LD (40AFH),A      ; set type flag to integer
LD A,50           ; range of random no. 1-50
LD (4121H),A      ; put range into WRA1
CALL 14C9H        ; get a random no. between 1
                    and 50
LD HL,(4121H)     ; load random no. from WRA1
LDL (RVAL), HL    ; to HL, and move to local
                    area
```

This routine crashes my system. I get an illegal function call (FC), which indicates, perhaps, the attempt to use a negative value for the argument. Looking at the code for 14C9H, the RND subroutine, it appears that the

The Software Dilemma

Recently I purchased Trakcess by Roxton Baker through Alpha Byte Stores. I am very impressed. Trakcess has lived up to my original expectations and much more. However, what really surprised me was that Mr. Baker actually encouraged people to share the cost of the program and use or make copies of it jointly.

As a full-time student with a part-time job, this is like a dream come true. Not everyone has money to shell out like candy for programs they do not even know if they can use or enjoy. If more people would follow this fine example, they would see increased sales as they would be opening their programs to a larger market.

Another point that Mr. Baker brought up is protected disks. Having written software myself, I think that the author has a right to be concerned about his software. However, this should not be at the consumer's expense. The consumer should be able to make as many backups of a program as he or she wishes. It is not fair to pay $100 for a program and then have to pay $10 plus shipping if the thing crashes.

David A. Roch
El Paso, TX

80 AID

routine does not access WRA1 for the argument, but HL instead:

```
14C94   C ALL 0A7FH; test for integer flag
        LD ,H
        OR
        JP n,1E4AH; Illegal FC if negative
        OR L
        JP Z, 147FOH; if parameter O, compute RND (o).
        . . .
14FOH            ;compute RND (o).
```

I have tried putting a parameter, or O, into HL directly, prior to calling 14C9H. The program does not crash, but the output in WRA1 (4121H) is not limited to the desired range. Does anyone know the answer to my problem? Any help will be appreciated.

Thomas R. Jones
235 East 73rd Street
New York, NY 10021

Network Help

Is there anyone out there who can show me how to build a switching network (or point me in the direction of commercially available hardware) to interconnect my Model I, Model II, Daisy Wheel II and Line Printer V?

My problem is the risk of damage to circuit board and connector plugs (not to mention the time that it takes) from continually plugging and unplugging printers when I need letter quality or high-speed printing as output from an application running on one computer or the other.

Peter C. Bennett
7139 Scott Road
Homer, NY 13077

On The Right Track

I just got my NEWDOS 80 to run on my 80-track drive and it turned out to be quite an undertaking. I had to use the MAKE 80 on the original NEWDOS 80 disk to make it readable to my 80-track drive. Then I received my LN Doubler. I tried to make Doublezap II work, but had no luck. It apparently needed more tracks. Then I decided to play with it and converted a NEWDOS copy to 40-track using the PDRIVE and changing the Gat table to eliminate all reference to tracks 40 to 79. I applied Doublezap II and it worked.

I also put it up with an expanded directory. When the run was completed, I had to use PDRIVE to get 80 tracks again and change the Gat table again. TR 17 SEC 0 byte 46 to byte 8DH got zapped to FCH and byte CCH to CFH became 8EH 50H 96H 42H. (See Table 1.) This is now an 80-track system disk with the free showing * 80 trks 114FDES 220 grans * on a side with sample 01/BAS removed and the Doublezap II utilities installed. It was really great to figure it all out.

John H. Fields II
Venice, FL

Stock Market Crash

I recently bought Color Space Traders from Spectral Associates. It involves buying stocks in hypothetical companies and trying to increase their market value. After playing the game a few times, and losing miserably each time, I began to wonder if all my work as a business undergraduate and weekend programmer had been in vain.

Then, much to the delight of my ego, I discovered a flaw in the game. When a stocks' value exceeds $3000, it splits two for one. The price is split in half, and the number of shares held by each player should be doubled. A For. . . Next loop is set up in line 3190 to do this, but a programming oversight left out the Next so that only player number 1 has his or her quantity of shares doubled, even though its value had been divided by two. Editing the program as I've done here makes it run correctly:

```
3190 FORI1 = 1TOP1:S(T1,I1) = 2*S(T1,I1):NEXTI1:
GOSUB50000:RETURN
```

Ken Teagan
Glendale, CA

Tom Rosenbaum of Spectral Associates tells us they've found the error, and have corrected it.—Eds.

Variance

C. Brian Honess' article "Vital Statistics" in the August issue contained one significant error. The mathematical form of the variance is generally taken to be:

$$\sigma^2 = \frac{\Sigma(x - X)^2}{(n - 1)}$$

This differs from Mr. Honess' equation in that the denominator contains an $n-1$ term instead of just n. In calculations involving a large number of data points (large n) the difference is insignificant. However, in smaller samplings it can be quite large.

The reason for the $N-1$ term instead of n is somewhat complex but it stems from the fact that each of the data points are first used to calculate the mean value of the sample (X in the formula) and the data points and the average are then used to calculate the variance. This two-step calculation is said to reduce the degrees of freedom of the system from n to $n-1$. Refer to any elementary statistics book for a thorough explanation.

Albert Stiegman
Department of Chemistry
Columbia University
New York, NY

Reader Service for facing page ✓581→

80 CALENDAR

November

● The University of California, Berkeley, will sponsor a two-day course titled **"Comparison of Recent Microcomputer Architectures"** for design engineers, programmers and technical managers at the San Francisco Airport Hilton, **Nov. 9-10.** Fee is $400. Information and registration is available through the office of Continuing Education in Engineering, University of California Extension, 2223 Fulton St., Berkeley, CA 94720.

● The New Mexico Computer Society will host the third annual **New Mexico Computer Fair Nov. 14** at the Albuquerque Civic Auditorium. Admission is free. Information is available from Ron Benninghoff, c/o New Mexico Computer Society, 515 Wyoming NE No. 2, Albuquerque, NM 87108

● Ken Orr and Associates, Inc., 715 E. 8th St., Topeka, KS 66607, will sponsor several courses and conferences in November. Subjects are: **Nov. 3-6 Structured Requirements Definition, San Antonio, TX; Nov. 3-6, Structured Program Design, Boston, MA; Nov. 10, Management Overview of Data Structured Systems Development, Denver, CO; Nov. 11, Management Overview of Data Structured Systems Development, Portland, OR; Nov. 13, Management Overview of Data Structured Systems Development, Seattle, WA; Nov. 9-13, Structured Systems Design/ Structured Program Design Combined Course, St. Louis, MO; Nov. 10-13, Structured Requirements Definition, Los Angeles, CA; Nov. 16-20, Structured Systems Design/Structured Program Design Combined Course, Washington, DC.**

Coming Next Month

In December, *80 Microcomputing* looks at the TRS-80 used in hobbies. Topics will include micro games, sports forecasting and collection filing systems.

Athletic Technical Editor Michael Vose will share his Runner's Log program. Dust off those sneakers and find your sweatsuit!

Review Editor Michael Nadeau has compiled a Buyer's Guide to peripherals and small electronics.

News Editor John Mello gets the scoop on Arcnet, the Model II network.

News Editor Bert Latamore has tracked down a model railroad club that uses a homemade micro.

Freelancer Robert Lloyd writes about the program he wrote to catalog his coin collection.

CORVUS

TRS-80 MOD I,II

Controller, Case/P.S., Operating System, A&T

5 Megabytes	3245.00
10 Megabytes	4645.00
20 Megabytes	5545.00
MIRROR BACK-UP	725.00

SPECIALS

SPECIAL NO. 1
TRS-80 DISK AND OTHER MYSTERIES, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE 44.50

SPECIAL NO. 3
NEWDOS/80 2.0, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE 149.00

SPECIAL NO. 4
MICROSOFT BASIC COMPILER, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE 190.00

SPECIAL NO.5
MICROSOFT BASIC DECODER AND OTHER MYSTERIES, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE 49.95

SPECIAL NO. 6
DOUBLE DENSITY SPECIAL NEWDOS/80, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE 279.00

COMMUNICATION SPECIAL MOD I & III
NOVATION MODEM, OMNITERM TERMINAL SOFTWARE, RS 232 CABLE
ACOUSTIC SPECIAL 239.00
DIRECT CONNECT SPECIAL 259.00

DISK LIBRARY SPECIAL
RACET DISK-CAT CATALOGING PROGRAM, FLIP SORT (50 DISK CAPACITY), 50 STICK-ON DISK LABELS. 59.95

TWO NEW 'OTHER MYSTERIES' BOOKS

THE CUSTOM TRS-80	24.95
MICROSOFT BASIC FASTER AND BETTER	24.95

TRS-80 GAMES

TEMPLE OF APSHAI	34.95
HELLFIRE WARRIOR	34.95
STAR WARRIOR	34.95
RESCUE AT RIGEL	24.95
CRUSH, CRUMBLE & CHOMP	24.95
INVADERS FROM SPACE	17.95
PINBALL	17.95
STAR TREK 3.5	17.95
MISSILE ATTACK	18.95
STAR FIGHTER	24.95
Z-CHESS III	24.95
ADVENTURE NO. 1,2 & 3	34.95
ADVENTURE NO. 4,5 & 6	34.95
ADVENTURE NO. 7,8 & 9	34.95
DUEL-N-DROIDS	17.95
STARFLEET ORION	21.95
INVASION ORION	21.95
OLYMPIC DECATHLON	24.95
ZORK	34.95
MONTY PLAYS MONOPOLY	31.95
SARGON II	31.95
BLACKJACK MASTER	27.95
ROBOT ATTACK	17.95
GALAXY INVASION	17.95
SUPER NOVA	17.95
TUESDAY MORNING QUATERBACK	26.95
LUNAR LANDER	18.95
THE MEAN CHECKER MACHINE	21.95
GAMMON CHALLANGER	18.95
SPACE ROCKS	18.95
PIGSKIN	17.95
ZOSSED IN SPACE	18.95
ARCADE-80	21.95
COMBAT	18.95
SPACE INTRUDERS	17.95
SILVER FLASH	17.95
MORTON'S FORK	26.95
PROJECT OMEGA	21.95

CP/M is a registered trademark of Digital Research.

TRS-80 MODEL III HARDWARE FROM MTI

INTERNAL DISK DRIVE KIT

A new low cost internal disk kit is available to expand your 16K Model III computer to a full 48K disk system. 40 or 80 Track drives are available. 40 Track drives store 175K and 80 Track drives store 350K. Up to four drives may be added, including two external drives. Using four 80 Track drives provides a total disk storage of 1.4 megabytes.

DRIVE NO. 1 - 40 TRACK SYSTEM 695.00

INCLUDES: Controller, power supply, cables, mounting hardware, 32K of RAM & installation instructions.

DRIVE NO. 2 - 40 TRACK SYSTEM 279.00

80 TRACK INTERNAL DISK SYSTEM

DRIVE NO. 1	789.00
DRIVE NO. 2	399.00

EXTERNAL EXPANSION DRIVES

Add up to two external drives to any two drive Model III system.

40 TRACK EXTERNAL DRIVES

DRIVE NO. 3	379.00
DRIVE NO. 4	359.00

80 TRACK EXTERNAL DRIVES

DRIVE NO. 3	519.00
DRIVE NO. 4	499.00

DOS PLUS OPERATING SYSTEM FOR MOD III

40 Track	99.00
80 Track	119.00
Expanded DOSPLUS 3.3.8 runs 40 and 80 Track	149.00

MOD III Cooling System 49.00

Reduce failures due to heat buildup, this kit insures proper air circulation for trouble free performance.

RS-232 Serial Interface 99.00

Direct connect Modem 199.00

We built a reputation on our prices and your satisfaction.

Alpha Byte STORES

69

We guarantee everything we sell for 30 days. If anything is wrong, just return the item and we'll make it right. And, of course, we'll pay the shipping charges.

We accept Visa and Master Card on all orders. COD orders accepted up to $300.00.

Please add $2.00 for standard UPS shipping and handling on orders under 50 pounds, delivered in the continental U.S. Call us for shipping charges on items that weigh more than 50 pounds. Foreign, FPO and APO orders please add 15% for shipping. California residents add 6% sales tax.

The prices quoted are only valid for stock on hand and all prices are subject to change without notice.

(213)706-0333

31245 LA BAYA DRIVE, WESTLAKE VILLAGE, CALIFORNIA 91362

Our incredible low prices just got 2% lower.

Have a merry computer and a happy software.

Take an additional 2% off our listed prices, until December 24.

PRINTERS

ANADEX DP 9500	1295.00
ANADEX DP 9501	1295.00
CENTRONICS 739	765.00
C-ITOH 25 CPS PARALLEL	1440.00
C-ITOH 25 CPS SERIAL	1495.00
C-ITOH 45 CPS PARALLEL	1770.00
C-ITOH 40 CPS SERIAL	1870.00
C-ITOH TRACTOR OPTION	195.00
EPSON MX-80	$CALL
EPSON MX-80 F/T	$CALL
EPSON MX-100 GRAPHIC	$CALL
EPSON GRAPHICS ROM	90.00
IDS-445G PAPER TIGER	779.00
IDS-460G PAPER TIGER	945.00
IDS-560G PAPER TIGER	1195.00
MALIBU 200 DUAL MODE	2695.00
NEC SPINWRITER 3510 SERIAL RO	2195.00
NEC SPINWRITER 3530 PARALLEL RO	2195.00
NEC SPINWRITER 7710 SERIAL RO	2645.00
NEC SPINWRITER 7730 PARALLEL RO	2645.00
NEC SPINWRITER 7700 D SELLUM OPTION	2795.00
NEC SPINWRITER 3500 SELLUM OPTION	2295.00
OKIDATA MICROLINE 80	399.00
OKIDATA MICROLINE 82A	569.00
OKIDATA MICROLINE 83A	799.00
OKIDATA MICROLINE 84	1199.00

16K RAM SPECIAL $13.95

SET OF 8 NEC 4116 200 ns WITH INSTRUCTIONS.
GUARANTEED ONE FULL YEAR.

MODEL I, III SOFTWARE

LAZY WRITER MOD I	119.00
LAZY WRITER MOD III	169.00
MICROPROOF SPELLING CHECK	84.95
PROSOFT NEWSCRIPT MOD I,III	99.00
PROSOFT MAILING LABELS	29.95
PROSOFT NEWSCRIPT & LABELS	115.00
SPECIAL DELIVERY MOD I,III	119.00
X-TRA SPECIAL DELIVERY MOD I, III	179.00
TRACKCESS MOD I	24.95
OMNITERM SMART TERMINAL MOD I,III	89.95
MAXI-MANAGER	89.95
DOS PLUS 3.3	89.00
LDOS	139.00

DISKETTES

ALPHA DISKS® 21.95

SINGLE SIDED, CERTIFIED DOUBLE DENSITY
40 TRACKS, WITH HUB-RING. BOX OF 10.
GUARANTEED ONE FULL YEAR.

VERBATIM

MD 525-01,10,16	26.50
MD 550-01,10,16	44.50
MD 577-01,10,16	34.80
FD 32 OR 34-9000	36.00
FD 32 OR 34-8000	45.60
FD 34-4001	48.60

DYSAN

5¼, SS/DD, BOX OF 5	24.50

DISKETTE STORAGE

5¼" PLASTIC LIBRARY CASE	2.50
8" PLASTIC LIBRARY CASE	3.50
PLASTIC STORAGE BINDER WITH INSERTS	9.95
PROTECTOR 5¼	21.95
PROTECTOR 8	24.95

TRS-80 MOD I HARDWARE

PERCOM DATA SEPARATOR	27.00
PERCOM DOUBLER II	159.00
TANDON 80 TRACK DISK DRIVE	429.00
TANDON 40 TRACK DISK DRIVE	299.00
LNW DOUBLER WITH DOSPLUS 3.3D	159.00
TRS-80 MOD I 32K EXPANSION INTERFACE	355.00

MODEMS

NOVATION CAT ACOUSTIC	145.00
NOVATION D-CAT DIRECT CONNECT	155.00
NOVATION AUTO-CAT AUTO ANSWER	219.00
UDS 103 LP DIRECT CONNECT	175.00
D.C. HAYES STACK Smart Modem (RS-232)	249.00
LEXICON LX-11	109.00

CP/M® SOFTWARE FOR MOD II

MICROSOFT

MICROSOFT BASIC-80	299.00
MICROSOFT BASIC COMPILER	319.00
MICROSOFT FORTRAN-80	399.00
COBOL 80	599.00

MICRO PRO

WORDSTAR	325.00
MAILMERGE (REQUIRES WORDSTAR)	110.00
SPELLSTAR (REQUIRES WORDSTAR)	199.00
SUPERSORT	195.00
DATASTAR	245.00

PEACHTREE

GENERAL LEDGER	595.00
ACCT. RECEIVABLE	595.00
ACCT. PAYABLE	595.00
PAYROLL	595.00
INVENTORY	595.00
PROPERTY MGMT.	799.00
CPA CLIENT WRITE-UP	799.00
MAGIC WAND	299.00

OTHER CP/M® SOFTWARE

SPELLGUARD	239.00
SUPERCALC	229.00
COMMX TERMINAL SOFTWARE	75.00
dBASE II	599.00
PICKLES AND TROUT CP/M® MOD II TRS-80	175.00

SUPPLIES

AVERY TABULABLES

1,000 3½ X 15/16	8.49
3,000 3½ X 15/16	14.95
5,000 3½ X 15/16	19.95

FAN FOLD PAPER (Prices F.O.B. S.P.)

9½ X 11 18lb white 3000 ct.	29.00
14 7/8 X 11 18lb white 3000 ct.	39.00

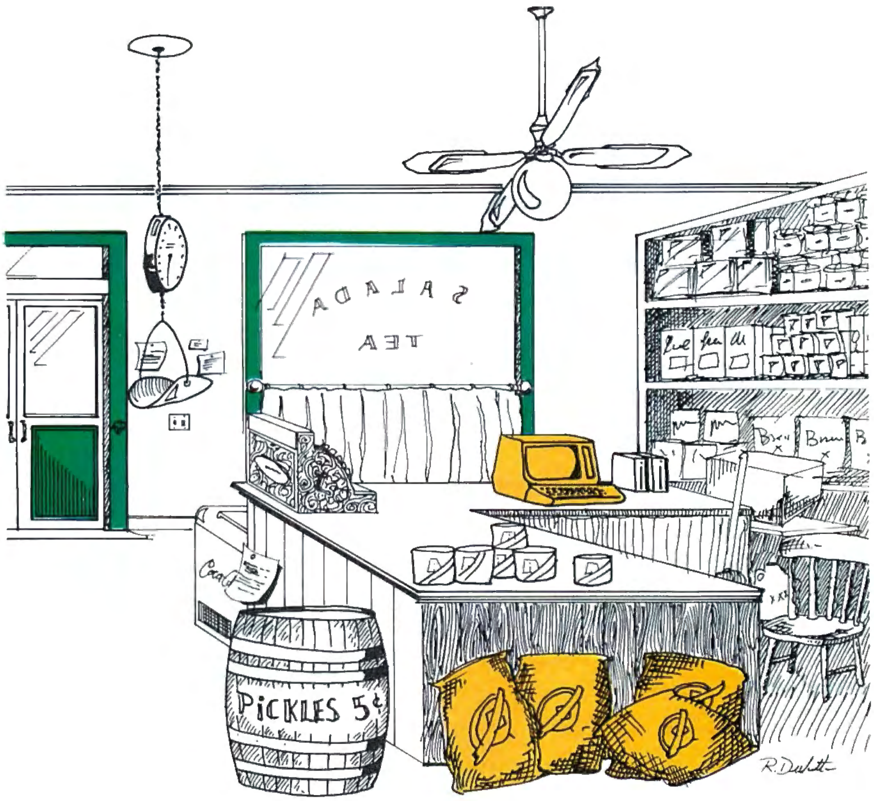
See List of Advertisers on page 402

80 REVIEWS
edited by Michael E. Nadeau

"For getting correspondence out quickly, this mini word processor is simplicity itself."

Speed Letter
12-Column Ledger
Three-Across Mailing Labels
Auto-Dialer and Toll Charge Monitor
Blechman Enterprises
Canuga Park, CA
$10 each
$25 for the package

by Dave Smith



Anyone who has ever operated a small business out of his home understands the difficulty of maintaining a ledger, handling letters, mailing lists, and keeping records of telephone usage and expenses. In response to these needs, numerous computer programs have been designed, which are often too extensive and complicated, not to mention costly. Fred Blechman, of Blechman Enterprises, has developed a package of programs for the home-based businessman which is characterized by effectiveness, simplicity and low price.

Designed for the Model I with 16K and an 80-column printer, the programs do not require disk drives, and are compatible with the Model III following a few simple line adjustments. The instructions for the Model III program changes are available at no charge from the author.

Speed Letter

For getting correspondence out quickly, this mini word processor is simplicity itself. Handling upper and lowercase, the word processing system is directed by a total of 10 single-letter commands. Speed letter can pack 150 lines of text into 16K of memory.

The program operates on lines of text rather than on individual words or characters, but within this single constraint, allows text review, line correction, replacement, insertion and deletion. Text is saved to, and read from cassette, or Exatron's Stringy-Floppy wafer. The left margin and the page length are defined just prior to printing, allowing the user great formatting flexibility. Printed output is also line selectable but vertical spacing can be achieved only by the inclusion of blank lines in the text.

This program's best feature is that it appears to be crashproof, eliminating the worry of lost text through entering an erroneous command. This is a real blessing to the writer who formerly became so involved in the content of his text that he forgot the complicated command structure of his word processor and bombed his text.

12-Column Ledger

Using column one for the dollar amount, and allowing the user to assign the remaining 11 column headings, 12-Column Ledger maintains and prints financial records for income tax and other purposes. All column headings are on display during data entry. The user simply types in date, check number, amount, recipient, purpose and column assignment, and the program prints out all column headings with page, month and grand totals. Characterized once again by ease of operation, this software performs all necessary arithmetic and produces a clear, easy-to-read printout.

Three-Across Mailing List

This program accepts up to 220 names and addresses (in 16K), sorts data according to user preference, and prints addresses on standard adhesive-backed label stock, laid out in three-column format. Each label contains four lines of 24 characters maximum length. While the Mailing List program is excellent for the small business, it should also be attractive to any organization, church, fraternal order or social group which issues a newsletter or other mailings to its membership.

Auto-Dialer

Auto-Dialer uses a $3 telephone inter-

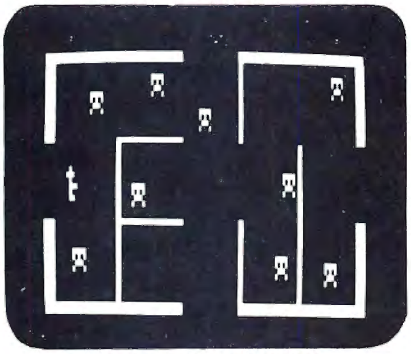
Reader Service for facing page ✓210

WHY IS THE ALPHA JOYSTICK SUCH A SUCCESS ?

A: Sofware support like this:

TALKING ROBOT ATTACK

NEW!



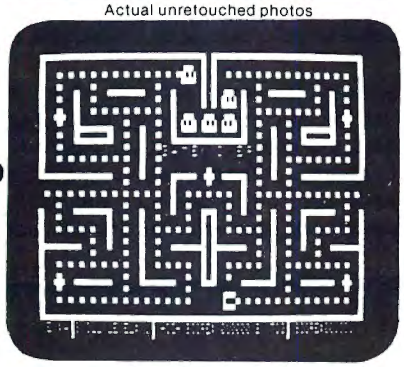
Actual unretouched photos

ALL GAMES:

16K Level 2, Mod 1 + Mod 3 Cassette: $15.95
32k Level 2, Mod 1 + Mod 3 Diskette: $19.95
10% discount for 2 games, 15% for 3 or more
Games may be played with or without joystick.

NEW!

SCARFMAN



TALKING ROBOT ATTACK

INCREDIBLE! This amazing game actually **TALKS** without a speech synthesizer, through the cassette AUX plug.

You are armed with just a hand held laser. In a remote section of the space station you encounter armed robots, some march towards you, some wait around corners. Watch out, the walls are electrified. Zap as many robots as you dare before escaping into a new section where more robots await you. The struggle continues. With Joystick action and **VOICE OUTPUT**, this game will amaze you.

NEW!

SCARFMAN

THE LATEST ARCADE CRAZE now runs on your TRS-80.

It's eat or be eaten. You control Scarfman around the maze, gobbing up everything in your path. You attempt to eat it all before the monsters devour you. Difficulty increases as game progresses. Excellent high speed machine language action game. From The Cornsoft Group. With sound.

CAUTION: Played with the Alpha Joystick, Scarfman may become addictive.

SUPER NOVA®

Asteroids float ominously around the screen. You must destroy the asteroids before they destroy you! (Big asteroids break into little ones.) Your ship will respond to thrust, rotate, hyperspace and fire. Watch out for that saucer with the laser! As reviewed in May 1981 Byte Magazine.

GALAXY INVASION®

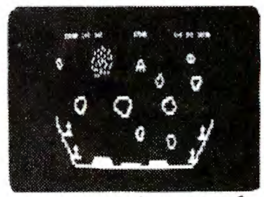
The sound of the klaxon is calling you! Invaders have been spotted warping toward Earth. You shift right and left as you fire your lasers. A few break formation and fly straight at you! You place your finger on the fire button knowing that this shot must connect! With sound effects!

ATTACK FORCE®

As your ship appears on the bottom of the maze, eight alien ships appear on the top, all traveling directly at you! You move toward them and fire missiles. But the more aliens you destroy, the faster the remaining ones become. If you get too good you must endure the "Flag-ship"... With sound effects!

COSMIC FIGHTER®

Your ship comes out of hyperspace under a convoy of aliens. You destroy every one. But another set appears. These seem more intelligent. You eliminate them, too. Your fuel supply is diminishing. You must destroy two more sets before you can dock. The space station is now on your scanner... With sound!



METEOR MISSION II®

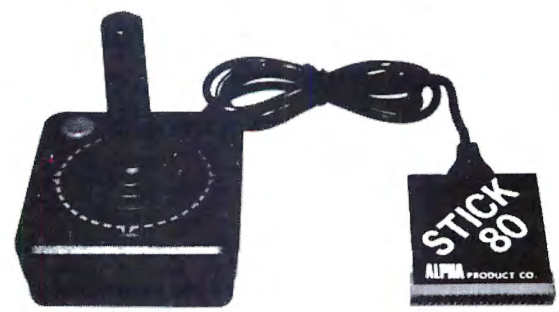
As you look down on your view, astronauts cry out for rescue. You must maneuver through the asteroids & meteors. (Can you get back to the space station?) Fire lasers to destroy the asteroids, but watch out, there could be an alien FLAGSHIP lurking. Includes sound effects!

THE ALPHA JOYSTICK: REAL ARCADE ACTION

ONLY $39.95

- Features the famous ATARI JOYSTICK
- Works with any Level II or disk system Model I
- Plugs directly into KB or E/I (next to printer port)
- Compatible with any other TRS-80 accessories
- Saves your keyboard
- Fun to experiment in BASIC. Simply use A = INP(0) to read the joystick
- Fully assembled, ready to plug in and use.

Price includes Atari Joystick + Alpha Interface + instructions + demo program. The Alpha Joystick is backed by an unconditional money back guarantee. If you are not delighted with it, return it within 14 days for a prompt and courteous refund.



Toll Free Order Line

(800) 221-0916

ADD $1.50 PER ORDER FOR SHIPPING AND HANDLING
ALL ORDERS SHIPPED FIRST CLASS MAIL
WE ACCEPT VISA, MASTER CHARGE, CHECKS, M.O.,
COD: ADD $2.00 EXTRA
QUANTITY DISCOUNTS AVAILABLE
N.Y. RESIDENTS ADD SALES TAX

ALPHA Products ✓210

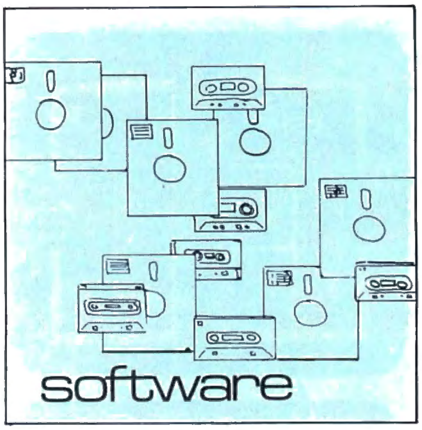
85-71 79th Street, Woodhaven, N.Y. 11421

(212) 296-5916

face relay to dial the number of any 500 names held in memory, upon keyboard entry of that name. The user enters each name and phone number one time. Subsequently, the program lists the names on the screen. The program matches the correct number to the name entered, and dials the number by toggling the cassette output from the computer. This signal is then processed by the interface relay and transmitted to the phone company's switching equipment.

The purchase price of the program includes a schematic diagram for assembling the interface relay and connection to the telephone line. The five relay components are listed with their Radio Shack part numbers for easy acquisition. Assembly of the interface relay should be a simple matter for most anyone.

This program will be of benefit to anyone whose business requires a lot of telephoning, especially when the user needs to know the cost of the calls. Once the user has entered the current schedule of telephone charges (obtainable from the initial pages of the phone book or from the phone company's business office), the program will time each call and calculate its cost. The duration and charges for the call are continuously displayed while the call is in progress.



An adjunct to the Auto-Dialer is the Phone Toll Charge program, which is included in the package. Very similar to the Auto-Dialer time and charge monitor, this program is intended for use with one-time, long distance calls.

Each of the four programs is available separately, but the combined price of one package makes it an exceptional buy. The accompanying documentation, while not super slick in format, is more than adequate in content, and all programs are self-documenting on the screen. All are simple to use.

For the small businessman, the person merchandising out of his or her home, the club secretary or any person required to do a lot of telephoning, this set of programs must be regarded as packaged convenience at a very low price.■

Crash Course in Microcomputers
Louis E. Frenzel, Jr.
Howard W. Sams & Co., Inc.
Indianapolis, IN
Softcover, 264 pp.
$17.50

by Debra Marshall
***80 Microcomputing* staff**

This book is a self-teaching course in the basics of microcomputers. It is designed to provide the newcomer with a solid basic knowledge about the construction, internal circuitry, and operation of micros so that the reader may proceed painlessly to more complicated and less generalized material.

The chapters are fairly short. Each chapter covers a very specific part of the microprocessor, such as memories, binary data, I/O operations, peripheral equipment and programming. Each topic is covered in a set of short explanatory sections, which are further divided into single concepts which are eventually assimilated into the whole picture of the topic at hand. Every paragraph is followed by a question designed to reinforce the key word or concept presented in the paragraph, and each chapter ends with a short quiz which again covers the concepts and vocabulary presented.

The chapters are built on one another; in other words, the concepts in previous chapters must first be fully assimilated in order to appreciate the information in subsequent chapters. I found this progressive method to be very effective; while new information is constantly presented, the old information is concurrently reinforced.

Although the questions asked as reinforcement following each chapter section often seem painfully obvious, the method does seem to work. By simply reading the chapter on binary data, I learned more about microcomputer number systems than I ever hoped to know. In fact, I learned enough to believe Dennis Kitsz when he says that machine language is a piece of cake.

"Although the questions asked as reinforcement . . . often seem painfully obvious, the method does seem to work."

The book assumes you have literally no prior knowledge of microcomputers. The first chapter explains what a microcomputer is, what it does, what it is made of, and presents basic terms such as microprocessor, data, memory, digital timer, and monitor. Explanations are accompanied by nicely executed illustrations and photos.

You do not need an electronics, math or technical background to use and appreciate this book. At the same time, it was aimed at an audience which would include professionals who wish to learn about the microcomputer field, and so the tone of the book is businesslike, to the point, and highly readable. Its physical make-up is also nicely presented: Type is well set, it is nicely laid out, and has been carefully edited. Figures are simple and well-documented.

In spite of its simplicity, the book speaks to readers who intend to pursue microcomputers further. The last three chapters cover the basics of machine language and Basic programming. Two appendices, covering the 8085 instruction set and the ASCII code, are also included.

The book meets its goal: Anyone who reads it has to come away with at least a working knowledge of the microcomputer, and one that will leave the reader well-versed in the lingo of the trade. This book would be an incredible boon to any business office which has just had one of those totally mysterious and incomprehensible machines foisted on it, as well as to anyone who wants to demystify the machine which is steadily making inroads into the most common areas of society. The book is well worth its price, and well worth the time it takes to read it.■

THE ALPHA I/O SYSTEM

a complete failure?

THE INSIDE STORY

It happened 3 years ago, when our President made a decision. At the time we specialized in custom analog and digital circuit design. The decision was to attempt to develop a line of standard interface hardware for the emerging microcomputers. At the time (1977) we had to decide which of the new machines could become the "industry standard" of the low cost micros.

Despite a few aggravating but minor deficiencies, the TRS-80 seemed tohave the most chance of success and it had the best price/performance ratio. Also, with some imagination their large sales organization could become the largest service network in the world, a reassuring throught for the many novices in this new field.

It became clear that the TRS-80 could be used (with our then hypothetical system) to solve problems in many fields where computers were not yet used, mostly because of their high cost.

The IDEA was simple! ALPHA PRODUCT would supply the missing link between the TRS-80 and the "outside world", (more about this "outside world" later).

Early Survival

DANGER! If Radio-Shack entered the same market, we probably would not have survived, but the expectation was that they would be too busy developing their basic line (drives, printers, modem etc.). Thanks to our more specialized products, we would not be competing with them.

BAD START! We began with a failure. Our first product was supposed to be a simple, low cost, general purpose device. It would allow the TRS-80 to accept inputs other than the keyboard. Many kinds of external devices (the "outside world" mentioned before) like photocells, sensors, thermostats, switches, contacts, etc., could be connected easily. In addition, there were two relays to control (on or off) external loads such as motors, lamps, appliances, heaters, etc., etc.. In other words, it would allow the computer to interact or interface with external devices. We called it the INTERFACER 2. What a mistake! It sounded too much like "expansion interface". Many enthusiastic TRS-80 users called thinking that our "INTERFACER 2" was a low cost Expansion Interface (at $85 that would have been a real bargain!) We wanted to change the confusing name. That meant reprinting the manual, changing the ad, scrapping the flyers, discarding the silk screened cases. Well, "INTERFACER 2" it would stay.

TROUBLE! We also found that the majority of TRS-80 users were AFRAID of the hardware. They could be very comfortable with fancy programming but thought you had to be a computer specialist or technically inclined to put the INTERFACER 2 to work. In truth, some IMAGINATION and a SCREWDRIVER is all you really need. Anyone able to wire a switch could use this device.

WORSE! There was also the fear of plugging a "foreign device" into the precious computer. This notion has all but disappeared as there are now so many quality products designed for the TRS-80 that plugging in a non Radio-Shack device has become common.

Our ad in Creative Computing (80-Microcomputing did not yet exist) hardly paid for itself.

We had a decision to make. Were we wrong or just too early? Our first INTERFACER 2 was sold to someone who wanted to, and succeeded in, controlling his fancy model railroad with his TRS-80. Interesting, but what made us stick with the concept was that some of our INTERFACERS began finding use in applications with fascinating possibilities. Space is lacking to describe them, but the most exciting was the successful use of the system in assisting a handicapped young boy. We were pleased to hear of such a meaningful application.

Today

Three years later, as you can see in our ads, The INTERFACER 2 is alive and well. The price went up a bit, and despite the introduction of the more powerful INTERFACER 80, the sales have been steady.

Then came the least understood product! the ANALOG 80. This $139, nicely designed module is an Analog to Digital converter with 8 input channels. Used with your TRS-80, it provides a powerful "data acquisition system". This jargon simply means that you can monitor, measure and record 8 independant varying voltages. Very few people realized its real power. Such a system would have cost over ten thousand dollars just a few years ago.

The possibilities in scientific and engineering environments are endless. This system could replace chart recorders, digital data recorders, programmable calculators, data analyzers and many other specialized and expensive pieces of equipment. Furthermore, up to 8 ANALOG 80's could be used simultaneously for a total of 64 channels of analog input! They simply plug into the TRS-80 using our "X" series of bus extenders (EXPANDABUS).

The idea was simple. We would supply the missing link between the TRS-80 and the "outside world".........

Our next product was to be a second generation, Input/Output interface, with more flexibility than the INTERFACER 2. Careful design and refinement yielded the INTERFACER 80, the most powerful real world interface on the market today. It has 8 inputs, each optically-isolated and 8 outputs, each with a relay contact. The INTERFACER 80 is fully compatible with our ANALOG 80, allowing these to be used together in order to create systems that control external devices based on "sensed" input under control of the TRS-80.

A FAILURE! in spite of our extensive advertising, very few are aware of the existence of the powerful ALPHA I/O SYSTEM.

The Facts Are:

- The ALPHA SYSTEM/TRS-80 combination forms an incredibly versatile and powerful tool for acquisition/processing/control.
- In spite of its moderate cost, the system is sophisticated and reliable.
- The entire system can be easily programmed in BASIC using INP(X) and OUT X,Y commands.
- The modular approach and our EXPANDABUS allow for instant expansion as requirements demand.

The following pages contain more information about the devices mentioned here. We invite you to call or write to discuss your particular application.

Device descriptions; NEXT PAGE

TIMEDATE 80

WHY LOSE PRECIOUS TIME ?

Neat, Compact Design 3 Years Battery Life | Slips Inside E/I (Y Option Shown) | Real Time Without Expansion Interface

- Complete, self contained "true" real time clock/calendar, TIMEDATE 80 continues to keep accurate time and date when the computer is turned off or experiences a power failure.
- **TIMEDATE 80** only needs to be set once, and it's two replaceable "AAA" batteries (not included) keep TIMEDATE 80 running in excess of 3 years. Costly Ni-Cad batteries and charging circuits are eliminated.
- The instant power is applied to the TRS-80, TIMEDATE 80 provides MO/DATE/YR, DAY of WEEK, HR:MIN:SEC and AM/PM information with quartz accuracy.
- **TIMEDATE 80** replaces the computer's internal clock. Extremely useful for automatic operation of remote systems with no operator in attendance. If the power fails and then is restored, only TIMEDATE 80 will update the system with current TIME and DATE information, an impossibility with the computer's internal clock.
- **TIMEDATE 80** is quartz crystal based with INTELLIGENT CALENDAR, including provisions for leap year! TIME display may be by 12 hour AM/PM or by 24 hour military and Eruopean format.
- **TIMEDATE 80** plugs directly into the rear of the TRS-80 keyboard and gives the "TIME$" function even without an Expansion Interface. For those with a disk system, it plugs into the left side panel of the Expansion Interface. An optional "Y" connector can provide for further expansion.
- **TIMEDATE 80's** small size keeps the computer table uncluttered. If you have an Expansion Interface, TIMEDATE 80 literally "DISAPPEARS" by slipping into the empty space in the bottom of the interface.
- Two sets of software, on cassette, come with TIMEDATE 80—"TIMESET" and "TIME$". "TIMESET" is a step by step set of simple instructions for setting TIMEDATE 80. "TIME$" is a set of poke routines which patch DOS and Level II TIME$ to read TIMEDATE 80 and is easily incorporated into any user software. "TIME$" will always print the time and date when LISTING a program—great for keeping track of revisions!
- Other valuable uses for TIMEDATE 80 are: accurate date and time information for business reports like payroll records, financial reports, etc., or to various I/O devices requiring 24 hour clock input, such as laboratory instrumentation, and to communication systems needing "Log In/Log Out" data (bulletin boards).
- **TIMEDATE 80**, fully assembled and tested. 90 day warranty, complete with instructions and software on cassette. $95.00 "Y" option, add $12.00

NEW: Computer to Computer ordering:(212) 441-3755 (24 hr. data line)

CALL
(212) 296-5916
FOR FAST SERVICE

ALPHA Products

85-71 79th Street, Woodhaven, NY 11421

ADD $2.50 PER ORDER FOR SHIPPING AND HANDLING
ALL ORDERS SHIPPED FIRST CLASS MAIL
WE ACCEPT VISA, MASTER CHARGE, CHECKS, M.O.
COD. ADD $2.00 EXTRA
QUANTITY DISCOUNTS AVAILABLE
N.Y. RESIDENTS ADD SALES TAX

✓210

Structured Basic and Beyond
Wayne Amesbury
Computer Science Press Inc.
Softcover, 310 pp.
$10.95

by John W. Liskey

How many times have program writers—novice and experienced alike—been admonished to design a program before committing a single statement or instruction to paper? In spite of this, how many succumb to that urge to reach for the coding form and gleefully write code off the top of their heads? And how often does this lead to gibberish, requiring hours of toil to debug?

In the early days (a couple of years ago), it may have been practical to make a program an individual work of art. Devoted hobbyists could afford to spend hours getting a tortured jumble of statements to execute properly. This is no longer a luxury business can afford; nor does it make any sense with the development of the structured approach to program design.

In *Structured Basic and Beyond,* Dr. Wayne Amesbury presents an approach making it possible for students to grasp the concept of designing a program before going anywhere near a keyboard. Professionals will find this book a new way of life if they have never seriously looked into structured design before.

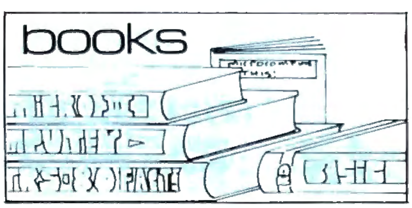
Dr. Amesbury presents pseudo-code as the primary designing aid. At no time is the discussion bound by any one version of Basic. The underlying principle of the book is to avoid imposing a particular version on the student. For this reason, there is a philosophical basis throughout the book completely independent of any language. From this book, anyone can progress to an understanding of Cobol, Fortran, PL/1, Pascal or Algol without difficulty. This book is not written only for the person who wants to understand the logic and process of a program. It causes one to understand structured design by simply following and trying the beautifully clear explanations. If any point is emphasized, it is the desirability of designing programs that are portable from one system and language to another.

While the theme of this book is structured design, you should not jump to the conclusion it might be vague on the practical applications of Basic. Dr. Amesbury follows each pseudo-code demonstration with examples showing the relationship between pseudo-code and Basic statements. He has also gone to great lengths to point out differences between the many versions of Basic, often illustrating how the same problem may be solved in different versions. The key element to solving these problems is understanding the logic, so when a different version is encountered, you can easily make the conversion and be on your way.

Today we are seeing the passing of a mini-era. Some programmers will view this with sadness, for they have traditionally winged it, writing line after program line, trying to figure out their program after trying to run it. The person emerging today in professional program design and development is one who never goes near a computer until his or her program has been completely designed and walked through; one who makes certain anyone else can follow his or her program design; one who is sold on the structured design concept.

If you enjoy debugging a convoluted program, this book could ruin your life. But if you want to write professional programs, it will open your eyes to a philosophy and approach of incalculable value and benefit. ■

> *". . . if you want to write professional programs, (this book) will open your eyes to a philosophy . . . of incalculable value."*



Introduction to PASCAL
Including UCSD PASCAL
Rodney Zaks
Sybex Inc.
Softcover, 422 pp.
$8.95

by Dennis Thurlow

While relaxing at the local inn I chanced to hear two fellow students speaking. One explained to the other the simplicity of learning programming with the help of Pascal. Being quick witted, and knowing the pay scale in the computer field, I decided to find out who this Pascal was.

I bought a copy of *Introduction to Pascal* by Rodney Zaks, a "tutorial and reference text," according to the author. By confining myself to a comfortable chair and following the advice given in "How to Read This Book," I was able to complete the introductory chapter with ease. The exercises seemed simple enough, so I continued, with only occasional help from a dictionary, through chapter six.

My understanding of computers, programming, Pascal, and the potential of the field were sufficient, at this point, for me to keep on reading without any problem.

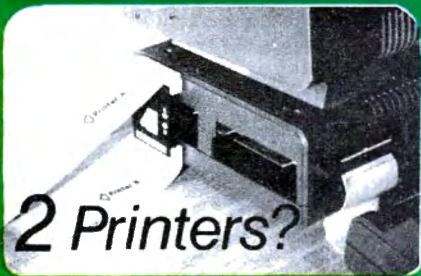
Further excursions into the book taught me about data types, files, arrays, and sets. (I still keep the book close by, as the appendices, syntax charts, and examples make it an excellent reference manual.)

Before completing my studies I learned of a new, interactive version of my old friend, called UCSD Pascal. I could now talk directly through a keyboard and get my answers immediately! The book was again helpful, with separate discussions in each chapter on implementing the material in the UCSD version, and an appendix on handling UCSD interactive files.

Sample answers to the programming exercises are given, the publisher is dedicated to updating any errors, the book is well indexed, and though the cover may only shine like silver, the contents glitter like gold. ■

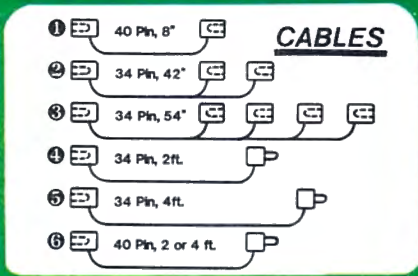
NOW MODEL I AND MODEL III!

Now Model III users can take advantage of the ALPHA I/O system too. Our new MOD III/I BUS CONVERTER allows most port based Model I accessories (such as our ANALOG-80, INTERFACER 2 and INTERFACER-80) to connect to the Model III bus. MOD III/I BUS CONVERTER, complete with all connectors, only $39.95.



PRINTSWITCH

Have 2 printers on line at all times and select printer 1 or 2 by means of a conveniently located switch. End the problem of constantly plugging and unplugging printer cables. PRINTSWITCH is a compact module that plugs onto the parellel printer port of your TRS-80 and provides an edge connector for each of your two printers. It works with any two types of printers: dot matrix, daisy wheel, plotters, TRS-80 converted selectrics, etc. Assembled, tested, ready to use with connector and instructions. For Model I or III (please specify). ONLY . . $59.00

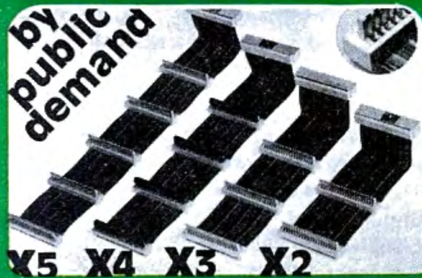


SUPERIOR QUALITY REPLACEMENT & EXTENSION CABLES

Highest quality cable and high force, gold plated contacts ensure the utmost in connection reliability.

❶ KEYBOARD TO EXPANSION INTERFACE . . . . $21.
❷ DISK DRIVE CABLE FOR 1 OR 2 DRIVES . . . . . . $32.
❸ DISK DRIVE CABLE FOR 3 OR 4 DRIVES . . . . . . $45.
❹ DISK DRIVE CABLE EXTENDER . . . . $22.
❺ PRINTER CABLE EXTENDER . . . . $24.
❻ 40 PIN BUS EXTENDER — 2 ft. . . $22 4 ft. . . $24.

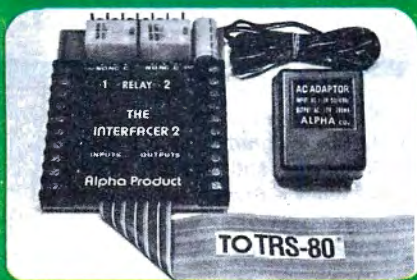
Custom cable configurations are also available. Call us.



YOU ASKED FOR IT: "EXPANDABUS" X1, X2, X3 AND X4.
CONNECT ALL YOUR TRS-80 DEVICES SIMULTANEOUSLY on the 40 pin TRS-80 bus. Any device that normally plugs into the keyboard edge connector will also plug into the "EXPANDABUS". The "X4" is shown with protective covers (included). The TRS-80 keyboard contains the bus drivers (74LS367) for up to 20 devices, more than you will ever need. Using the E/I, it plugs either between KB and E/I or in the Screen Printer port. Professional quality, gold plated contacts. Computer grade 40 conductor ribbon cable.
X2 . . . $29. X3 . . . $44. X4 . . . $59. X5 . . . $74.
Custom configurations are also available, call us.

ANALOG-80: A WORLD OF NEW APPLICATIONS POSSIBLE.

8 DIGITAL MULTIMETERS PLUGGED INTO YOUR TRS-80!!!
Measure Temperature, Voltage, Current, Light, Pressure, etc. Very easy to use: for example, let's read input channel #4: 10 OUT 0,4 'Selects input #4 and also starts the conversion 20 A = INP(0) 'Puts the result in variable "A" Voila! Specifications: Input range: 0-5V. to 0-500V. Each channel can be set to a different scale.
Resolution: 20mV (on 5V range). Accuracy: 8 bits (.5%). Port Address: jumper selectable. Plugs into keyboard bus or E/I (screen printer port). Assembled and tested. 90 day warranty. Complete with power supply, connector, manual . . . . $139.



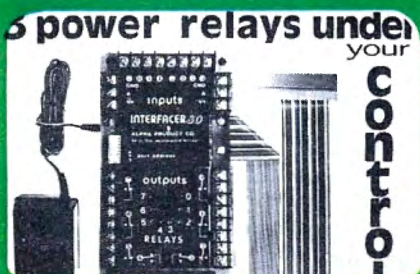
INTERFACER 2: LOW COST INPUT/OUTPUT MODULE.
Still the best value in sense/control devices. Use it for energy control, burglar alarm, darkroom, selectric drive, model trains, robots, Skinner box . . .
—8 latched TTL outputs. 2 relays SPDT 2A, 125V. contacts
—8 TTL/CMOS inputs. Input 0 and 1 are optically isolated.
—Neat and compact design, very easy to use:
10 A = INP(0) 'Reads the 8 inputs (if A = 0: all inputs are low) 20 OUT 0,X 'Controls the outputs and the relays
Assembled & tested, 90 day warranty. Price includes power supply, cable to KB or E/I, superb user's manual, free phone dialer program. $95. Manual only $5.

DISK DRIVE EXTENDER CABLE, FREE YOUR MINI-DRIVES.

End the daisy-chain mess once and for all. Fits all mini-drives: Percom, Aerocomp, Shugart, Micropolis, MTI, Vista, Pertec, Siemens, BASF. East to install: just remove the drive cover, plug in the EXTENDER CABLE and replace the cover.

Now you can change and move your drives without disassembly. Keep the cover on and the dust out. High reliability gold plated contacts, computer grade 34 conductor cable. Tested and guaranteed.

Get one for each drive . . . ONLY $8.95



INTERFACER-80: the most powerful Sense/Control module.
- 8 industrial grade relays, single pole double throw isolated contacts: 2 Amp. @ 125 Volts. TTL latched outputs are also accessible to drive external solid state relays.
- 8 convenient LEDs constantly display the relay states. Simple "OUT" commands (in basic) control the 8 relays.
- 8 optically-isolated inputs for easy direct interfacing to external switches, photocells, keypads, sensors, etc. Simple "INP" commands read the status of the 8 inputs.

Selectable port address. Clean, compact enclosed design. Assembled, tested, 90 days warranty. Price includes power supply, cable, connector, superb user's manual . . . $159

GREEN SCREEN WARNING

IBM and all the "biggies" are using green screen monitors. Its advantages are now widely advertised. We feel that every TRS-80 user should enjoy the benefits it provides. But **WARNING**: all Green Screens are not created equal. Here is what we found:
- Several are just a flat piece of standard colored Lucite. The green tint was not made for this purpose and is judged by many to be too dark. Increasing the brightness control will result in a fuzzy display.
- Some are simply a piece of thin plastic film taped onto a cardboard frame. The color is satisfactory but the wobbly film gives it a poor appearance.
- One "optical filter" is in fact plain acrylic sheeting.
- False claim: A few pretend to "reduce glare". In fact, their flat and shiny surfaces (both film and Lucite type) ADD their own reflections to the screen.
- A few laughs: One ad claims to "reduce screen contrast". Sorry gentleman but it's just the opposite. One of the Green Screen's major benefits is to increase the contrast between the text and the background.
- Drawbacks. Most are using adhesive strips to fasten their screen to the monitor. This method makes it awkward to remove for necessary periodical cleaning. All (except ours) are flat. Light pens will not work reliably because of the big gap between the screen and the tube.

Many companies have been manufacturing video filters for years. We are not the first (some think they are), but we have done our homework and we think we manufacture the best Green Screen. Here is why:
- It fits right onto the picture tube like a skin because it is the only **CURVED** screen **MOLDED** exactly to the picture tube curvature. It is Cut precisely to cover the exposed area of the picture tube. The fit is sucn that the static electricity is sufficient to keep it in place! We also include some invisible reusable tape for a more secure fastening.
- The filter material that we use is just right, not too dark nor too light. The result is a really eye pleasing display.

We are so sure that you will never take your Green screen off that we offer an unconditional money-back guaranty: try our Green Screen for 14 days. If for any reason you are not delighted with it, return it for a prompt refund.

A last word: We think that companies, like ours, who are selling mainly by mail should •list their street address•have a phone number (for questions and orders)•accept CODs, not every one likes to send checks to a PO box•offer the convenicnce of charging their purchase to major credit cards. How come we are the only green screen people doing it?

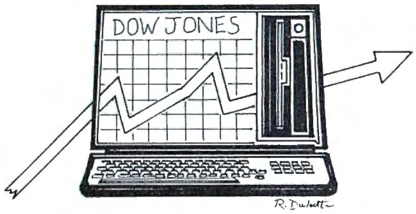
Order your **ALPHA GREEN SCREEN** today . . . $12.50

ALPHA Products

✓210

ADD $2.50 PER ORDER FOR SHIPPING AND HANDLING
ALL ORDERS SHIPPED FIRST CLASS MAIL.
WE ACCEPT VISA, MASTER CHARGE, CHECKS, M.O.
COD: ADD $2.00 EXTRA
QUANTITY DISCOUNTS AVAILABLE
N.Y. RESIDENTS ADD SALES TAX

85-71 79 St. WOODHAVEN N.Y. 11421 ☎ Info and order: (212) 296•5916



New Uses for the Home Computer in the Stock Market
Thomas V. Lenz
Spring Creek
Elko, Nevada 89801
Softcover, 254 pp.
$19.95

by Dave Smith

Are you looking to improve your score in the stock market? Would you like to have an edge over the experts on Wall Street? Thomas Lenz has provided the means for you to do so in this fascinating collection of 34 Level I programs for the TRS-80.

Possessing a degree in business administration and more than 20 years experience as a securities broker, the author has developed these programs over the course of seven years, borrowing from the field of physics the methodology of hybrid analysis and applying it to the field of stock market timing.

The Truth Behind Myths

The book explores the analysis of both short (seasonal) and long-term cycles in the market, using data which has been collected over the course of 96 years and which are available to everyone. Using the personal computer as a sophisticated tool, Lenz demonstrates the truth behind numerous market myths and shows you how to quantify business cycles that are unknown to many expert market analysts.

The book presents the 34 programs with a textual explanation of each. The programs are progressive, with each succeeding unit built upon previous ones. Typical programs run average, percentage series, statistical oscillograph, strong cycle test, random trading test, upper limit of earnings, statistical resonator (for locating cycles), statistical wave filter and extensive statistical analysis. The text accompanying each program is clear and well presented. Lenz details the creation, function and significance of each program as it is presented. A glossary provides clarification of unfamiliar market and mathematical terms.

The programs which are central to the purpose of the book let you create, edit and duplicate data files. Computer sophisticates will deplore the use of the lowly cassette tape as the data file storage medium, but Lenz makes no apologies for its use. Since all programs are designed to operate within 16K of memory (and most within 4K), the use of a cassette is appropriate.

From Level I to Level II

The programs are printed in large, easy-to-read type, making them easier to type in. Two problems arise, however, concerning the format of the programs themselves. The first is that they are all written in Level I Basic. The best estimate I can find says that only between three and five percent of the TRS-80s sold operate on Level I. That leaves several hundred thousand potential users of these programs who will be required to translate the programs into Level II.

The author in response to this problem included in the book, a Level I to Level II translation guide which is where the second problem arises. The guide is seriously deficient in dealing with Level II array definitions (no mention of the DIM statement requirement) and does not comment on the translation of logical operators at all. Only those benighted individuals who, like me, moved to Level II after mastering Level I will recall that in Level I "*" and "+" are the equivalent Level II operators And and Or. Level II users who are not programmers are likely to be severely disappointed to find, after translating and entering some of the more extensive programs, that the programs either do not work or worse, that they provide erroneous analysis.

Translation problems notwithstanding, Lenz has certainly provided in this book all the justification needed for purchasing a personal computer. Whether you are a stock market amateur seeking to broaden your knowledge or a daily trader on Wall Street, you can only enlarge your understanding of business cycle analysis through reading and applying this book. ■

Data File Programming in Basic
LeRoy Finkel and Jerald R. Brown
John Wiley & Sons
New York, NY
Softcover, 338 pp.
$9.95

by Hal Knippenberg

Have you ever wished someone would write a book about Basic offering more than a simple introduction to the language—a book for those of you already writing simple programs that would teach you more advanced programming tricks?

Two well-known educational consultants on computers, LeRoy Finkel and Jerald R. Brown, have done just that. Their new book, *Data File Programming in Basic*, is a self-teaching guide for using data files with your microcomputer. This book explains data files used in TRS-80 Microsoft and Northstar Basic. And the explanations are simple enough to be applied to other versions of Basic with little effort.

Although not written for the novice, this publication begins with a review of key Basic programming statements. Because the authors believe programs should be readable by computers *and* people, they show several ways to use "prettyprint" to improve program clarity. They believe in organizing programs from top to bottom and using many remark statements to explain program logic and make it into modules. Their review of Basic statements is notable because it concentrates on string and substring statements. Strings are seldom covered thoroughly in introductory books on the subject, but are essential to the effective use of files.

After their excellent Basic review, Finkel and Brown focus on writing data file programs. They show how to build data entry and error-checking routines—especially helpful information considering most program errors occur at data entry time.

"Sequential files are ideal for cassettes."

Once error-free data entry is mastered, the authors show how to create and read back sequential data files. From this point on, you can write programs using more data than your computer has memory. Your disks or cassettes become a working extension of your computer, not just a means for storing programs.

The skills mastered working with sequential files are used to help you build utility programs. You learn how to write programs for copying files and for adding, changing or deleting file data. You even learn to combine data from more than one file.

Many of you do not have disks, but are able to put data on cassettes. Sequential files are ideal for cassettes, and chapter six is devoted entirely to cassette data files.

The book completes your data file education with a thorough discussion of random access files. You learn how to create, verify, copy and change random access disk data files. You also learn to write programs that will convert sequential files to random access files.

The final chapter, "Random Access File Applications," shows several ways to use your new-found skills. One very useful technique utilizes sequential pointer data files to index large random access files. By the time you master this technique, you will be writing quite sophisticated data file programs.

This is the first book I have read that shows how to write effective data file programs. After studying this publication, you should be able to write these programs with ease, modify any commercial programs, and adapt data file programs found in computing magazines. In short, I think *Data File Programming in Basic* is an excellent book and well worth your study.■

The Small Computer in Small Business
A Guide to Selection and Use
Brian R. Smith
Hardcover, 144 pp.
The Stephen Green Press
Brattleboro, VT
$12.50

by Bert Latamore
***80 Microcomputing* staff**

Author Brian Smith promises that his book, "requires no prior understanding of computers or computer terminology. It is written for those who want to know how a small computer can help them run a business successfully."

Smith fulfills this promise and has written a book belongs on every small businessperson's reading list.

Smith focuses completely on the issue at hand. Except for one excursion into a brief explanation of computer electronics, he avoids talking about technical subjects that are not directly involved in the use of computers in small businesses. When he does use a technical term it is one that cannot be avoided and he carefully explains its meaning and significance to business.

He educates the reader to all those computer issues and terms needed so that the business person can talk with computer store personnel on an equal footing. Smith also provides a glossary of technical terms in the back of the book for handy reference.

All of this is relayed in a solid journalistic writing style which is unfortunately rare in computer literature. As a result, the book is extremely easy to read.

Definitions

Smith starts his book logically with a discussion of what a computer is and what it is not. He points out at the start that the computer "is nothing more than a very fast, very dumb, adding machine."

He also points out that the computer almost never makes a mistake, that the mistakes are almost always made by the programmers and operators. This point illustrates one of the basic issues he takes up later in the book; the businessperson has to know exactly what he wants to accomplish with a computer.

Smith spends his second chapter giving the reader a look at how the computer works. He explains base 2 math and why it is used in computers, and he discusses OR, AND, NOR and NAND gates, half adders and the like.

This is the one place where Smith really does get more technical than he probably needs to be. The more technical aspects of this chapter, particularly the wiring discussion, would probably have been better placed in an appendix or left out altogether. However, since he never refers to the circuitry again in his book, the reader does not really have to understand this bit of wiring technology he discusses.

Do not skip the whole chapter, however. He does explain what a central processing unit is, how the size of its main memory is measured and what that means to the business user. He discusses input and output devices, floppy and hard disks, tape storage and the usefulness of each.

Once through these basics, Smith defines data processing and discusses how this is accomplished with the computer. He works with the simple example of a payroll clerk equipped with a calculator and typewriter and discusses exactly what data the clerk will need and how to calculate it to determine employees' pay. Smith then diagrams the steps in a simple flowchart. He gives a chart showing exactly how the computer would handle the same task from the reading of time-card data to the printing out of the checks.

Machine Capacity

Smith then introduces one of the important themes in his book: machine capacity, which he returns to several times throughout. He states that one of the most common errors is to underestimate a small business' computer needs. He strongly advises the businessperson first study the business carefully to determine exactly how much data the machine needs to handle. Then Smith advises him to buy a machine that has at least twice that capacity or one which can be easily expanded to allow for growth of the business.

He points out that a computer will represent a large up-front expense for a small business and it will only pay for itself over time. If the business outgrows a machine in a year, that machine has not paid its way.

Software

In chapter 3 Smith starts with the basics—what software is and why it is needed. Smith again avoids confusing the beginner by presuming the reader has no knowledge of a microcomputer and its operation.

Smith goes on to discuss operating systems, particularly CP/M, compilers and interpreters. He also briefly discusses Fortran, Basic, Cobol, Pascal, PL/1 and C. These discussions are extremely brief—he devotes one paragraph to each of the last three. He makes no attempt at teaching the reader to program. He points out that there are many good books on the subject already available. Beyond that, he recommends that the businessperson avoid learning to program, and he warns that a businessperson who spends his days figuring more elegant ways to have his machine perform mathematical calculations is not spending his time on the computers doing what he should be doing—running his business. The computer can become an expensive and very distracting toy.

Chapter 4, "The Coming of the Computer in Small Business," is a summary, and represents a study done by Bob Vitelli, a former student of Smith's at Franklin Pierce College in Rindge, NH. The chapter presents the rational for the book which, Smith says, was inspired by Vitalli's survey of the attitudes of small business owners towards computers.

"The entire Vitelli study pointed out the need for knowledge and information. In many cases, either the information that business owners had about computers was insufficient, was misleading, or both. This book was written to dispel some of the myths that the study found business owners have about computers, and to provide some basic knowledge so that intelligent questions could be posed by business owners before a decision on a computer is made," Smith wrote.

In chapter 5, "Applications," Smith discusses which application programs are available and what to do if you need one that is not.

Smith starts by pointing out that while too often the businessman interested in computerization buys a computer, then buys software, and finally tries "to shoehorn both the hardware and the software to fit the business," the sequence of events should be just the opposite.

Smith says the first step should be to "define what it is within the business that needs automating." Then find software that will handle the jobs you have and finally find a machine that the software will run on.

Therefore, Smith starts the chapter on software with a discussion of those areas of business that are susceptible to computerization. He concentrates on the basic financial areas of receipts and disbursements, accounts receivable and billing, accounts payable, inventory control, budgeting, customer mailing lists and payroll.

Smith goes on to explain exactly how to discover how large a file each will need, using the simple example of an employee file for a firm with 15 employees. This is very important since the software and machine the company buys must be able to handle various size files.

He then defines the four sources of software: turnkey systems, packaged programs, learning-by-doing with kits, and custom written material. He explains the importance of error messages, warranties and program maintenance contracts.

Having examined the relative merits of all these approaches, Smith proceeds to a detailed analysis of the basic business functions. He explains the correct procedures involved for each using flowcharts and sample ledger pages, and examines the ways that the computer approaches each job. He explains what the effect of computerization will have on the accountant's work load and gives some idea of what amounts of money the average business can expect to save on each task because of computerization.

Chapter 6, "Justifying a Computer for Your Business," was my one major disappointment in an otherwise excellent book. I had hoped for a cogent discussion of the various financial issues involved in determining the cost effectiveness of a small business computer.

While each business is different, and no hard rules can be made to cover all of them, I am sure there are good methods for analyzing just how much a computer will save in terms of time and money in various applications of business. And I am sure there are good rules of thumb concerning how large a small business should be before it considers computerizing. Of course these are usually estimates, but they are important guides in making the decision on whether to computerize.

Instead of discussing these in a meaningful way, Smith took the attitude at the beginning of the chapter that he had never seen a feasibility study that was not biased in favor of buying a computer. Therefore, he said, in essence, that the businessman will base his decision to computerize on his feelings even if he has hard data to go by, and any study will automatically be designed to do nothing more than justify those feelings.

My contention with this chapter is that, if I were a business person I would want to know exactly what I could expect in costs versus savings before I put a penny down.

Summary

I strongly recommend this book to businesspersons, computer technicians and students. If you do not have a computer you stand to learn a great deal about how they may help you if you are in business. Those with machines may well learn how to better utilize them. Computer technicians in general, and especially computer salespeople and programmers working for the business market, can also learn from this book because it explains basic business issues using the computer as an aid.

Finally, it can be a good book for the person interested in a home computer but not necessarily interested in computers as a hobby. Although the book discusses business issues, much of its material is directly applicable to the home on a smaller scale. ■

Disk Storage: It's No Place to Compromise.

A dependable disk system means everything to the high-performance operation of your computer. At Percom, we know this. And we do something about it.

For example, the connectors on our TFD drive systems are gold plated.

Why? Because solder-plated contacts oxidize, forming an insidious insulation that can cause erratic operation. And then total failure.

Gold-plated connectors are just one way we make better drive systems than the competition.

Now #1 for the Model III

Quality Percom TFD drives provide more features, cost less. Add-on drives start at only $439.00. Complete First-Drive Systems start at only $749.95. Features: **"Flippy" Capability** — Record your programs and data on either side of a diskette. **Greater Storage Capacity** — Store 180 Kbytes — formatted — on one side of a 40-track TFD drive. Store over twice as much on a TFD 80-track drive. **Think Megabytes** — Your TFD drive controller will handle up to four drives. You can access almost 1.5 million bytes of on-line program and data files. **Either Add-in or Add-On** — Your TFD First Drive System can be **either** internal (add-in) or external (add-on). In either case, you get a complete system including the disk controller, drive, power supplies and disk-operating software. **BASIC-Language DOS** — Included on diskette with each First Drive System is Percom's OS-80/III™. This easy-to-use disk-operating system loads into 7 Kbytes of RAM and then frees the drive for other use. With OS-80™ programs, you can have full, read-write interchangeability between Model I and Model III diskettes. **Or Model III TRSDOS*** — TFD 40-track drives work with Tandy's Model III TRSDOS, without modification. For 80-track operation, TRSDOS can be easily modified with diskette patches supplied.

Still #1 for the Model I

Besides greater storage capacities, more quality control measures and lower prices, all Percom Model I drives are rated for double-density operation.

Plug a DOUBLER™ in your Expansion Interface and enjoy the same double-density disk storge capacity as Model III owners. Included with each DOUBLER is a TRSDOS compatible double-density disk-operating system. DOUBLEZAP programs are available for upgrading other popular DOSs for DOUBLER operation. And our double-density version of OS-80™ costs just $49.95.

Of course you don't **have** to upgrade your Model I for double-density operation to use Percom disk drives. But it's nice to know you can.

Percom TFD drives for the TRS-80* Model I are available in 40-, 77- and 80-track versions, in 1-, 2- and 3-drive configurations. Prices start at $399.00.

System requirements: Model III TFD drives work with a 16-Kbyte system (min) and Model III BASIC. The initial drive must be a first-drive system. An optional interconnecting cable is available for expanding with external drives #3 and #4. Model I TFD drives work with a 16-Kbyte system (min) equipped with an Expansion Interface, Level II BASIC and DOS software, and an interconnecting cable. Two- and four-drive interconnecting cables are available from Percom.

Yes. . . I'd like to know more about the best for my TRS-80 computer. Send me free literature about quality Percom products.
☐ Model I ☐ Model III

Send to
PERCOM DATA COMPANY, Inc., 8-C
11220 Pagemill Rd. Dallas, TX 75243

name

street

city state

zip phone number

MAIL TODAY!

To order products, or for the name of your authorized Percom retailer, call toll free:
1-800-527-1222

PERCOM

PERCOM DATA COMPANY, INC.
11220 PAGEMILL RD. DALLAS, TX 75243
(214) 340-7081

™trademark of Percom Data Company, Inc.
*trademark of Tandy Radio Shack Corporation which has no relationship to Percom Data Company

PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE.

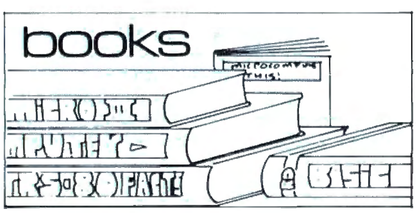
Morse Code, Baudot and ASCII Radio Teletype Programming for the TRS-80 Model I and Model III Microcomputers
Richcraft Engineering Ltd.
Chautauqua, NY
Softcover, 276 pp.
$18

by Dr. William M. Laird

Here is a book that will turn the hair white of firms selling Morse code and radio teletype Assembly programs for the past few years for $149 and up.

As only a modestly experienced Assembly programmer, I found the expanded commentary in each chapter, plus the object code and source code printouts with comments for each program extremely helpful in understanding the programs' logic and flow. These programs are definitely not for the beginning Assembly programmer, who must learn the difference between a JR and JP instruction, but they are easy to follow if you have read this publisher's self-programmed learning series of books. *(Disassembled Handbook for TRS-80,* Volumes I, II and III. This book is actually Volume IV of the series.)

The authors have simplified following the program flow by using a minimum of jumps. The logic is straightforward for each program. The programs have been designed so that once loaded from cassette or disk, the average radio amateur can master their operation with minimal practice.



Chapters I-IV tell you how to generate perfectly timed Morse-code, something seldom heard on amateur bands. For transmission speed, one inputs the words per minute desired. For reception speed, the operator is asked to input two numbers corresponding to dot length and typical space length. This may sound difficult, but it is quite easy to master. The two inputs allow the reception program to accommodate swing fisters—amateurs sending less than perfectly timed Morse.

After about 30 minutes practice, I could not find a station whose Morse the program could not copy. The merged programs use the cassette-in line to copy 2,000-cycle audio from the station receiver, the cassette-out line to drive a 2N2222 transistor. That device keys the station's transmitter key jack with TTL logic, and the cassette motor control relay (through a buffer relay) to control the stations transmitter/receiver and antenna relay, when switching from transmit to receive with the Clear key.

Chapters V–VII are worth five times the price of the entire book. Having been a radio teletype buff for many years, I've finally found a home. Gone are the whirring and clanking machines of past times. All the operator does is tell the program the equivalent Baudot speed desired and nearly everything else is automatic. The clear key is the T/R switch.

The book includes 22 prepared messages may be called from the menu by pressing shift and the up arrow in the transmit mode. If you do not wish to look at the menu, then just press the up arrow and the letter for the message desired.
transmit mode. If you do not wish to look at the Menu, then just press the up arrow and the letter for the message desired.

Chapters VIII–X are similar to programs found in Chapters V–VII, except for 110-Baud, eight-bit ASCII code instead of five-bit Baudot code. If you study and enter the programs in Chapters I–X, you should have earned your master's degree in code conversion, as you will have mastered a number of difficult subjects.

Further Notes

None of the programs in this volume require the RS-232C interface unit for the Model I or the Model III, since they generate their own serial to parallel and parallel to serial conversion.

For the programs in chapters V–X, a digital port interface (such as the Design Solution AN-511, Telesis VAR/80 or Alpha Product Interfacer 2), is required. All work with or without the expansion interface on the Model I, and with the adaptor on the Model III.

Nearly all available memory is grabbed by EDTASM. This program details an interesting approach to writing a mini-program in low memory, so your printer prints out 64 characters or spaces per line.

The book covers using an adaptor, allowing the use of almost any Model I ancillary port operated device (40-pin connector) with the Model III (50-pin connector). Only one active TTL chip is necessary to switch the I/O bus from output to input, plus the 40- and 50-pin connectors.

Ever wish you could start printing an editor/assembler program being assembled at any line without starting at the beginning? A simple 39-cent modification described in the book may be installed in about 10 minutes and does the job very efficiently.

"None of the programs require the RS-232C interface."

A demonstration program allows you to generate the 2125-cycle radio teletype mark tones and 2295-cycle space tones with software rather than a terminal unit. It uses only a single chip transducer from Alpha Product Company. The program outputs the radio teletype test signal RYRYRYRY from the transducer at an equivalent speed of 60 words per minute.

A fascinating, yet short, program prints out the speed in equivalent words per minute of any Baudot radio teletype signal being received—60, 66, 75 or 100 speed. It also tests for 110 or 300 Baud ASCII RTTY.

Summary

This is a teaching text and a book full of functional, working programs for the radio amateur and computer buff who would like to use his TRS-80 to communicate with the fascinating world of ham radio. All programs are divided into individual transmit, receive and combined chapters to allow the reader the opportunity to assimilate the concepts being presented in reasonably sized bites, rather than choking to death with too big a mouthful. The programs may be used as is by inserting only your call letters, name and address in the appropriate prepared message locations with the Radio Shack Editor/Assembler.

All three transmit/receive programs are available on two 35-track disks, though this book is also required for operating instructions. On special order, Richcraft will insert your call letters, name and address in the proper locations of all three transmission and reception programs for Morse, Baudot and ASCII RTTY.

The appendices are extremely useful and could have been the subject of another volume.

Volume IV deserves an excellent rating and should be invaluable to any microcomputerist who wishes to understand the concepts of code conversion and use the TRS-80 in telecommunications. ■

Reader Service for facing page ✓307→


ADVENTURE has gone GOLD!
AND WE WANT YOU TO CELEBRATE WITH US.*
The 12 Scott Adams' Adventures are presented in our **Limited Gold Edition.** Each tape and disk is individually numbered, and guaranteed until July 10th, 2001. Each package contains a certificate of authenticity, a registration card and an autographed, (rather interesting) photo of the author.
The 12 Adventures normally retail for $239.40 individually on tape and $159.80 for 4 triple-pack disks.
The **Limited Gold Edition** is yours . . . **forever** . . . for $100.00 tape or disk, value for value.
To Order: The **Limited Gold Edition** is available in very limited quantity, from interested Software Retailers. Ask your local dealer. If he does not have The **Limited Gold Edition** . . . Then call toll free 1-800-421-5770. In California 1-800-262-4242 (Local 213-670-9461) . . . And we will direct you to a dealer who does have the **Limited Gold Edition.** ***Supply Is Limited!***
The **Limited Gold Edition** is available in the following different configurations.
• Apple 2 Disk with TRS-80 Disk in same package
• Atari Tape
• Atari Disk
• TRS-80 Tape
$100.00 Each
Limited Gold Edition
Scott Adams Adventure Series
Autographed and Individually Numbered
ADVENTURES 1•2•3•4
*In keeping with the spirit of **Adventure,** $1,000.00 in Gold Coins is hidden within The **Limited Gold Edition.** Its waiting for **you.**
The **Limited Gold Edition** from Adventure International is distributed exclusively by: SOFTSEL
8295 SO. LA CIENEGA BLVD. • INGLEWOOD, CA 90301

The Creator
Complete Business Systems, Inc.
Chicago, IL
$295

by Mary S. Gasiorowski

Have you ever invested time and energy in a data management program only to find it did not fit your needs?

Now's your chance, without knowing a bit of Basic, to create your own personalized data management program. Called "the programmer on a disk, for non-programmers and small systems house use," the Creator is a disk-based TRS-80 program which writes Basic programs to collect and manage data. Further, the created programs are modular, well-documented, compact, well-designed and easily modified by the user without knowing the techniques of developing such programs.

Is the Creator really for non-programmers? Well, yes, though for the documentation you may need an interpreter. Here is another example of a good program made difficult by weak documentation. The documentation seems to still be in a rough-draft stage. Page 50, for example, talks about showing the relationship of record size to the number of records on a disk in a table "on the next page"—the table is actually on page 14.

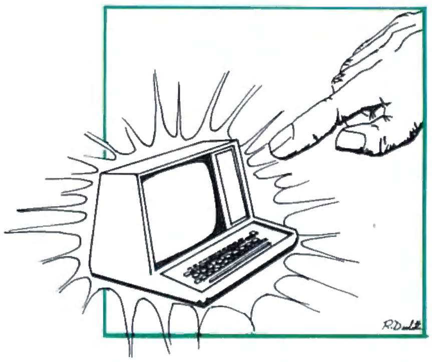
Also, after exclaiming several times that the Creator is not technical, the documentation goes into several somewhat technical and vague discussions on what packed numbers, records and fields are; how many records will fit on a system disk versus a formatted disk; how to calculate (approximately) how many records you can fit on a disk and still allow for expansion; and how to shorten or compress your program.

Data Setup

The programs created by the Creator are ideal for handling a mailing list or an inventory of items—any list of information with any combination of numbers and letters.

A handy form is included in the documentation to help you determine the length of each piece of information (field) in the data item (record). For example, in a mailing list, one complete address is a record, made up of various parts, or fields, such as a zip code. You might allow the last name field a length of 15, the address field 20, the state field only 2, etc. Also using the form, you decide the order in which you want the data to be entered—such as last name first or first name first.

Another major planning step is to determine what checks (the Creator calls them edits) you want included in the program to make sure the data typed in is the proper information. To go back to our mailing list example, it is wise to make sure the zip code is not omitted, or that it has exactly five numbers and no letters. Some of the other things that can be checked for are numbers or letters within a certain range, all numbers or all letters, and inclusion in the data of a certain combination of numbers or letters or both.



You must also indicate the key field, the section (field) of data which determines where in the entire list of data each record should be. In our mailing list, the key field would probably be the last name, so that the mailing list can be sorted by alphabetical order.

Running the Creator

Believe it or not, running the Creator is as simple as typing Run Creator and answering some questions about your data.

Type in a file name for your created program—this is the name you will call up to run your program—and a title to appear on the screen.

Then you will be asked for an exit code. When your program is running and the user is typing in information, typing the exit code lets the program know to stop what it's doing and return to the menu. A common exit code is the word End.

The next step is to describe the data fields. First the length of each field is entered. The Creator keeps track of how much room is left—if you make a mistake, you can reenter the length. Next, indicate which is to be the key field. The Creator will then display information on the types of data and checks you can use.

The Creator will review your outline of data fields, reminding you of the length, and ask you for a prompting message for the user, the kind of data, and the checks you want to use on that data.

In a mailing list, a prompting message for the last name of the addressee might be: "Type in the addressee's last name."

The kind of data can be any character: numeric, numeric and integer (whole numbers between -32767 and +32767), numeric to six significant digits, or numeric to 16 digits.

Finally, enter the checks you want on the data and any messages you want to appear on the screen if the checks fail. For example (our mailing list), if you are looking for a last name, you want to check for No Entry (hitting the enter key without typing in anything), Not Alpha (any numbers instead of letters), and perhaps Length >15 (to restrict last names to 15 characters). Each check produces its own error message; for example, if a last name has 16 characters (flagging the check Length> 15), you might have the message, "Please shorten the name to 15 characters."

The Creator will proceed through all of your data fields, and then suggest a way to end your data collection program.

In this way, you can set up what kind of data you want and what the computer should do if the data is incorrect, without knowing any Basic programming. Once you have run the Creator to create a program, you need only load and run your new program.

The created program is well-designed, easy to modify, well-documented by remark statements, modular and efficient; it is easy to use because it is menu driven and incorporates error trapping. You can update or amend your data files by choosing that option in the menu. You can change one data field without having to delete the entire record.

The Creator is an easy and inexpensive alternative to hiring a consultant to develop data management programs. It is worth the effort to work through the documentation to create a personalized and useful program.

The Reporter

With the Creator came the Reporter, a program which generates programs to write reports. You can set the title, column headings, tabs, column and field relationships (such as column(5) = column(2) + column(4)/2), subtotals, totals, pagination and whether you want the report sent to the screen or the printer. The Reporter is very flexible in its handling of data; for example, if you want to exclude data below a certain range, you can say, "If Column(3)< 10 Then Skip." Just as in the Creator, use of the Reporter requires no knowledge of Basic programming to develop a useful, personalized program.

As in the Creator, you must give a file name for your report program, as well as a title to appear on the screen. You must also determine whether you want the report on the screen or the printer. Unfortunately, the Reporter is not capable of showing the report on the screen, verifying it with the user and then sending it to the printer. (This is one change I suggest the author make for his next version of the Reporter.)

Then type in the name of the data file you want used for the report (don't forget the extension /DAT if the file was created by your data program created by the Creator). Specify the record length of your data—refer to the data form you used for the Creator—then describe the various field lengths.

You can then set the column headings for your report. Note that the number of columns may be different than the number of fields of data, since a column may be defined as a combination of fields or as just part of a field. For example, two data fields may be the number of items in stock and price per item. Two columns may exactly reflect that data, and a third column may be the product of those—the value of the inventory.

You need to plan ahead for the tab settings for the columns. Use a piece of graph paper to determine how much space to set aside for each column. And don't forget to set aside space for left and right margins.

The Reporter will number your pages, if you want, as well as putting the title and column headings on each page. The documentation claims you can have up to 999,999 pages, if you live that long and are that patient, though I only tried five pages.

Type in the type of data in each of the data fields, then describe how the columns are related to these fields. This is similar to setting up the checks or edits in the Creator. You can set a column to be exactly equal to a specific field, or any algebraic or trigonometric combination of any numeric fields or columns. Character (alphabetic) fields can also be manipulated; fields can be reordered, combined or split up to define columns. For example, in a report based on our mailing list, where the last name is in data field(1), first name in field(2), and middle initial in field(3), you can define column(1) as follows: column(1) = field(2) + field(3) + field (1). This would print out the name in correct order in column one in the report. A column can also be defined as part (left, middle or right part) of a character field. For example, if field(5) contains area codes and telephone numbers (and is defined as character not numeric), a column for the area code can be set up: column(3) = left$(field(5)).

The variable V and combinations of it are reserved for saving subtotals and totals. If you want the total of field(6) for all the data, you might define V2 = V2 + field (6). You will be allowed to print any totals at the end of the report.

The Reporter is not as well developed as the Creator. The screen/printer routine should be improved to offer both, and the tab setting could be done automatically by the program after you specify the column headings. The Reporter is not as easy to use as the Creator, and it does use many more Basic terms and commands, and so, requires more from the user.

The Creator is worth the effort (don't get bogged down in the documentation—just run the program). The Reporter is unnecessarily difficult and lacking in several useful features. By the time you figure out the Creator, you may be able to tackle the Reporter. The idea behind these programs is good. Each is not just a single program, but a creator of programs. Perhaps by the time you read this, the author will have revised these programs to make them easier to use. ■

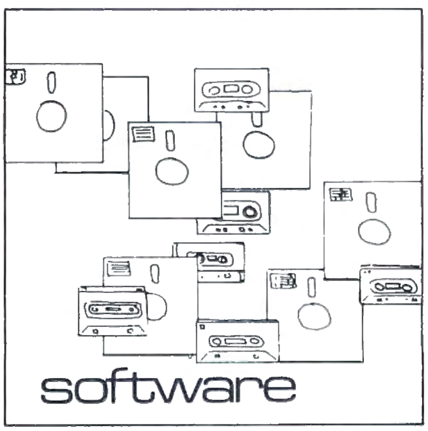
Computer Acquire
Microcomputer Games, Inc.
Avalon Hill Game Co.
Baltimore, MD
$20

by Mary Gasiorowski

Have you ever imagined yourself a financial wizard, the head of a vast hotel chain involved in power struggles and stock mergers? Do you like exciting, fast-paced games? Then keep looking—this is not the game for you.

Computer Acquire is the microcomputer version of the board-game Acquire. There are three versions on the cassette tape: TRS-80 Model I (Level II), Apple II, and Commodore Pet 2001—all require 16K RAM.

Computer Acquire is a game of high finance—a game of strategy and luck, in which from zero to six players can participate (including playing the computer and watching the computer play itself). The object is to build and expand hotel chains. At the same time, you can buy stock in any chain to try to get a majority. Merging two chains will give someone a profit, with a bonus going to the majority stockholder.



"Computer Acquire is a game of high finance."

The game ends when all the chains are merged together or are too large to merge. The hotels and stocks are then cashed in, and the person with the most money is the winner.

Does it sound complicated? It is. The instructions suggest that the game will take two hours to play. It does.

I played Computer Acquire on the TRS-80 Model III (it is compatible) and the Apple. It seemed fairly well debugged. If you answer a question with an absurd number or letter, you are asked to try again.

Why was this board game put on a microcomputer? Oh, the game is okay, but little attempt was made to take advantage of the various features of the microcomputer.

You can play the computer, or, if you are a fanatical game watcher (rather than a game player), you can watch the computer play itself.

There are no graphics, no color, no sound. The screen is harder to read than the playing board included in the original (non-computer) version; it is difficult to distinguish the numbers of the squares from the letters standing for hotel chains since there is no spacing between squares. If you really want to play this game, I suggest you buy the board version ($6 for the board, $5 for a set of marker tiles).

I hope Avalon Hill finds a new programmer before they convert any more of their games for the microcomputer. ■

ACCEL2: Compiler for TRS-80 Disk Basic
Allan Gelder Software
San Francisco, CA
$89

by Bruce Powel Douglass

Machine code instructions can be executed directly by the Z-80. Run-time for programs written in machine code tend to be very fast. High-level languages, such as Basic, make the machine more accessible to the user by being easier to remember.

These languages must also be translated into the machine code for execution. There are two ways to go about this: interpretation and compilation.

An interpreter fetches the Basic instructions that tell it what to do, and changes them to machine code. This must be done every time a statement is executed. And, although it can take a lot of time, there are benefits. Basic programs can be easily edited, changed, and rerun with a minimum of user effort.

A compiler, on the other hand, only translates the program once, so the CPU can execute the program directly. The run-time is shortened since statements are pre-digested into byte-sized bits. But if the program bombs it's harder to debug. For many time-critical applications, however, a compiler is the best choice, even though they usually require large amounts of memory and cost a lot of money.

ACCEL2 is a compiler which offers a unique approach to accelerating execution spend of Basic programs for the TRS-80. Coupled with its low price, its capabilities make it a reasonably good alternative to other options, such as: writing machine language programs, embedding machine-executable code in Basic programs, or spending a lot of money on big compilers.

ACCEL2 comes on a self-relocatable system tape. It loads starting at 18944 and occupies 5120 bytes. To save it on disk, enter Basic2 from DOS and load the tape. Relocate it where you want it to reside, and reboot. Now you can save ACCEL2 as a core image onto disk.

One of the nice things about ACCEL2 is that 16K tape users can now own and operate a compiler that works. Compare that with Microsoft's BASCOM, in which the run-time system alone is over 14,000 bytes!

The lovable aspect of ACCEL2 is that you are free to sell programs you compile with it without having to pay royalties. Microsoft requires nine percent or $195 per year for programs sold that used their compiler.

To use ACCEL2, you must call it via the USR function. When you load it from Level II, it automatically loads its address into locations 16526 and 16527, so you may immediately call it by A=USR(0). ACCEL2 has the built-in capability to save and load programs to and from disk. If you want to write a self-contained System tape, you must purchase TSAVE for $10.

ACCEL2 takes the Basic program in memory and compiles certain statements into REM statements with the embedded, compiled code. It leaves the statements that it cannot compile for the Basic interpreter to execute. Thus, it will compile almost all Basic programs without any revision on your part.

This allows the Break key to still work. This is the only compiler that I know (I have three) that has this capability, although they may be simulated in the others. TRON and TROFF, the Trace function, also work, but they only trace the uncompiled sections of the program.

The run-time system is only 1K, and yet ACCEL2 compiles full floating point for elementary operations (+, -, *, and /), as well as single-dimensioned arrays and some string functions.

When you are compiling a program that you don't want to type again, be sure to save a copy. Compiled programs can't be edited. If you enter the edit mode for any line of your compiled program, whether that line is compiled or not, your system will hang up, requiring a reboot or reset, which is very inconvenient. If the compiler sends you a compile-time error message, your program has been destroyed. You must type in New and reload your program.

Error messages are limited, but useful. You may receive one of six messages: SN (syntax error), OM (out of memory), FC (illegal function), NF (Next without For), LS (string too long) or UL (undefined line for GOTO or GOSUB).

The compiler also supports tape or disk I/O, whereas BASCOM only allows disk I/O. This enables you to keep tape files, and use your compiler at the same time.

The program does contain an error in how it reads data statements. After compiling a program that contains data statements, the program will run (all other things being correct). But if you call the same compiled program off disk to run, you will receive an OD error. This is not much of a problem for me since I only have one program with data statements, but it would be a real pain for someone who dearly loves them. It necessitates recompiling the program every time you use it.

Compile times are relatively short, except for long programs. A very short program will compile in a few seconds. A 6K program takes five to 10 minutes, and a long program should be compiled while you take your mother out to dinner.

The Manual

The manual is mediocre. It is 11 pages long and contains useful information, but does not go into enough detail. No mention is made of the data-statement error. It does talk some about pitfalls and about speeding up execution.

ACCEL2 does not compile several statements, such as Print, and the transcendental functions, such as LOG, SQR, and SIN. But it will compile double precision for the elementary arithmetic operations +, -, *, and /. This makes ACCEL2 very useful for speeding slow double-precision arithmetic. You can even write approximation routines for double-precision transcendental functions and compile them.

How much ACCEL2 will speed your Basic programs depends on program structure. If your program contains mostly non-compiled statements, execution time will be similar to Basic's. If the major reason for slow execution is comparing and branching, then execution time will be significantly faster.

The manual gives a table for relative speeds of execution for a variety of compiled statements. Integer arithmetic operations are said to run 47 times faster for addition and subtraction, and two to three times faster for multiplication and division. Double-precision arithmetic execution times are much closer to Basic, ranging from 1.5 for addition and 1.02 for division.

The string operations are quicker as well. LEN executes 43 times faster, MID$ four times faster, LEFT$ three times faster, RIGHT$ 2.8 times faster, and CHR$ almost five times faster.

Summary

In conclusion, ACCEL2 offers an inexpensive, and efficient method for decreasing run times for general Basic programs. Tape and disk I/O are supported, as well as double precision. Rewriting is not necessary, although execution speed may often be enhanced by writing more optimal Basic code. Its only real problem is in its execution of Read statements when executing compiled programs called off disk. (If someone does discover the correction for this error, please pass it on.)■

In Memory Information 3.0
Tandy/Radio Shack
Fort Worth, TX
$19.95

by Fritz Milhaupt

Radio Shack has been selling its In Memory Information card filing system since the TRS-80's introduction in 1978. Since then many versions have been offered, the most recent being version 3.0 released last fall for Models I and III. It is a fairly good system and lacks nearly all of the bugs of the earlier versions.

The In Memory Information System is based on storing information on pages of memory referred to as *cards*. Each card may contain up to 14 fields of one line (41 characters) each definable by the user. The card has a maximum length of 255 bytes which somewhat limits the amount of data per field. In some applications it may be necessary to break up some entries into two or more cards. This can pose a problem if the cards are sorted later.

This package consists of three programs: the Initialization program, the Sort program, and an Update and Retrieval program. All are written in machine-code.

The Initialization program sets up the number and the length of the fields and creates the initial data file. During the card formatting procedure, the program asks whether to justify short entries to the left or to the right. Unless left justification is specified, leading blanks will have to be included in search strings when using the Find function later on when using the Update and Retrieval module. This is not mentioned in the manual.

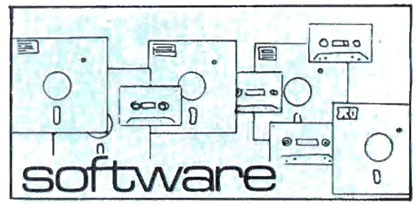
The Sort module serves three tasks. It sorts the file in ascending or descending order to user specification, it prints the data file on optional 80-column parallel printer, and it can also split a long data file into two or more sections to avoid overrunning available memory.

The Update and Retrieval program is the most used of the three in this system. With it you can add, delete, or update the data cards. Updating can be quite tedious. First the card is found (manually or using the Find command) and the Update mode is entered. The entire card may now be updated. To skip a line that you do not wish to change, hit Enter and the cursor will drop one line. To skip more than one line, hold down the Enter key until the cursor reaches the desired line. Once the last line is passed, the Update mode is exited. Since there is no global replace or delete, updating can get quite tiring.

Each program allows for the repetition of key entries by holding down the proper key until the desired number of characters has been printed. This is something of a mixed blessing since it allows the rapid entry of multiple characters. But if you are a heavy-fingered typer like myself, it is quite an annoying feature, because it acts like a badly bouncing keyboard. The only real operational bug is that if the amount of available memory drops below twice the length of a data card (both of which are displayed constantly), all of the data in memory will be lost because the program restarts.

As a package, In Memory Information makes extensive use of tape files. In order to pass from one module to another, the data must be saved, the desired module loaded, and the data reloaded. Tape data may be saved at 500 baud on the Model I and at 500 or 1500 baud on the Model III. The programs themselves are loaded by their own special loaders.

This program is exactly as Radio Shack advertises, a computerized card filing system. It is possible to save and access great amounts of information with it, particularly in systems with more than 16K memory. It has potential for those who wish to keep brief records of customer



payment or status in very small businesses. For most small businesses though, Radio Shack's Versafile, which is a shirttail cousin of In Memory Information, would be better suited as long as disks were available.

In Memory Information is a good data base for its price, and for most home and hobby applications would probably be adequate. Compared to other data bases in the same price range it ranks favorably. ■

Rats Revenge
Med Systems Software
Chapel Hill, NC
16K Level 2 Models I and III
$12.95 cassette
$16.95 32K diskette (Model I)

by Darren DeVigili

What makes Rats Revenge stand out from the many maze games already on the market? It's interesting. Up until now every maze I traversed led to tedium. I never found going from one maze to another very stimulating.

Apparently the author of Rats Revenge felt likewise, as the program is loaded with things its predecessors lacked. Suddenly I'm a rat, with the desire to scurry around, looking about a maze for a morsel of Swiss cheese. I can move one space at a time, run down straight hallways, turn in place (to get my bearings), and even consult a Hint Guru as to my relative location to the maze's main trail. There is also competitive scoring, though obviously only one person can play at a time. (Try to figure out a maze while taking turns.)

Everything I see is in 'rat' perspective. I don't get a top view of the maze until I find the cheese or quit. Oh yes, there is one other way to leave the maze—on little phosphorescent wings. Not that merely dying is enough, some twisted mind included the dreaded cheese hallucinations.

Besides what you see as a rat, there are assorted messages, some good instructions, and the mysteriously appearing (and somewhat familiar) "hmmmmm." Better wait until you see it yourself, no reason to tell you everything. The mazes are randomly generated before each game and take from one to six minutes depending on what size maze you choose. When you either win, quit or starve, the maze is drawn for you and a small dot traces the path you took. Small mazes are drawn with a double-pixel line, large ones with a single-pixel line. Those of you who think it isn't all that big should remember you're inside the maze rather than over it, and you have a limited number of moves to find the cheese.

All of Med's software comes with a two-week guarantee. They don't just guarantee the program will run, they guarantee you'll like it. That's something I appreciate. Who else has the confidence to make guarantees like that? Rats Revenge, as with all the Med software I've seen, is worthwhile. With its well-drawn graphics, total rat perspective, variable skill levels, humorous touches, and more, is one game you won't play once and shelve. ■

Hayes Stack™

Microcomputer Component Systems

A beautiful match, the Smartmodem and the TRS-80. *Your TRS-80 can talk with other computers,* over the telephone lines. And with no acoustic losses or distortions.

Access time-sharing systems and information utilities such as the Source,* CompuServe† and MicroNet.

Direct hook-up with no interference noises. The Smartmodem hooks to the telephone line just like a modular telephone, simply insert in a wall jack.

"Love at first sight" – your TRS-80 and the Smartmodem!

Brawny – because it does so many things. *Auto-dial and auto-answer features built in.* With the Smartmodem, your TRS-80 can *automatically* dial the telephone, answer the telephone, receive and transmit, and hang up the telephone. Completely unattended.

*Pulse dialing or Touch-Tone.*** The Smartmodem can be connected to any telephone system in the U.S. because it allows pulse-dialing, Touch-Tone dialing or a combination of the two. FCC approved.

Program controllable in any language using ASCII character strings. This is a unique feature of the Hayes Smartmodem.

Brainy – because it does them all so simply. *Seven LED indicators on the front panel* give you visual signals of the status of the Smartmodem: MR – Modem Ready, SD – Send Data, CD – Carrier Detected, etc.

The audio monitor feature lets you "listen in" on the call being dialed and the connection made. You are immediately alerted to busy signals, wrong numbers, etc.

Over 30 different commands can be entered directly from your TRS-80 keyboard, including the unique "Set" commands which allow you to select and change various optional parameters such as dialing speed, escape code character, length of time for a dial tone, and number of rings to answer. There are 17 "Set" commands.

The Smartmodem is completely compatible with the Bell-103 type modems, the type of modem most time-sharing systems have. Operation can be in full or half-duplex, with a transmission speed of 0-300 baud.

The Smartmodem is ready to "get-together" with your TRS-80. TRS-80 Model II and TRS-80 Color Computers have RS-232 serial ports and can immediately interface with the Smartmodem. Expansions that permit use of the Smartmodem with TRS-80 Model I and Model II are available through your TRS-80 dealer.

Match your TRS-80 with a Hayes Smartmodem for a sophisticated, high performanced data communication system. Available at computer stores nationwide (except TRS-80 dealers) – call or write for the location nearest you. And don't settle for anything less than Hayes.

Hayes

Hayes Microcomputer Products, Inc.
5835 Peachtree Corners East,
Norcross, Georgia 30092 (404) 449-8791

Your TRS-80 computer and the Hayes Stack™ Smartmodem. Beauty, Brains, and Brawn!

™Hayes Stack is a trademark of Hayes Microcomputer Products, Inc. *Trademark of Source Telecommunication Corporation. **Trademark of American Telephone and Telegraph. †Trademark of CompuServe Information Services, Inc.

XBE
Computer Applications Unlimited
Rye, NY
$24.95

by Bruce Powel Douglass

"You have a blinking cursor with reasonably complete ease of movement. . ."

XBE makes editing Basic programs much simpler than any other program of its type. I almost never write or edit programs without using XBE.

Some of its capabilities are selective renumbering of a line or a block of lines, finding the occurrence of any string of characters in the program changing the string to any other string of characters, block deleting lines, definable macro keys for quick entry of Basic commands or oft-used words, page the program, scan the program a line at a time, or jump to any line directly.

The program comes on tape with a relocating loader to place the program at the top of memory, with 16K, 32K or 48K. Once placed into the appropriate memory location, it may be saved on disk as a core image, or on tape with T-Bug. It occupies 5,120 bytes. Once loaded and initialized (with a System command), it may be invoked by entering the command XEDIT.

You must have at least one line to invoke the editor. Once invoked, the editor clears the screen and reformats it, presenting the line numbers in columns zero through seven, and the Basic commands beginning in column nine. No statement will be displayed below the twelfth line, and at the bottom of the screen an area is boxed off. This is for the extended commands mentioned later.

You have a blinking cursor with reasonably complete ease of movement and two speeds to move the cursor along a line using the right arrow. The movement keys are repeating. To backspace (non-destructive), you press the left arrow key. Shift left arrow places the cursor at the beginning of the line. To go to the next line, press Enter. This is also a repeating key. To go to a previous line, press shift up arrow. An up arrow without the shift is a recognizable character, and so it must be accompanied by a shift to move the cursor to the previous line. Shift down arrow lists the next page of your program on the screen.

Editing

When you get to a line you wish to edit, you merely type over the offending errors. To insert characters, use shift I. This will allow you to insert characters until you press Enter. The full page remains visible and the line wraps around as you insert the new characters. To delete, shift D will do the trick. It is a repeating key, so you can hold it down until you have deleted as much as you want. Be careful though—there is no way to recover characters that have been accidentally deleted. They must be reinserted.

Shift X has the same function it has in Level II editing—it extends the line by placing the cursor at the end of the current line and entering the insert mode.

If you enter the insert mode, shifted characters are now macro keys. They have default strings which may be changed to your desire. Thus you press only a single key to enter Print, LPRINT or CHR$(. You are limited to six characters with the macro keys.

Clear Command Mode

You have an entire mode called Clear command mode, because it is initiated by pressing the Clear key at the right of the keyboard. The following commands must first be initiated in this manner.

To position the cursor at the first line of the program, press Clear T. To exit the editor and return to Basic, press Clear E. To find a previously defined string, press Clear F.

To delete a line, press Clear D. To insert a line, press Clear I. The default increment is five, but that is easily changed. If there is no room for your line (with the given increment), you will be given the message "No Room Between Lines." You may then renumber the entire program with larger increments between lines, or change the increment for the insert function.

You may mark a statement for a variety of purposes. Clear C marks a statement for copying, while Clear M marks it as a statement to be moved elsewhere (effectively, a selective renumbering of that line). Copy makes a copy of the line where you tell it to, while Move moves the line, thus with the former function, you have two copies of the line, and with the letter, you have only one. Clear H says, "This is where I want the line (or block of lines) moved or copied to." You place the cursor at the line preceding the location where you want the line(s) to appear, and press Clear H.

Clear B is a marker for blocks of statements to be moved, copied or deleted. You mark the first line with Clear B, and then the last line with the Clear command function you wish performed. A square lights up at the lower right corner of the screen to remind you that you have something marked.

More Features

There is also an Extended Clear command mode. You enter this with Clear space bar. The cursor is then positioned at the bottom of the screen. From here you can: change the increment value on lines, renumber the entire program, search for a line number, search for a string, change a character string, or redefine your macro keys. Enter executes the command.

By entering In, where n is a number, you can set the increment value for inserting lines. With Sn, where n is a number, you can look for line number n and position it at the top of the page. You may also use Fstring, which tells the editor to find the first occurrence of string, and the string may be up to 20 characters. Remember, you may invoke Clear F to search for further occurrences of the string. Once the string is located, you may reenter the extended Clear command mode and enter Cstring. This will change every occurence of the string used in the Find command to the string given in the Change command. The strings needn't be the same length either—the lines will adjust themselves (unless the length becomes greater than 240 characters).

This mode also allows you to renumber the entire program with the extended Clear command Nn1,n2, where n1 is the starting number assigned to the first statement in the program, and n2 is the increment.

Finally, you can enter D(key)(string) and redefine any letter key A through Z to the macro key of your choosing. This macro key may be invoked anytime you are in the insert mode by pressing Shift. Thus it may be used when in insert-in-a-line, X, (extend a line), or insert-a-line modes. You are limited to six characters.

I have used this program extensively since I received it in the mail, and I am still impressed. It is an excellent piece of well written software that is long overdue. ■

Reader Service for facing page ✓357→

EDAS
Misosys
Alexandria, VA
$79 Disk

by Paul Weiner

There are several editor/assemblers available for the TRS-80. Each trade-off among conflicting design goals such as mainframe features, size of text buffer, speed, I/O options, flexibility and ease of use.

One way to avoid excessive memory consumption without sacrificing features which require lengthy code is to use an overlay system. With this method, a large program is broken up into two or more smaller parts so that not all the code need occupy memory at anyone time. The most familiar example of such a system is TRSDOS. In editor/assemblers, the simplest application of the overlay approach is to use two main overlays, one for the editor, and the other for the assembler.

Since most assemblers perform more than one pass through the source code during an assembly, the assembler itself can be broken up into several overlays. This technique does offer lots of source space without skimping on features, but the frequent loading of the various overlays tends to make performance sluggish.

At one end of the fancy versus quick-and-easy spectrum is Radio Shack's Editor/Assembler. It supports only tape I/O and does nothing flashy, but is straightforward and easy enough to use. At the other end of the spectrum is Microsoft's ALDS (Assembly Language Development System). ALDS has all the high-level features referred to and more, but is awkward to use, hard to comprehend and slow. Because of these drawbacks, surprisingly few assembly language programmers seem to be satisfied with ALDS, despite its advanced features.

In the middle is a variety of software products. Near the low end are various modifications and upgrades to Radio Shack's editor/assembler. These would include AsPatch, DiskMod, and Apparat's EDTASM. At the high end are completely new products, such as Microsoft's new editor/assembler, and EDAS. I think that for most purposes, EDAS has accomplished the most practical trade-off of design factors, and is currently the best available choice.

EDAS

EDAS, written by Roy Sotoff, is a disk-based editor/assembler. Tape I/O is not supported. Unlike ALDS, the current release of EDAS does not support macros, relocatable code generation, or conditional assembly. The next release of EDAS, which will be sent free to all purchasers of earlier releases, will implement some conditional assembly and other new features. However, it is unlikely that macros or relocatable code generation will ever be supported. But don't assume that EDAS is not for formidable programming projects. The LDOS operating system was developed on EDAS. And the new releases of Kim Watt's SuperUtility, a 24K machine-language program, were edited and assembled on EDAS.

". . . EDAS assembles directly from disk."

One of the problems for assembly-language programmers is the size discrepancy between object and source code. Even on a 48K system, a program which assembles down to only a few K of machine language may have a symbol table and source code which occupy more room than is available in an editor/assembler's buffer. This is especially true if the code is highly commented and labeled. ALDS' relocatable module generation solves this problem by letting the author preassemble various segments of his/her program in relocatable form and then having Link-80, ALDS' linking-loader, link them together into a run-time program.

Unfortunately, this usually involves multiple assemblies and seemingly endless disk I/O operations, all at a surprisingly slow disk data transfer rate.

EDAS has a different and remarkably effective approach to many of the problems which ALDS attempts to solve. For instance, EDAS is not organized in overlays. Normally, it's all there in memory at the same time. However, if you're cramped for text space, EDAS' X(extend) command allows the text buffer to overwrite the assembler. This, in effect, achieves an overlay system, since the assembler can be reloaded and run after the source has been moved to disk. But you will rarely need to bother with that procedure because EDAS solves the long source file problem in another way.

Via the Get statement, EDAS assembles directly from disk. One assembly may use any number of Get statements. Thus, EDAS lets you break your source program into several modules, each of which is stored in a separate disk file. Since a file assembled as the result of a Get isn't actually buffered in memory, a module may be larger than 48K.

If your in-memory source code consists only of a series of Get statements and an End statement, most of your computer's memory is available to hold the symbol table. This solves the source size problem and permits editing and assembling of large programs. It also provides the basis of a subroutine library management system.

In fact, if there are no ORGs in any of the Get modules, they will all be sequentially arranged in memory relative to the ORG in the text buffer. In essence, this arrangement achieves the same effect as does ALDS' relocatable code generation, except that the relocatable modules exist in the source state instead of the object state.

The EDAS assembler also accepts a View (filename) command which causes it to display the requested disk file on the video screen. This allows you to make sure you are Getting the right file. And to further enhance its disk utility, EDAS allows viewing disk directories (including file size allocations and free space) and killing files without leaving the editor.

Other Features

Three other nice features of EDAS have a synergy which results in a breakthrough in debugging ease. I am referring to the /IM assembly switch, the Branch command, and EDAS' warm start capacity. The IM/ switch lets you assemble a program directly into memory instead of to a disk file. The Branch command lets you jump to any specified address, such as the entry point of the program just assembled into memory. And EDAS' reentrant nature lets you escape from test execution of your object code (by reset if necessary), and return to EDAS with your source intact (providing your testing hasn't run wild and squashed it).

One word of caution with regard to reentering EDAS. EDAS keeps its text pointers in low memory—starting at about 5300H. Therefore, going to DOS and using any of the library commands is likely to result in a confused EDAS after reentry. This will probably be corrected in EDAS' next release.

In the meantime, if you have used EDAS to assemble some code to memory and want to debug it, the best way to leave EDAS is to Branch to 30H. This puts you into the restart routine, which invokes Debug without issuing the DOS Debug library command. From Debug you can operate on your program in the usual fashion and finally jump back to EDAS with your test intact.

Upon entering EDAS, you are presented with a memory size question similar to that of Level II Basic. If you default by hitting Enter, EDAS uses the address pointed to by HIGH$ (DOS' pointer to the highest available byte of memory) as its memory ceiling. If you answer the memory size question with a hex address, EDAS uses that address for its ceiling. This allows you to protect driver routines, monitors, and such in high memory. In either case, the text buffer and symbol table are cleared for a new session. Finally, if you answer the memory size question with zero, or by pressing Break, EDAS makes no restoration of its pointers. Thus, you may go on editing the same text you were working on before you last left EDAS, providing nothing has stepped on it in the meantime.

EDAS also supports lowercase. To avail yourself of this support, you must, of course, have a lowercase keyboard (no problem on a Model III) and lowercase driving software (again, no problem with the Model III). If necessary, EDAS' Memory Size feature can be used to protect the driver in high memory.

Once set up for lowercase, EDAS offers two modes of operation. In one, both upper and lowercase characters are maintained in the text buffer just as they were keyed in. This allows you to use EDAS' powerful editor to prepare word processing files.

In the other mode, certain lowercase letters are converted to uppercase as they are inserted, because assembler syntax requires instructions, mnemonics, and labels to all be uppercase. Even in this mode, EDAS does not convert material in quotes or comments, since lowercase in those areas would not generate assembly errors.

EDAS has several other improvements over what is available in editor/assembler. For example labels may be up to 14 characters long. E/A's limit is six letters. The extra letters are important to those us who would like to choose labels with a mnemonic quality or which contribute to the self-documenting nature of the program.

Another such improvement is EDAS treatment of DEFM, DEFB, and DEFW. Each of these now allows you to concatenate values.

Each value is separated from others in the value list by a comma. ASCII values are surrounded by single quotes. For example

```
10 DEFM 'This is example',' ','# ',1 assembles as 54H
68H 69H 73H 20H 69H 73H 20H 65H 78H 61H 6DH 70H
6CH 65H 20H 23H 20H 01H.
```

This string of hex characters, when looked at through ASCII-colored glasses, becomes: This is example # 1. (Note: the digit 1 in this example assembles as a hex 1, not an ASCII 1.) Items in the value list may also be expressions, for example: 20 DEFM 'T' + 20H,' = t' assembles as 74H 3DH 74H, which in ASCII is t = t.

Another nice thing about EDAS' handling of DEFM and DEFB is their compact presentation in the assembled listing.

One of EDAS' smallest, but most useful, advantages over E/A is its enabling of the clear key. I use this feature almost constantly, as I like to present myself with neat cohesive screenfuls of data when I'm programming. With E/A, I've had to hand scroll superfluous material off the screen by pressing the down-arrow 15 times in succession.

EDAS' documentation is well written and complete. Amazingly, it is printed on 70-pound lined paper, which makes merely turning the pages a luxurious experience. It does not contain an instruction-by-instruction explanation of the Z-80 opcodes. Those who need a course in Z-80 machine language will have to supplement the EDAS manual with additional material. The EDAS documentation quite correctly restricts itself to detailing the operation of the editor/assembler, and that it does quite well. There is a source file on the EDAS disk which has an uncommented list of all the Z-80 instructions.

"EDAS' documentation is well written and complete."

Finally, I would like to mention that a few minor bugs have been discovered in EDAS. They are cosmetic or documentational. None of them interfere with the basic functioning of the program. For instance, when explaining the Write command, the documentation indicates that a comma should be used to separate the starting and ending line numbers. Actually, a colon (:) must be used.■

Just as this article was about to be typeset, I learned that Misosys is ready to release EDAS 3.5. All registered EDAS owners will receive a notice to mail in a disk for a free upgrade. Here are the major improvements in the new version of EDAS.

There is now a cross-reference (X-REF) utility. It generates files containing every lable declaration and reference. For each declaration, the value of the symbol is given. For each reference, the file name and line number of declaration are recorded. A source file of EQUs for all referenced symbols can be generated. The user can specify a page heading (and other page formatting specifications) for X-REF to use when printing its files.

A new utility, called Tape to Disk (TTD) has been added to the EDAS package. It converts EDTASM or EDTASM + sourcefiles from tape to disk.

Another new utility is CMDFILE, which allows the user to transfer object files from tape to disk and vice versa, merge object files, and perform similar manipulations.

*A number of improvements have been made to EDAS itself. It now accepts the additional logical operators: integer *, integer /, logical OR, and logical XOR. Four conditional operators for conditional assembly are now available.*

The following improvements have been made to provide compatibility with Microsoft's Macro-80 source files: A dollar-sign ($) now may be the first character of a label. A label may be alone on a line. EDAS can now read and write Macro-80 source files with or without line numbers, and with or without Macro-80 headers.

A couple of new pseudo-ops are provided including COM, which forces a comment line to be written to a load module. (A comment line does not get copied into memory when the load module is loaded.)

EDAS 3.5 allows an object code filespec to be put into an assembly command line. and Gets a file that starts with a comment. Also, you now have a better chance of reentering EDAS from DOS with your source code intact, because the crucial vectors are being stored higher in memory. EDAS can now be run from a JCL (Job Control Language) file.

Certain prompts (like 'Mem Size' and 'End of Page') have been dispensed with. Users may specify the pertinent information when entering EDAS from DOS. Since this relies upon DOS' parameter scanning routine, the new version of EDAS is not compatible with NEWDOS-80.■

EDUCATION 80

by Earl R. Savage

"Fortunately, these and similar procedures may be done in the computer's memory."

Last month we discussed ways of chaining programs so one can call up another. Examples of the "master menu" type of operation (where one program calls up any one of several programs) were given.

The same principle may be used to chain sequentially through a string of programs. As soon as a student finishes one program, the next is automatically called up.

The chaining technique can be effectively used in demonstrations and in instruction, where it is often advantageous to carry information from one program to the next or to accumulate data across a series of programs.

The First Chained Program

As an example, suppose the first chained program asks the student for his name. You would not want subsequent programs to ask for the same information; it is better to arrange for all of them to "know" the answer. Another use is to have each chained program store the student's score so you can access his achievement on each one. This could be accomplished by printing the scores, but who has a printer connected to every computer?

Fortunately, these and similar procedures may be done in the computer's memory (RAM). There must be extra memory—some memory at the top not required by any of the programs in the series, in which to store any information to be passed from program to program.

The first step is to reserve memory. Do this manually before loading the first program—answer the memory size question with the proper number. You will find, however, that there are several advantages to having the first program set the memory automatically. (If you have forgotten how to do this, review the procedure in the October Education 80.)

Once you have set aside memory you must make a special effort to store and retrieve data in that area. (A good thing, too; it prevents all but the most sophisticated student from changing it!) We'll use POKE and PEEK to place and retrieve the data.

The statement examples which follow were developed for a 48K Model III. Use on the Model I should require no changes. However, either machine with only 16K or 32K will require that the POKE, PEEK, and memory size addresses be changed accordingly.

```
10 POKE 16562, 255 : POKE 16561, 228
```

This first statement sets the memory size at 65510 just as though the question had been asked and answered with that number. Twenty-five RAM locations have been reserved for special use.

```
30 IF PEEK (-3) THEN 310
```

This example is taken from a master-menu type program chain. If any value other than zero is found at −3 (65533), program execution transfers to line 310. Any time the menu program is run after the first run, the introductory material is skipped and the program goes straight to the menu.

"... for the difference in cost... they will provide you with the Model III version."

There are times the locations above 16K must be referred to as negatives. Beyond the 16K boundary, PEEK and POKE require negative numbers which decrease to −1 at 65535 (see your TRS-80 manual for further details).

```
150 INPUT B$ : REM FOR STUDENT NAME ENTRY
```

The name in B$ will be lost as soon as another program is loaded. In order to keep it, we use the following technique: 160 Y = LEN (B$) − 1 : IF Y > 20 THEN Y = 20. This statement sets the value of Y equal to one less than the length of the student name. Because of the amount of RAM reserved, Y is limited to a maximum value of 20. 165 POKE −3, 1 sets location 65533 to the value of 1, to be read by line 30 the next time the menu is run. 168 POKE −4, Y POKEs the length of the name (Y) into 65532.

```
170 FOR X = -26 TO -26 + Y
    : POKE X, ASC (MID$ (B$, X + 27, 1))
    : NEXT
```

This line POKEs the student's name (one letter at a time) into reserved memory, beginning at 65510. Here is how it is retrieved:

```
310 Y = PEEK (-4)
315 FOR X = -26 TO -26 + Y
    : B$ = B$ + CHR$ (PEEK (X))
    : NEXT
320 PRINT"HELLO, "B$". IT'S GOOD TO SEE YOU
    BACK. HERE IS THE MENU AGAIN."
```

Line 310 gets the name length from reserved memory. Line 315 concatenates the name letter by letter into the variable B$. The student will be addressed by his name every time he returns to the menu to make another selection.

This technique can be used to pass any kind of information from one program to another. By putting line 315 in any subsequent programs, they will "know" the student's name or whatever data you have stored.

If you wish to accumulate the student's scores, have each program POKE its score in a different RAM location. You can use a special program to PEEK at the numbers and present them to you at any time.

When It Won't Work

There are three circumstances which will prevent this technique from working. If any program in the chain POKEs into the area where you are storing data, that data will be lost. If the student presses the Reset button, all will be lost. Finally, if the

computer is turned off and then on, there will be no information stored. You can take steps to minimize the possibility of these things occurring.

To clear out stored data before another student uses the program chain, you can depend upon newly stored information which is POKEd in to replace the data from the previous student. Should you decide not to do so, you can turn the computer off, press Reset, or POKE in a string of zeros.

Passing values from one program to another has many advantageous uses in your CAI (Computer Assisted Instruction). It also opens up another dimension in your programming.

Word Processor Update

In an earlier column I mentioned Delmer Hinrichs' excellent word processor, listed in the June 1980 issue of *80 Microcomputing*. It was (and is) the best "freebie" word processor I have come across. It does a good job in preparing term papers, reports and the like. It has two major disadvantages.

A lot of time is required to key it into the computer the first time. (But, then, one hears that student labor is cheap!) Also, the response time is a bit slow because it is written in Basic. (One also hears that few teachers or students are such great typists anyway!)

If you want to avoid all the typing and get a word processor that reacts faster, you might want to get in touch with Delmer (2116 S. E. 377th Avenue, Washougal, WA 98671). He has several improved versions available for tape or disk operation. Better yet, his prices are low. I would not imply that his word processors are equivalent to Scripsit or the Pencil, but for the price, they are terrific.

Delmer even has a word processor which will print a letter repeatedly, pausing to pick up a new name and address and use the first name in the text wherever you wish. Could be very handy when you are writing original letters to parents or students.

Upgrading from Model I to III

There is no question that Radio Shack is selling a lot of Model III machines. As time goes by, more and more schools will be replacing their Model I machines. When we upgrade from a Model I to a Model III, we have a different situation from those of you who start off with a Model III.

Model I users have accumulated a variety of utility programs. Some may not be needed after upgrading, because their functions are built into the Model III. Most, however, would continue to fill a need if they would work on the Model III.

Since I upgraded, I have found some of my old utilities work well; others function only partially and still others, not at all. Replacing a lot of utilities can be impossible after the school or activity budget has been broken in the upgrading process!

Fortunately, some producers of utilities are very cooperative in these circumstances. The general policy seems to be that for the difference in cost between the two versions, they will provide you with the Model III version.

To date, I have found two producers to be quite helpful. One of these is Cottage Software (if you aren't using Packer, you must not be writing programs). The other is Computer Applications Unlimited.

This is not to say that the other manufacturers are uncooperative in solving this problem; I have written to some others and have not yet given up on receiving a reply.

To help each other through the exchange of information, let me know your experience with utilities in the upgrading process. Which have you found that work just as well on the Model III? Which won't? What reactions have you gotten from the producers? Be sure to give me your Model I set-up and the new Model III configuration, as well as the names of the utility and the producer.■

80 REMARKS

from page 6

tially. When we cut costs that generally means a few more pages of magazine for you.

We are seeing more and more articles on the use of word processors, and even hear them mentioned on television. It didn't hurt when Jimmy Carter (remember him?) turned to word processing for his writing.■

Making Money

Ever since the microcomputer business began I've been asked for advice on which firms appear to be the best for investment. With the recent media attention to the industry my calls from people with venture money have been increasing.

I've been cautious about making any recommendations for several reasons. There have been some products which seemed to have good possibilities, but my confidence in the management of the firms was low. It is a sad fact of life that marketing and promotion go more to making a product a success than the value of the product itself. Indeed, we've watched some very good products go down the tubes as a result of mismanagement.

With the entry of IBM into the micro field there will be a new rash of business opportunities. Just as the Radio Shack system has generated an industry dependent upon it—as you can see from the 160 or so pages of ads in this magazine—so we will be seeing a similar phenomenon with the IBM (and the Xerox, and so on). Some of these peripheral firms are already into the tens of millions of dollars in sales, so this "support" type of business can be very profitable.

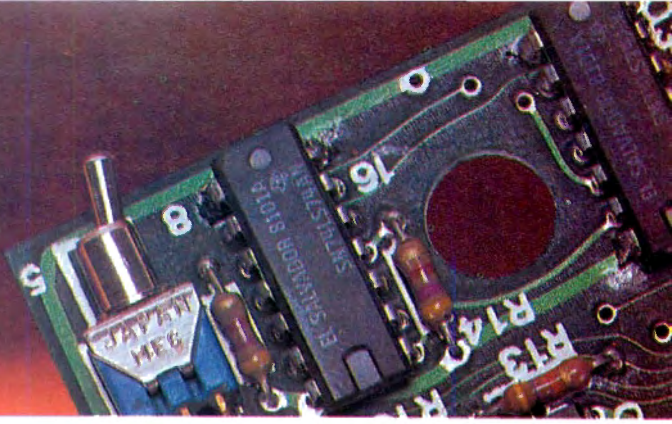
In recent weeks I've talked with a couple of small firms which have products or ideas which could, I think, quickly develop into massive money makers. Each need the money to get started with the promotion and production of their product.

Most of these firms need between a quarter and a half million dollars to really get going quickly. Each of three that I can think of are well established firms, but none have that amount of cash on hand for getting a new product out fast. The product sales should mushroom, when you consider the 300 percent yearly industry growth pattern which seems to be continuing and perhaps accelerating.

I'm going to look into the legal ramifications of setting up an investment venture capital fund for getting needed products and services started with each share at $1,000. The fund would own a share of the businesses and be on the board of directors, thus keeping an eye on the management of the firms. With some diversification, even if every investment didn't grow into a bonanza, it would be difficult to miss on all bets. At $1,000 a share, I'll bet there are a lot of people who would like to get in on the action. I'll let you know if this looks feasible.■

"It didn't hurt when Jimmy Carter. . . turned to word processing for his writing."

THE SWITCH



SWITCH TO 5"/8" DOUBLE DENSITY

LNDoubler 5/8

FEATURES

- 5- and 8-inch* disk drives
- Single- & double-density
- Any size and density in any mix
- Read Model I, II* and III disks
- 5- or 8-inch* system disk
- Single & double sided disk drives
- DOS+ 3.3.9 included, with Disk BASIC.
- 6 month warranty
- Up to 3.75 megabytes online
- Easy installation – plug-in & run
- Analog phase lock loop data separation
- Precision write precompensation
- Regulated power supply
- Guaranteed operation at 4MHz
- All contacts gold plated
- Solder masked & silk screened
- Runs under DOS+ 3.3.9, TRSDOS 2.3, NEWDOS 2.1, NEWDOS/80 1.0, LDOS, NEWDOS/80 2.0, and ULTRADOS
- Reads 40- and 35-track disks on 80-track drives
- FD1791 controller + your FD1771
- Fits Model I expansion interfaces
- Fits LNW expansion interfaces
- Track configurations to 80-tracks
- 5 inch disk storage increased to:
  161,280 bytes – 35-track SS/DD
  322,560 bytes – 35-track DS/DD
  184,320 bytes – 40-track SS/DD
  368,640 bytes – 40-track DS/DD
  368,640 bytes – 80-track SS/DD
  737,280 bytes – 80-track DS/DD
- 8 inch disk storage increased to:
  591,360 bytes – 77-track SS/DD
  1,182,720 bytes – 77-track DS/DD

SS: single-sided DS: double-sided
SD: single-density DD: double-density

COMPLETE – The *LNDoubler 5/8*, switches your Model I or LNW-80 into the most versatile computer you can own. The LNDoubler's switch allows you to boot from 5- or 8-inch system disks, and it's accessible from outside the interface. The *LNDoubler 5/8* comes with a double-density disk operating system (DOS+ 3.3.9), complete with BASIC and utility programs . . . ready to run your software NOW!

VERSATILE – Whether you want single-sided, double-sided, single- or double-density, 5- or 8-inch operation, complete versatility is here today! Any combination of 5- and 8-inch disk storage is possible with the *LNDoubler 5/8*. Each of your present 40-track, single-sided 5-inch drives will store up to 184,320 bytes (formatted storage) – that's an 80% increase in storage capacity for only half the cost of just one disk drive. With three 8-inch double-density, double-sided drives your Model I will have 3.75 Megabytes of online storage – that's more storage than a Model II or Model III!

ADVANCED – The *LNDoubler 5/8* is the most technically advanced, tested and reliable double-density board you can buy. The *LNDoubler 5/8* has more features, more options and more software support than any other product of its kind.

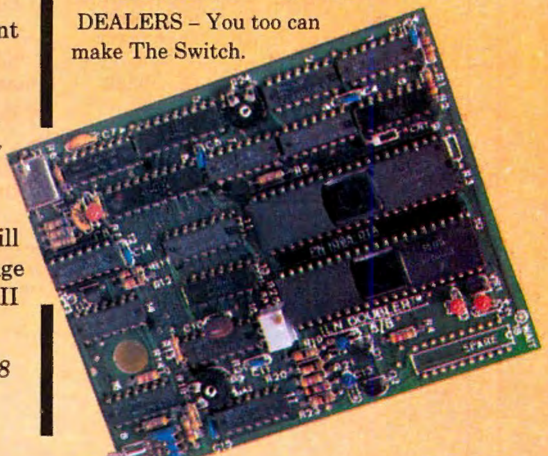
EASY TO INSTALL – The *LNDoubler 5/8* is easy to install. There are no traces to cut, no wiring to do, just a screwdriver and a few minutes of your time is all that is required. The instructions are fully illustrated for all interfaces. In minutes you will be 'up-and-running', and enjoying your computer as never before.

COMPARE – Compare features, compare quality, compare value, and make the **SWITCH** today!

Immediate delivery from stock – at your dealer **NOW** for only

$219.95

DEALERS – You too can make The Switch.



LNW RESEARCH CORPORATION

2620 WALNUT Tustin, CA. 92680 (714) 544-5744 (714) 641-8850

*8" drive operation requires special cable, 8" double-density requires 3.55MHz CPU speed-up modification or LNW-80 4MHz computer.

TRS-80 is a trademark of Tandy Corporation.

THE ASSEMBLY LINE

by William Barden, Jr.

"When we last left our wimpy hero, the Color Computer, he. . . was trying to pass an argument. . ."

Last month we covered some preliminary ground on the use of machine language subroutines in the Color Computer. This month we'll complete the course and send you your gold-embossed Certificate of Completion for 6809 machine language programming.

When we last left our somewhat wimpy hero, the Color Computer, he had successfully run some machine language programs and was trying to pass an argument to the machine language subroutine. As we tune in this month, we confront the floating-point accumulators.

There are two floating-point accumulators in page 0 of RAM (locations 0 through 255). They are designated FAC 1 (locations 79 through 84) and FAC 2 (locations 92 through 97). Each of these is six bytes long. They hold floating-point numbers, numbers that represent a form of "scientific notation." Let's review what we know about scientific notation. . .

Scientific Notation

Scientific notation uses a mixed number and power of ten to represent any size number. A somewhat standard form for Basic is shown by this example: 1.2E + 10.

	STA	$2000	B7	2000	183,32,0
	STX	$2001	BF	2001	191,32,1
	LDB	#6	C6	06	198,6
	LDY	#$2003	108E	2003	16,142,32,3
LOOP	LDA	,X+	A6	80	166,128
	STA	,Y+	A7	A0	167,160
	DECB		5A		90
	BNE	LOOP	26	F9	38,249
	RTS		39		57

Program Listing 1. Store A/C/FAC

A 10 is taken to the 10th power, or 100,000,000,000, and multiplied by 1.2. The actual number represented is 120,000,000,000. Negative numbers can also be represented, as in 5.333E – 5, which is 1/1,000,000 × 5.333 or .00000533.

To convert any number to scientific notation form, move the decimal point to the right of the first *significant* digit and count the number of places. The count becomes the power of ten, either positive (original number greater than one) or negative (original number less than one). The following are examples:

12345 = 1234.5E + 1 = 123.45E + 2 = 12.345E + 3 = 1.2345E + 4

.0000234 = 0.000234E – 1 = 0.00234E – 2 = 0.0234E – 3 = 0.234E – 4 = 2.34E – 5

The advantage of scientific notation is that arithmetic operations can easily be carried out on a range of numbers.

Basic uses a modified form of scientific notation internally for floating-point numbers. In place of powers of 10, powers of 2 are used. In place of a mixed number, a fraction is used for the mantissa. To see the form used, we'll use still another machine language program with a Basic driver.

Program Listing 1 shows the machine language program. Let's go through the instructions step by step: The first two instructions store A and X as before, into $2000 and $2001, $2002. The next instruction loads the B register with six. B is used as a loop counter for six passes through the loop.

The next instruction loads the Y register with a value of $2003. This value points to the $2003 area. The LDA instruction loads

```
ARGUMENT= 0
 0     0     79    0     128   0     0     0     0
ARGUMENT= 1
 0     0     79    129   128   0     0     0     0
ARGUMENT= 2
 0     0     79    130   128   0     0     0     0
ARGUMENT= 3
 0     0     79    130   192   0     0     0     64
ARGUMENT= 4
 0     0     79    131   128   0     0     0     0
ARGUMENT= 7
 0     0     79    131   224   0     0     0     96
ARGUMENT= 10
 0     0     79    132   160   0     0     0     32
ARGUMENT= 100
 0     0     79    135   200   0     0     0     72
ARGUMENT= 1000
 0     0     79    138   250   0     0     0     122
ARGUMENT= 10000
 0     0     79    142   156   64    0     0     28
ARGUMENT= 100000
 0     0     79    145   195   80    0     0     67
ARGUMENT= 1000000
 0     0     79    148   244   36    0     0     116
```

Fig. 1. FAC Values

```
ARGUMENT=-1
 0     0     79    129   128   0     0     0     128
ARGUMENT=-2
 0     0     79    130   128   0     0     0     128
ARGUMENT=-3
 0     0     79    130   192   0     0     0     192
ARGUMENT=-15
 0     0     79    132   240   0     0     0     240
```

Fig. 2. Negative FAC Values

the memory location pointed to by the X register. Since we entered the subroutine with X pointing to the FAC area, the first byte of the FAC (79) will be loaded into A. The plus sign means that the X register is incremented by one to point to 80. The STA stores the A register contents into the location pointed to by the Y register. Since we just loaded Y with $2003, the contents of 79 will be transferred to $2003. The Y register is then incremented by one by the plus sign.

Next, the count in B is decremented by one to five. If the count is not equal to zero, the loop is again executed. Each time through the loop, one more byte will be transferred from the FAC area to the $2003 area, until all six bytes are transferred. (We used six bytes on blind faith from the Extended Color Basic manual.)

The Basic driver is shown in Program Listing 2. This program transfers the decimal data form of the machine language code to the $2009 through $201C area. (The $2000 through $2008 area is

```
100 DATA 183,32,0,191,32,1,198,6,16,142,32,3,166,128,167,160,90,38,249,57
105 DEFUSR0=&H2009
110 FOR I=&H2009 TO &H201C
120 READ A
130 POKE I,A
140 NEXT I
145 INPUT X
150 A=USR0(X)
160 FOR I=&H2000 TO &H2008
170 PRINT PEEK(I);
180 NEXT I
190 PRINT
200 GOTO 145
```

Program Listing 2. Basic Program to Display FAC

reserved for A, X and the six bytes of FAC.) Next, the input X command is executed so we can use various values in the USR0 call.

After the USR0 call is made with X as the input argument, the area from $2000 through $2008 is printed. Typical values are shown in Fig. 1.

The first value represents the contents of A coming into the machine language program. It is always zero, indicating a numeric (rather than string) value. The next two values are the pointer in X coming into the machine language program. X always points to location 79, the FAC.

The next six bytes represent the number in floating-point format. The first byte of this is the power of two plus 128. What? The 128 bias is used to simplify arithmetic operations on the floating-point number, and additionally, to cloud programmers' minds. Look at an input argument of one. The exponent value is 129. A value of $129-128=1$, so the power of two represented is one.

The next four bytes are the fraction. An input value of three, for example, has 192 or 11000000 for the first byte, followed by zeroes. The fraction is therefore .110000 000000000000000000000000. (The fractional powers of two are 1/2, 1/4, 1/8, . . .). The floating-point form of three is therefore $130-128=2^2 \times .11=3$. The floating-point form of 100 is $135-128=2^7 \times .11001000=1100100=100$.

The last of the six values is the sign of the fraction. A value less than 128 is a positive sign, while a value greater than 127 is a negative sign. (This byte is a *signed* eight-bit value representing −128 through +127 in two's complement form.) Fig. 2 shows representative negative values for the FAC.

That is the format of numbers held in the floating-point accumulator. This does not mean we must work with floating-point numbers in the machine language subroutines. As a matter of fact, most processing will *not* be using floating-point numbers, but simple integer numbers that can be held in 16 bits. Why then the interface to floating-point numbers—Only God, Radio Shack, and Microsoft know the reason. This USR call is similar to the Model II, which also uses a pointer to a floating-point number. The Models I and III, however, pass only 16-bit integer values, which, to my mind, is much more convenient.

```
100 DATA 189,179,237,183,32,0,247,32,1,57
105 DEFUSR0=&H2003
110 FOR I=&H2003 TO &H200C
120 READ A
130 POKE I,A
140 NEXT I
145 INPUT X
150 A=USR0(X)
155 PRINT#-2,"ARGUMENT=";X
160 FOR I=&H2000 TO &H2001
170 PRINT#-2, PEEK(I);
180 NEXT I
190 PRINT#-2
200 GOTO 145
```

Program Listing 3.

INTCNV to the Rescue

Reading on in the Extended Color Basic manual, we find it's possible to call a ROM subroutine named INTCNV, which will convert the argument to a 16-bit value. The code in Fig. 3 shows this. JSR $B3ED calls INTCNV with X pointing to the FAC. The INTCNV subroutine converts the floating-point value in the FAC area to a 16-bit integer value and puts it in the D register (the A and B registers taken together). The STA stores A and the STB stores B into locations $2000 and $2001, respectively.

Program Listing 3 is the Basic driver for these four instructions. The equivalent data values are first POKEd into locations $2003 through $200C. The argument is then input, and a USR0 call is made. After the call, locations $2000 (A) and $2001 (B) are printed. Fig. 4 shows the A and B values after the INTCNV conversion. They represent the values in the USR0 call. The numbers are signed 16-bit values, in which the most significant bit is the sign (1 = negative, 0 = positive).

The two bytes make up 16 bits. To find the equivalent binary value, write the first value in eight bits, immediately followed by the next value in eight bits. 117, 48, for example, becomes 01110101 00110000. If the sign bit is zero, the number is correct as it stands −117,48 represents $7530, or 30,000. If the sign bit is one, change all the ones to zeroes, all the zeroes to ones, and add one to get the negative value represented (138,208 becomes 10001010 11010000, with the sign bit = 1). "Complementing" produces 01110101 00101111, and adding 1 gives a final result of 01110101 00110000, or 30,000, negated.

INTCNV is one way to pass an integer value to the machine language subroutine, albeit a somewhat roundabout way. Another way would be simply to POKE an eight-bit or 16-bit value into dedicated memory locations and then have the machine language code pick up the value directly. This would work well for multiple arguments. Make certain the dedicated location is in the protected area of RAM and does not overlay the program area.

```
      STA   $2000   B7    2000  183,32,0
      STX   $2001   BF    2001  191,32,1
      LDB   #5      C6    05    198,5
      LDY   #$2003  108E  2003  16,142,32,3
LOOP  LDA   ,X+     A6    80    166,128
      STA   ,Y+     A7    A0    167,160
      DECB          5A          90
      BNE   LOOP    26    F9    38,249
      RTS           39          57
```

Program Listing 4.

```
JSR  $B3ED  BD  B3ED  189,179,237
STA  $2000  B7  2000  183,32,0
STB  $2001  F7  2001  247,32,1
RTS         39        57
```

Fig. 3. INTCNV Example

```
ARGUMENT= 0
 0  0
ARGUMENT= 1
 0  1
ARGUMENT= 5
 0  5
ARGUMENT= 30000
 117  48
ARGUMENT= 32767
 127  255
ARGUMENT=-1
 255  255
ARGUMENT=-2
 255  254
ARGUMENT=-30000
 138  208
ARGUMENT=-32768
 128  0
```

Fig. 4. A + B Values after the INTCNV Conversion

Passing String Arguments

How about the case in which the argument passed by the USRn call is non-numeric? A non-zero value in A indicates a string argument will be passed. In this case, X will not point to the FAC area, but will point to a string parameter block.

The string parameter block is shown in Fig. 5. It is made up of five bytes. Byte 0 is the length of the string, up to 255 bytes. Byte 1 is reserved. Bytes 2 and 3 contain the address of the string. This may be in a Basic statement for strings, such as: 100 A$ = "SUCH STRINGS AS. . ." or the address may point to the string storage area at top of (non-protected) RAM. Byte 4 is reserved.

This column is loaded with programs,

BYTE 0	String Length
1	Reserved
2	Address
3	Of String
4	Reserved

Fig. 5. String Parameter Block

but I suspect you (and I) need all the 6809 programs we can get. Program Listing 4 shows another machine language program built along the lines of preceding ones. It stores A in $2000, X in $2001,$2002 and the string parameter block in $2003 through $2007.

The Basic driver for this machine language wonder is shown in Program Listing 5. As before, the data values representing the machine language code is transferred to $2008 through $201B. A test string of $A is entered and passed to the program via the USR0 call. The output argument B$ is a dummy string rather than a numeric variable.

After the USR0 call the string starting address SA is picked up from the parameter block copy at $2005, $2006. The value of A, X, and the parameter block is printed. SA is used to print the length of the string by a series of PEEKs, shown in Fig. 6. The first entry of Fig. 6 shows an A value of 255 (string variable), an X value of 31,149 (8085 = string parameter block location), and the five bytes of the string parameter block, length 22, string location 63, 129 or 16257.

The same scheme as above can be followed to find the location of Basic text strings. This is a handy feature if the machine language subroutine is searching or sorting strings, or if the strings are dummy strings containing data other than text, such as multiple arguments to be passed to the machine language processing.

Using VARPTR

We now come to a confusing paragraph in the six pages on the use of VARPTR. VARPTR in the other TRS-80s will point to a variable, string parameter block, or the first element of an array. For example:

```
100 B = VARPTR(AA)
110 C = VARPTR(ZX(0))
120 D = VARPTR(S$)
```

will set B equal to the location of variable AA, C equal to the location of the first element of array ZX, and D equal to the location of the string parameter block of string S$.

As it turns out, that is exactly the way VARPTR works in the Color Computer.

```
 255  31  149  22  0  63  233  0
STRING ADDRESS= 16361
STRING=TANDY TOWERS OVER ALL!

 255  31  149  30  0  63  181  0
STRING ADDRESS= 16309
STRING=HAVE YOU HUGGED YOUR CC TODAY?
```

Fig. 6. String Parameter Block Format

```
100 DATA 183,32,0,191,32,1,198,5,16,142,32,3,166,128,167,160,90,38,249,57
110 DEFUSR0=&H2008
120 FOR I=&H2008 TO &H201B
130 READ A
140 POKE I,A
150 NEXT I
160 INPUT A$
170 B$=USR0(A$)
180 SA=PEEK(&H2005)*256+PEEK(&H2006)
190 FOR I=&H2000 TO &H2007
200 PRINT PEEK(I);
210 NEXT I
220 PRINT
230 PRINT "STRING ADDRESS=";SA
240 PRINT "STRING=";
250 FOR I=1 TO PEEK(&H2003)
260 PRINT CHR$(PEEK(SA+I-1));
270 NEXT I
280 PRINT:PRINT
290 GOTO 160
```

Program Listing 5. Basic Driver for String Analyzer

```
        STA   $2000    B7    2000   183,32,0
        STX   $2001    BF    2001   191,32,1
        JSR   $B3ED    BD    B3ED   189,179,237
        TFR   D,X      1F    01     31,1
        STX   $2003    BF    2003   191,32,3
        LDB   #5       C6    05     198,5
        LDY   #$2005   108E  2005   16,142,32,5
LOOP    LDA   ,X+      A6    80     166,128
        STA   ,Y+      A7    A0     167,160
        DECB           5A           90
        BNE   LOOP     26    F9     38,249
        RTS            39           57
```

Program Listing 6. Store A/C/FAC

```
100 DATA 183,32,0,191,32,1,189,179,237,31,1,191,32,3,198,5,16,142,32,5,166,128
110 DATA 167,160,90,38,249,57
120 DEFUSR0=&H200A
130 FOR I=&H200A TO &H2025
140 READ A
150 POKE I,A
160 NEXT I
170 INPUT B
180 A=USR0(VARPTR(B))
190 FOR I=&H2000 TO &H2009
200 PRINT PEEK(I);
210 NEXT I
220 PRINT:PRINT
230 GOTO 170
```

Program Listing 7. Driver for VARPTR Analysis

The output or VARPTR *is* a pointer to the variable. The Extended Color manual states "X = USR0(VARPTR(A)) passes a pointer to the variable A." It isn't quite as easy as picking up the pointer in X on entry to the machine language subroutine thinking it will point to the variable.

For a USRn call with a numeric call, the X register always points to the FAC area at location 79. That's true even if VARPTR is used, as in: 100 X = USR0(VARPTR(A)).

The pointer is converted from a 16-bit integer value into a floating-point number in the FAC area. If VARPTR(A) is 7988, indicating that variable A is located at location 7988, the 7988 will be converted into floating-point format and put into the FAC at 79. To find the VARPTR location we must use INTCNV to convert the FAC value to an integer value. After we do this, the D register (A and B) will hold the address of the variable.

Sure enough, we have another machine language program to show you how this works. Program Listing 6 stores A into $2000 and X into $2001,$2002 upon entry to the machine language subroutine. INTCNV is called at $B3ED to convert the FACX value to integer. The integer value in D is transferred to X and stored in $2003, $2004. B is then loaded with a loop count of five and Y with a pointer to $2005.

The loop is executed five times. Each time through the loop, the byte pointed to by X is stored into $2005 through $2009, transferring the five-byte variable into the $2005 area. At the end of the subroutine, we have stored data as shown in Fig. 7.

The Basic driver for this program is shown in Program Listing 7. As usual, the data making up the program is first transferred to $200A through $2025. Next, variable B is input. The machine language subroutine is called by USR0(VARPTR(B)). The data in $2000 through $2009 is then printed.

When the above program is run with sample values (Fig. 7) for B, we find some interesting results. A is always zero, indicating a numeric variable. X always points to location 79 on entry to the machine language subroutine. The next two values are the location of B in RAM. The two bytes of 31,52 are 00011111 00110100, or 7988, the location of variable B. The next five bytes are the floating-point representation of the variable. As the manual states, "floating-point variables are stored in the variable table in a slightly different format than they are stored in the FAC."

The format here is very similar to the FAC format. There are five bytes rather than six; the last byte represented the sign in the FAC format. The sign, however, can be represented in one *bit*! The sign bit becomes the first bit in the mantissa, replacing the first mantissa bit. Since this first bit is always a one, we haven't lost any information! But why is the first bit always a one?

Every floating-point number in the variable table is "normalized." Normalization simply means adjusting the number until the first fractional bit is one, similar to our scientific notation normalization; this is done by shifting right or left, with each shift adding or subtracting one to the exponent. The format of variables, therefore, has all the precision of the FAC, but is a little harder to read.

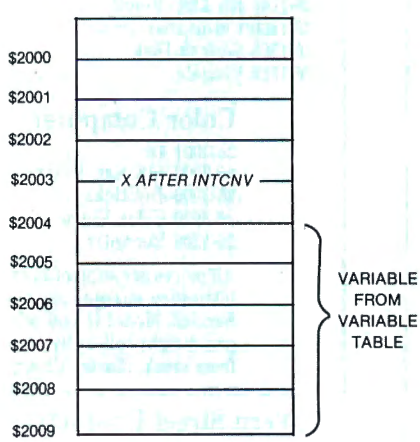


Fig. 7. Store Variable Program Storage

Consider the value of B = 100, in Fig. 8. The exponent is 135. When the bias of 128 is subtracted, the power of two is seven. The fractional part is 72, or 01001000 000. . .00. The sign bit is the first fractional bit, or zero (positive). When the one bit that should be there is added, the fraction becomes 11001000, or 1/2 + 1/4 + 1/32, or 200/256. Two to the seventh is 128 and 128 x 200/256 is 100.

Negative values must be two's complemented as in previous examples.

When VARPTR is used with string variables, the pointer in X is to a string parameter block as previously discussed.

Passing Arguments

The last page and a half of the six pages discuss passing arguments back from the machine language subroutine. The Basic interpreter assumes that the FAC area will hold the argument passed back.

```
100 A = USR0(1000)
110 PRINT A
```

In the above example, A will be set equal

```
B= 0
 0  0  79  31  52  0    0    0  0  0

B= 1
 0  0  79  31  52  129  0    0  0  0

B= 2
 0  0  79  31  52  130  0    0  0  0

B= 3
 0  0  79  31  52  130  64   0  0  0

B= 100
 0  0  79  31  52  135  72   0  0  0

B=-1
 0  0  79  31  52  129  128  0  0  0

B=-2
 0  0  79  31  52  130  128  0  0  0

B=-3
 0  0  79  31  52  130  192  0  0  0

B=-100
 0  0  79  31  52  135  200  0  0  0
```

Program Listing 8. Divide by Two Program

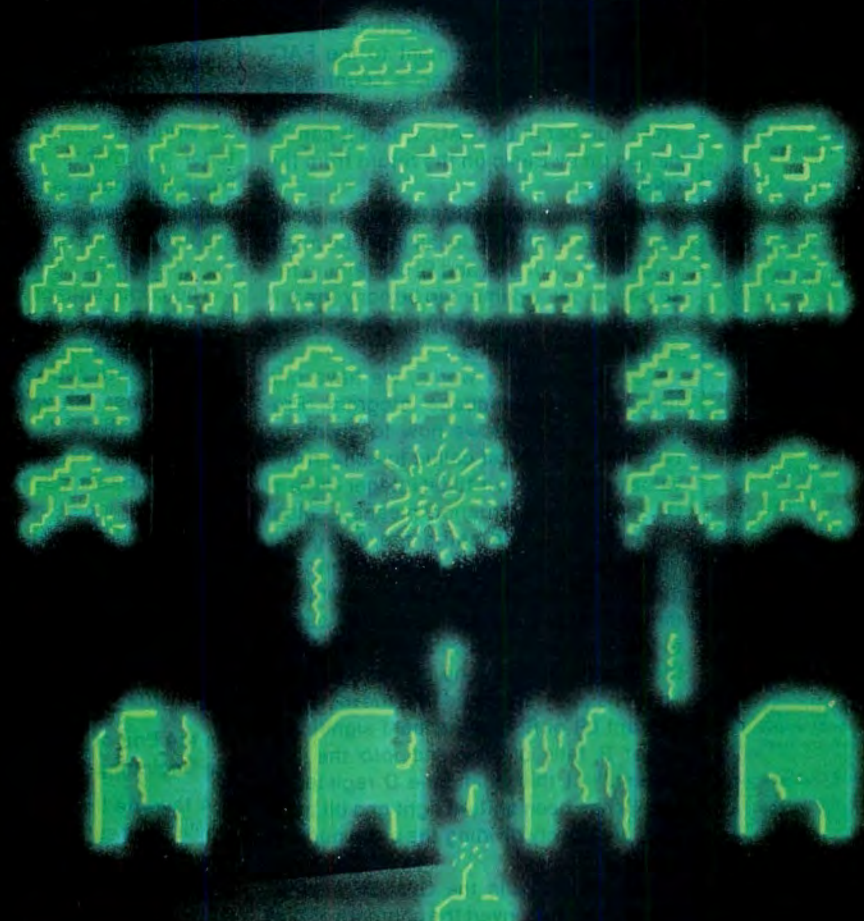
Reader Service for facing page ✓224→

Even if you have one of the other versions on the market, you'll still consider this program a MUST for your collection.

SPACE INTRUDERS

10940

SCOTT ADAMS SEAL OF APPROVAL ™



WE BELIEVE THIS IS THE BEST PROGRAM OF ITS KIND. SHOULD YOU BE DISSATISFIED IN ANY WAY, RETURN IT TO ADVENTURE INTERNATIONAL FOR A COMPLETE REFUND.

Adventure INTERNATIONAL
A DIVISION OF SCOTT ADAMS, INC.
BOX 3435, LONGWOOD, FL 32750
(305) 862-6917
WRITE FOR OUR FREE CATALOG OF 150 PACKAGES

SPACE INTRUDERS IS FOR THE

TRS-80 MODEL 1 & 3 TAPE 16K	010-0116	$19.95
TRS-80 MODEL 1 & 3 DISK 32K	012-0116	$20.95

TO ORDER: SEE YOUR LOCAL DEALER, IF HE DOES NOT HAVE THE PROGRAM YOU WANT ... THEN CALL; TOLL FREE 1-800-327-7172 (Orders Only) OR WRITE US.

THE SPECTACULAR ARCADE VERSION

THE ASSEMBLY LINE

to 1000, as the Basic interpreter picks up 1000 from the FAC.

As I mentioned previously you will probably not be working with floating-point numbers in your machine language programs. You will typically be picking up eight and 16-bit integer arguments, processing them, and returning eight or 16-bit integer arguments. The question, then, is how to return an argument to the FAC. Fortunately, INTCNV has a counterpart that converts an integer into a floating-point number in the FAC, where it can be picked up by Basic on the return from the USRn call.

The subroutine is GIVABF, and is located in ROM at $B4F4. A call must be made to GIVABF with D (A and B) containing the 16-bit integer argument to be converted. If the argument is only eight bits, A is zeroed and B holds the value.

To see how this works, we have one final machine language program. Program Listing 8 shows how to divide by two. INTCNV is first called to convert the value in the FAC to a 16-bit integer value in D. The LSRA instruction means Logical Shift Right A, and shifts the A register right one bit position. A zero goes into the most significant bit, and the contents of the least significant bit are shifted out into the carry condition code. RORB rotates the B register right one bit position. The bit in the carry is rotated into the most significant bit of B, and the least significant bit of B falls off the end into the "bit bucket." At this point, the D register (A and B) has been shifted right one bit position, effectively dividing the entry number by two.

JSR $B4F4 calls the GIVABF subroutine in ROM to convert the contents of D to a floating-point number in FAC. The RTS then returns. Program Listing 9 is the Basic driver for this program. It simply transfers the data values as before, asks for input, calls the machine language subroutine, and prints the output value.

Passing Back Multiple Arguments

The six pages finish with "passing back multiple arguments" and cautions about the use of string variables on output. I don't know how practical it would be to pass back modified variables, as again, we would primarily be concerned with integer arguments. Probably the reserved block approach is best here—dedicate a block of memory for input and output arguments, and use POKEs for input arguments and PEEKs for output arguments.

The comments on string use are excellent, and if the cautions are followed, machine language code can process strings nicely in sorts and searches. Bear in mind that strings, like variables, move! Make certain VARPTR is used directly in the USRn call; if it is not, a garbage collection routine in Basic may have shuffled around the string locations. Also be wary of changing strings within Basic programs. Basic lines terminate on a zero, and returning a zero value in a dummy string may wreak havoc upon return from the USRn call. In general, I wouldn't recommend using strings for passing output arguments.

```
JSR    $B3ED   BD    B3ED   189,179,237
LSRA           44           68
RORB           56           86
JSR    $B4F4   BD    B4F4   189,180,244
RTS            39           57
```

Program Listing 8. Divide by Two Program

```
100 DATA 189,179,237,68,86,189,180,244,57
110 FOR I=&H2000 TO &H2008
120 READ A
130 POKE I,A
140 NEXT I
150 DEFUSR0=&H2000
160 INPUT A
170 B=USR0(A)
180 PRINT B
190 GOTO 160
```

Program Listing 9. Basic Driver for Divide by Two Program

It's the End of Him, But Just the Beginning for Us...

A favorite line from a Grade C monster movie... Yes, "Assembly-Line" readers, this is my last column. I've enjoyed writing very much, and sincerely appreciate the many letters I've received (except for the one from Manny, who sent me a 250-page listing with a demand to "straighten out his eight-dimensional string sort"). You'll see me again in some other articles—there's a lot of exciting things happening and I intend to keep writing. Keep on assembling... ■

The editors of 80 Microcomputing *regretfully say goodbye to Bill Barden as author of "The Assembly Line." The Assembly Line will be kept rolling in a new column that will begin next month, to be written by Roger Fuller (*Supermap *author). To keep our advanced Assembly programmers happy, we are planning a new column, to begin in January, which will be written by a variety of programmers.*

And we don't plan to let Bill go without a struggle. We have him on line for a feature article on Color Graphics, which will run in our March Color Graphics issue.

By Sparky Starks from Adventure Int.
As mercenary and galactic police officer, you must maintain the condition and control of all parts of your spacecraft. You sit at the controls while peering out of the digital space-view port. Suddenly something appears on your screen: is it a Starpirate or a friendly merchant ship? You can't tell yet, and at this speed you may have only a fraction of a second to make an attack/no attack decision.

Model I & III, 16K tape...$24.95
Model I disk version......$29.95

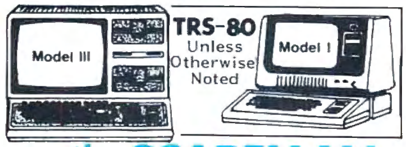
MISSILE ATTACK

By Philip Oliver from Adventure Int.
You must use your twin silos of ABMs to fend off barrage after barrage of enemy missiles that rain down toward your cities. As your skill increases so does the difficulty and speed of this machine language arcade game. Watch the skies and may your aim be true! MISSILE ATTACK has sound and fast-moving graphics galore.

16K tape...$14.95 32K disk...$20.95

By Wall & Moncrief from Adventure Int.
You get a vast lunar landscape, graphically depicted in both long range and close up, with many choices for landing sites. Choose a more difficult site and get more points -- if you can land successfully. You have complete control of your LEM via main engines and small side thrusters, and a successful landing is heralded with a flag raising ceremony. Great graphics and sound add to the realtime challenge and fun.

16K tape...$14.95



NEW SCARFMAN

From Cornsoft Group
Action-filled arcade game that pits you against the monsters. Race your Scarfman around a maze, gobbling up scoring dots. You are pursued by five monsters: if you eat a "+" they'll lower their eyes and you can eat them, otherwise they'll eat you!

With exciting graphics and sound, SCARFMAN may be played using the keyboard or Alpha Product's Joystick. WARNING: MAY BE HABIT-FORMING!

Tape...$15.95
Disk (specify mod. I or III)...$19.95

INVADERS FROM SPACE

By Carl Miller from Acorn
A fast machine language approach to this classic (and addictive) space game. The aliens drop bombs, move around, and try to overrun your bases.

You choose the speed, enemy bomb frequency and accuracy, and how many shots and bases you have. Unlike other games of this type, you can move your base and simultaneously fire at the invaders. Fun for all ages and skill levels, it has full sound effects for even more excitement.

16K protected tape....$14.95
16K protected disk....$20.95

SUPER NOVA

By Hogue & Konyu from Big Five
Asteroids surround your ship. You must shoot the asteroids, as well as any of the five types of alien spaceships. Use your thrusters for full movement and rotation of your ship -- if you are overwhelmed, you can even jump to hyperspace! Written in fast machine code with superb graphics, this game is GREAT!

16K tape $15.95 32K disk $19.95

By Steven Kearns from Acorn
Gigantic antimatter rocks appear on the Tactical Display Screen of your spacecraft. You blast away with lasers and they just explode into smaller chunks. To score in this fast arcade game with sound, you must destroy the rocks. To stay in the game at all, you must avoid them!

To add to your woes, time bombs appear periodically. If their timers reach zero -- BOOM! And if that's not enough, the aliens will be glad to send out some spaceships loaded with antimatter torpedoes. Fire thrusters to move, shoot laser cannon, jump to hyperspace -- anything to avoid the onslaught. One or two players can compete, with nine levels of difficulty.

16K protected tape...$19.95
16K protected disk...$19.95

By Hogue & Konyu from Big Five
One of the top names in TRS-80 arcade games adds a new dimension: voice sound effects! It's you against the robots in this fast-moving shoot-em-up. Electrified Mazes and the "Flagship" complicate things as you stalk the evil androids.

The innovations built into ROBOT ATTACK take your TRS-80 near the limits of its capabilities. You MUST see and hear it!

16K tape $15.95 32K disk $19.95

COSMIC FIGHTER

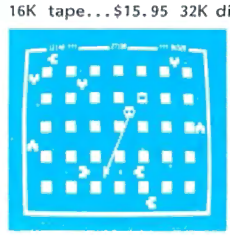
By Hogue & Konyu from Big Five
Terrific sound, graphics and unique challenges mark this space game a winner! While fighting off the alien convoys -- each more skillful than the last -- you must keep track of your rocket fuel or risk explosion. Finally your space station appears. Can you dock immediately, or is the station overrun by aliens? Find out by ordering Cosmic Fighter today.

16K tape...$15.95 32K disk...$19.95

METEOR MISSION 2

By Hogue & Konyu from Big Five
Six astronauts are stranded on a desolate planet. You must undock from your command module and maneuver your rescue shuttle through the asteroid field to save them. You can only save one at a time, and each landing burns away parts of your landing sites. Order this realtime action game now or live with the astronauts' pitiful screams forever.

16K tape...$15.95 32K disk...$19.95



ATTACK FORCE!

By Hogue & Konyu from Big Five
Unlike the usual "shoot-em-ups," Attack Force lets you control both speed and direction as you maneuver all over the screen in search of the alien Ramships and Flagships. Enemy ships chase you everywhere, and the Flagships' lasers can fire in any direction! The Ramships can even impersonate your spacecraft, so don't look away even for an instant. Machine language action with sound.

16K tape...$15.95 32K disk...$19.95

TO ORDER CALL TOLL FREE 800 424-2738 For information Call (202) 363-9797

THE PROGRAM STORE

✓ 17

The Program Store carries more than 1000 programs for TRS-80, Apple, and Atari. If the program you want is not listed here, call us: we probably have it in stock!

MORE-»

THE PROGRAM STORE

For SCRIPSIT

SUPERSCRIPT

Now you can easily print **BOLDFACE**, underlined text, slashed Ø's, special characters, superscripts & subscripts (A^c X $\log_e[A_1-A_n]$) and more!

By Richard Wilkes from Acorn
Using your SUPERSCRIPT modified SCRIPSIT Word Processor and a compatible printer, you can now underline, boldface, insert text during printout, slash zeros, set type pitch, subscript and, of course, superscript! You can even read your directory and kill files without ever leaving SCRIPSIT.

Includes drivers for popular printers and easy instructions. For TRS-80 model I (requires 32K, 1 drive, SCRIPSIT/LC) or model III (requires 32K, 2-drives for initial transfer, model I version of SCRIPSIT/LC).

$49.95

PROOFREADER™

From Soft-Tools
Computer may not make misteaks, but people do. That's why you need PROOFREADER if you use your computer for word processing. In about five minutes, this program can check any word processing disk file against its dictionary of more than 38,000 words. Any words that don't agree with the dictionary are listed on the screen and, optionally, to a disk file for later retrieval.

For words words you use that aren't in PROOFREADER's vocabulary, an easy-to-use auxiliary file is provided. Five minutes with PROOFREADER can save hours of time proofreading.

Model I, 32K disk (1-drive min.)...$54.00
Model III, 32K disk (1-drive min.)..$64.00

GRAMMATIK

BEYOND SPELLING CHECKING

From Aspen Software
Actually checks your word processing document for common grammatical errors. GRAMMATIK comes with its own error dictionaries which can be expanded to encompass additional grammar use of your choice. It will pick up errors of usage, wordiness, redundancy, punctuation, and much more -- including use of "sexist" terms! It even offers suggestions for change. A versatile utility, this program belongs right alongside PROOFREADER in your word processing toolkit.

32K minimum (48K optimum)
Model I: $49.95 Model III: $59.95

ACCEL 2

BASIC COMPILER

From Allen Gelder
Turns your BASIC program into a machine language/BASIC hybrid that may run many times faster. For those who plan to sell their programs, compiling by ACCEL 2 offers the additional advantage of protection: the source code and REMarks are not included in the compiled version.

While all compilers may require some modification of the BASIC program (usually because of improper structuring), we have found that ACCEL 2 requires the least, and even works with program "tricks" like string-packing, etc.

ACCEL 2 works with models I or III, requires a minimum amount of memory, supports either disk or tape (with TSAVE, optional at $9.95), and does not require extensive rewriting of your BASIC programs. Unlike other compilers, no royalty is required when selling ACCEL-compiled programs.

Supplied on tape for 16-48K...$88.95

PACKER

From Cottage Software
Packer's five commands allow tremendous control over the readability and efficiency of your BASIC programs. Specify "PACK" and the program will compress text into multiple statement lines. This really speeds up storage, load, and execution time. PACKER is also useful in compressing code to be compiled by ACCEL 2 (above).

Also included are four other handy utilities: "MOVE" lets you relocate program lines, "RENUMB" allows program renumbering, "SHORT" deletes unnecessary words and REMarks, and "UNPACK" separates multi-statement lines to ease editing.

16K, 32K & 48K tape...$29.95

BOSS

By V. Hester from Soft Sector
This utility is the perfect tool for creating and debugging BASIC programs. It allows single stepping through the Basic program, setting up to five breakpoints within the program and tracing of program logic using only a small portion of the display screen. With BOSS, you can review selected variables during program execution and return to the program with the display restored. Allows storing programs in high memory for later retrieval. For Level II, TRSDOS, NEWDOS+, NEWDOS/80, VTOS.

Model I or III (specify) tape: $18.95

DDT Disk Drive Timer

GRAPHIC DISPLAY OF MOTOR SPEED
DRIVE NO.: 0 RPM RANGE: 10
EACH MARK REPRESENTS 0.17 RPM.
(SLOW) (CORRECT) (FAST)
295.00 296.67 298.33 300 301.67 303.33 305.00

from Disco-Tech
Analyze and adjust your disk drive motor speed with a realtime graphic display. Manual details use for virtually all popular drive units. Avoid the many errors that improper drive timing can cause. All you need is DDT, two screwdrivers and five minutes.

Model I disk...$19.95

LDOS

From Lobo

"...LDOS is by far the best disk operating system for the Model I TRS-80."
Paul Wienner,
June 80-MICROCOMPUTING

"...if you are a serious programmer, you should have LDOS."
George Blank, Editorial Director
September CREATIVE COMPUTING

A totally new and advanced operating system that offers benefits to everyone from novice to advanced system programmers. LDOS comes with one of the most complete and well written documentation books ever, with easy-to-follow instructions to get started plus a wealth of supporting information for when you need it. This, plus a toll-free customer service number means that LDOS supports you as no one has before!

Completely device independent, LDOS provides for routing, linking, setting, and filtering of the input/output of a number of peripherals. Up to eight disk drives are supported in almost any mixture: 5", 8", single/double density, up to 80 tracks, even hard disks.

"Job Control" can end the tedium of repetitive keystrokes needed to call up and manipulate programs. Once you tell LDOS what it needs to do to get you up and running, it executes the series thereafter with a simple command. And LDOS notes the date all files were created or modified -- even flagging files that haven't been backed-up after modification.

We've just scratched the surface of LDOS' capabilities, and we'll gladly send you a pamphlet that goes into more detail. But better yet, order LDOS now and begin enjoying the power that this system will bring to your computer use.

Disk with manual: $169.95
Specify model I or III

BASIC Programming Asst. ✓156

From Instant Software
What an aide to the writing and debugging of BASIC programs! With BPA you can list out your program variables, with a notation of variable type, whether each is an array or not, and a notation of line numbers where the variable occurs. What's more, BPA even tells you the line numbers where any variable changes value! You can also produce a cross--reference to all GOTOs, GOSUBs, and IF..THEN statements for easy visualization of program flow. Want to find that last INPUT statement? BPA will give all line numbers where selected BASIC keywords reside. Save time and frustration by using BPA when you program.

Tape for 16K, 32K & 48K...$14.95

Visit our other stores: Seven Corners Center · Falls Church, VA & W. Bell Plaza · 6600 Security Blvd. · Baltimore, MD

TO ORDER CALL TOLL FREE 800 424-2738 For information Call (202) 363-9797

THE PROGRAM STORE ✓17
4200 Wisconsin Avenue NW, Dept. E11 Box 9609
Washington, D.C. 20016

MAIL ORDERS: Send check or M.O. for total purchase price, plus $1.00 postage & handling. D.C. residents, add 6% tax. Charge card customers: include all embossed information on card.

«-MORE

Get the most from your micro with software and accessories from one of the world's largest selections.

The Program Store

CALL TOLL FREE 800 424-2738

Just Released

ASYLUM

From Med Systems
You are sitting alone at 2 AM. Your eyes are bloodshot as you peer into your computer's screen and cry, "I must be CRAZY!" If this has never happened to you, you've never tried ASYLUM. It's Med Systems most ambitious 3-D graphics adventure yet!

ASYLUM places you on a cot in a small (padded?) room. Periodically the janitor lobs a hand-grenade through the window. What you do next could mean escape -- or disaster.

16K tape...$14.95

DEATH-MAZE 5000

From Med Systems
A new breed of adventuring! Venture through a graphically represented 3-D maze, with halls that could dead end -- or recede to infinity. Step through the doors or drop into the pits. Will you encounter monsters and mayhem, or will you be treated to useful objects and information? Will you ever get out alive?

16K tape...$14.95 32K disk...$19.95

LABYRINTH

From Med Systems
A nightmare of an adventure in graphically depicted three dimensions. Corridors stretch toward infinity right on your TRS-80 screen as you search this maze for treasures. If you get the feeling you're not alone, it's because you're not!

You use the arrow keys, plus two-word commands to move, manipulate objects and avoid the many pitfalls (pun intended) that await you in Labyrinth.

16K Tape...$14.95

MICROWORLD

From Med Systems
One of the strangest adventures yet, with wierd-sounding rooms and baffling tools n' treasures -- it all takes place inside your TRS-80! You're in the keyboard, exploring the inner structure. Go from chip to chip as if you were a piece of data trying to get from one end of the bus to the other. Fascinating and instructive.

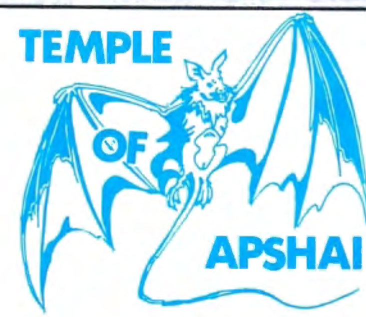
16K tape...$19.95 16K disk...$24.95

FORBIDDEN PLANET I

By Wm. Demas from Fantastic Software
The first TALKING adventure! With skill, luck, and tenacity -- and a little help from your chatty TRS-80 -- you may survive Part One of this multipart adventure! You don't need a voice synthesizer, this program talks to you via the cassette port. And it's a good thing it does, 'cause otherwise you'd get mighty lonesome on desolate FORBIDDEN PLANET.

48K disk...$39.95

TEMPLE OF APSHAI



TEMPLE OF APSHAI
The first of the DunjonQuest series, and still one of the most popular. In exploring over 200 rooms in the magical labyrinth, you will encounter more than 30 kinds of fearsome monsters guarding over 70 treasures. Some of the treasures will help you in your quest, but you must still watch out for the many monsters and traps that spring out from the walls and shadows.

16K tape or 32K disk...$39.95

LOWER REACHES OF APSHAI: Four more levels. Requires above program. $19.95

HELLFIRE WARRIOR
Sequel to APSHAI, this thriller gives you four levels to explore with sixty rooms each. New monsters, new traps, new challenge!

16K tape or 32K disk...$39.95

KEYS OF ACHEREON: Four more levels. Requires above program. $19.95

EVEREST EXPLORER

By Godwin & Knowlton from Acorn
Not for everyone. One reviewer said, "...don't bother with Everest Explorer." Another commented, "It holds your attention for quite a while and I have yet to get bored with it."

Most people here love it. This is a game of logistics in which you try to lead a team up Mount Everest. If your skill, the weather, and luck hold out, you'll make it. But remember, you'll also have to get back down safely.

16K protected tape...$19.95
32K protected disk (with "save game" feature)...$19.95

LOST COLONY

By David Feitelberg from Acorn
The scene is the world's first attempt at colonizing a planet in deep space. An election was held for an economic manager to straighten things out, and guess who won!

You must either make things better or be voted out of office in shame. A remarkable simulation, LOST COLONY arms you with maps and charts as tools for resource management. You assign human and robotic labor forces, explore new land, and allocate production quotas. Input to the computer is free form: you can use terse commands or full sentences. A challenging game, LOST COLONY might give you insight into real life management as well.

16K protected tape...$19.95
32K protected disk...$19.95

Special Save $10

MICROSOFT ADVENTURE

Prepare yourself for the adventure of your life as this classic fantasy/logic game takes you into the world of the Colossal Cave. Your computer is your guide as you search for treasures, solve puzzles, explore, and avoid the dangers that lurk within.

Complete version of the original Adventure, originally written for the DEC PDP-10 in FORTRAN. The program has been translated to bring you all the enjoyment in your home computer.

Model I, 32K disk...~~$29.95~~ Special

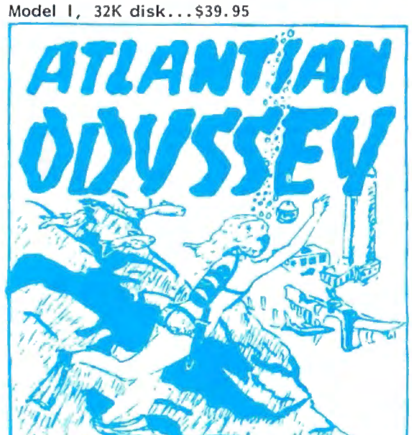
SPECIAL SAVE $10: $19.95 thru 12/31!

ZORK

By Infocom from Personal Software
In Zork, the Great Underground Empire, unearthly creatures guard 20 treasures. Bring all the treasures back to the trophy case and you can leave alive! You must pick your way through intricate mazes, collecting objects that may help or hinder you in your quest. You may use standard 2-word responses or speed things up by using multipart statements. But keep your wits about you, because in Zork, they take no prisoners!

Model I, 32K disk...$39.95

ATLANTIAN ODYSSEY



From Interpro
An illustrated adventure for the TRS-80. It's said that "a picture is worth a thousand words," and in this program you'll see why. While still in the classic text-type mold, the graphics give you a new perspective and aid your sense of direction. This saga of the sea contains a 150 word vocabulary and depicts 32 graphic locations.

Model I 48K disk...$29.95
16K tape (Text only, No graphics)..$14.95

Prices Subject to Change

THE PROGRAM STORE · Dept E11 Box 9609 · 4200 Wisconsin Ave, NW · Washington, D.C. 20016 17

Item ____ Price ____

Postage $1.00
Total ____

☐ CHECK ☐ VISA
☐ MASTERCARD MC Bank # ____

name ____
addr ____
city ____ state ____ zip ____
Card # ____ Exp ____

MORE→

THE PROGRAM STORE — CALL TOLL FREE 800 424-2738

NEW

VOYAGE TO VALKYRE

By Leo Christopherson from AOS
Combine the animation and music techniques pioneered by Christopherson with the challenge of his first fast-moving arcade game and you have VOYAGE TO VALKYRE!

You speed through a magical maze guarded by ferocious birds that swoop down to attack if you don't get them first. To list all the play and options of this exciting game would take the 16 pages of instruction included.

16K tape...$34.95
16K disk...$39.95

DUEL «N» DROIDS

By Leo Christopherson from Acorn
Your 'droid has already learned NIM, so now it's time to teach it how to wield a laser sword! Starting out as a lowly clown, you teach it how to use a laser sword by controlling its movements -- advance, attack, even retreat if necessary. After training it to be a "Grand Master," you enter the tournament against the program's skilled 'droid. Revel in the fanfares of the victorious -- or hear the funeral dirges of the defeated! Entertainment for all ages.

Protected tape...$14.95
Protected disk...$20.95

Our Most Popular

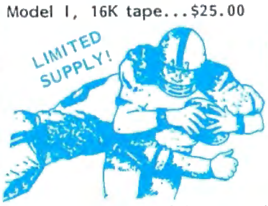
Unbelievable Realtime 3-D Graphics!

FLIGHT SIMULATION

From Sub-Logic
The wait is over! If 3-D graphics seem impossible on the low resolution TRS-80, you haven't seen this brilliant program. During FLIGHT SIMULATION, you instantly select instrument flight, radar, or a breathtaking pilot's-eye-view. But be sure to strap yourself in -- you're liable to get dizzy!

Once you put in some air time learning to fly your TRS-80, head for enemy territory and try to bomb the fuel depot while fighting off five enemy warplanes. Good Luck!

Model I, 16K tape...$25.00



PRO-PIX '81

By Tally-Ho from Adventure Int.
This ingenious program provides a computer projection of the winners and point-spreads for the 1981 professional football season. Each week you enter the current scores into this self-modifying program, and it updates the strengths and weaknesses of each team. Forecasts are designed to become more accurate as more data is accumulated.

32K disk...$24.95

NEW

TANKTICS

By Chris Crawford from Avalon Hill
In this combination computer and board game, you control up to eight German tanks or anti-tank guns against the computer's Russian forces. It all takes place on the eastern front of WWII and rages through a landscape of forests, lakes, roads, and rough terrain. To beat the computer you must develop good armored tactics and judgement, but have a willingness to take risks.

16K tape including colorful wargame board...$23.95

By John Allen from Acorn
You have to be fast to keep up with the action as you try to outscore your opponent in five minutes of one-on-one basketball against a friend or your TRS-80 model I or III.

16K protected tape...$14.95
16K protected disk...$20.95

Game of the Year

OLYMPIC DECATHLON

By Timothy Smith from Microsoft
The graphics capabilities you were promised when you bought your computer are finally utilized in this marvelous series of programs. Just like the real Decathlon, you compete in 10 demanding games that encompass different forms of running, jumping and throwing. Play alone or with as many as eight competitors.

You MUST see this system in action. Otherwise, you simply won't believe the combination of outstanding graphics, fast-paced action, nail-biting intensity, and even a touch of comedy you'll experience with Olympic Decathlon!

Model I, 16K tape..$24.95 32K disk..$24.95

NEW

By John Allen from Acorn
More features, thrills, and sound than even John Allen's famous PINBALL. Once you load ASTROBALL into your TRS-80, the arrow keys become flipper buttons, the screen becomes the play board, and you become the "Pinball Wizard!"

A flying saucer, spaceships, meteors, and black holes add to the fun as your ball realistically zings around the board. ASTROBALL will have all your family and friends lining up for the pinball action and challenge. Five skill levels.

16K protected tape...$19.95
16K protected disk...$19.95

PINBALL

By John Allen from Acorn
Get your flipper fingers ready for action in this real-time, machine language game.

Lots of sound and flashing graphics make this fast action game so much like the real thing that you'll have to remind yourself not to shake your TRS-80. Choose from five playing speeds to match your skill. Can you beat your friends' scores? Will you avoid the infamous "Bermuda Square?" Get PINBALL today and find out.

16K protected tape...$14.95
16K protected disk...$20.95

NEW

Crush, Crumble and Chomp!

From Epyx
It's a monster movie, and you are the monster! You can be The Glob, Kraken, Mantra, Mechismo, Arachnis, or Goshilla -- or even design your own "custom" monster (disk version only). This hilarious action game is loaded with graphics and sound as you practice your villany. With 6 monsters, 4 cities, and 5 game objectives, you get a choice of more than 100 possible scenarios. A monster's life is not all carnivorous crunching, though: The combined resources of the police, science, and armed forces are bent on your destruction.

16K tape or 32K disk...$29.95

Visit our other stores: Seven Corners Center · Falls Church, VA & W. Bell Plaza · 6600 Security Blvd. · Baltimore, MD

TO ORDER CALL TOLL FREE 800 424-2738 For information Call (202) 363-9797

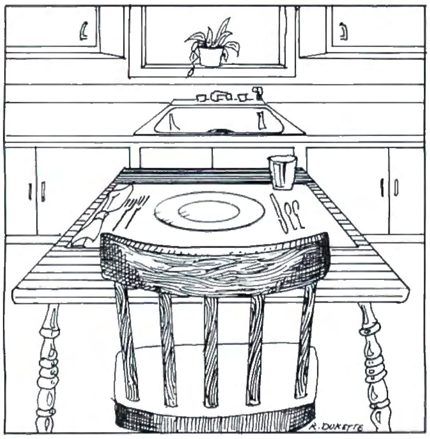
MAIL ORDERS: Send check or M.O. for total purchase price, plus $1.00 postage & handling. D.C. residents, add 6% tax. Charge card customers: include all embossed information on card.

THE PROGRAM STORE ~17
4200 Wisconsin Avenue NW, Dept. E11 Box 9609
Washington, D.C. 20016

«- MORE

News From KITCHEN TABLE SOFTWARE, INC.

by David Busch



The TLS-8E microcomputer—flagship of Kitchen Table Inc.—has reached full maturity with the recent announcement by that fictitious firm that a new comprehensive line of retrofixes, enhancements and optional repairs for the model's design defects will be offered later this year.

The new upgrades are welcomed by all serious S-80 buffs. Astute observers will remember that as long as Radio Shack was able to keep creating cassette loading modifications, buffered expansion interface cables and mods, lowercase fixes, and other corrections, the TRS-80 Model I remained a popular, viable product. It was only after the computer was semi-perfected it became necessary to introduce the Model III and a corresponding new round of debugging.

So I viewed with great pleasure the 213-page addenda, errata and corrections manual arriving at my home late last week. It meant the United States' only Z79-A-based microcomputer was gaining new ground. I quickly sifted through the mass of data to glean the items of most interest to *80 Microcomputing* readers and put through a collect phone call to KTI to ask for clarification on a few points. Here are the results of my research.

A Trip to Sri Lanka

The most important offshoot of the modification campaign is the institution of a brand-new network of authorized repair service centers for the TLS-8E. My Tandy-loving friends crowed about how simple it is to take their computers to their local Radio Shack for repair. After little more than an hour disconnecting cables and unplugging AC cords, they can cart their portable, 97-pound microcomputers for a session with diagnostics specialists. My TLS-8E, my friends point out, has to be shipped to the KTI factory in Sri Lanka just to replace a fuse in the power supply.

No more. Qualified, independent service shops throughout the nation have been recruited by KTI. Each repair center is supplied with a free TLS-8E technical manual, exploded drawings showing how the computer looks after it has exploded, and a hot-line number the service technician can call at any time—night or day—to find out the correct time and temperature.

An intensive training course, conducted by telephone, ensures each authorized repair center has at least one person who can recognize the TLS-8E on sight as a computer.

Having nearby service can be extremely comforting. In my town, repair is offered by Vito's TV Sales & Service. Last week I had to take my TLS-8E to replace a defective fuse. I went back the next day for an estimate, and they told me they'd have to send away for parts. Now, that's what I call knowledgeable, fast service.

Your local repair center can install any of the following mods for you or if you can tell the business end of a soldering iron from the handle, you can do the work yourself.

Many of you have been highly vocal in your complaints about keyboards installed in the early TLS-8E units. For some unknown reason, a significant minority prefer the QWERTY key layout found on some other computers and many office typewriters.

To meet these needs, KTI has introduced an expansion keyboard for the TLS-8E. It can be described in one word: huge. In addition to a full complement of keys silk-screened with the usual alphanumeric characters, it has a bunch of other keys with their tops obscured with masking tape. I found strange markings under the tape: QPOUNDER/CHEESE, FF, LGCOKE, FILET and BMAC. I couldn't figure what these special function keys or the cash drawer located under the keyboard could be used for. Then it hit me: KTI had cleverly bought a bunch of surplus keyboards from some supplier and recycled them for lucky TLS-8E users.

Name That Tune in Five Bytes

Those of you who elected to stay with the stock TLS-8E keyboard haven't been forgotten. Kitchen Table has introduced a pair of modifications of interest to you. The first is a keyboard debounce kit. It is a jar of rubber cement and a foam rubber pad which can be glued to the bottom of the keyboard.

More practical is a sound-feedback kit. When installed, the mod produces a tone when a key is pressed. This lets the user know an entry has been made. A clever touch has been added by KTI. Each key depressed produces a different sound, so users with perfect pitch can determine which key was hit without looking at the keylabel.

This innovation introduces some interesting possibilities. Typing in certain keywords produces recognizable bits of song. For example, entering "print using" generates the first two bars of Johnny Carson's theme song. "On error Goto 1200" produces the opening measures of Beethoven's Fifth Symphony. My family and I spent an entire evening playing a game we invented: "Name That Tune in Five Bytes."

I discovered, by accident, how the inverse procedure works. By referencing sheet music and entering the tune at the keyboard, some interesting program lines result. Barry Manilow's "Mandy," yields a pretty nifty data base management program. I invested two days inputting Handel's "Messiah," and found it produced a Basic version of "Mystery Fun House." Quite a revelation!

17-Jewel, Swiss-Made

Another hardware defect in the TLS-8E has been the lack of a real-time clock. The system has always had its own internal clock—even if access was denied to the user. Because the TLS-8E has this internal clock, it would seem a simple matter to make it available to the user. However, KTI has offered a hardware solution: an add-on clock. The unit is described as the only "17-jewel, Swiss-made real-time clock available for a microcomputer." I plan on adding this feature to my TLS-8E as soon as it becomes available.

A second generation of TLS-8E ROMs has been introduced by KTI. They reduce available memory by 255 bytes and add no new features. The firm has also unveiled a new, one-chip ROM nearly identical to the previous version in function, but with completely new jump vectors to the ROM routines. Our usually reliable KTI source explained this change was made to confuse programmers who insist on using undocumented addresses in their software.

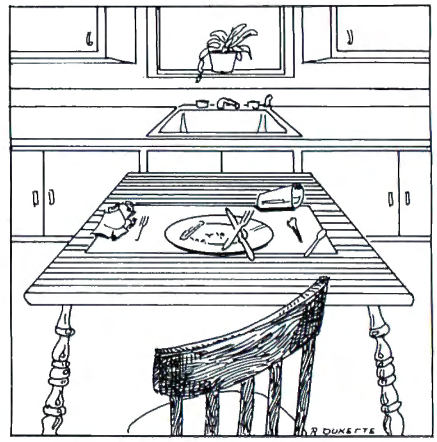
The latest ROMs, with the routines shuffled in an interesting manner, were introduced for two reasons, KTI reports. The company has a new SUPERDUMPMAP II it hopes to sell to all SUPERDUMPMAP I owners. In addition, Kitchen Table has found the previous ROMs had an alarmingly short shelf-life and needed to be reprogrammed at intervals. It seems the Sri Lankan factory thought ROM meant Remember Only Momentarily.

I won't go into the other retrofixes and modifications. They are obscure changes of interest to only a few TLS-8E users. One, for example, provides a means of isolating the CRT chassis from the keyboard. Even a non-hardware type like me can see the CRT is already isolated from the keyboard. In fact, nothing connects the two at all, except for few feet of wire and ribbon cable. Why bother with this one?

Red-Blooded American Modifications

My recent talks with KTI personnel indicate the company is firmly committed to improving both its hardware and software expertise in the future. The firm is not relying totally on the design skills of its Sri Lankan factory team. Several of the new modifications were developed by red-blooded Americans. And the KTI double-density board (The DOUBLOON) was designed and built by a KTI shipping clerk who gained considerable electronics knowledge by building three HEATH kits (including his own color television that can predict the weather.)

With dedication like that, I have nothing but the greatest confidence in the future of an innovative firm like Kitchen Table Inc.■



Reader Service for facing page ✓311→


StarFighter
The PENULTIMATE Space War Game is HERE!
TRS-80*
Model 1 & 3
16K and up — Tape
010-0102 . . . . $24.95
32K and up — Disk
012-0102 . . . . $29.95
ai Adventure INTERNATIONAL
A DIVISION OF SCOTT ADAMS, INC.
BOX 3435 • LONGWOOD, FL 32750 • (305) 862-6917
ORDER FROM YOUR FAVORITE DEALER
or CALL TOLL FREE (800) 327-7172
SHIPPING & HANDLING CHARGES ARE EXTRA
PRICES SUBJECT TO CHANGE WITHOUT NOTICE
OTO COPYRIGHT - DON DIXON
*TRS-80 is registered Trademark of the Tandy Corp.

80 ACCOUNTANT

by Michael Tannenbaum C.P.A.

If the microcomputer suffered from an image problem, it doesn't now. The Great Computer Company IBM has joined Tandy, Commodore and Apple and spawned a microcomputer. Suddenly, small computers are now legitimate. Prestigious publications like the *New York Times* and the *Wall Street Journal* are running feature articles on the coming revolution in personal data processing. For those of us who have been participating in the data-processing revolution, this is heady stuff. After years of defending our "clever little toys," it is difficult to restrain a triumphant grin and cheery "I told you so."

Unfortunately, these new small computers complicate the market. Instead of a small selection of microcomputers available from a limited number of vendors, there will be an abundance of devices available from any number of major retailers. For people selecting a computer, this market will be a strange and confusing place.

The Competition

An early question will be: Compare Radio Shack equipment to IBM. Based on initial information, the IBM two-disk system will cost more than a Model III with about the same capacity. The cheapest IBM unit will be three times the price of a Color Computer with the same capacity. No personal computer configuration comparable to the Model II is available from IBM.

Can the IBM Personal Computer meet the requirements for successful business installation? The Great Computer Company has the resources and skill to market its product successfully. A problem will be selling the unit through retailers, not directly to the user. The firm will lack the short communication channel between vendor and consumer proven important in correcting problems in Tandy hardware and software.

It will take time for IBM and other new vendors to overtake Tandy's lead in applications software. The some 300,000 computers sold by Tandy have created an enormous market. There is also a large amount of user-designed software available. It is safe to say there is a program for almost any need for Tandy equipment.

Aware of this, IBM and other vendors have announced their hardware is compatible with the CP/M operating system. Almost all CP/M software will be usable on these new computers. This will no doubt make CP/M the defacto standard for microcomputers and encourage an explosion of new CP/M applications.

That explosion should not affect Tandy models. Those products are CP/M compatible, although the Models I and III are limited due to their read-only lower memories. That restricts available memory and could disqualify those models from some CP/M applications. The Model II has none of those limitations and functions effectively in CP/M.

It is a data processing axiom every new piece of computer equipment requires a settling down period until the kinks are worked out. Veteran Model I users recall the early days when TRSDOS was flaky and the central processing unit often lost contact with the expansion interface. Every session was a new and uncertain adventure. Purchasers of new IBM equipment will no doubt go through a similar experience. They will be unable to go directly to IBM for assistance; they must return to the primary vendor. If he or she cannot repair the computer, the owner may have to send the machine to an authorized repair station at his or her expense.

Although the record of Tandy's repair centers has not always been good, at least they are in place. There have been enough problems and upgrades in Tandy's 300,000 computers to build a knowledgeable national repair staff. Tandy's philosophy calling for the repair centers to support the retail operation benefits the customer. A retail store manager is an important ally when a repair center can't fix the problem.

At this point, I believe Tandy products are still a good choice for anyone impatient with the problems linked to new computers.

New Mailing List Program

This month's software mailbag included MAIL-X from Micro Architect Inc. The system from the Arlington, MA firm includes a report writer generating up to 10 report formats applied to the data base. Each format specifies the report's name and filtering criteria used to limit the records accepted for printing. The report writer includes four sort fields to arrange data in proper sequence.

MAIL-X has a greater capacity than Radio Shack's MLS II. The Micro Architect product can pack up to 3,200 records on a disk (10,000 on a four-drive system) and files can span disks. However, MAIL-X does not interface with Scripsit, Radio Shack's text editing program.

The system from Tony Pow's firm is also more flexible than MLS. MAIL-X can be set to indicate the dates of a record's creation, expiration and last time referenced. Using this feature, the Micro Architect system can act as a data-base manager for a subscription fulfillment system. If you select the expiration date option, the system will automatically enter a date 365 days in the future when a new record is added. To provide an audit trail, the system offers the option of creating a new data entry log in the file maintenance mode. This printout is essential to process inquiries from subscribers.

The system also offers more flexibility than MLS when defining file formats and record sizes. Field sizes and names can be altered allowing the program to be used for other list-processing applications. I adopted the program to control equipment deliveries during installation of a large data-processing system.

Micro Architect's long experience with Radio Shack hardware is apparent in MUTIL, one of eight programs in the system. This file utility can extend the size of a data base, delete bad records, append files and change passwords.

I found moving from program to program in the system difficult. Unlike MLS II, this system is a series of unlinked programs. Some menus are present, but often a program can only be called by returning to TRSDOS and executing a do-file.

Each program signs on with an initialization menu and options list. After going through this sequence many times, the sign-on protocol becomes annoying. This system needs one-time initialization and a transfer file. If parameters need to be changed, the routine in the transfer file could be called up and changes made. All sign-on parameters should be satisfied by the transfer tile.

I cannot comment on the system's documentation; I received an early release of the system with a preliminary manual.

I did not have time to enter a large enough data file to test the sort-merge features of the system. This routine is called when a file is greater than available memory. The largest file I tested was about 70 records. With a file that size, the sort is almost instantaneous.

MAIL-X requires a more knowledgeable operator than the MLS II. However, an experienced Model II owner should be able to work with the system without difficulty. Since the programs (with the exception of the sort) are in Basic, the user may customize them. The code is well-written and easy to follow—even a file layout is provided for the programmer. ■

Xerox, IBM storm market, pull wraps off their micros

Xerox and IBM have challenged the hold Tandy/Radio Shack, Apple Computer Inc. and Commodore International Ltd. have on the small business microcomputer market. A 16-bit microcomputer selling for $1565 and a data and word processing system with multiple work station options have been introduced by IBM, while Xerox' 820 desktop model has word processing, printer and software options. Both companies are competing for the small businessman, who is, in many cases, a first-time computer user.

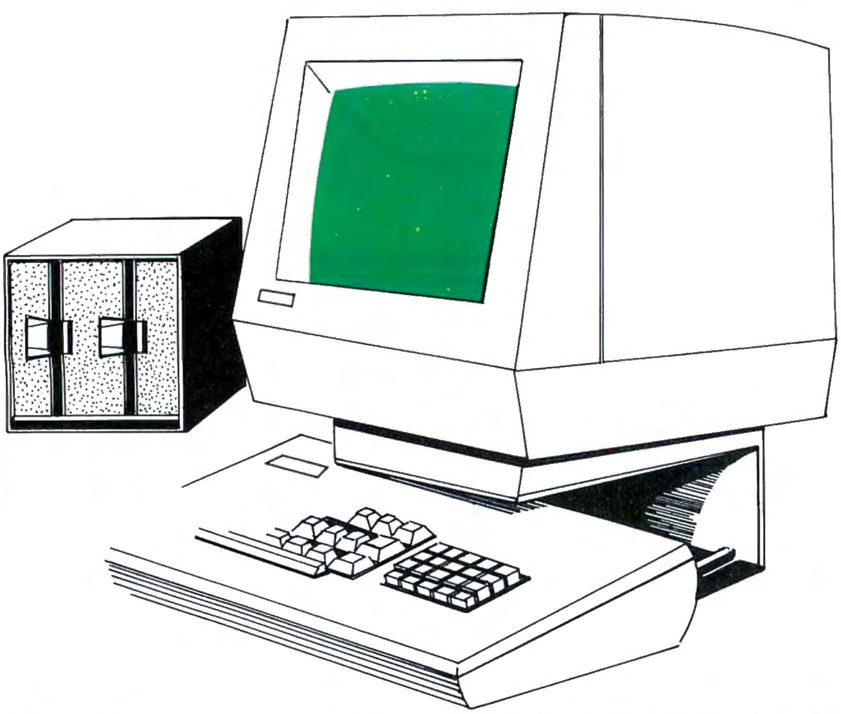
Thus far Tandy's reaction to their new competition has been blase. "I don't think we're going to lose any business because of it," says Jon Shirley, vice president of the Fort Worth, TX, firm's computer division.

Though Tandy won't release sales figures, market analysts estimate they hold about 25 percent of the personal computer market, with Apple of Cupertino, CA, garnering about 22 percent and Norristown, PA-based Commodore 20 percent. Shirley says about 96 percent of their Model II units have been sold to small businesses.

What can Xerox and IBM offer the business user that Tandy cannot? Both companies have a respected service network and a sales force with many contacts in the business world. Their micros are also compatible with other company products. They will compete with the Model II, but the question is how much and how soon.

These new machines "will not have an immediate effect," says market analyst Al Hirsh of Datapro Research Corp., Delran, NJ. Hirsh feels that the new computers will have the swiftest impact on Tandy's major accounts because their competitors have so many business contacts.

Other marketing people think the new computers—particularly the IBM personal computer—will affect Apple computer's sales more than Tandy's. "The IBM personal computer is aimed smack at Apple," because its price and capabilities are similar, says Gerald Hallaren of the Yankee Group, a Cambridge, MA, market consulting firm. Nonetheless, all three new computers will be marketed with the small businessman in mind and are bound to cut into Tandy's share of this boom market.



Datamaster

At almost $10,000, the most expensive of the new competitors is the IBM Datamaster, but experts don't agree on whether the cost will affect sales. Some think the IBM name will be enough to sell the system. David Folger, a consultant with International Resource Development Corporation, Norwalk CT, thinks Datamaster will be bought by "people willing to spend more money than they have to."

"IBM's cashing in on their reputation," agrees Stan Green of Datapro. He contends users are willing to pay $5,000 to $6,000 more for an IBM product based on the Armont, NJ, firm's reputation alone.

Consumers will pay a base price of $9830 for a single IBM computer work station and an 80 character per second printer. The computer features 128K of RAM and 112K of ROM. The terminal display is the standard 24 by 80 and features upper and lowercase and five character sets, including Nordic and Spanish.

The system's dot-matrix printer is bi-directional. A second printer is available at $3200. To achieve letter quality printouts, IBM uses a strike-over mode that reduces speed to 40 cps. This option costs $3500.

Single or dual disk drives are available and users can add another disk unit to the system. Each drive contains 1.1 megabytes of storage; when the 5246 disk unit is added, each work station will have 4.4 megabytes of storage. Total memory with two work stations is 6.6 megabytes.

What IBM considers the system's most attractive feature is the ease additional work stations, disk drives and printers can be added as a company's business grows. Up to four printers (and two disk units) can be used with two work stations.

The Datamaster also comes with a word-processing option ranging from $1100–$2200. The circuit card costs $600 and the software is $500. If the user wants a letter quality printer, it's $1100 more than the 80 cps printer.

Smaller IBM

IBM's smaller personal computer offers many of these features at a much lower price. The standard system includes 16K of memory, keyboard, cassette jack, speaker, and Basic language interpreter. IBM has made it easily adaptable to word processing. There are 10 keys for scrolling and editing; an upper and lowercase display; and a dot-matrix printer option with page spacing and column skip. The printer runs at 80 cps and features 12 type styles. Easywriter, the word-processing software, is a comprehensive text editor without automatic hyphenation, repagination or scrolling.

A 64K system with single disk drive and a display costs $3005. A full business system, including two drives and a printer, costs $4500.

Up to 256K RAM is available, with 40K ROM; up to two disk drives can be attached with 160K per disk. An RS-232C asynchronous adapter hooks the system to data bases and other computers. Because the personal computer uses the CP/M operating system, a host of programs are available for it, including the Visicalc financial package, Peachtree Software's accounting programs, and communications software allowing access to data bases and other computers.

Xerox Entry

Xerox' 820 computer also supports the CP/M system and has word-processing capabilities. Though less powerful, its features are strikingly similar to IBM's—a 96-character ASCII format with a 10-key numeric pad; dual 92K byte single-sided mini-floppy Shugart disk drive; and standard 64K RAM with 4K ROM. A dual RS-232C port connects printers and other communicators. Xerox is pushing the daisy-wheel Diablo 630 printer as an 820 option; the $2995 printer is bi-directional with 40 cps letter quality output.

Their $495 word-processing package has a long list of features. Text editing functions include global find and replace, standard insert and delete, auto return and a command moving copy from file to file. Text formatting includes pagination, headers and footers, reformats, tabs, margins, auto centering, justification and super- and subscripts.

The Key

Software will be the key to whether these systems are marketable, says Karen Horowitz of the Venture Development Corp., Wellesley, MA. "People won't be buying 64K memories, they'll be buying solutions to problems."

Since IBM's Datamaster is not CP/M based, the firm is offering its own business accounting package—for a whopping $6150. The six program package includes billing, accounts receivable, accounts payable, inventory accounting, payroll and general ledger. A unique feature of the package is the ability to enter information from one program and have it automatically transferred to another without re-keying.

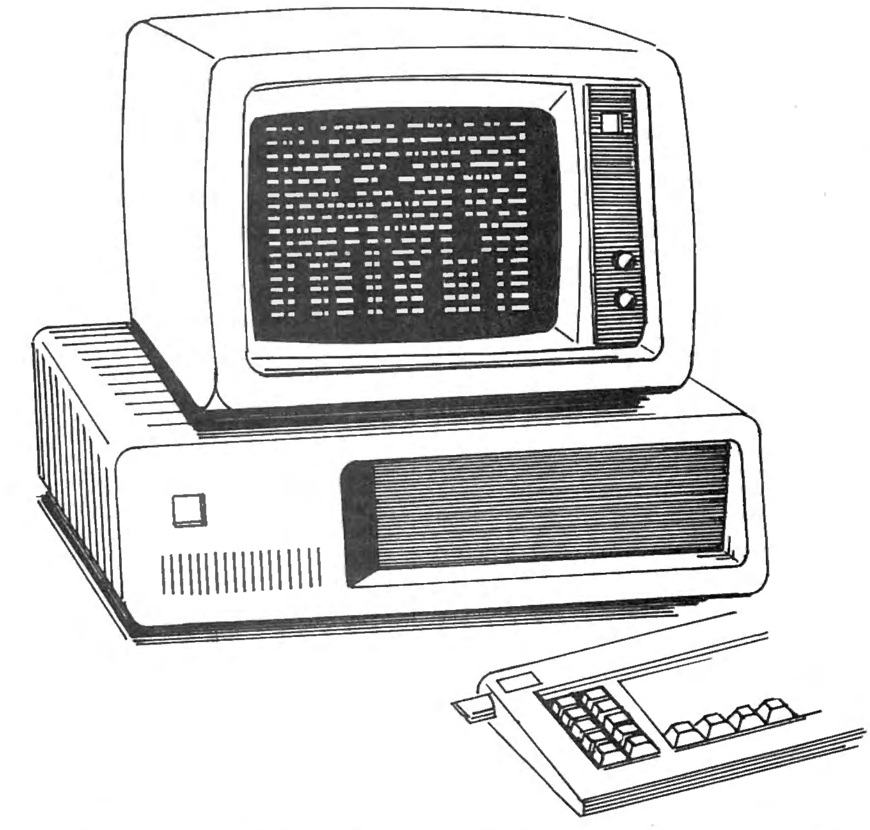
Another software package, Brads III, will soon be available. It's designed for unique business applications like sales analysis, and costs $1145. An IBM spokesman said he was confident people would be writing software for the system.

Though the Datamaster's software is more expensive, some say the small businessman would rather pay the extra price in exchange for adequate user support. Hirsch of Datapro praises the IBM software package for its integration and says many first-time users prefer company-backed software to dealing with third-party distributors.

With the Datamaster and personal computer so far apart in price range, there's also the question of their competing with each other for the small business market. But marketing experts seem to agree the two will appeal to different camps in the business field.

"The type of guy who keeps his own books will go for the personal computer," says Hallaren of the Yankee Group. Larger businesses with a bookkeeper or two would be better suited to the Datamaster because "it's fantastically easy to use" and has multiple work stations.

Others say the personal computer is the more attractive of IBM's offerings. "The Datamaster is over priced (and) there's nothing technologically advanced about it," says Hirsch. The personal computer, on the other hand, has word processing and communications links with other systems and could even be convert-



ed to hard disk. One million of them will be sold by 1985, he predicts, while after 50,000 Datamasters are sold by next year, the machine will be "put to rest," because of its expense and incompatibility with other units.

But IBM's promotion of the Datamaster will be heavy. They're sinking $3 million into a 90-day television ad campaign, as well as a radio and print blitz.

Different Views

Experts differ on the importance of price to the small businessman. Some say the first-time user is reluctant to make a large investment. "The first-time user isn't going to pay that money as readily. He has to be convinced of the economic value (of the product)," says Francis O'Reilly, an independent marketing analyst in Ardsley, NY.

An analysis by Venture Development Corporation also stressed the "need for small businesses to increase profits without making a large capital investment."

Both IBM and Xerox have respected and recognized names in business circles. "Like we used to say in the data processing business, nobody ever got fired for buying an IBM computer," observes Bishney. Their reputation for service and reliability could hurt Tandy, whose pitch for major accounts is "in its infancy," Hirsh says.

Sales Approach

Ultimately, what may matter most in the micro market is not cost and function, but how the companies sell and distribute their products. The appeal of Xerox and IBM to businessmen will be a key factor in product promotion. Both are disseminating their products in a variety of ways. Dallas-based Xerox is relying on their 500-strong sales force, office dealers, distributors, and 25 outlets. Datamaster will be sold via IBM's General Services Division, using 50 outlets and their three main product centers—Baltimore, Philadelphia and San Francisco. A personal computer marketing organization has been set up within the Data Processing Division. They also have a mammoth sales force.

Because it costs IBM $100 every time a salesperson calls on a customer, the company is establishing seminars bringing clients to the company according to Hirsch. Whether these seminars will accomplish as much as visits to customers remains to be proven.

Retail Outlets

What's new for both companies is the use of retail outlets to distribute their products. Xerox will sell the 820 at retail outlets such as Computerland; IBM is selling its personal computer through 200 Computerland stores, five Sears, Roebuck and Co. business machine stores, IBM product centers and a sales arm of its Data Processing Division.

But retail stores may not be the best way to reach the small-business owner. A Venture Development survey indicates only a small percentage of users bought their systems at a retail outlet; an even smaller percentage of potential users would consider buying a retail computer.

Apparently, small businessmen—particularly first-time computer users—consider company back-up more important than any money they save at retail outlets. Buying direct through salesmen or company outlets appeals to the novice, who needs a lot of "hand-holding" from the company.

Anticipating this, IBM is offering complete service back-up for the personal computers sold through Sears outlets. Maintainance contracts will cost 13 percent of purchase price per year. They have also set up a toll-free line for Data master users.

How will Tandy's distribution match up? With 2,000 dealers, 168 computer centers and 4,800 retail stores, they've pretty much got the field covered. They also have some direct accounts sales people, though this is certainly not their strongest point. Shirley of Tandy thinks their retail units are the key to escaping the influx of IBM and Xerox. "They're selling them in stores where they sell Apples and PETs," he says, theorizing those two producers will feel the brunt of the competition.

Micro Versus Mainframe

Though they are making a healthy stab at the micro market, both Xerox and IBM may have problems divorcing their microcomputers from larger office products. Marketing analyst O'Reilly says IBM must alter its marketing approach to maintain all its product lines. "I think IBM should divide itself like an amoeba," to accommodate microcomputers. However, he is skeptical of their mass market capability. "I don't think IBM has had any experience in high volume production."

O'Reilly also questions Xerox' credibility because of their abrupt withdrawal from the mainframe business several years ago. At that time Xerox bought Scientific Data Systems, whose mainframe product had an "esoteric" application, according to Xerox spokeswoman Marcie Williams. Xerox' forte is mass production, not customized products, she says. The new 820 blends in much better with their mix of office products.

Both corporations have worked hard to fit the micros into larger offices as well as small businesses. Datamaster and the

IBM micro's nitty gritty

Computer boasts speedy microprocessor

The new IBM Personal Computer is a machine designed for business, school, and home use. This is the lowest-priced computer system IBM sells, with prices ranging from $1565 to $6300.

Standard equipment includes: a detachable 83-key keyboard; a cassette tape jack; five expansion slots for additional memory and display, printer, communications and game adapters; built-in speaker for music programming; automatic self-test of components after power-on; an enhanced version of Microsoft Basic; 16K characters of user memory; and a high speed 16-bit microprocessor.

The system can accept two 5.25-inch disk drives. The memory is expandable up to 256K bytes. Either a monitor or a television with radio frequency modulator may be used to provide a display of 25 lines of 80 characters per line.

Software already available for the IBM Personal Computer includes: a disk operating system; a Pascal compiler; CP/M and the USCD p-system; VisiCalc; general ledger; A/R; A/P; a word processor; Microsoft Adventure; and communications software to use the optional RS-232C asynchronous adapter.

Service for the IBM Personal Computer will come from IBM and from a nationwide network of authorized dealers.

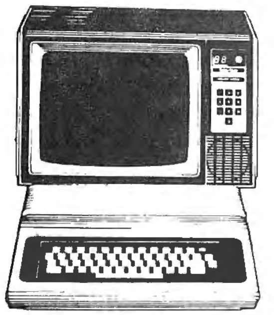
First deliveries of the IBM Personal Computer are scheduled to begin in October. ■

by Bruce Metzger
Data Processing Manager
Wayne Green Inc.

TEXAS COMPUTER SYSTEMS

Offers Lowest Prices on

FOR PERSONAL BUSINESS
OR ENTERTAINMENT



Color Computer

4K Level I $319
16K Level I $439
16K Extended Basic $489

With TCS Memory:

16K Level I $369
16K Extended Basic $449

Expansion Interfaces

Zero K Interface $254
16K Interface $359
32K Interface $469

With TCS Memory:

16K Interface $318
32K Interface $388

Epson Printers $Call

Letter quality matrix printer has full software control of 40, 80, 66 or 132 columns. 80 cps bidirectional tractor feed, disposable printhead. $300 less than nearest competitive printer. Lists $645. Call for our low price.

MX-80 Tractor Feed
MX-80 FT Friction and Tractor
MX-100 Graphtrax, Friction and Tractor up to 15" wide.
Graphtrax for MX-80, MX-80 FT, graphics option.

SOFTWARE

EPSON MX-80 PRINTER DISKETTE

A complete diskette with 6 complete files for use with your Epson MX-80 Printer. . ready to use.

M/CMD - Send all of the Epson Printer commands to the printer directly from DOS or as part of an auto booting system. Set up the printer from DOS to the typestyle you want: then run the program.

BIGLETT/BAS - Produce large graphic letters on your printer and screen Dump the letters to printer or to disk Allows 3 different size letters to the printer. Program is menu driven. You may also construct large letter titles for MERGEing into your basic program

EPSON/SUB - Merge this small module into any Basic program and then issue any of the printer commands with a 2 letter command with letters representing the purpose of the command. You need not refer back and forth to documentation. Extremely easy to use and makes using the printer much easier than ever before.

JKL PATCH - This patches NEWDOS 2 1 and NEWDOS 80 so that the JKL function will print the TRS80 graphics while the printer remains in the standard mode rather than the TRS80 mode. This allows all of the features of the printer to be used but you can dump the screen to the printer at any time with all graphics.

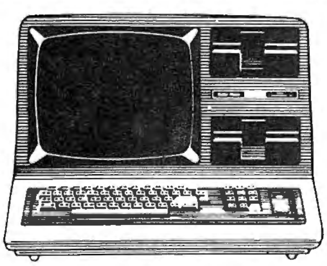
DEMO/BAS - A super tutorial program with hundreds of REMs and hundreds of screen prompts that teach you how easy it is to create miracles with your Epson MX80 printer

LABEL/BAS—Custom label making program including graphics. Makes custom mailing labels an easy job.

DEFSTATE/SUB - A series of one word printer commands that can be merged into any Basic Programs that allow various automatic functions to occur. Print titles with graphic borders around them with one word. Center any text with one command. etc

Special Price: All seven of the above modules on one diskette or on tape (less JKL) for your Model I or III for only **$29.95**.

Or, take a $10 discount if you buy this item with purchase of the Epson MX-80.



TRS-80

We have all Radio Shack computers and accessories, and a number of other brand computer items. Call us for your specific needs and get a copy of our catalog.

Model III 16K $835
Model III 32K $979
Model III 48K $1089

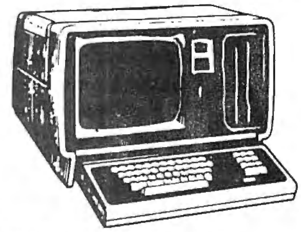
With TCS Memory:
Model III 32K $909
Model III 48K $969

Model III 48K 2 Disk RS232 $2100
Model III 32K 1 Disk $1729
Model III 48K 1 Disk $1849

TEXAS COMPUTER SYSTEMS

Offers Lowest Prices on

TRS-80 COMPUTERS



Model II 64K $3288

An excellent computer for your business needs. Easy expandability & compatibility. No formal operator training needed. All accessories available—disk expansions, printers, software, at our low discount prices. Our fast, fully insured air freight service can assure most deliveries within seven days after payment is received.

Corvus Hard Disks $Call

5, 10, or 20 megabytes of storage for the Model I, II, or III. We can configure a system for you with either TRSDOS, NEWDOS 80, or CP/M for one, or several computers sharing a single hard drive simultaneously. For example, from 1 to 8 Model II's may share a single large data base with the Corvus Constellation Multiplexer. This also allows you to use different manufacturers computers on the same hard disk system at the same time. Also fits Apple, Superbrain, Altos and most other computers. As your business grows, additional storage and terminals may be added easily, inexpensively. Why wait months for other's promises? We can deliver these reliable proven hard disk systems NOW! Corvus has sold and shipped ten times more hard drives for microcomputers than all their competitors combined. **Call us for the lowest price!**

MICROSOFT

BASIC-80 - Extended disk Basic, long variable names chaining variable length records. ANSI compatable interpretative version **$294 w/manual. $30 manual only.**

BASIC COMPILER - Same Basic language as BASIC-80 except compiles Basic program into a directly executable, relocatable machine language file Operates 3 to 10 times faster than regular Basic Includes Macro-80 Assembler Specify CPM or TRSDOS **$325 w/manual. $30 manual only.**

FORTRAN-80 - Supports ANSI 66 plus many extensions including MACRO-80 Has library with manager, relocatable object compiler and linking loader **$400 w/manual. $30 manual only.**

Word Processor Package $2679

Includes 2 Disk Model III with 48K, Epson MX-80 Tractor Feed with cable, and word processing software ready to operate. Lists at $3300. Our low price special this month: **$2679.** For MX-80 FT Tractor and Friction, add $99.

Special—With TCS Memory
★ **Model III 48K 2 Disks $1895**

Uses proven MPI drives modified for flippy operation and Percom controller Limited 90 day warranty Call for details

Printers

Daisy Wheel II $1695

Carbon ribbon typewriter quality for word processing. 43 cps. Standard Centronics parallel interface Full 16" width, individual sheets or friction feed. Tractor feed $250 extra. Fastest Daisy Wheel available at this price

Line Printer IV $825
Line Printer V $1599
Line Printer VI $995
Line Printer VII $350

Pocket Computer & Acc. $Call

Pocket Computer Printer Interface in stock.

- Payment. Money Order, Cashier's Check, Certified Check. Personal checks take 3 wks VISA, MC, add 3%
- Prices subject to change any time
- No tax out-of-state Texans add 5%
- Delivery subject to availability.
- Shipping extra, quoted by phone ✓25

TEXAS COMPUTER SYSTEMS

Box 951, Brady Texas 76825

For fast, efficient service, we can air freight from Dallas to major a/p near you. Call for information.

Toll Free Number 800-433-5184
Texas Residents 817-274-5625

✓See List of Advertisers on page 402

Xerox 820 are compatible with other company products.

The Datamaster, which many feel is a redesigned version of the ill-fated 5100 series computer, features business level Basic compatible with the IBM System/34. Also available is an $875 software package that converts 5110 and 5120 Basic to Datamaster Basic.

Local Networks

Xerox' 820 can be used with Ethernet, the company's communications network. A user simply needs an RS-232C board and communications server. Some manufacturers, including Intel Corp., are working on a chip for a direct connection.

It's also rumored IBM may soon release a communications system similar to Ethernet to be linked with the personal computer. An IBM spokesman denies this however.

With the business end of the microcomputer market growing the fastest—Venture Development predicts it will total $2 billion annually by 1984—companies will have to meet the needs of the small businessman. This means cost-efficient systems, adequate software, user support and ample memory. But to capture the brunt of the market, companies like Tandy must compete for the major accounts sales to larger companies. That's where Xerox and IBM hit the hardest.

Everyone but Tandy is taking the entrance of these corporate giants seriously. Speculation was rampant on Wall Street after IBM's 16-bit personal computer debut. But Tandy remains non-plussed. Shirley of Tandy contends they have the market cornered and "everybody else can go fight it out among themselves."■

by Betty Thayer
80 Microcomputing *staff*

Bill targets computer crooks

Nelson of Sunshine State takes aim at purloining programmers

A bill making many kinds of computer crime a federal offense has been reintroduced into the US House by Rep. Bill Nelson (D-FL). The bill would mobilize the resources of the FBI and federal judiciary system to fight the rise in computer crime.

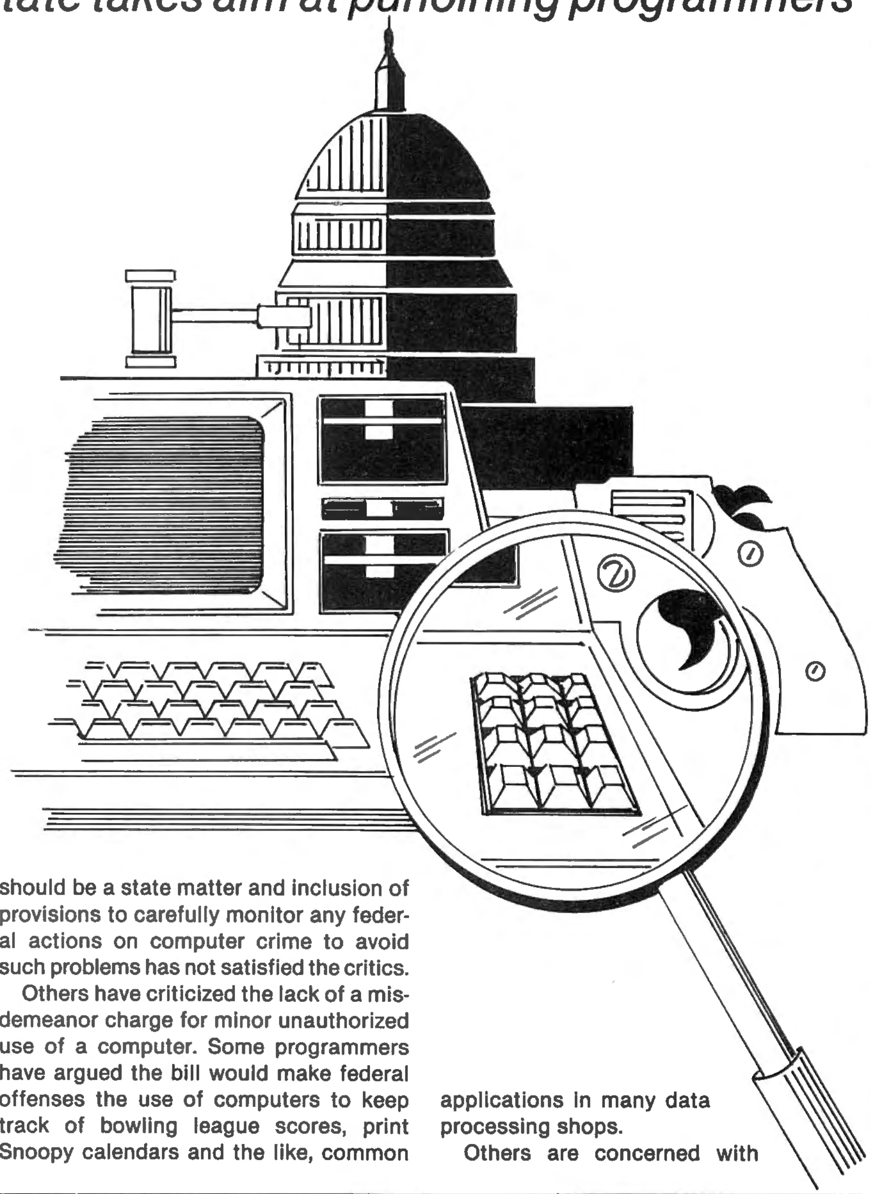
Known as the Federal Systems Protection Act (H.R. 3970), the new bill is presently before the House Judiciary Committee's Subcommittee on Civil and Common Law. Hearings have not been scheduled on the measure yet.

The bill is the latest version of a proposed act sponsored for four years by retired Sen. Abraham Ribicoff (D-CT) and reported out of a Senate subcommittee at the end of the 96th Congress last year.

The bill makes it a federal offense to use, for illegal purposes, a government computer or a private computer owned by a firm involved in interstate commerce or owned by a financial institution insured by the federal government or in the Federal Reserve System. It also makes acts of destruction against any such machines a federal offense.

The bill provides a fine up to twice the proceeds from the illegal act or $50,000, whichever is greater, and imprisonment for up to five years.

Congressman Nelson represents an area of Florida hosting several high-technology companies. He sponsored Florida's first computer crime legislation while a state legislator. That bill, the Florida Computer Crimes Act, became law in 1978 and was the first state law to deal directly with computer crime.



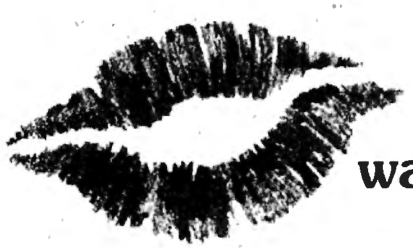
Critics

Some critics feel the bill in Congress will create federal intrusion into what should be a state matter and inclusion of provisions to carefully monitor any federal actions on computer crime to avoid such problems has not satisfied the critics.

Others have criticized the lack of a misdemeanor charge for minor unauthorized use of a computer. Some programmers have argued the bill would make federal offenses the use of computers to keep track of bowling league scores, print Snoopy calendars and the like, common applications in many data processing shops.

Others are concerned with

KISS BASIC GOODBYE



Announcing the new quick and easy way to write custom business applications in hours instead of weeks.

At last, you can get a microcomputer business application development system that is designed for one purpose only . . . fast production of bug-free professional-looking custom business applications.

"Thinks" the Way Business Thinks
Quic-N-Easi is the revolutionary screen-format-oriented application development system that thinks in terms of transactions, records and fields. The same way business thinks. **Quic-N-Easi** expects an application to include custom formatted key entry. It lets you set up the character, field, and record validity checks business wants by merely filling in the blanks. It automatically signals in real time when errors occur and "explains" what is wrong in plain English messages you select for each custom application.

Much Faster than BASIC
Quic-N-Easi programming is much faster than BASIC because the standard business applications (key input, file handling, and output reporting) are handled via pre-programmed parameter driven sub-routines. To produce finished professional looking custom business programs, you merely:

- Draw the business formats directly on the CRT in minutes.
- Fill in the blanks for field attributes, validations, tables, etc.
- Invoke the **Quic-N-Easi** interpreter to check fields against tables, check limits, access data files, and perform business calculations, etc.
- Optionally define file and output formats right on the screen.

Pays for Itself in a Week
By eliminating the coding drudgery of writing screen, field, file, and format programs in BASIC, programmers are free to concentrate on the unique business aspects of each custom application. No professional programmer can afford to develop one more business application without **Quic-N-Easi**. The savings are so significant with **Quic-N-Easi**, it can actually pay for itself in only one week.

Gives You Much More than a Screen Builder . . . the Only Complete Business Development System for Microcomputers
Quic-N-Easi handles the entire application development job from key entry to final output. You get everything you need, including detailed documentation to begin writing professional programs the first day. **You get . . .**

- A singularly capable **Quic-N-Easi** screen builder
- A comprehensive parameter driven **Quic-N-Easi** content editor
- A full **Quic-N-Easi** interpreter language
- A complete **Quic-N-Easi** file management system with Index Sequential, Random, and Sequential File Access Method
- A complete **Quic-N-Easi** print format handler
- A detailed **Quic-N-Easi** reference manual
- A **Quic-N-Easi** self-teaching guide
- A **Quic-N-Easi** quick reference card
- An interface to other program files

ORDER NOW—Don't Waste One More Day Coding BASIC
Your time is too valuable to waste hours on end writing BASIC code. The first week you use **Quic-N-Easi**, your finished business programs will look better, run better, and return more dollars to you. Don't delay. Order **Quick-N-Easi** today.
Phone . . . 215 968-0689

quic-n-easi™

STANDARD MICROSYSTEMS INC.
136 GRANITE HILL COURT, LANGHORNE, PA 19047

✓317

Pricing Information
- Complete **Quic-N-Easi** system **$395**
- Manual only **$60**
- **Visa** and **Mastercard** accepted
- **Dealer Inquiries Invited**

Minimum System Requirements
48K • CP/M • Z80 • Floppy Disc

Other Disk Formats • 8" Single Density Vector Graphics • Micropolis Model 2

Customized Versions • TRS80 Mod II, TRS80 Mod III, APPLE, OSBORNE, INTERTEC, VECTOR , ZENITH

✓See List of Advertisers on page 402

the broad definition of computer in the bill. The bill defines computer as "any device that performs logical, arithmetic, and storage functions by electronic manipulation, and includes any property and communication facility directly related to or operating in conjunction with such a device..." The definition excludes automated typewriters and typesetting equipment, microcomputers and portable hand-held electronic calculators.

Bill supporters have argued a broad definition is necessary to cover future generations of computers.

Sees a Need

In spite of the criticisms, however, Nelson believes the federal bill is needed. He says the federal government, alone, owns more than 10,000 computers. These are used, among other things, to pay billions of dollars annually to federal workers, defense contractors and recipients of Social Security benefits.

Private industry has thousands more computers handling all kinds of financial and sensitive information. Nelson notes a programmer stole $10.2 million from Security Pacific Bank in California and transferred it to a Swiss bank account with a single telephone call. An employee of Kelly Air Force Base in San Antonio, TX, stole $100,000 through false computer billings for aviation fuel. A business in Virginia stole information from US Department of Energy computers in Maryland by telephone. The total direct financial loss in the United States due to computer crimes is estimated at between one and several hundred million dollars annually.

However, Nelson said, no federal law clearly covers the use of computers in crime or their damage by terrorist-type attack. The 40-odd federal laws covering various aspects of the issue were all written long before computer crime existed as a major problem.

Nelson admits computer crime is a small part of the total problem of white collar crime in the United States. However, he maintains thousands of Americans today are familiar with the strengths and weaknesses of computers and have both the knowledge and opportunity to commit computer-based crimes. For this reason, he has reintroduced the bill in spite of its less than spectacular progress through Congress in the past.

No Great Following

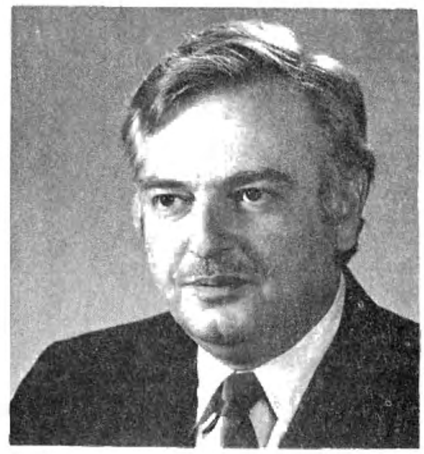
Although Ribicoff sponsored the bill in the last two Congresses and succeeded in getting it reported out of subcommittee, it was never acted on by the Senate. With the start of the new Congress, it would have to go back to the subcommittee if someone sponsored it in the Upper Branch. So far no one has. In the House, it never got out of subcommittee, partly because the House Judiciary Committee's Chairman, Peter Rodino, Jr. (D-NJ) does not favor the bill. Nelson's administrative assistant, Jim Southerland, said that situation has not changed and admitted the bill has not attracted a great amount of interest in Congress.

This time around Nelson hopes to open communications with concerned persons to learn their views on computer crime and inform them on the need for the bill and its purposes. His office is preparing a mailing to government officials, business leaders and other involved individuals to determine the level of interest and need for the bill. Southerland said Nelson would welcome comments from interested parties.

Persons interested in the bill can contact Southerland at Room 307, Cannon House Office Building, Washington, DC 20515. Those interested in contacting the Subcommittee on Civil and Common Law may write to its chairman, Rep. Don Edwards (D-CA), and those interested in communicating with the full committee may address their remarks to Chm. Rodino.■

Dartmouth fetes John G. Kemeny, father of Basic

The father of the popular computer language Basic was honored recently by the university he has served for more than 25 years. Dartmouth's John G. Kemeny, who last fall stepped down after 11 years as the college president, has had an academic chair established in his name in the mathematics department.



The John G. Kemeny Parents Professorship in Mathematics is a $1 million distinguished professorship. The endowment is a gift from three non-alumni parents, who have chosen to remain anonymous, and given in appreciation of Kemeny's service to the university and to the country.

Pioneering Author

Kemeny was a pioneer of computer time-sharing techniques and one of the first educators to recognize the importance of computers in higher education. His time-sharing system at Dartmouth became the model for other systems in academic and business environments. In addition, Kemeny's Dartmouth Basic found great popularity among his students and proved so easy to learn it was adopted by the computing public as the most comprehensible computer language available.

Fleeing the German persecution of Jews in his native Hungary in 1940, Kemeny came to this country as a 13-year-old student. His affinity for mathematics and science was soon obvious to his teachers at George Washington High School in New York City, and by the time of his graduation, he was accepted at Princeton University in nearby New Jersey. While at Princeton, he was recruited by the government to work on the Manhattan Project at Los Alamos, NM. Here, as he assisted in the development of the atomic bomb, he rubbed shoulders with some of the best scientific minds of the time including Jon Von Neuman, Edward Teller and Eugene Wigner.

When his work at Los Alamos was completed, Kemeny returned to Princeton to pursue his doctorate in mathematics. While completing his graduate studies, he was research assistant to Albert Einstein and became an advocate of World Federalism—a movement for the sensible human control of nuclear technology.

In 1952 Kemeny was recruited by Dartmouth and in two years became chairman of its aging math department. As he brought new blood into the department, he also worked on developing an easy to learn computer language. He called it the Dartmouth Beginner's All-purpose Symbolic Instruction Code (Dartmouth Basic). During this time Kemeny also set up an elaborate time-sharing system for the college computer and encouraged under-

graduates to get familiar with computers through game playing. As a result, over 90 percent of Dartmouth's undergraduate students had exposure to computers before leaving school.

Storms on Campus

In 1969 Kemeny was appointed the college's 13th president. That year and through the early 1970s, his cool style of leadership, encouraging dialog rather than rhetoric, stood him in good stead with students. Though the bombing of Cambodia and the killings at Kent State University in Ohio rocked Dartmouth as they did most other American campuses, Kemeny managed to weather the storm by suspending classes for a week, establishing a series of teach-ins and seminars and providing transportation to Washington, DC, for outraged demonstrators who sought President Richard M. Nixon's ear.

In 1972 Kemeny wrote the book *Man and the Computer*. In it he argued for a new symbiosis between man and machine. In Kemeny's view the computer was man's key to managing the increasing complexity of his future. In the book Kemeny also sought to allay the fears of the layman concerning technology. He maintained that, if introduced properly, man and computer could become fast friends and partners in the future.

The Commission

Kemeny's best known contribution in the arena of popular science is the "Report of the Kemeny Commission on the Accident at Three Mile Island." Tapped by President Jimmy Carter in 1980 to head the commission investigating the near melt-down of a nuclear reactor outside Harrisburg, PA, Kemeny later said, "As we looked at the NRC (Nuclear Regulatory Commission) overall, our most important judgement was that it is an agency hypnotized by equipment. It had a firm belief that equipment could be made fail-safe, and as a result it totally ignored the human element in nuclear power."

For many years Kemeny has sounded a similar theme in his work and today he contends man, by himself, is unable to cope with the extremely complex systems he has created. For Kemeny, the computer must play a role in helping man cope with his increasingly complex world. Kemney's contributions toward popularizing that machine have made coping easier and computer literacy attainable for many a microcomputerist. ■

by Chris Brown
80 Microcomputing *staff*

News in bits and bytes tests waters for future of electronic newspapers

Two years ago if you wanted to read a newspaper, you either subscribed to it or bought it at the corner store. If you wanted to read your home town paper from 1,000 miles away, you had it mailed to you and read it three days later. Not so today—at least for microcomputer owners.

More and more newspapers and news services are becoming available to micro owners. At least three major databanks—CompuServe, The Source and Dow Jones Electronic News Service—offer electronic newspaper services, and a small Midwestern newspaper has a public database.

CompuServe of Columbus, OH, offers electronic editions of 10 newspapers: *The New York Times, Washington Post, Los Angeles Times, San Francisco Chronicle* and *Examiner, Columbus* (Ohio) *Dispatch, Minneapolis Star* and *Tribune, Norfork* (Virginia) *Pilot and Ledger, St. Louis Post-Dispatch, Atlanta Constitution and Journal* and *Middlesex News* in Framingham, MA.

The Source in Maclean, VA, does not carry individual newspapers, but does carry several electronic newsletters and a wire service, United Press International. Stories from UPI go into The Source seconds after being transmitted on the wire.

The Princeton, NJ-based Dow Jones Electronic News Service includes access to the international Dow Jones business and financial wire. This service is designed for financial and investment professionals and has been available to them for several years.

Recently, Dow has made the service available to the general public and formatted some of the databases for TRS-80, Apple and Commodore microcomputers.

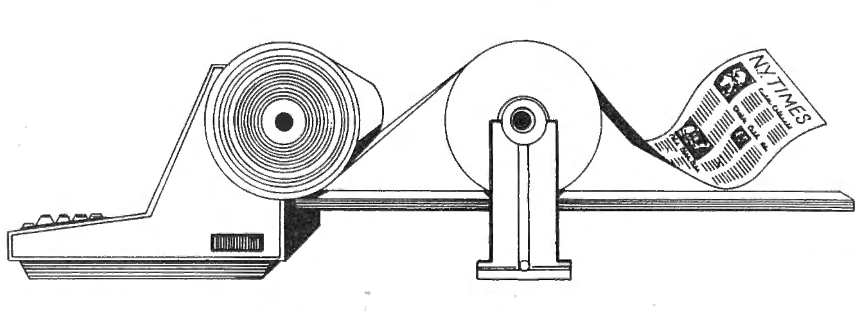
According to Jerry Lipski, head of computer services for the *Atlanta Constitution and Journal*, electronic service is less expensive and more convenient for the newspaper but more expensive and less convenient to the reader. You can't fold up a microcomputer and read it on the train home or under a hair dryer, and you can't line a bird cage with it. Then there's the expense of buying a microcomputer or dumb terminal and subscribing to the database.

For a newspaper, electronic editions cut out some of the major costs of the business: typesetting, printing and distribution. They also allow for more flexable deadlines, since the database can be updated instantly. However, an industry source familiar with one of the existing electronic news providers said there are delays of up to three hours between filing a story and its appearance in the database.

In the CompuServe system, most electronic editions are story by story renderings of the day's newspapers. When a user asks for an edition, he or she chooses an area of interest and is given a menu of headlines. After selecting a headline, the story appears on the screen.

Lipski said the *Constitution's* electronic edition is unusual in its format and reason for existence. It only includes the full stories of the front page and other special pages. Most of the paper's stories are summarized in a few paragraphs, as is done on radio. The idea, Lipski said, is to encourage people to buy the *Constitution* instead of having the electronic edition compete with the paper.

In addition to news stories, he explained, the *Constitution* database includes a guide to Atlanta's hotels, a television guide with reviews, a schedule of airline service to Atlanta and other general interest items. ■



Height games

Simulations eyed during strike

Reporters looking for a new lead for the continuing story of the air controllers strike hit on microcomputer simulations of airport traffic control last August. For awhile, the public read and heard about products like *O'Hare,* a program marketed by Instant Software, Peterborough, NH, and *Air Traffic Controller* by Creative computing, Morristown, NJ.

The sudden media interest in the subject apparently started with a newsman for the National Public Radio network familiar with one of the games. As a result, a commentator for "All Things Considered," the network's rush-hour news and comment show, interviewed the author of *O'Hare.*

Then the *Wall Street Journal*, in the second section of its August 17 edition, ran a two-column story "For Some Would-Be Controllers, Guiding Airplanes is just a Game." After that, things really started rolling.

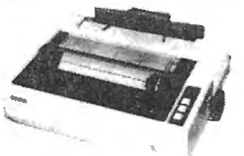
Marketing Manager Mary Reed said Instant Software and its distributors had contact with about 15 major news organizations including *Time* magazine and the CBS television network. A CBS affiliate in Denver, CO, had pictures of the Instant Software product, *80 Microcomputing* and *Kilobaud Microcomputing* on their evening news broadcasts. A major Boston radio station ran a four minute telephone interview with Reed on the subject. ■

SMALL BUSINESS COMPUTERS

118 SO. MILL ST. • PRYOR, OK 74361 • (918) 825-4844 *TM TANDY CORP.

SUPERPRINTER THE EPSON MX-80

520



This is the new Epson MX-80 dot matrix printer. It does just about everything you could ask a printer to do-quickly, quietly, reliably. In fact, for OEM installations, the MX-80 may be the single best, all-around printer you can buy. But that's not the best reason to buy it.

The MX-80 prints bidirectionally at 80 CPS in a user-defined choice of 40, 80, 66 or 132 columns. And if that's not fast enough, its logical seeking function minimizes print head travel time. The MX-80 prints 96 ASCII, 64 graphic and eight international characters with a tack-sharp 9 x 9 matrix. For a long time. Epson printers are known for reliability and the MX-80 is no exception. But that's not the best reason to buy it, either.

The print head has a life expectancy of up to 100 x 10^6 characters, and when it wears out, just throw it away. A new one costs less than $30 and the only tool you need to change it is attached to the end of your arm. The MX-80 is compact, weighs only 12 lbs., and the whole unit, including the two stepper motors controlling carriage and paper feeding functions, is precisely controlled by an internal microprocessor. But even that isn't why you should specify the MX-80.

The best reason is this: because Epson makes more printers than anyone else in the world, we can afford to sell each one for a little less.

IF YOU BUY AN MX-80 AND DON'T CHECK OUR PRICE — YOU PAID TOO MUCH!

OKIDATA MICROLINE 80 PRINTER — $399.00

522

The Microline is built on a rugged cast aluminum base to withstand the rigors of continuous business use. It is driven by two motors and will run all day with no duty cycle limitations. Microline printers use a dense 9 x 7 dot pattern to produce crisp, clean copies, first copy to last. The seven pins in the head are "fired" using energy stored in tension members. This technology permits the use of short, low mass pins made with an extremely hard alloy. The head produces less heat, thereby extending its life.

THE REST OF THE FAMILY

523

OKIDATA MICROLINE 82A — The same as the Microline 80 but prints at 120 CPS and has lower case descenders. **$599**

OKIDATA MICROLINE 83A—The wide carriage big brother to the microline 80 and 82A. Features both RS-232 and parallel interfaces, 120 CPS bidirectional printing, 9 wire print head and adjustable forms length. An excellent printer for only **$899!**

The **EPSON MX-70-$ Call**
The **EPSON MX-80 FT-$ Call**
The **MX-100-$ Call**
ALL OTHER EPSON OPTIONS

IF YOU BUY ANY EPSON PRODUCT AND DON'T CHECK OUR PRICE — YOU PAID T O O MUCH!!

Call 918-825-4844 for prices.

AW...WHAT THE HECK

521

RAM Memory Chips for the TRS-80 *Apple* and others

It is the policy of American Business Computers to offer merchandise at the lowest price possible. Several months back, we began selling RAM Memory Chips for the TRS-80 for $45.00 per set. Someone else sold chips for $44.00. We sold them for $38.00. They sold them for $37.95. So we say "AW...WHAT THE HECK!" Let's see the other guys beat this price.

$19.95 PER 16K SET

These chips are brand new "4116's". These 200 nanosecond chips are fully compatible with all TRS-80 products. Instructions for insertion are included; however, the dip shunts required for converting a 4K Model I to a 16K Model I are not included at this low price.

★ RETROFIT YOUR ★ MX-80 TO AN MX-80FT

524

ADD A PLATEN TO YOUR MX-80 TO ALLOW USE OF LETTERHEADS OR ROLL PAPER. **$69.95**

DIRT CHEAP ...$265

***TRS-80 MODEL I VERSION**
Complete with Power Supply & Cabinet

TEAC DRIVES

But Highly Reliable. Yes, our Mini Disk Drives have an estimated 8000 Power-On-Hours Mean Time Between Failure (*That's 8 Hours A Day, 5 Days A Week, for 3.85 YEARS)*. Our drives also feature a lead screw head positioner - *not a cheap band-type positioner*. If you purchase a Mini Disk Drive with a band-type positioner, you are buying an INFERIOR PRODUCT! If you have any doubts about our drives, our *ONE YEAR* Warranty should convince you that *we* know this unit is the *best* drive at the *best* price on the *American* market. *This is one example where you can pay less and get more.*

*TM TANDY CORP.

NOTICE
BY THE TIME YOU READ THIS, OUR **DIRT CHEAP MINI DISKS** SHOULD BE AVAILABLE FOR THE MOD III

525

918-825-4844

80 APPLICATIONS

by Dennis Kitsz

"When you order Extended Color Basic be sure to ask for the kit."

Say, folks, I've got a Radio Shack Color Computer, but one that's just a little different . . . it has Extended Color Basic with 32K memory, provides a clean, composite video output, and displays uppercase and normal lowercase letters. No, you can't buy one exactly like it at Radio Shack (yet?), but you can have one for less than the cost of an off-the-shelf 16K Extended Basic machine. Want one?

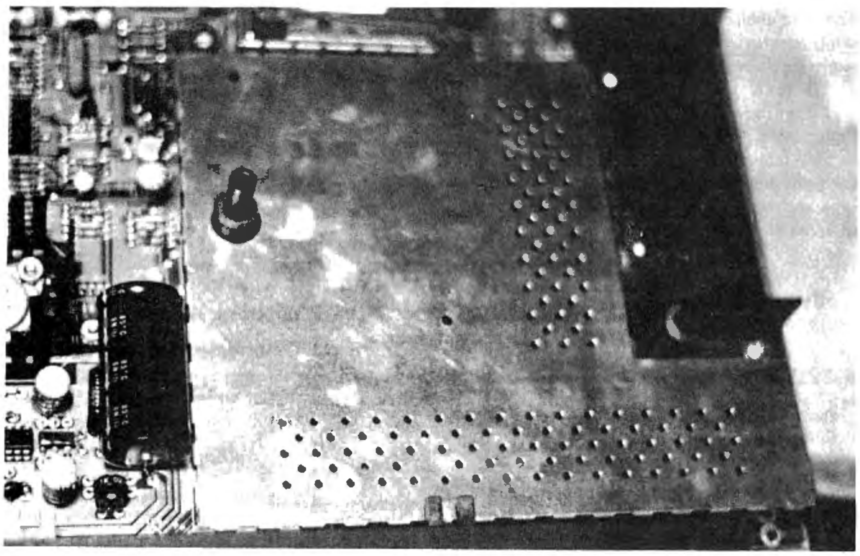


Photo 1. Main CPU activity is shielded by a metal cap, and by plate on the underside of the board, to prevent radio frequency interference (RFI) with televisions and radios.

This month's column is a potpourri of simple Color Computer expansions I have collected over the past few months. As sold, the machine is a fine investment for $400, but for a few dollars more you can install Extended Color Basic ($99 from Radio Shack), two banks of 16K RAM ($18 each from non-Shack suppliers), composite video ($2, build it yourself), and a plug-in character board (about $50, using the printed circuit pattern in this column).

Some folks have suggested that I make modifications just for fun. Fun isn't enough. When I purchased the Color Computer I was ready to be convinced that it needed no changes at all. It looked like the perfect second-generation personal computer. I anticipated with excitement installing my Extended Color Basic and seeing those rumored amazing high-resolution graphics come alive. Then I learned that Radio Shack didn't want me to install Extended Basic (but they would do so for an additional fee); and that the extended version required 16K RAM (which they also would install at a high price and for an additional fee).

About the same time, I borrowed a color monitor from a friend, only to find that this beautiful display had no RF input . . . and my Color Computer had no composite vid-

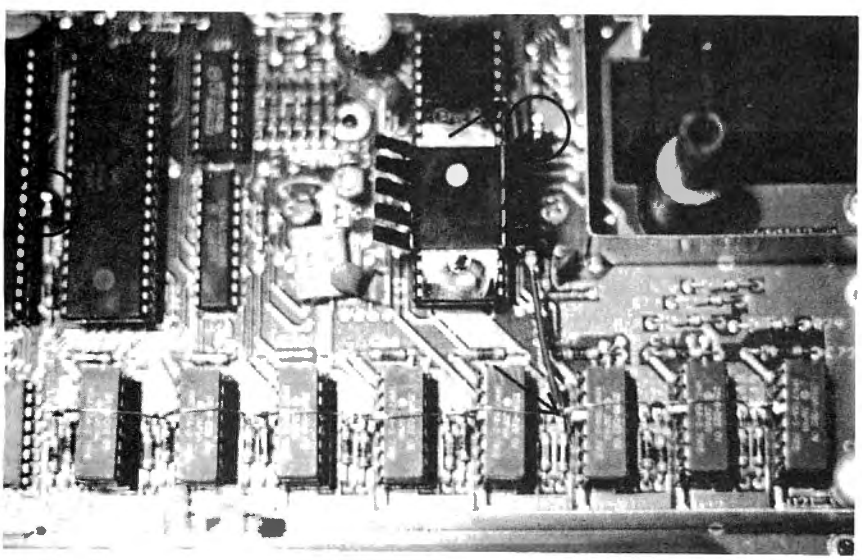


Photo 2. CPU area uncovered. Circles show 4K/16K jumpers, arrows point to 33-ohm resistor connecting SAM pin 35 to the second bank of memory.

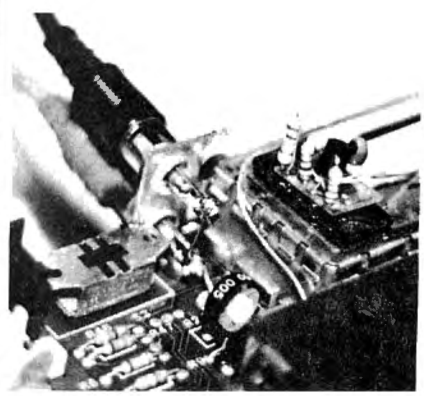


Photo 3. Video buffer amplifier affixed to the modulator unit. Dual phono jacks provide composite video output. Y-connectors can be used to drive ten or more monitors.

eo output. It was back to black-and-white. Finally, I tossed together a short text editing program in Basic. I felt it would be more appropriate to write about the Color Computer while actually using it. A few minutes in front of a screen full of normal and reversed uppercase characters convinced me that real lowercase was the only way to go.

"A few minutes... convinced me that real lowercase was the only way to go."

Extended Color Basic

Reluctantly I filled out my waiver of warranty (see August Applications), paid $99, and waited for my Extended Basic kit. The "kit" turned out to be an off-the-shelf Extended Color Basic manual and a single integrated circuit. When you order Extended Color Basic, be sure to ask for the kit; don't specify an "Extended Color Basic ROM," because that's all you'll get. At

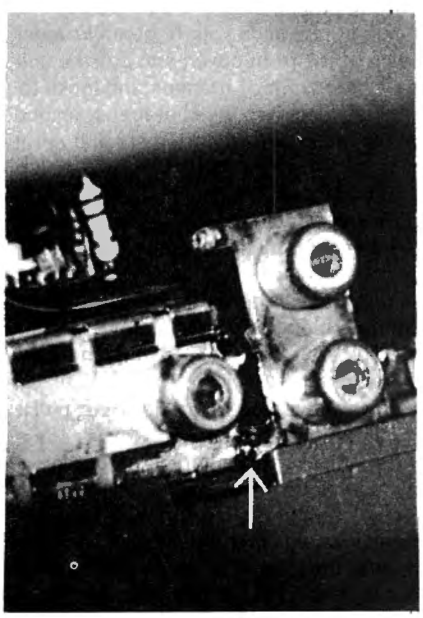


Photo 4. Arrow shows output jacks soldered to case of modulator unit (not the top). Two solder connections have been made for strength.

See List of Advertisers on page 402

least with the kit they throw in a book.

When you receive the Extended Color Basic (ECB) ROM it will be packed in a plastic tube or on a piece of foam. Leave it there until you install it. The device is static sensitive—not as frighteningly delicate as earlier integrated circuits, but still be cautious.

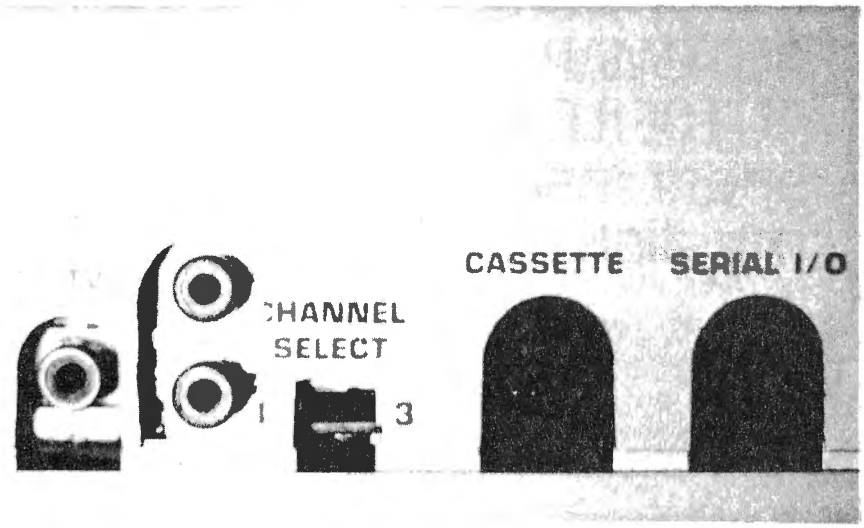


Photo 5. Video output jacks fit between television output and channel select switch. Carpentry should duplicate roughly the computer's style.

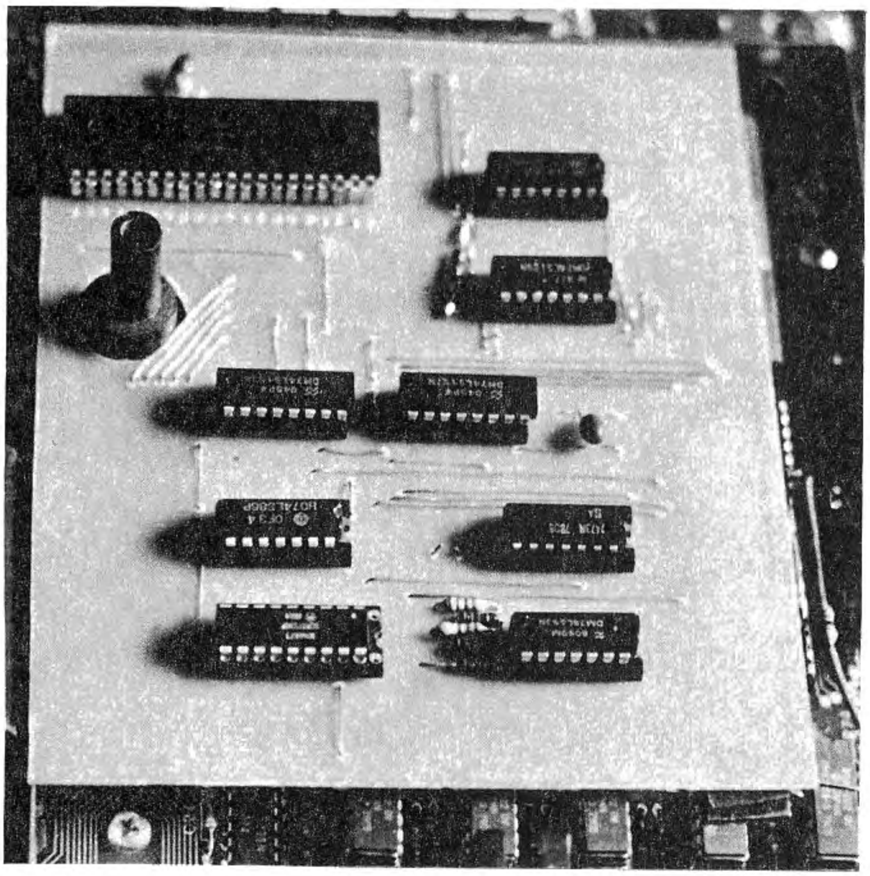


Photo 6. Complete lowercase board fits into VDG socket and around support port, and rest on banks of RAMs. When ICs are soldered to the board directly (they are socketed in this prototype), the RFI shield cover can be fit back in place.

Before attempting to install ECB, you need a computer with 16K memory. If yours is 4K, then order a set of 16K memory chips. These should be type 4116, uPD416, or equivalent (generally advertised as TRS-80 compatible memory expansion). Expect to pay about $18 a set of eight, but certainly no more than $25. Purchase 200 to 250 nanosecond (nS) memories. These will permit you to operate your computer successfully at twice normal speed. High speed is a software action, by the way, put in effect by entering POKE 65495,0. A 4K machine will probably hang up. A 16K machine might lock up too. More later on how to do it.

The Installation

When you have your ECB chip and eight 16K RAMs at hand in their original packages—together with a Phillips screwdriver, a nailfile, and a 4-inch piece of aluminum foil—you can begin. First, open your computer and flip it over. There are nine visible screw holes, but (on the units I have seen) only seven are used. There is an additional hole under a label which warns, "Opening case will void warranty. See owner's manual for warranty information." Yes, you will void the warranty if you open the machine, so don't proceed unless you are willing to take the risk and save over $100. When you open it you will find little to damage. Unlike the Model I, this machine is not delicate; it is physically sturdy and all sections are made to unplug, snap apart, or unscrew.

Unplug the machine, slowly flip the machine upright, slide your fingernails under the lip of the front cover, and pull the unit apart gently. Since it snaps and locks together, expect it to resist at first, then pop up; lift the cover up and back, and set it aside.

You will see: the keyboard, resting on posts and attached underneath with a removable cable; a circuit in a metal case at the very back; the cartridge connector at the right rear; and a large area covered by a metal shield (Photo 1). It is inside this shielded area that the ECB chip and RAM will be installed.

Lift the keyboard up off its posts, pull it forward about an inch, and look for some white plastic straps holding the metal shield in place; these may not be present. If they are, clip them off. Slowly work the metal cover up and off, examining how the shield is fitted in place so you can replace it later. Inside you will find several significant parts in large, 40-pin packages. Reading clockwise from the back right, these are: the 6809E central processing unit; below it a memory manager (Synchronous Address Multiplexer), type 6883; next, a row of eight RAM chips; up from there, two 6821 port chips which handle keyboard, cassette, audio, and other input/output work; and a 6847 Video Display Generator. Note whether the 6847 is in a socket or soldered to the board; we will return to it later. Finally, there is an empty 28-pin socket, and to its right a socket containing Color Basic.

Referring to Photo 2, find the two small blue or black jumpers (one is to the right of the S.A.M. and the other is between the 6821s). Associated with each jumper are three pins in a row, and the jumpers that currently connect the middle pin to one marked 4K on the board. The other pin is unconnected. To begin the conversion to 16K, lift each jumper off and move it so it connects the middle pin to the 16K pin, leaving the 4K pin free. Now slip the nail file under each 4K RAM, rock it slowly from each end alternately, and lift it out of the socket. Place it, pins down, on the piece of aluminum foil.

Remove a 16K RAM from its package, holding it by the ends of the integrated circuit (not by the pins). Notice that one end has a notch, dot, or both. Orient it in the same direction as the other 4K RAMs (the notch pointed away from the keyboard) and carefully insert it in the socket. Be sure that no pins slip outside the socket or bend underneath the IC. Do this for each memory chip in turn. Double-check your work, make sure all your tools are out of the way, plug it in, and power up; you need not replace the cover for this test, but keep your hands out of the works.

The screen will clear, but the Color Basic sign-on message will take a little longer than normal to appear. Enter PRINT MEM, and you should discover bet-

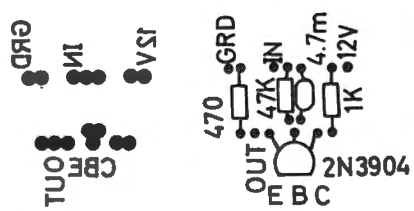


Fig. 1(a). Video buffer amplifier printed circuit layout, viewed from above through the board, and (b) parts mounting diagram, viewed from above.

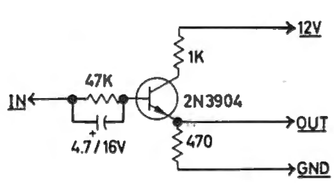


Fig. 2. Video buffer amplifier schematic.

See List of Advertisers on page 402

ter than 14K available memory (the screen uses your regular memory, so this takes away from memory available for Basic use). Problems will show as a locked-up machine, only 4K of memory still reported, far under 14K memory reported, or random crashing. A locked-up machine indicates RAM inserted backwards, pins crushed underneath or outside sockets, or (least likely) a bad memory chip. Recheck your work. If 4K memory is still reported after PRINT MEM, make sure the two jumpers are properly placed in the 16K position. Too little memory reported might point to a bad memory chip; power up several times and see if this value changes, suspect a chip which is too slow (be sure to get 200 nS RAMs). Finally, if the system randomly crashes, you probably have a RAM which is not fully inserted in the socket; check again.

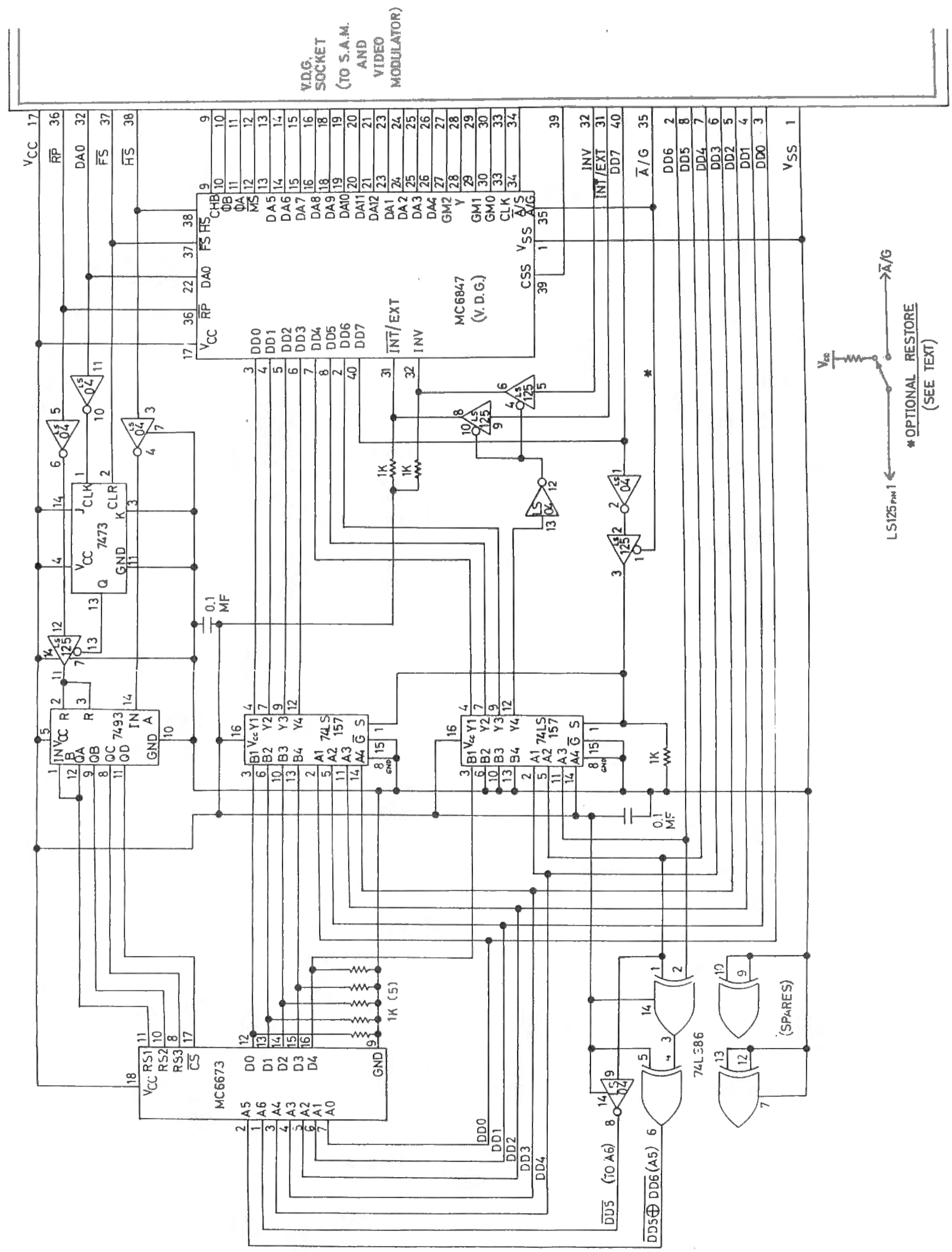


Fig. 4. Schematic of the lowercase modification for the Color Computer.

If all is well, remove the line cord. This is important, because the power supply transformer inside the case is always "hot." Insert the ECB ROM chip in the empty socket next to the Color Basic ROM. Position it carefully, with the notch in the same direction as that of the Color Basic ROM. Press it into the socket, being sure not to bend any pins underneath or letting any slip outside the socket. You have just performed Radio Shack's $20 installation.

Plug in and power up the computer. Your screen will now display an Extended Color Basic copyright notice. PRINT MEM will report 8487. 8487 on a 16K machine? Yes, because initially ECB sets aside 6K of memory for high resolution use. PMODE0:PCLEAR1 <ENTER> will give you about 14K of usable RAM.

Now power down, unplug the machine, and look over your work. Replace the metal cover. This is important because it prevents your television or radio from picking up interference from the computer. Place the keyboard back on its pegs, replace the cover and re-insert the screws. Your 16K ECB computer is complete. Wrap up the package of 4K RAMs in the aluminum foil; you may be able to trade or sell them someday.

About High Speed

As I mentioned, the Color Computer has a built-in high-speed mode, accessible by POKEing 65495 with any value. POKE 65494,0 will return to normal speed. With 200 to 250 nS memories, your computer should be able to run at high speed, which will be evident if the cursor flashing rate increases. The hangup is the pair of 6821 port chips which handle keyboard scan, cassette I/O, certain aspects of video, etc. These are rated for 1 MHz operation (the Color Computer normally operates at 0.9 MHz), and occasionally won't do the job at 1.8 MHz. Try it and see; if you are sure you'll need the higher speed in the future, then opt for replacing the 6821s with type 68A21, a faster device.

32K RAM for the Adventurous

Little effort is required to add another 16K of memory. Some solder, and iron, needlenose pliers, wire snips, one 33-ohm resistor, a few inches of light solid wire, nine pieces of wire insulation, three feet of aluminum foil, and another 16K RAM set are all that is necessary. You also must have time, patience, dexterity and perhaps a magnifying lens, but the process is not formidable.

In short, you will fit each memory IC atop those already in the computer, and solder it there. One pin will be connected not to the RAMs below, but together with the other RAMs in the new set. A resistor will run from the SAM chip to this connection in order to select the second 16K block of memory.

For safety, line your worktable or desktop with aluminum foil, and set your tools, parts, and computer on it. Ideally, use a non-static felt workstation (#WS-9010, sold for about $15 plus shipping by Wescorp, 1155 Terra Bella Avenue, Mountain View, CA 94043), which includes a grounding wrist strap. Open the computer case according to directions above, and put the top and metal shield aside. For reference, locate the eight memory chips (socketed, nearest the keyboard), and the Synchronous Address Multiplexer (SAM, the 40-pin chip U10 just above and to the right of the memories).

Examine your new set of 16K RAMs. If they have gold pins, they should fit in place as is. Otherwise, you will see that the pins fan out slightly from the body. Keeping your arm grounded against the aluminum foil, hold the sides of the IC between thumb and forefinger, and squeeze until the pins are perpendicular to the body or canted very slightly inward. Keep the pins in line with each other. Orienting the notch in the proper direction, temporarily fit one of these RAMs atop one of those in the computer. Make sure the pins line up precisely with those already in place, that they fit snugly, and that no pins slip in between others. Form each memory chip this way.

Identify pin 4 of the new memory chips. Looking at the top of the chip with the notch facing away from you, this is the fourth pin down from the upper left. With the needlenose pliers, bend this pin gently (not sharply) out and upward so it rises above the top of the IC. Do this for each of the eight chips.

There are three ways to continue with this project, depending on your soldering technique.

- First method: Fit all eight memory chips into place. Check that in bending pin 4 you have not altered any other pins. With a small, hot soldering iron, flow a minimal amount of solder on pins 1, 8 and 9. It should join the new chip with the one below it; be careful not to melt the socket of create globs of solder.

- Second method: Fit all eight memory chips in place. With a small, hot soldering iron, flow a minimal amount of solder on all pins (except pin 4, which is pointing upward). Solder alternate pins as you work your way around the chip; allow time for it to cool if it feels uncomfortable to the touch. Be very careful not to melt the socket or create solder shorts or globs.

- Third method: Remove the eight memory chips from the sockets, piggyback the new chips on these, and solder them pin-for-pin (except pin 4) on the workbench. Since they are out of their sockets, you will have to be doubly careful not to overheat them. Then re-insert the piggyback pairs in the sockets, orienting them correctly.

A wire must be connected to all the pin 4's. The easiest way is to snip off a piece of bare wire long enough to connect all the pins together in a line. Solder this wire to the first memory chip to the left, then slip a piece of insulation on the wire. This piece should be long enough to cover the wire only as far as the next chip. Solder the wire to the next pin 4, slip on another piece of insulation, then move on to the next. One exception: between the fifth and sixth memory chips (U23 and U20), leave a little extra wire and put two shorter pieces of insulation in place.

You will now use the 33-ohm resistor. Slip a piece of insulation over each end so about 1/8 inch of bare wire is visible. Solder one end to the point in the middle of the two pieces of insulation between the fifth and sixth memory chips. Locate the SAM chip (U10); identify pin 40 (the top right-hand pin), and count backward until you get to pin 35. The other end of the resistor will comfortably reach this pin. Using a very small amount of solder, attach the other end of the resistor here.

Refer to Photo 2 for details on this mod-

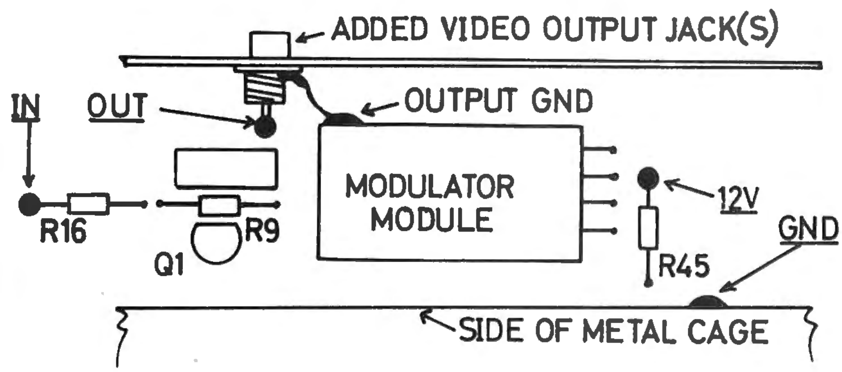


Fig. 3. Four wires connect the video buffer amplifier to main PC board at R16 (output to amp), R45 (12-volt power supply), metal cage wall (ground), and phono jack center pin (video output).

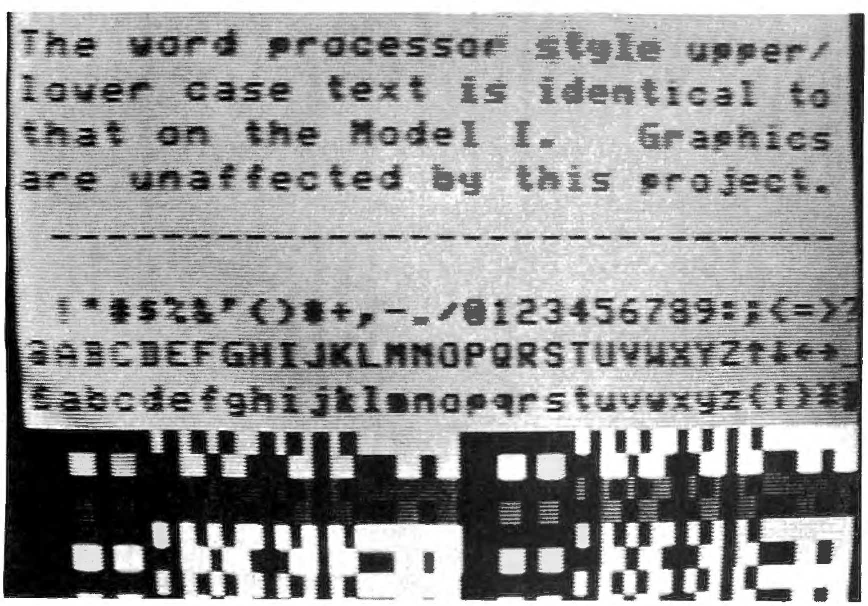


Photo 7. Latest character generator used for the Model I TRS-80 produces 5-by-7 dot matrix characters with descenders.

See List of Advertisers on page 402

ification. The arrows show connection points for the 33-ohm resistor. Cavalierly I have left off the insulation on my connection wire; don't do that, because the RF shield cover can short against the wire. The additional bank of 16K memory is now ready to use. Plug in and power up the computer. The sign-on message should take a while to appear; when it does, enter PRINT MEM. Over 24K of memory should be revealed, the rest (as I mentioned above) being reserved for high-resolution graphics.

Problems and Causes

Problems? They may include: a locked-up machine with no sign-on message; random crashing and lockup after the sign-on message; no evidence of the second 16K bank of memory; an improper amount of memory display upon PRINT MEM.

Probable causes: A locked-up machine points to memory inserted backwards, shorts between pins, memory lifted out of its socket, or memory damaged by heat (the last is unlikely but possible). Solder globs or hairs on the SAM chip will also cause this. Random crashing after sign-on probably results from pins not seated properly on the memory chips below, not fully bent, corroded, or otherwise not making proper contact. No evidence of the second 16K memory bank is caused by one pin not making any contact at all with the one below, or a bad solder connection on the 33-ohm resistor. An improper amount of memory may be caused by a bad or slow memory chip, or, if the displayed amount of memory changes, by an improperly seated or soldered chip. Replace the RF shield and computer cover. The 32K memory modification is now complete.

By the way, here's the theory of what you have done. All the memory lines (address, data input, data output, refresh/select, and power) are identical but one. Therefore, the second set can be paralleled (piggybacked here) on the first set. Pin 4 is the only exception, and it is the memory address line MA7. Pin 35 of the SAM chip provides this signal whenever a memory address from 4000 to 7FFF (decimal 16384 to 32767) is requested.

Sinking SAM

Look again at Photo 2. Something is attached to the top of the SAM chip, with a white substance smeared on it. After your computer has been operating for a while, place your finger on the SAM chip; if it is uncomfortably hot, you may want to extend its life with a heat sink. Only two machines I have seen (felt?) have this heat buildup in the SAM, so you may not need this mod.

Obtain a finned, black, TO-220 heat sink and flatten the fins somewhat. With white silicone heat-sink grease, cover the top of the SAM and the bottom of the heat sink. Press it in place, fins up; the grease will hold it there. Replace the metal RF shield. It should just touch the heat sink if you have bent the fins properly.

Sweetening the Video

The television (RF) output of the Color Computer is excellent, and if you have no access to a color monitor, it gives a fine video image. But there is a purity in coupling video directly to a monitor without

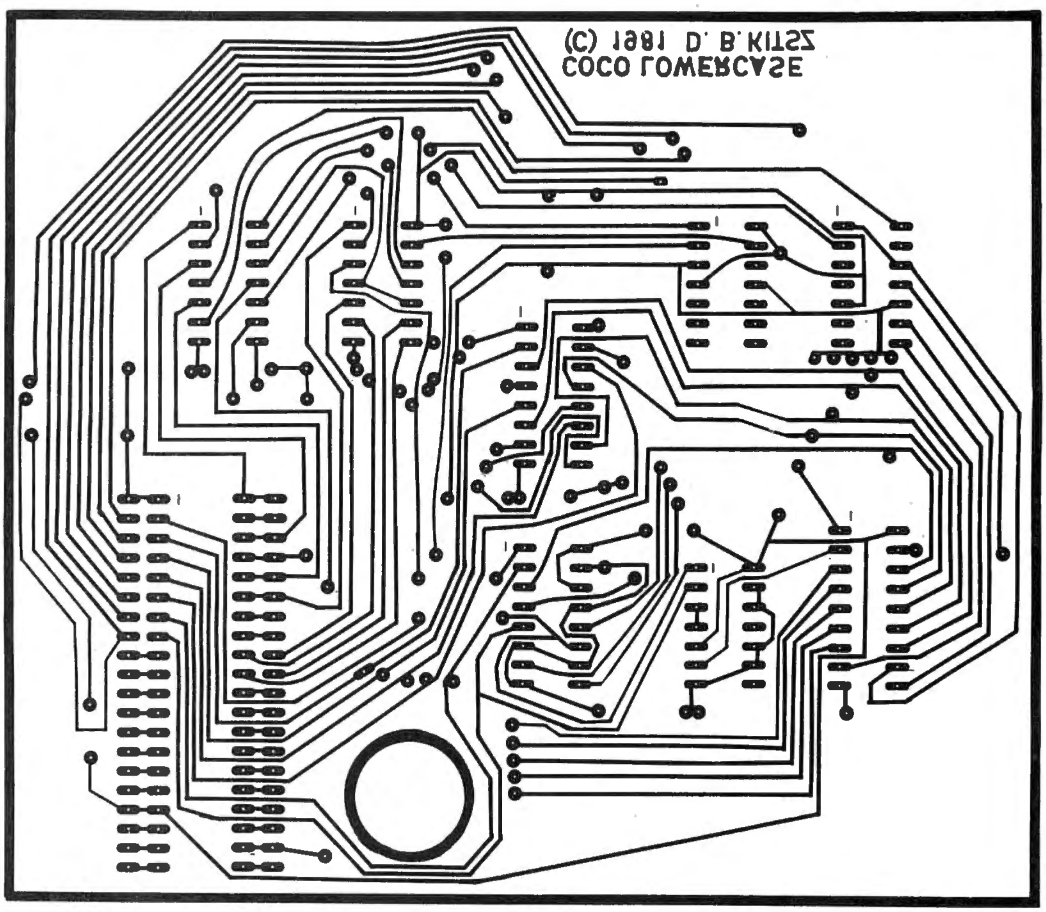


Fig. 5. Complete printed circuit board layout for the Color Computer lowercase modification. Post hole must be cut before mounting parts on the board. Note that this is the bottom of the PC board, viewed from above and through the board.

converting to RF and back again at the television. Further, the signal sent out by the Color Computer is too weak to power a number of sets for display purposes or group use, but the video buffer/amplifier shown here can run a dozen sets or more. High-resolution graphics are more finely detailed, particularly on black-and-white sets.

A printed circuit board layout for a ¾-inch-square board that affixes to the video modulator case with double-face tape is shown in Fig. 1. The schematic for a simple transistor buffer/amplifier which attaches ahead of the RF modulator and provides a clean, fast, composite video output is shown in Fig. 2. The unit can be assembled on perfboard as well; be sure to keep leads short. Radio Shack sells 2N3904 transistors (276-2016, or in packs of 15, 276-1603), and the resistors. A common electrolytic capacitor may be used (Shack number 272-1024), though I use a bead tantalum type for cleaner signal (no longer available from Radio Shack).

For video output, I installed a pair of ordinary phono jacks similar to those used for the computer's TV output. Solder can be flowed directly to the modulator's case to ground these jacks, but don't solder the cover in place (in case video alignment is needed at some future date). All that remains is the carpentry. The plastic case is soft, so it can be cut easily with a fine coping saw blade and filed smooth. Photos 3, 4 and 5 show the results of mounting the video buffer/amplifier, fastening the jacks, and cutting the case. The locations where the four leads from the video buffer/amplifier must be soldered (input, output, 12-volt supply, and ground), are identified in Fig. 3.

If you have a parts "junk box", this is a free project; otherwise, the investment is about $2. If you have no time at all, a complete kit of parts is available from MSB Electronics (see the end of this article for details).

Real Lowercase

If the Color Computer has any truly serious deficiency, it is the lack of reasonable lowercase letters. For programming and prompting, that lack is not too bad;

✓See List of Advertisers on page 402

but for text editing of any kind, a screen full of black letters and green letters in black blocks is not merely inconvenient, it is ridiculous. You can provide legitimate lowercase with the Color Computer, though by no means as easily as on the Model I.

On the Model I, the lack of lowercase characters was merely a result of the absence of one bit of video storage. By adding that bit, lowercase characters already present in the Model I's character generator became available to the user. The Color Computer uses a vitally different method of accessing characters: the 6847 Video Display Generator (VDG).

Within this integrated circuit the characters are created, and both coarse and high-resolution graphics modes are produced. Together with the SAM chip, flicker-free displays of alphanumerics and graphics are possible. The VDG contains an internal 64-character generator, and no lowercase letters. To simulate them on the Color Computer display, inverted characters are selected when bit 6 goes low. Software converts the internal character set (00 to 1F) to correct ASCII equivalents with green letters on black for lowercase (60 to 7F), and the inverted character set (20 to 3F) to correct ASCII equivalents with black letters on green for uppercase (40 to 5F). The numbers and symbols from the black-on-white set are not printed. If this sounds a little confusing, it is. Enter this short program:

```
10 CLS:PRINT:PRINT:PRINT:PRINT
20 PRINT:PRINT:PRINT:PRINT
30 FOR X=1024 TO 1151
40 POKE X,Y : Y=Y+1 : NEXT
50 FOR X=0 TO 127
60 PRINT CHR$(X); : NEXT
70 GOTO 70
```

The top four lines in the first group show the character set actually received by the processor; the second group of lines presents the character set converted to normal ASCII by the Basic screen display routines.

When bit 7 is set high (hex 80 to FF), the semigraphics mode is selected, allowing eight groups of graphics characters to be displayed. These graphics blocks are identical in both displays above since standard ASCII values are not assigned to these codes.

Table 1 describes the 6847 display generator and the functions of each of the pins, and how the VDG fits into the Color Computer system. Each input or output used for the modification will be described later.

The key to displaying an independent set of characters is the INT*/EXT connection (pin 31). According to the Motorola data sheet, this mode "uses an external character generator as well as a row counter. Thus, custom character fonts and graphic symbol sets with up to 256 different 8×12 dot 'characters' may be displayed." To display lowercase on the Color Computer, then, the VDG must be fed information by some external device (such as the Model I character generator); a row counter must be added to select the lines of dots to be displayed on the screen; and all other modes of graphics must be maintained unchanged for compatibility with normal Color Computer operations. Ideally, the whole thing

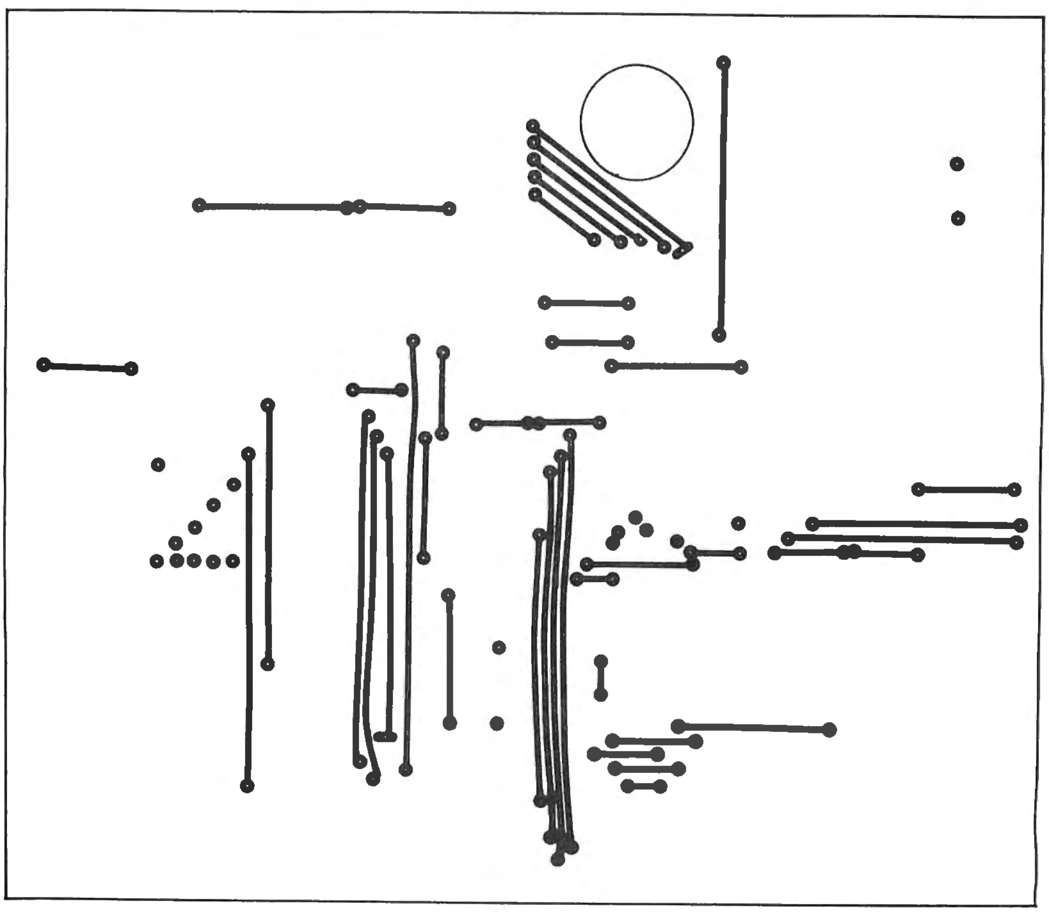


Fig. 6. Jumper arrangement for lowercase board, viewed from above. Jumpers should be soldered in place before the integrated circuits are mounted. Note that this figure will produce a double-sided PC board.

should be switch-selectable.

The complete schematic for a lowercase addition is shown in Fig. 4. The VDG is intercepted and connected to several outboard integrated circuits. The heart of the external selection process is the pair of 74LS157 multiplexers in the center of the diagram. A multiplexer is like a traffic light—only one pattern of vehicles may proceed at a time. In this case, eight lines of data are switched from the internal mode (the multiplexer A inputs) to the external character generator (the B inputs). In charge of this selection process is data bit 7 (DD7, pin 40) together with the alphanumeric/graphic select line (A*/G, pin 35). When alphanumeric mode is selected, the signal from DD7 (inverted by the 74LS04) passes through the 74LS125 buffer. If DD7 is high, semigraphics are selected (coarse graphics with hex codes 80 to FF), which causes the multiplexers to connect their A inputs to the Y outputs. By following data lines DD0 through DD6 from the socket, you can see that these seven signals will pass through the multiplexer directly to the VDG. The eight A input is tied high, and passes through an inverter. This eight signal opens up the other 74LS125 buffers to the INV signal (pin 32) and the INT*/EXT signal (pin 31). In other words, when semigraphics are selected, all signals proceed normally to the VDG, as if there were no other circuitry in place. When high-resolution graphics are selected (A*/G is high), the result is the same—completely normal operation of the computer.

When line DD7 goes low (meaning alphanumerics are being printed on the screen), the B inputs of the multiplexers are connected through the Y outputs to the VDG. Five dots are output from character generator lines D0 to D4, as selected by the address inputs A0 to A6. The computer's video display is made up of 12 rows of dots per screen line. Because the character generator must know which set of dots to output for each line, a counter is needed which counts from 1 to 12 in binary to select these dot rows. Since the Model I character generator is only a 5×7 matrix of dots, however, this chip must be deselected (turned off) during the final counting stage.

Count Sheep Instead

All of this counting is done by the combination of a 7493 binary counter and a 7473 flip-flop. The 7493 counter is configured to count from one to 16. Display scan lines are selected by sending a clock signal provided by the horizontal synchronization (HS*) pulse, and the count is reset to zero (cleared) by the row preset (RP*) pulse, where it begins counting again. This produces a complete row of characters. Counting stops completely when the video beam is out of the frame by the field sync (FS*) signal, and counting begins again when the first character is fetched from memory (DA0). In fact, this counter only counts to 12 before it is cleared for the next screen line. Since this character generator produces a 5×7 matrix, the chip must be turned off to avoid redisplaying the first five lines of any letter immediately below it. The generator is turned off by bringing the chip select line (CS*) high; a count from 8 to 12 brings the QD output of the 7493 high—perfect for deselecting the character generator at that point. The five resistors hold the character generator's data outputs low during the deselection phase, and assure that the characters will be presented on a clean screen, and not one filled by characters underscored with black rectangles.

The final difficulty is the conversion process accomplished by the software. As noted earlier, since the internal character generator uses only six bits of data, these characters fall outside the normal ASCII positions. They are input to the CPU which then performs a software change to make them compatible with usual ASCII codes. The Model I character generator, however, provides proper ASCII codes. To make the character generator look like the VDG, then, the lowercase must be swapped with control characters, numbers must be swapped with lowercase, and uppercase remains the same. The 74LS86 XOR gate configured as an exclusive-NOR gate correctly swaps signals that enter address line 5; DD5 is inverted and presented to address line 6. This conversion (see Table 2) provides the

Signal Name	Pin Number	Description of Function
Vcc	17	Power supply voltage, +5v.
Vss	1	Power supply ground.
CLK	33	Color burst clock input, 3.58 MHz, provided by the SAM chip.
DA0-DA12	22-26, 13-16, 18-21	Address lines to display video memory, set to high impedance during MS*. These address lines are not used in the Color Computer since the SAM chip provides the characters as needed.
DD0-DD5	3-8	Data from display memory, provided by the SAM chip.
DD6	2	Data from display memory in graphic mode, color data in semigraphic mode. Tied to the INV line in the Color Computer. Information is provided by the SAM chip.
DD7	40	Data from display memory in graphic mode, color data in semigraphic mode. Provided by SAM.
phase A	11	Chrominance and luminance analog
phase B	10	outputs to video modulator chip
Y	28	MC1372, and chroma bias reference
CHB	9	to phase A and phase B levels.
RP*	36	Row preset timing output for an external character generator. Not used in the Color Computer.
HS*	38	Horizontal sync timing output for an external character generator, and used by the SAM chip for memory timing.
INV	32	Inverts video in alphanumeric mode. Normally in invert position for uppercase letters in the Color Computer, and tied to DD6.
INT*/EXT	31	Switches to external character generator in alpha mode and between semigraphics modes 4 and 6. Handled by SAM.
A*/S	34	Switches between alphanumeric and semigraphics modes. Handled by SAM.
MS*	12	Memory select line input disables address buffers. Not used in the Color Computer.
A*/G	35	Switches between alpha/semigraphic modes and medium/high-resolution display modes.
FS*	37	Field synchronization output goes low at bottom of display area, and returns high at top of video display. Not used in the Color Computer.
CSS	39	Color set select chooses colors for alpha display in semigraphics 6 and full graphics mode.
GM0-GM2	30,29,27	Select for one of eight graphics modes. Color Computer software does not access them all, although the hardware makes them available.

Table 1. 6847 Video Display Generator pin assignment in the Radio Shack Color Computer.

BASICS II/CMD — Mod I or III — $69.95

BASIC/S II is a BASIC compiler for a powerful subset of TRS-80 Disk BASIC (Mod I/III) which is itself written in BASIC, but now is a stand alone / CMD file compiled by BASCOM(c), and is FAST! (previously sold as BASIC/S 3.1) It runs under almost ANY DOS, Mod I or III (as do the /CMD files it generates).

BASIC/S II will compile BASIC programs up to about 10K in size (but since one compiled program can run another with no loss of variables, this is not a serious limitation). Variables allowed are integers and strings (A-Z and A$-Z$) and also arrays of integer and string (1 or 2 dimensions). Array names can be any length, with all characters significant. Integer arithmetic is limited to addition, subtraction, multiplication, and division with at most two operands on the right hand side (eg A=B*C or Z=D/F). Array elements may be used here as usual, as may constants (as in A=ARRAY (7) *5, for example).

Most Level II string functions are supported: INSTR, STR$, VAL, CHR$, ASC, LEFT$, RIGHT$, MID$, INKEY$. Complex string formulae are not allowed (thus you can't say A$=LEFT$ (X$+Y$,2); you would need to say something like B$=X$+Y$:A$=LEFT$(B$,2), instead.

Disk I/O IS supported, both sequential and random (with LRL < 256 as well). You can have up to 10 files open at one time, and two of them may be random files. Disk statements that are supported include OPEN, CLOSE, LINE INPUT #, PRINT # FIELD, LSET, GET, PUT, LOF, EOF, CVI, and MKI$.

LZAP/CMD — Mod I or III (LDOS only) — $39.95

LZAP/CMD is the most powerful, easy to use ZAP routine ever written, and it is for LDOS Mod I or III! LZAP uses the resident drivers of LDOS, thereby taking FULL ADVANTAGE OF THIS POWERFUL OPERATING SYSTEM. LZAP will auto-configure itself to drive capacity. It will support any # of drives from 0-7, any cylinder count up to 255, and any sector # to 255. Will page between double sided drives, 8" drives, LX-80, and hard drive, 5" floppies of all sizes at same time without getting lost!

Contains these features:

(R) ead	(E) xit	(D) ebug)	(R) emove Password
(A) scii Modify	(C) compare	(G) o Directory	(I) Go Extent
(M) odify Hex	(W) rite	(B) Locate Byte	(Z) ero buffer
(F) ind file	(H) ash code	(L) ocal search	(P) rint screen
(S) earch Global	(X) Escape	(T) oggle	0-7 page drive

(space bar) brings up a command menu for learners and QUICK reference at any time! Also contains an ERROR DICTIONARY that can be turned off to override those crashed or protected files for examination.

QSD Utility Disk #1 for LDOS (c) (Mod I only) — $69.95

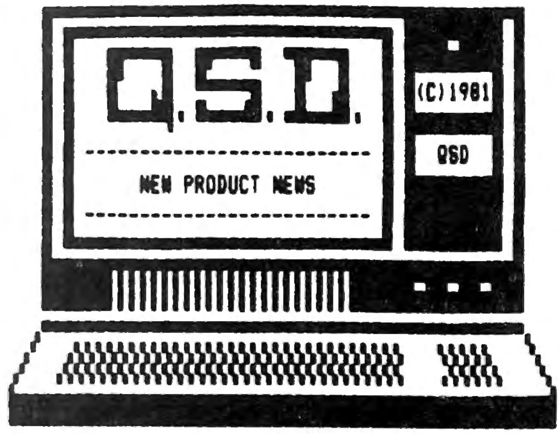
Increase the speed and power of LDOS!! Contains:

LZAP/CMD* RESIDELD/CMD* CLONE/CMD
CHANGE/CMD DCAL/CMD* VDISK/CMD*
STAT/CMD* BINHEX/CMD*

And also contained on the disk, by Kim Watt (author of Super Utility):

LCOPY/CMD RECOVER/CMD DVORAK/CMD LPURGE/CMD*

*means "works in Double Density"



SUPER UTILITY!

This fantastic 24K stand alone machine language program allows rapid and easy access to controlling data. Super Utility, written by Kim Watt of Breeze Computing, Inc., comes with 30 pages of easy to understand, step-by-step instructions assuring maximum use of your valuable time.

Seven main menus combining 43 major programming or disk repair options makes this single program incredibly versatile. For fast and easy reference we've capsuled some of the best options below.

ZAP
Display Sector (Disk, File)
Display Memory
Compare Disk Sectors
Copy Disk Sectors
Verify Disk Sectors
Zero Disk Sectors
String Search
Sector Search

PURGE
Kill Selected Files
Get Disk Directory
Zero Unused Directory Entries
Zero Unused Granules
Remove System Files
Kill By Category
Change Name, Date, Password, Auto Command
Change File Parameters

DISK FORMAT
Standard Format
Format Without Erase
Special Format
Read Address Marks

DISK COPY
Standard Copy With Format
Standard Copy Without Format
Special Copy (To back up most protected disks - for original purchasers use only)

TAPE COPY
Back up most TRS-80 tapes, no matter how it is recorded. (for original purchaser use only)

DISK REPAIR
Repair Gat Table
Repair Boot
Read Protect Directory Track
Recover Killed Files
Check Directory

MEMORY
Move Memory
Exchange Memory
Compare Memory
Zero Memory
Test Memory
Input Byte From Port
Output Byte To Port
Memory To Disk
Disk To Memory

Residing from 4,000H to 9FFFH, Super Utility has its own I/O drivers and does not use any ROM or DOS calls. So go to the heart of your protected disks and read and/or modify data using simple 1-key commands. Now time is on your side!

To Order... *Send check or money order for only $49.95 plus $2.50 shipping and handling

SUPER UTILITY 2.0 (SINGLE/DOUBLE DENSITY) $74.95

LDOS — MOD I OR III — $149.95

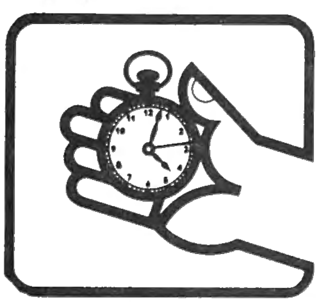
LDOS is the latest generation of sophisticated operating systems for the TRS-80 computers. LDOS is completely documented in a 252-page reference manual. Total support is provided from some of the most knowledgeable people in the industry. You owe it to yourself to investigate this system. Call or write for details.

ST80-III The Ultimate Communications Package: Price $150.00

The package includes the ST80-III™ smart terminal program and nine other communication utilities: Fully documented with easy to follow instructions, ST80-III™ is by far the best terminal product on the market today.

EDAS

EDAS is a sophisticated Editor and Assembler for the '80 Model I or Model III. All commands and SOURCE text can be entered in upper or lower case. Direct assembly for memory or multiple disk files by means of *GET assembler directives provides the capability of assembling huge source files with 30,000 bytes of symbol table. Direct assembly to disk or memory for faster debugging operations. DOS functions DIR, KILL, and LIST are available from within EDAS. The Editor provides block move & global change with BASIC syntax editing. EDAS provides power with ease of use. $79 + $S&H.



TWO Printers on ONE Computer

- FOR MOD I,II, AND III...
- PARALLEL ONLY
- SCHOTKEY CIRCUITRY
- FULLY BUFFERED...
- SWITCH SELECTABLE...

$139.95 IN STOCK!

— DEALER INQUIRIES INVITED —

QUALITY SOFTWARE DISTRIBUTORS

11500 STEMMONS EXPRESSWAY, SUITE 104
DALLAS, TEXAS 75229
PHONE (214) 484-2976
MICRONET 70130,203 ✓27

TRS-80 & Scripsit are Trademarks of Tandy Corp.

VISA

Price Does Not Include Postage

ADD $2.50 SHIPPING (USA)

MENTION THIS AD AND DEDUCT 5%

proper non-ASCII codes needed by the Color Computer software.

Some considerations remain. Ironically, the very attractive type font of the Model I character generator (actually a Motorola 6670 series shifted generator, type MC6673) contains a few non-standard characters in the high upper and lowercase positions (see Fig. 9), and two differences between generation of the Model I character chip. Especially unfortunate is the replacement of the carat or up-arrow with a down-arrow (!) and the replacement of the square brackets with an up-arrow and left-arrow. Programming in Basic looks mighty strange when exponentiation is represented by a arrow pointing down, and when program tracing frames line numbers with up and left-arrows. For these reasons, Fig. 4 shows an optional restore switch. A 1K resistor pulls the multiplexer select line high when the external generator is unwanted; the entire display system then looks unmodified. The switch mounts comfortably and without interference between the cassette and printer output jacks.

Another minor bother is the fact that the Model I character generator with descenders shifts the entire character set up one line on the screen in order to accommodate the descending characters when they come along. This means, when alphanumerics and graphics are mixed, the tops of the letters bump into the bottom of any graphic above them. They don't overlap, but they do touch, just as they do in the Model I. There are two solutions to this, one ugly, and one very attractive.

The first solution is to use the old Radio Shack character generator (if you can get one), without lowercase descending tails. A more elegant solution is to use a 7 × 9 matrix character generator, Motorola type 66740, which matches all the characters in the Color Computer set correctly, but provides lowercase with descenders and a much prettier 7 × 9 matrix of characters. Since there are more rows of dots to display, the 66740 device has data outputs D5 and D6. These are connected to the B2 and B3 inputs of the second multiplexer instead of connecting these inputs to ground. The chip is always selected (because it outputs blank lines where no character is present), so the four chip select lines (CS1 through CS4) are connected to ground. The extra dot data is chosen by feeding the 7493's QD output to row select 4 (RS4). Those are the only changes necessary to use the 7 × 9 generator, but note: the 66740 character generator has 24 pins and will not fit on the PC board layout in Fig. 5.

Building the Board

This is the first time I have presented a printed circuit (PC) board layout in this column (Fig. 5). Because all of us are only skilled amateurs at best, I have made this board single-sided using jumpers instead of double-sided, demanding plated-through holes. The jumper arrangement in Fig. 6 can be used to create a double-sided board. A complete board and kit of parts is available for this project (see end of article for details).

Making a printed circuit board is time consuming, but quite simple. Full directions are available from PC suppliers, but here is a summary; take note that all the chemicals used in this process are either flammable or toxic, so work with care. To make a PC board, take the layout in Fig. 5

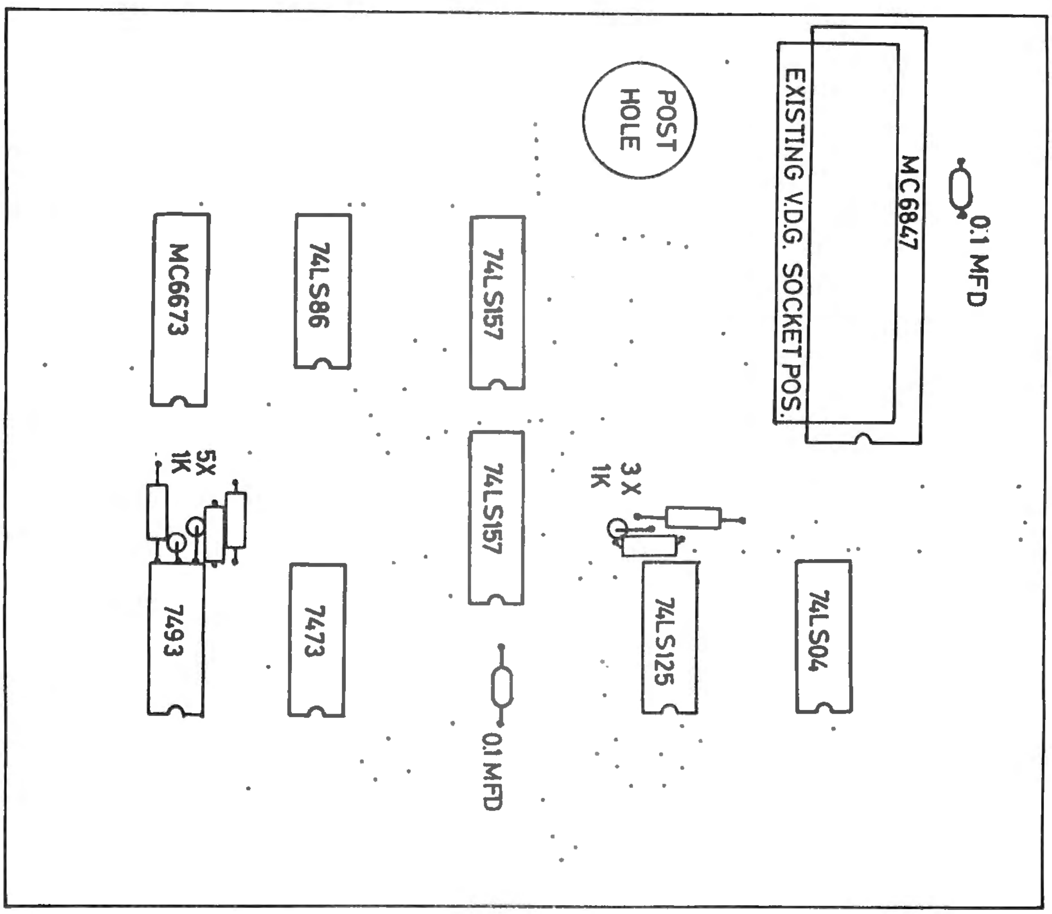


Fig. 7. Parts layout for lowercase modification, viewed from above. Three resistors are mounted vertically, but enough resistor lead should be left to bend the parts over after soldering.

✓See List of Advertisers on page 402

to your local newspaper or printing office and obtain an exact-size (100%) film negative. Clamp the negative firmly against a piece of negative photosensitive PC board (from Kepro or Vector) in a photo proofsheet frame, under a vacuum table, or just under glass weighted all around so there is no space between the negative and the PC board. The words "Coco Lowercase" should read correctly.

You may make your own board with negative PC photosensitizer. Work only under darkroom lighting or on the other side of the room from a shaded, dim, yellow bug light. Expose the board with a 3200 degree tungsten photoflood bulb (available at any photo store) at a distance of two feet for seven to ten minutes. Develop the board for one to two minutes in PC board developer (Kepro or Vector), and harden the emulsion in an oven at 200 degrees for about one half hour or by drying overnight. Drop the board into a glass or enamel tray of ferric chloride, face up, and gently agitate about a half hour until all the copper has been removed except where the traces are present. Wash the board thoroughly, dry it, and drill all the holes with a #66 hobby drill; this is most easily done in a small drill press. Cut the post hole to the outside edge of the copper band. Remove the remaining emulsion with steel wool, buffing until the copper is very shiny. Wipe the board clean with a dry cloth, and begin mounting the parts.

Mount the jumpers first; there are 42 of them. Put the resistors and capacitors in place next. Then mount the integrated circuits. If you use sockets, the RF shield will need extra help to remain in place.

The final few steps are a bit tricky. Insert a 40-pin wire-wrap socket three-quarters of the way into the board; refer to the parts layout (Fig. 7) to be sure you put it into the correct set of 40 parallel holes. Solder all 40 pins into place so that about ½ inch of wire wrap pin protrudes from the bottom of the board. Now clip the socket off, snipping close to the top of the board. Pull all the pins out of the socket, and slip the socket over the pins which are soldered to the board. It becomes a 40-hole grommet, helping the pins stay straight and guiding them for insertion in the VDG socket.

A 40-pin, solder-tail socket is inserted in the remaining holes on the board. It should fit easily in place; if its pins do not protrude through the board, double check that you have clipped the tops of the wire wrap pins closely.

Open the computer and remove the metal RF shield. Find the VDG, a 6847 chip mounted at the back left of the metal cage. Now remove the VDG from its socket. Lift it very gently with a nail file, rocking the file from side to side. If the chip moves a little and then sticks, push it back down and start again. Gradually it will work free. Insert it in the 40-pin socket on the new PC board. The wire wrap socket pins are positioned over the VDG socket and the post goes through the hole in the board. Shine a bright light through the hole, and look through. Fit the wire-wrap pins into the VDG socket, and press gently but firmly until it secures in place. The modification is now complete.

If you read this last paragraph with consternation, it probably means that your VDG is soldered in place. The first three of four thousand Color Computers were produced this way, mine among them. Unsoldering the VDG and inserting a socket is an unmitigated pain, demanding patience and over an hour. You will need—and don't do this without—solder wick or other flux-impregnated solder-removing braid, and fine solder. *Do not* attempt to remove this circuit unless you have the proper materials ready and are willing to take the time to do it correctly; the process cannot be rushed.

Make sure the line cord is unplugged, then pull off the power supply connectors from the main board and from the power cord. Mark down their positions as you do this. Remove the screws holding the power supply in place (two are recessed into the board against the power transformer). Lift out and set aside the power supply.

Unplug the keyboard from either the computer or keyboard end, and set it aside. Remove the screws from the PC board (there are a dozen), and lift the entire board out of the case. Easy so far. The bottom of the PC board is completely covered with a metal plate, but this removes easily. It is held in place by nifty little expansion rivets. Push on each one, very gently but firmly, with the eraser end of a pencil until it works out of the board. Eventually the metal shield will fall free.

Lay the solder wick on the VDG pins on the underside of the board, and heat it and the pins with a soldering iron until the solder flows off the VDG pins into the solder braid. Check the temperature of the VDG as you work, and stop to let the IC cool off completely if it gets uncomfortable to the touch. Its pins are rated 325 degrees for 10 seconds, so don't exceed that. Eventually there will be very little solder remaining, most of it having been sucked out of the PC board holes into the braid. Test each VDG pin from the bottom with your fingernail to see if it is free from the hole; a slight push should snap it away from the hole wall. Otherwise, desolder a bit more.

When the chip is finally free, lift it out and set it aside on some aluminum foil or black protective foam. Clean the solder flux off the board with 3M belt cleaner #78-9020-0006-2. Insert a 40-pin socket and solder hairs, balls, or shorts. Push the plate back on the PC board, pressing each rivet back in place. Replace the board in the case, screw it down, reinstall the key-

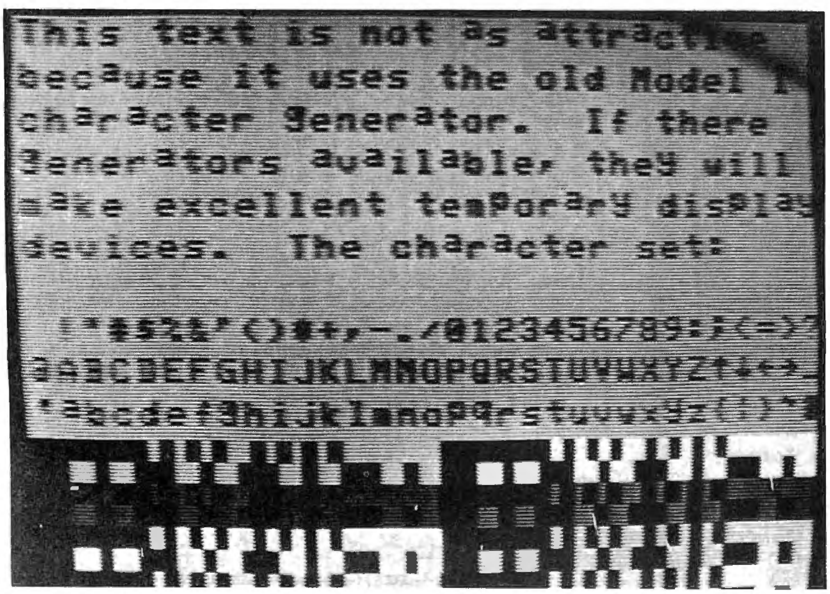


Photo 8. Older Model I character generator produces text more acceptable than the color machine's inverted system, but does not have descenders. Text shown in this photo has peculiar grammar.

board and power supply, and power up. All should be well. Breathe deeply.

Lowercase at Last

Now the test. Double check for correct insertion of all integrated circuits, jumpers, resistors, and the VDG itself. Also recheck the wire-wrap extender pins, making sure they fit into the socket exactly. The finished installation should look like Photo 7. Now restore power to the Color Computer. If you have set the switch in the new generator's position, or if you chose not to install the switch option, your sign-on message should appear. It will look mostly the same, except that the O's are no longer square and the zeros are slashed. The top row of letters will also touch the top border of the screen. Enter this program:

```
10 CLS
20 FOR X=0 TO 255
30 PRINT CHR$(X);
40 NEXT
```

The screen will clear and all your printable characters will appear, including real lowercase letters with descenders, as in Photo 8. If you have the old Model I character generator, the character set will look instead like Photo 9.

If you have installed the switched option, flip back to the original mode. The reversed uppercase characters reappear in the place of the lowercase ones. To demonstrate the difference dramatically, leave the old system switched in, and enter this program:

```
10 CLS
20 A$=INKEY$
30 PRINTA$CHR$(95)CHR$(8);
40 GOTO 20
```

This is a simpleminded text display. Begin typing, using shift-0 to produce lowercase (reversed) letters. Punctuation and uppercase are normal. Fill the screen with text until it is a blotchy mess. Now flip the switch to engage the new lowercase generator. What a good feeling.

Problems? If you have a black, jittery screen with unidentifiable characters, check to see that the VDG is seated properly in its socket, and that the external generator board is pressed firmly into the PC board socket. Also check for cold (improper) solder joints on the board, or broken, cracked or scratched PC traces. If the screen displays the wrong characters (mixing lowercase with uppercase, numbers or control characters), check the soldering on the 74LS86 or the 74LS04. If there is no display, see that the external generator board is pressed into the Color Computer socket correctly, and that the pins are not offset by one position to the right. If alphanumerics and graphics are mixed oddly, make sure that the 74LS04 and 74LS125 are inserted in the correct direction. If the screen is a garbled mess, look closely at pin 33 of the VDG, checking for solder hairs or shorts. For any other problems, recheck all of your work.

Where to Get Parts

A complete kit of parts for the video buffer/amplifier is available for $7.50 postpaid from MSB Electronics, Drawer 766, Barre, VT 05641. The lowercase modification board is available complete and ready to install for $89.95 (7×9 matrix).

Printed circuit board supplies are sold by Vector Electronic Company, 12460 Gladstone Avenue, Sylmar, CA 91342 or Kepro Circuit Systems, 630 Axminister Drive, Senton, MO 63026

Integrated circuits and transistors for these projects are commonly available; see the classifieds in computer and electronics publications. The 5×7 Model I character generator is part number AXX3027 (type SCM37530P/8046673), from Radio Shack, also known as the word processor generator; the price varies. The 7×9 character generator is sold by Jade Computer Products, 4901 West Rosecrans Ave., Hawthorne, CA 90250, part number ICP-MCM66740P, $13.25.

For removing soldering flux, 3M Belt Cleaner (part number 78-9020-0006-2) is sold by 3M, Visual Products Division, St. Paul, MN 55144. Eight-ounce samples (enough to last years) can be obtained at no charge from service representatives.

Updates

- I continue to get mail asking about the Memory Sidecar (February Applications). It is available complete or as a PC board from the Peripheral People, P.O. Box 524, Mercer Island, WA 98040. Write for information.
- Thanks to the readers who sent in new and innovative solutions to the Game of Life, using Hooper's algorithm. I will be presenting some of those intriguing solutions from time to time in the future.
- My sincere apologies to any readers who waited to receive a copy of my book, *The Custom TRS-80*. Due to a 3,000-mile hassle over editorial control, the book was inexcusably delayed.
- Corrections to high-resolution graphics board (The Detailer, July Applications):

1. In Fig. 1, Z8b is shown as a three-input NAND gate. It is a three-input AND gate, type 74LS11.
2. Pin numbers were omitted from Z32. Z32A, inputs 1-2, output 3. Z32B, inputs 4-5, output 6. Z32C, inputs 9-10, output 8. Z32D, inputs 12-13, output 11.
3. Pin numbers were omitted from Z33. Z33A, inputs 2-3, output 1. Z33B, inputs 5-6, output 4.
4. The TRS-80 SYNC connection was omitted. It is found at Z32 pin 11 in the TRS-80.
5. The 10 mF 16V capacitor connected

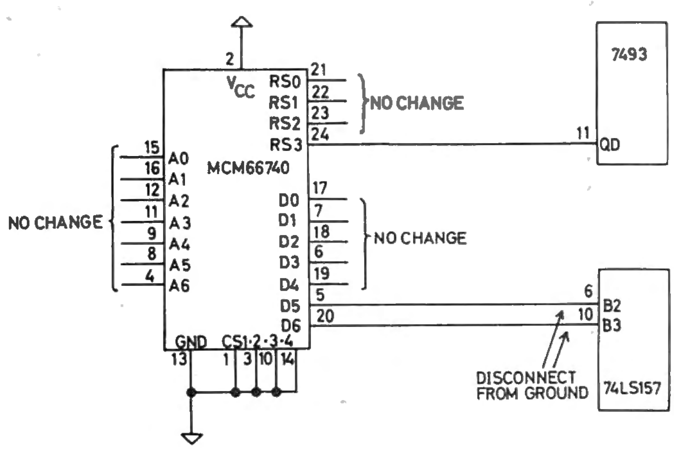


Fig. 8. Schematic changes for using the MCM66740 (or other members of the MCM66700 family) character generator, providing a 7-by-9 dot matrix.

INTERNAL	[	\	]	↑	←	@	[	\	]	↑	←
EXTERNAL	↑	↓	←	→	_	£	{	¦	}	¥	#
OLD VERSION	↑	↓	←	→	_	‘	{	¦	}	~	#
ASCII VALUE	5B	5C	5D	5E	5F	60	7B	7C	7D	7E	7F

Fig. 9. Comparison of internal VDG character set with Model I character generators. The earliest generator also differs in that the lowercase letter "a" is above the baseline.

to Z33A should be labeled C8.

6. In Fig. 2, pin numbers were omitted from Z31. Z31A, inputs 1-2, output 3. Z31B, inputs 4-5, output 6. Z31C, inputs 9-10, output 8.
7. In Fig. 4 and the parts list, the values of C6 and C7 are omitted. They are 750 pF and .022 mF, respectively.
8. The values for VCR1 and VCR2 were omitted. They are both 100K ohms.
9. IC Z30 includes the transistor immediately to its right on the schematic.
10. In Fig. 5, the 19.8 VDC line is shown connected to the case of the DIN plug. It should be connected to pin 2 of the plug.
11. The resistor in the power supply is unmarked. It should be listed as R28, 100 ohms, ½ watt.
12. The bridge rectifier value is missing. It is 2 amps, 100 volts.
13. In Figs. 1 and 2, the lines labeled MRA5 and MCA5 should read MRAS and MCAS.
14. The lines labeled M, MA, C, CA, MD, and CD were not clear. "M" means a connection to the matching line on the high-resolution board; "C" indicates a connection to the computer's edge connector. For example, CD1 means computer data line 1, and MA13 means high-resolution board address line 13.
15. In the text in the first paragraph on page 60, a V-Sync and H-Sync control is mentioned. These references are to an earlier working version of the board; none are needed in the version shown.

There has been some reader interest in a PC board for The Detailer. If you are firmly interested in a board priced at the $50 level (less parts), please drop me a post card—not a letter. PC boards are frightfully expensive to produce. I found that out when the 70 folks demanded a PC board for the Micro Front Panel project and it up and up and disappeared into the postal twilight zone.

- Speaking of the Micro Front Panel (ahem), it is now available as a complete kit of parts for $45, or PC board alone for $15, from MSB Electronics (see above).

- Finnally, some humor. I've spoken occasionally of bruised machines which nestle temporarily on my workbench. When I opened the computer Dan sent, I was reminded of illustrator B Kliban's "How To" cartoons. So, with apologies to Kliban and sympathies to Dan, I present Photos 9 and 10—before and after shots of a poor Model I. ■

Upcoming

Buffers, bubbles, and blips.

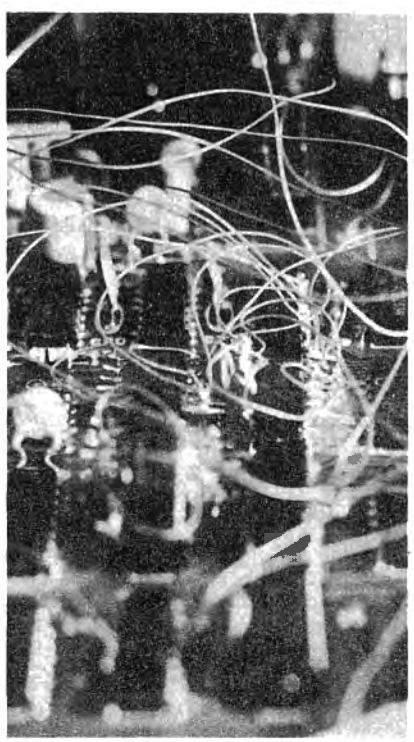


Photo 9. Wrong.

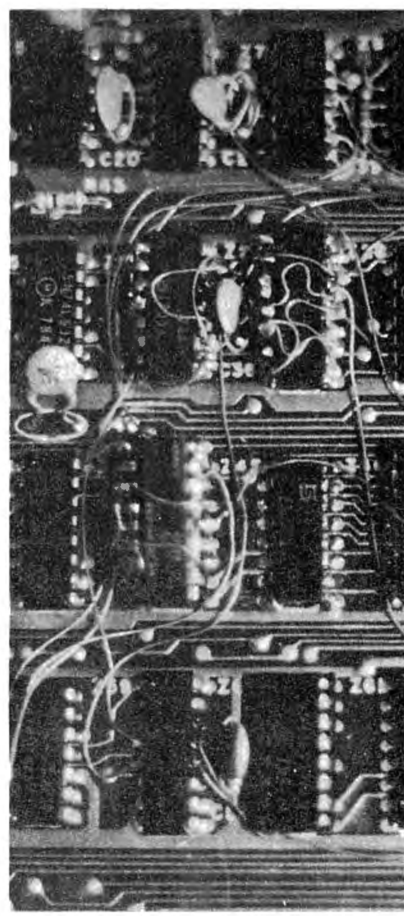
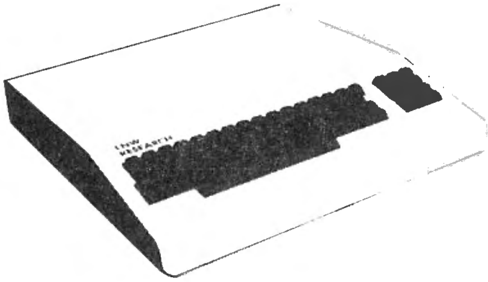


Photo 10. Right.

★★

4MHZ, DOUBLE DENSITY, COLOR & B/W GRAPHICS . . THE LNW80 COMPUTER



When you've compared the features of an LNW80 Computer, you'll quickly understand why the LNW80 is the ultimate TRS80 software compatible system. LNW RESEARCH offers the most complete microcomputer system at an outstanding low price.
We back up our product with an unconventional 6 month warranty and a 10 days full refund policy, less shipping charges.

LNW80 Computer $1,450.00
LNW80 Computer w/B&W Monitor & one 5" Drive $1,914.00
All orders must be prepaid, CA residents please include 6% sales tax.
Contact us for shipping charges

* TRS80 Product of Tandy Corporation.
** PMC Product of Personal Microcomputer, Inc.

COMPARE THE FEATURES AND PERFORMANCE

FEATURES	LNW80	PMC-80**	TRS-80* MODEL III
PROCESSOR	4.0 MHZ	1.8 MHZ	2.0 MHZ
LEVEL II BASIC INTERP.	YES	YES	LEVEL III BASIC
TRS80 MODEL 1 LEVEL II COMPATIBLE	YES	YES	NO
48K BYTES RAM	YES	YES	YES
CASSETTE BAUD RATE	500/1000	500	500/1500
FLOPPY DISK CONTROLLER	SINGLE/ DOUBLE	SINGLE	SINGLE/ DOUBLE
SERIAL RS232 PORT	YES	YES	YES
PRINTER PORT	YES	YES	YES
REAL TIME CLOCK	YES	YES	YES
24 X 80 CHARACTERS	YES	NO	NO
VIDEO MONITOR	YES	YES	YES
UPPER AND LOWER CASE	YES	OPTIONAL	YES
REVERSE VIDEO	YES	NO	NO
KEYBOARD	63 KEY	53 KEY	53 KEY
NUMERIC KEY PAD	YES	NO	YES
B/W GRAPHICS, 128 X 48	YES	YES	YES
HI-RESOLUTION B/W GRAPHICS, 480 X 192	YES	NO	NO
HI-RESOLUTION COLOR GRAPHICS (NTSC), 128 X 192 IN 8 COLORS	YES	NO	NO
HI-RESOLUTION COLOR GRAPHICS (RGB), 384 X 192 IN 8 COLORS	OPTIONAL	NO	NO
WARRANTY	6 MONTHS	90 DAYS	90 DAYS
TOTAL SYSTEM PRICE	$1,914.00	$1,840.00	$2,187.00
LESS MONITOR AND DISK DRIVE	$1,450.00	$1,375.00	---

LNW80

- BARE PRINTED CIRCUIT BOARD & MANUAL $89.95

The LNW80 - A high-speed color computer totally compatible with the TRS-80*. The LNW80 gives you the edge in satisfying your computation needs in business, scientific and personal computation. With performance of 4 MHz, Z80A CPU, you'll achieve performance of over twice the processing speed of a TRS-80*. This means you'll get the performance that is comparable to the most expensive microcomputer with the compatibility to the world's most popular computer (TRS-80*) resulting in the widest software base.

FEATURES:
- TRS-80 Model 1 Level II Software Compatible
- High Resolution Graphics
  - RGB Output - 384 x 192 in 8 Colors
  - NTSC Video or RF MOD - 128 x 192 in 8 Colors
  - Black and White - 480 x 192
- 4 MHz CPU
- 500/1000 Baud Cassette
- Upper and Lower Case
- 16K Bytes RAM, 12K Bytes ROM
- Solder Masked and Silkscreened

LNW SYSTEM EXPANSION

- BARE PRINTED CIRCUIT BOARD AND MANUAL $69.95
WITH GOLD CONNECTORS $84.95

The System Expansion will allow you to expand your LNW80, TRS-80*, or PMC-80** to a complete computer system that is still totally software compatible with the TRS-80* Model 1 Level II.

FEATURES:
- 32K Bytes Memory
- 5" Floppy Controller
- Serial RS232 20ma I/O
- Parallel Printer
- Real Time Clock
- Screen Printer Bus
- On Board Power Supply
- Solder Masked and Silkscreened

LNDoubler & DOS PLUS 3.3D

- Assembled and Tested W/DOS PLUS 3.3D. $175.00

Double-density disk storage for the LNW Research's "System Expansion" or the Tandy's "Expansion Interface". The LNDoubler™ is totally software compatible with any double density software generated for the Percom's Doubler***. The LNDoubler™ provides the following outstanding features.

- Store up to 350K bytes on a single 5" disk
- Single and double density data separation
- Precision write precompensation circuit
- Software switch between single and double density
- Easy plug in installation requiring no etch cuts, jumpers or soldering
- 35, 40, 77, 80 track 5" disk operation
- 120 day parts and labor Warranty

*** Doubler is a product of Percom Data Company, Inc.

DOS PLUS 3.3D

Micro Systems software's double density disk operating system. This operating system contains all the outstanding features of a well developed DOS, with ease in useability.

KEYBOARD

LNW80 KEYBOARD KIT $84.95

The Keyboard Kit contains a 63 key plus a 10 key, P.C. board, and remaining components.

CASE

LNW80 CASE $84.95

The streamline design of this metal case will house the LNW80, LNW System Expansion, LNW80 Keyboard, power supply and fan, LNDoubler™, or LNW Data Separator. This kit includes all the hardware to mount all of the above. Add $12.00 for shipping

PARTS AVAILABLE FROM LNW RESEARCH
- 4116 - 200ns RAM
  - 6 chip set $26.00
  - 8 chip set $33.50
  - 16 chip set $64.00
  - 24 chip set $94.00
  - 32 chip set $124.00
- LNW80 "Start up parts set" LNW80-1 . . . . $82.00
- LNW80 "Video parts set" LNW80-2 . . . . $31.00
- LNW80 Transformer LNW80-3 . . . . $18.00
- LNW80 Keyboard cable LNW80-4 . . . . $16.00
- 40 Pin computer to expansion cable $15.00
- System Expansion Transformer $19.00
- Floppy Controller (FD1771) and UART (TR1602) . . . $30.00

LNW RESEARCH
C O R P O R A T I O N

2620 WALNUT
TUSTIN CA. 92680 ~571

ORDERS & INFO. NO. 714-544-5744
SERVICE NO. 714-641-8850

VISA MASTER CHARGE ACCEPTED

UNLESS NOTED ADD $3 FOR SHIPPING

●●

NEW PRODUCTS

edited by Janet Fiderio

Business Feature—The CR-180 Cash Register/POS Expansion System

The CR-180 cash register/POS expansion system

Cash Register Expansion System

The CR-180, a cash register/POS expansion system for the TRS-80 Model III has been announced by FutureSoft and ICR. The system includes an electronic cash drawer and receipt printer which plug directly into the TRS-80 Model III. (The software will support Radio Shack or other printers.) Audio has been added for keystroke confirmation.

The CR-180 software provides the operator with continuous instructions at the point of sale and allows management to rapidly change promotional messages printed on customer receipts and displayed for the operator.

CR-180 stores transactions for up to 100 employees, saves eight methods of payments and provides inventory control and complete reporting. The system produces reports including daily sales and cash reports by employee and by transaction type, inventory usage and gross profit computation. Price and shelf labels are also printed.

Software for the Cash Register Expansion System was developed by FutureSoft and is available in foreign language versions. Prices range from $900 to $1900. Contact ICR/FutureSoft Southern Region, P.O. Box 1446, Orange Park, FL 32073, for information.

Reader Service ✓ 161

Word Processor For the Color Computer

C. C. Writer is a line-oriented word processor designed for the TRS-80 Color Computer with Extended Color Basic and 16K. The program may be initialized in a 32K format to take advantage of the expansion products available. C. C. Writer is written in Extended Basic.

The menu allows full control of margins, line spacing, justification, paragraph indentation and page pause for single sheet feeding. Commands embedded in the text control include line skipping and forcing, new paragraphs, page forcing, centered texts, tabs and indentation of both margins. ASCII control codes may be inserted in the text to control "smart" printers or create print graphics.

Editing abilities allow insertion, deletion, or modification and blocks of text to be moved on command.

The package is priced at $30 from Transformation Technologies, 194 Lockwood Lane, Bloomingdale, IL 60108.

Reader Service ✓ 325

Software is for Business

Occupational Software is offering two new programs of interest to the business community.

The business Accounting Control System for Models I and III features accounts receivable, billing, inventory control, accounts payable, payroll, job costing and general ledger programs. This comprehensive software system is priced at $4500.

The Restaurant Accounting Control System features accounts payable, payroll, sales analysis, labor planning, inventory control and general ledger programs. It runs on the Model II. The system, sold by module, is priced from $1000–$7500.

Both systems are available from Occupational Software Co. Inc., 22311 Ventura Blvd., Suite 123, Woodland Hills, CA 91364.

Reader Service ✓ 336

Add CP/M to Mod III

The Freedom 3 is a modification that adds CP/M capability and more to the Model III. It includes added memory, memory restructuring and a battery-backed date calendar clock.

The memory addition replaces the ROM in 2K segments up to the entire 14K ROM size. The alternate memory can be either EPROM/ROM or RAM. This system is compatible with TRSDOS and other operating systems, will operate on intermixed drives and may be ordered on either 40 or 80-track disks.

Freedom 3A provides address restructuring and the operating system for CP/M for $199. Freedom 3B is a 40K user space system also, but may be user upgraded by adding ICs to a full RAM system, with battery back-up of either or both the clock and memory for $340. Freedom 3C provides the full RAM, battery backed-up clock and a CP/M-type operating system that provides 54K of user space for $490.

For more information contact Field Engineering Consultants, Ltd., Box 2368, Woburn, MA 01888, (617) 944-5329.

Reader Service ✓ 326

Ecology Simulations I

Use of a new computer simulation game, Sterl, has computerists trying to eradicate the Mediterranean fruit fly. This simulation permits the use of two methods of pest control—pesticides and the use of sterile males, either alone or in combination.

Sterl is one of four simulations in the package Ecology Simulations I. Other programs in the package include: Buffalo, which simulates the Buffalo population of 1850; Tag, a simulation of the tagging and recovery method of measuring animal populations; and Pop, a simulation of animal population management.

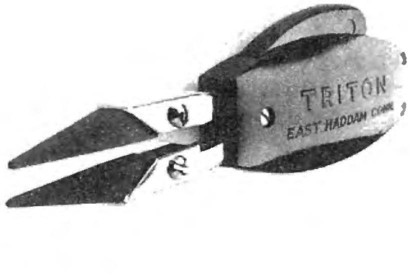
Available for Model I, Level II, it costs $24.95 from Creative Computing, 39 E. Hanover Ave., Morris Plains, NJ 07950, (201) 540-0445.

Reader Service ✓ 168

"HOT LIPS"— A Soldering Tool

Hot Lips is a hand held electrical resistance tool for intricate soldering operations. Its unique plier-like design enables the user to produce perfect soldered connections in tight places.

This Triton Manufacturing tool is six volts safe and is furnished with a remote transformer power supply. It's instant-heat on, instant-cool off means minimum power use. Hot Lips is sold for $50.20 from



Triton's "Hot Lips" soldering tool.

The TRS-80 Pocket Computer Printer and Interface.

Triton Manufacturing Company, P.O. Box 263, East Haddem, CT 06423.

Reader Service ✓ 178

Printer/Cassette Interface For The Pocket Computer

Radio Shack now offers a printer and cassette interface for its Pocket Computer. This product comes with batteries, an ac adapter/charger, cassette recorder connecting cable, replaceable printer ribbon cartridge, additional rolls of paper and an owner's manual.

The cassette interface allows programs and data to be loaded, saved and recalled. A remote switch allows manual control of the tape recorder. Connections between the printer/cassette interface and the computer are made directly, without cables.

This printer/cassette interface is priced at $149.95 from Radio Shack, 1800 One Tandy Center, Fort Worth, TX 76102.

Reader Service ✓ 337

Educational Games

Jabbertalky is a programmable word game for one or more players. It includes two game features, a utility program and eight levels of skill.

The first, Alphagrammar, an anagram game, challenges players to unscramble an entire grammatically correct sentence. The second game, Cryptogrammar, is a code-breaking game in which players are given a computer-generated sentence where each letter of the alphabet is substituted for another letter. You must try to determine the correct arrangement before your time runs out. The special utility program, Jabbergrammar, stores lists of words for each part of speech.

Jabbertalky is available on cassette for the Model I with 16K, or on disk for 32K. Contact Automated Simulations, Inc., Box 4247, Mountain View, CA 94040, for more information. The retail price is $29.95.

Reader Service ✓ 345

Magikube, the Computerist's Version of Rubik's Cube

Magikube is a game available on cassette for the Radio Shack Color Computer.

Based on the Rubik's cube, you have the option of scrambling the cube yourself or letting the computer do it. The objective is to restore the cube back to its original state. Magikube also has a tape save feature allowing you to save your cube and continue at a later time.

Magikube costs $19.95 and is available from Computerware, Box 668, 1472 Encinitas Blvd., Encinitas, CA 92024, (714) 436-3512.

Reader Service ✓ 162

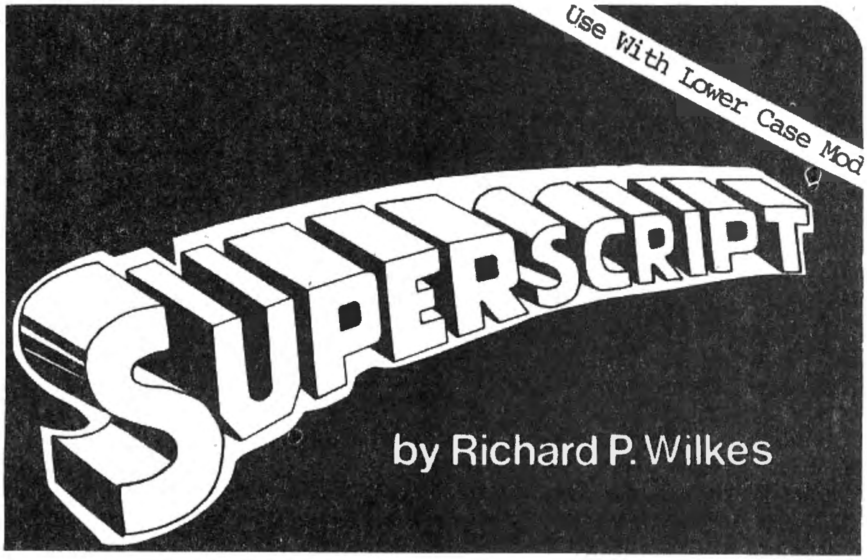
Program Evaluates Solar Systems

Sunheat1 is used to put together and evaluate a variety of solar hot water systems. It will determine the effectiveness of various types of solar collectors, heat exchangers, preheat tank sizes and hot water tank sizes for your needs.

Once the system has been configured, the program calculates the amount of useful solar energy that can be collected each month and the percentage of hot water needs that can be supplied by solar means on a monthly and yearly basis.

Sunheat1 is written in Basic and requires a Model I Level II with 16K; the cost is $16. For more information contact Solartek, Box 298, Guilderland, NY 12084.

Reader Service ✓ 344



Superscript

Superscript For the Model III

Acorn Software has developed a Model III version of Superscript for the TRS-80 with 32K of RAM, expansion interface, disk drives and lowercase modification.

The product features: directory and kill-files from within Superscript; ability to pause while printing and insert text; serial drivers which use the ETX/ACK protocol for 1200 baud communications; special drivers for various printers; ability to superscript, subscript, underline, boldface, select 10/12 pitch, or use brackets, braces, carets, and more.

Superscript costs $50 and is available from Acorn Software Products, Inc., 634 North Carolina Ave., S.E., Washington, D.C. 20003, (202) 544-4259.

Reader Service ✓ 350

The Newest Versamodem

Bizcomp Corporation's newest Versamodem, the Model 1080, provides users needing only a minimum feature modem with a direct-connect alternative to older acoustic couplers.

The modem can be used with personal computers at rates of 300 baud and below using Bell standard 103 protocol. The Versamodem is FCC registered and supplied with a modular plug for direct connection to the telephone network.

The price of the modem is $119 and is available from the Bizcomp Corporation, Box 7498, Menlo Park, CA 94025, (415) 966-1545.

Reader Service ✓ 335

Precision Template

ABS Suppliers is offering a new product that will increase your disk storage capacity 100 percent. The ABS-Doubler is a precision template that permits the conversion of single-sided disks to double-sided disks.

The Doubler is sold by ABS Suppliers, Box 8297, Ann Arbor, MI 48107, (313) 971-1404, for $3.50.

Reader Service ✓ 171

Disk Speed Measurement For the TRS-80

RPM measures the rotational speed and variation of disk drives on the TRS-80 Models I and III. RPM makes it very easy for the user to see the speed, percent error, and history of speed variation on any mini-floppy connected to the TRS-80.

Readouts are given in real-time, and are shown in numbers, percents, words and graphs. With one keystroke any drive may be selected. RPM automatically shows all speed ranges and can recover from severe errors without requiring a system reset.

The product is available on disk for $24.95 from Prosoft, Box 839, North Hollywood, CA 91603, (213) 764-3131.

Reader Service ✓ 327

Graphics Editor and Programmer

The Graphics Editor and Programmer is a hybrid program that is part text editor, screen drawing utility, and Basic program creator. This product requires no knowledge of basic programming skills.

GEAP contains a full command library which allows you to create screen images and formats. The software will save your drawing by constructing a Basic program. The program will stand alone, does not require any special drivers or operating systems and will recreate the screen image you have drawn.

The Graphics Editor and Programmer is available for the Model I priced at $16.99 from J F Consulting, 74355 Buttonwood, Palm Desert, CA 92260.

Reader Service ✓ 331

Eliminate Tedious Commands

Autofile allows the Basic programmer to automate the task of moving data elements to and from a direct file.

Autofile has eliminated the Field statement thereby eliminating guessing as to where the Fielded variable is. In addition the ASC and CHR$ function references will be performed automatically.

The price of this product is $75. For more information contact Snappware, 3719 Mantell, Cincinnati, OH 45236, 1-800-543-4628.

Reader Service ✓ 341

District Courseware Authoring System

CASE, the District Courseware Authoring System, consists of 14 programs on two disks and two cassettes. It is of interest to school districts.

CASE provides for the centralized development of customized courseware for controlled distribution throughout an entire school system. This system does not require a knowledge of programming; includes automatic lesson to lesson branching, performance records on disk by date, time, student ID, lesson and question, and more.

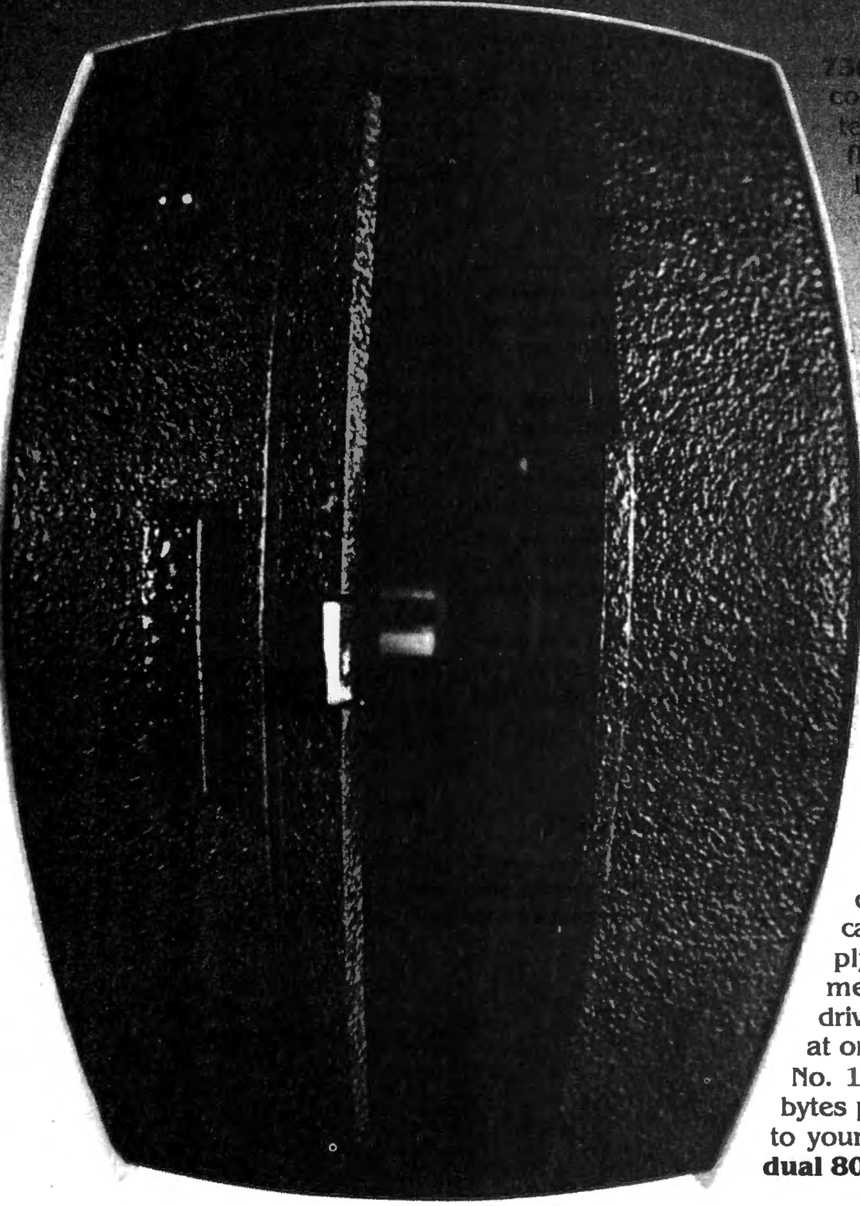
For information contact the MicroGnome Division of Fireside Computing, Inc., 5843 Montgomery Road, Elkridge, MD 21228, (301) 796-4165. The cost of this courseware is $395.

Reader Service ✓ 160

Software Development System

The SDS80C is a complete 6809 editor, assembler and monitor package contained in one Color Computer program pack.

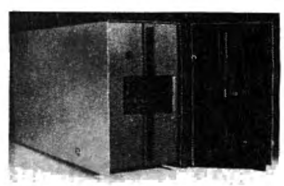
APPARAT OFFERS
More bytes per buck!



750 K/bytes of storage. Apparat has combined its Newdos/80 operating system and a dual-sided 80 track mini-floppy drive to give you up to 735,440 bytes of storage in a single volume. Newdos/80 version 2.0 expands the capability of double density drives, so you'll have greater applications for your TRS-80® model I and III.

Drives plug directly into an expansion interface (requires installation of a double density controller) or the model III disk bus with our single volume cable so you can now have over 2 megabytes of storage on-line with standard mini-floppy diskettes. Each drive has up to 573 free grans, for a total of 1,719, on a maximum of three 80 track drives, which can be added to a TRS-80 model I. Model III's can have up to 4 dual 80's on-line (almost 3 megabytes).

These drives can "read" standard 35 or 40 track diskettes using Newdos/80 version 2.0 which will allow skipping every other track.



Drives come complete with case, power supply and documentation. The drives are priced at only **$515** (Cat. No. 1-705, specify I or III). Now, at 1424 bytes per buck, it just might be the answer to your storage problems. **Special – Two dual 80 track drives only $999.**

Apparat, Inc.

4401 So. Tamarac Parkway, Denver, CO 80237 (303) 741-1778

VISA master charge

264

"ON GOING SUPPORT FOR MICROCOMPUTERS"

The screen-oriented editor features finds, changes, moves, copies and more. All keys have auto repeat, and no line numbers are required so the full width of the screen can be used to generate commented code.

The assembler features a complete 6809 instruction set, a complete 6800 set supported for cross-assembly, conditional assembly, local labels, assembly to cassette tape or memory, listing to screen or printer and mnemonic error codes instead of numbers.

The ABug monitor is a compact version of CBug. It features examine or change of memory or registers, cassette load and save, break-points and more.

SDS80C is priced at $89.95 from The Micro Works, Box 1110, Del Mar, CA 92014, (714) 942-2400.

Reader Service ✓ 332

Software for Physicians

Medical Office Management is a software applications package which provides medical offices with a system which maintains general patient information files, office appointment schedules, maintains and prints the daily transactions log, prints procedure-by-procedure management reports, prepares and prints private patient bills, and prepares patient insurance claim forms.

The system includes an online Help function in addition to extensive documentation. This software requires either two or three disk drives, a 130-column printer, and a 48K Model I Level II or Model III (Basic III) with TRSDOS. The price is $499.95. For more information contact Charles Mann and Associates, Micro Software Division, 55722 Santa Fe Trail, Yucca Valley, CA 92284, (714) 365-9718.

Reader Service ✓ 347

Sales Tax Calculator

For small businesses reporting state sales taxes quarterly, Manhattan Software has released the TRS-80 Sales Tax Calculator.

The program automatically separates entries coded taxable, calculates and displays the tax on screen, and adds sales categories and tax due. The user may optionally enter tax actually charged on each sale. The program will hold up to 500 entries in 16K, and 1500 in 32K.

The cassette is priced at $14.95 for Models I and III. For more information contact Manhattan Software, Box 1063, Woodland Hills, CA 91365, (213) 704-8495.

Reader Service ✓ 330

Create Formatted Screen With a Single Command

Automap simplifies the programmer's task of communicating and displaying information with the user-operator. Simple send and receive statement commands will communicate and display the information you want.

Automap is available for Models II and III for $100. For more information contact Snappware, 3719 Mantell, Cincinnati, OH 45236, 1-800-543-4628.

Reader Service ✓ 342

For your Typewriter

Dynatyper/Dynastar is a new typewriter system designed to generate printouts directly from computer output through any electric typewriter.

The interface fits directly over the keyboard and requires no modification to your typewriter. It is RS-232 serial, Centronics parallel, IEEE 488(GPIB) and 15-pin current loop compatible. Features include a 3.5K RAM buffer, 37.5 to 9600 switch-selectable baud, top-of-form option, 80 or 132 option, and downloading capability to four special character sets.

It is available for Models I and III for $699 from Rochester Data, 3000 Winton Road, South, Rochester, NY 14623, (716) 244-7804.

Reader Service ✓ 164

Model II Gets SoftCare

SoftCare, a medical billing system, previously used only on minicomputers and larger Z80 systems, is now available for the Model II.

It prepares patient bills and insurance claims, maintains accounts receivable and transaction details and is user-friendly.

It is available for $1995 from Professional Business Software, 119 Fremont St., San Francisco, CA 94105.

Reader Service ✓ 340

Energy Saving Analysis

The Home Energy Cost Saving Analysis program for the Models I and III is designed to evaluate the cost/benefit of heat saving home improvements.

To accomplish this the program models the current surface area of your home using the "R" factor of each material used to construct your home. The computations for 29 building materials and seven types of heating fuels are used to compute the cost of heating and the fuel saving payback period for the improvement being considered.

This program is available for $9.95, from Demi-Software, 6 Lee Road, Medfield, MA 02052, (617) 359-4502.

Reader Service ✓ 167

Timing Program for Stock and Commodity Traders

Stock Timer is a program that calculates and plots up to three uniform or exponential moving averages of the user's choice on a common set of price or volume data. The interaction of these moving

The Dynatyper

SAVE/on Software for TRS-80®

NEWDOS/80 Version 2.0

The most sophisticated DOS ever produced for the TRS-80® Models I and III. It provides the user with "MAINFRAME" power on a "MICRO".

Some Features available are:

- Jobstream Control Language
- Mod I/Mod III Diskette interchangability
- Double Density Support on Model I
- Pagenation of BASIC listings on the screen
- Basic program single stepping
- Dynamic variable manipulation
- Multiple array sorts
- Complete technical support provided

All this plus much more for only

$149.00

MICRO ACCOUNTING SYSTEM

Accounting by the book for the non-accountant. Menu driven for ease of operation. Single or Double entry bookeeping, complete system allows INTERACTION between General Ledger, Accounts Receivable, and Accounts Payable or each sub-system can be ran stand alone. Maintenance programs and detailed reports support numerous applications. User friendly support provided by the authors. Write for a complete description and sample printouts today!

Complete System	**$489.00**
General Ledger	**$159.00**
Accounts Receivable	**$159.00**
Accounts Payable	**$159.00**
Check Register	**$ 79.00**

CHEXTEXT®

Let your TRS-80® do the proofreading on your SCRIPSIT® text files!!

Features of this program include:

- Complete dictionary maintenace including the addition and deletion of words.
- Menu driven for ease of operation.
- FREE expanded dictionaries available, depending on your drive storage capabilities.

NEW LOWER PRICE $59.95

THE MICRO CLINIC

by Dave Stambaugh

"An ounce of prevention is . . ." Routine system checkout will help prevent that dreaded loss of data. a thorough system checkout includes both Memory and Disk diagnostics. The Micro Clinic offers the most exhaustive set of routines known of for the Model I or III. Don't take chances with your data, a routine system checkout is your best friend.

". . . worth a pound of cure."

Model I version	**$24.95**
Model III version	**$29.95**

MUSIC TO YOUR EARS!

For your Model I disk or tape

Orchestra 80	$ 79.95
Orchestra 85	$129.95
Orchestra 90	Available Soon!

Assorted Items of interest

Lazy Writer (Mod I disk)	**$125.00**
Lazy Writer (Mod III disk)	**$175.00**
Meal Master (Mod I & III disk)	**$ 24.95**
Scarfman (disk)	**$ 19.95**
(tape)	**$ 15.95**
Flextext I (mod I & III disk)	**$ 29.95**
Flextext II (mod II disk)	**$ 29.00**
Big five games (mod I & III) disk	**$ 19.95**
tape	**$ 15.95**
MTC CALCS IV (mod I disk)	**$ 39.95**

ATTENTION: Our marketing department is constantly searching for high quality professionally written software.

ATTENTION DEALERS: Write for our dealer packet today.

PROFESSIONAL FOOTBALL PREDICTIONS

$29.95

- Makes predictions for all 14 NFL games each week.
- Game predictions are incredibly accurate.
- Program's data base is updated weekly from your local newspaper's game statistics.
- Keeps complete record of all final scores and standings.
- Re-useable year after year.
- Very sophisticated program, yet easy to operate.

FOR YOUR TRS-80® Model I or III

*NOTE: Start saving your game results at the time of your order, APPARAT will provide the data base for the previous weeks games on the disk.

MISCELLANEOUS SUPPLIES

DISKETTES

Double density-soft sectored-replacement guaranteed-Hard Hole protected. (5¼" only)

Verbatim Datalife 5¼" 40 track	$24.95
Apparat's No Name 5¼" 40 track	$21.95
Verbatim Datalife 8" model II	$39.95

PAPER

9½"x11" blank white, tractor feed paper, full box	$24.95
9½"x11" blank white, tractor feed paper, half box	$14.95
14⅞"x11" green bar, tractor feed paper, full box	$34.95
3½"x15/16" tractor feed mailing labels	$19.95

OTHER

5¼" plastic library case	$ 3.95
8" plastic library case	$ 5.95
5¼" Flip-sort	$29.95
8" Flip-sort	$39.95
16K memory kits	$19.95

UNPRECEDENTED 90% DISCOUNT SALE

On MODEL II Commercial Grade Business Software

(LIMITED Quantities Available)

TITLE	LIST PRICE	SALE PRICE
General Ledger	$300.00	$30.00
Accounts Receivable	$300.00	$30.00
Accounts Payable	$300.00	$30.00
Inventory	$300.00	$30.00
Order Entry Processing	$300.00	$30.00
Sales Analysis	$300.00	$30.00
Word Processing	$400.00	$40.00

*All systems include extensive situation oriented documentation, but is supplied on diskette only.

*Above software sold as is and require a minimum of 64K and two drives.

(Sorry, but at these rediculously low prices our usual software support will not apply.)

ADD $3.00 for postage and handling for these items.

VISA

Apparat, Inc. ✓264

"On-going Support for Microcomputers"

4401 S. Tamarac Pkwy. • Denver, CO 80237 • (303) 741-1778 • (800) 525-7674

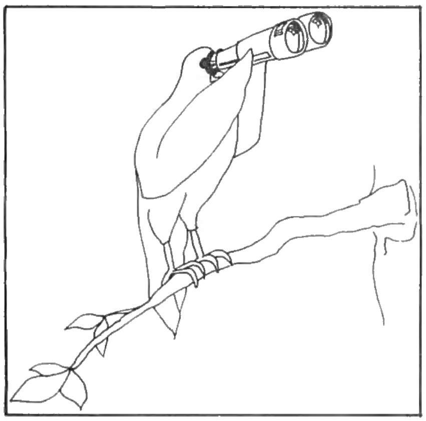
Scripsit & TRS-80 are a registered trademark of Tandy Corporation.

Freight F.O.B. Denver-call for shipping charges. Foreign Orders shipped Air Freight averages produces the customary buy and sell signals.

Stock Timer also includes a data base management system for the necessary price and volume information.

It is available for Models I, II, and III in tape or disk versions with 16K or more of memory. For additional information contact Bayesian Investment Services, 757 Santa Rosita, Solana Beach, CA 92075. The tape version is priced at $26.95, the disk at $31.95.

Reader Service ✓ 343



For the Bird-watcher

Life List is a program for the bird-watcher. It is a one-disk record of bird sightings which holds bird name, sighting date, location and notes for 450 sightings.

Data in the program's search files can be edited to reflect species name changes or later sightings. The program will display individual records on-screen or an index of all sightings. A special search feature allows the user to find all birds seen on any date, in any month or in any year and display them together on the screen. Similar searches will find all sightings at a particular location or all birds of one name group.

Life List is available on disk for the Model I for $24.95 from Manhattan Software, Box 1063, Woodland Hills, CA 91365, (213) 704-8495.

Reader Service ✓ 334

Mailing List System

Mail-M3 is an advanced mailing list system for the Model III and TRSDOS. This program features multiple labels (up to four) across a page, form input, a report writer, fast multi-sort keys, duplicate label checking, random access, search and more.

Mail-M3 is not "on-memory" so the entire database does not have to be sorted each time you add records. One disk can store about 500 records of 127 bytes each, or more if you have additional drives.

Mail-M3 comes in two versions. The enhanced 48K version is priced at $79, while the 32K version is $59. Both are available from the Micro Architect, Inc., 96 Dothan St., Arlington, MA (617) 643-4713.

Reader Service ✓ 173

For Manufacturing Cost Estimating

Two software packages designed for manufacturing, engineering, consulting and analytical applications are available from Centec Process Systems, Inc.

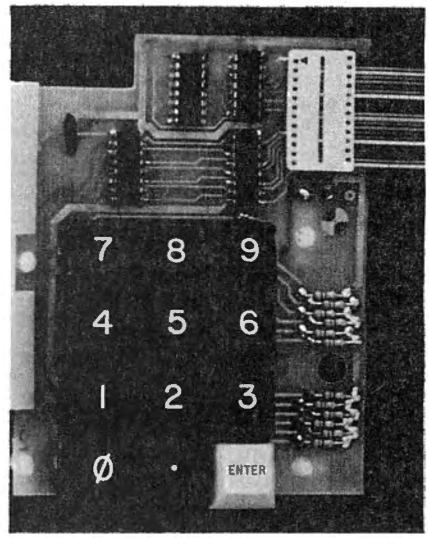
The MCE package allows the user to estimate detailed manufacturing costs based on individual specifications. Information is organized under both fixed and variable costs including categories such as labor, utilities and raw materials.

The DCF package is designed for corporate managers, financial advisors, certified public accountants, tax consultants, and real estate and building analysts. Applications include detailed cash flow, rate of return and payback, escalation of income and individual costs, and more. Separate interest rates can be used for borrowed and working capital.

The MCE system is priced at $75, the DCF at $125. Both systems are priced at $149 from Centec Process Systems, Inc., 11260 Roger Bacon Drive, Reston, VA 22090, (703) 471-5999.

Reader Service ✓ 165

Software from Centec Process Systems



The Control Key Hardware

Control Keys For the TRS-80

The Control Key system is a hardware and software combination that allows control of the Model I 32K, or 48K disk system with single keystrokes.

The hardware converts the numeric pad on the TRS-80 into a set of 12 programmable special function keys that can be accessed by either machine language or Basic programs. When used in conjunction with the four currently available Control Key programs, each of the numeric keys perform specific functions such as execution of the most frequently used DOS functions, rapid input and debugging of Basic programs, and more.

The cost of the system assembled is $150. Kits are also available. Contact Clockwork Software, Box 704, Colorado Springs, CO 80901, for additional information.

Reader Service ✓ 346

New Software Catalogue

Charles Mann & Assoc. has released a new software catalogue of office management software for several microcomputers including the TRS-80 Models I and III.

The catalogue covers the field of small business office management applications from simple desk management and appointment systems to detailed job cost accounting applications.

It is available from Charles Mann & Assoc., Micro Software Division, 55722 Santa Fe Trail, Yucca Valley, CA 92284.

Reader Service ✓ 339

Master Disk Index

The Master Disk Index (MDI), creates and maintains a master file of disk directories updated by automatically reading each disk's directory.

MDI records may be selectively displayed, printed, or deleted by either string matching or record number. The entire file may be sorted by a fast machine code sort.

It is available on cassette with minimum machine requirements of 32K, 1-disk, TRSDOS, NEWDOS, or NEWDOS80 for $16. For more information contact Custom Software, Box 3241, Poughkeepsie, NY 12603.

Reader Service ✔ 170

CBug Monitor

The CBug monitor, offered by the Micro Works, has 19 commands and is relocatable and reentrant.

The monitor allows you to: examine or change memory using a formatted hex display, save areas of memory to cassette in binary, download/upload data or programs to a host system, move the video display page through RAM and send or receive RS-232 at up to 9600 baud.

It will also investigate and activate features of your computer, such as high-resolution graphics and machine-language music, or use the computer as an intelligent peripheral for another computer.

Available on cassette, the monitor cassette price is $29.95. For additional information contact The Micro Works, Box 1110, Del Mar, CA 92014, (714) 942-2400.

Reader Service ✔ 333

T-ROM II Utility

Microbyte Research and Development Corporation currently is marketing the T-ROM II.

The T-ROM II features include: blinking block cursor, keyboard debounce, auto repeat on any key, audible key-beep, autoline feed, lowercase driver, video to printer echo, shiftlock, screenprint and the dual-case typewriter function. Graphics functions are retained in the lowercase mode, and normal Level II Basic commands are unaffected.

The cassette version sells for $15.95, while the disk version is $19.95. Contact Microbyte Research and Development, 105 Rackley Drive, Greenville, NC 27834.

Reader Service ✔ 349

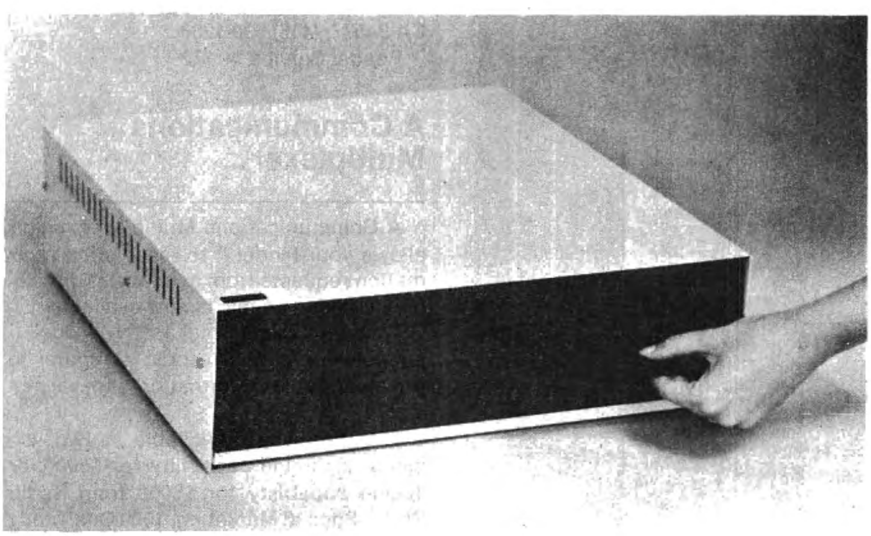
New Disk Subsystem

Matchless Systems now introduces the MS-800C disk subsystem compatible with the TRS-80 Model II and other microcomputers.

The subsystem includes case, power supply, fan, cables and a choice of four different drive configurations: one single-sided drive for $1050, two single-sided drives for $1595, one double-sided drive for $1395, and two double-sided drives for $2095. Single and double-density controllers and software are available at an additional cost.

For more information contact Matchless Systems, 18444 South Broadway, Gardena, CA 90248, (213) 327-1010.

Reader Service ✔ 328



Matchless Systems MS-800C

For the Software Author

Programmer is a new monthly newsletter for those interested in the software market. Scheduled for debut this month, it will cover what kind of software is selling and who is selling it, with market updates of opportunities in the software field.

Programmer's monthly columns will bring together professional writers and published programmers who will be talking about the principles of good programming and the techniques that make a program saleable. The editors of *Programmer* will be examining contracts from software producers, surveying royalties and offering the programmer advice on good business practice.

A six month subscription is $13. For additional information contact *Programmer* Box 3210, Manchester, NH 03105.

Reader Service ✔ 179

Lunar Lander— Based on Physics

Lunar Lander is a game based solidly on physics. The computer displays the lunar landscape, your landing craft and navigational information—all of which are changing constantly. Lunar Lander teaches the concepts of acceleration and velocity.

Minimum requirements are 16K Level II Basic. A cassette is available for $10 from Custom Software, Box 3241, Raymond Ave., Poughkeepsie, NY 12603

Reader Service ✔ 169

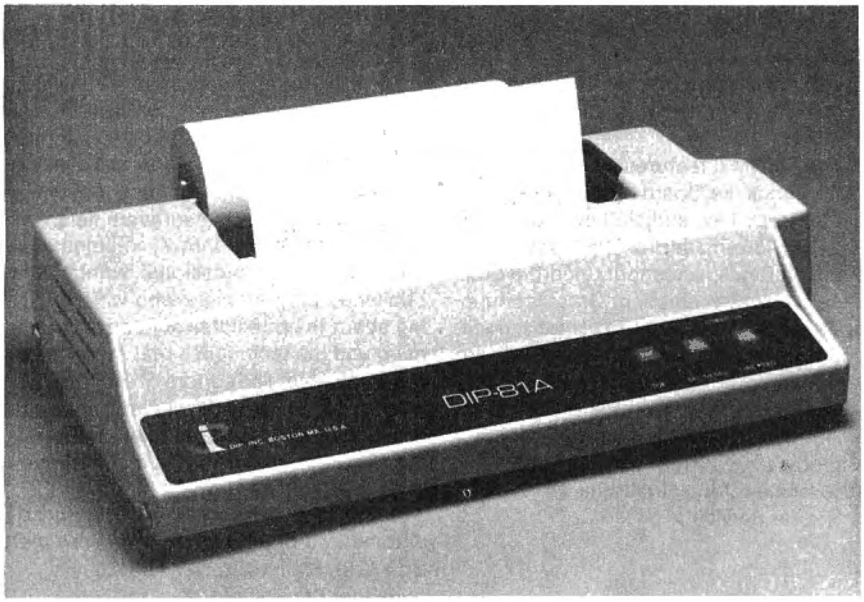
Scriplus Modifies Scripsit

Scriplus is a modification to Scripsit which enables you to take advantage of the special functions, features and print formats of your printer while your document is being printed, all in accordance with the limitations and capabilities of the printer.

Scriplus will not crash programs held in high memory and provides the following features: It will alphabetically list a disk directory from within Scriplus; optionally select line feed after carriage return; support custom printer drivers; and is compatible with VTOS, LDOS, TRSDOS, DBLDOS and NEWDOS.

For more detailed information on Scriplus features contact Quality Software Distributors, 11500 Stemmens Exp., Suite 104, Dallas, TX 75229. Scriplus is priced at $24.95.

Reader Service ✔ 329



The DIP-81A

New Data Impact Printer

DIP, Inc., is now marketing a new low-cost data-impact printer, the DIP-81A.

This dot-matrix printer features a full 96-character ASCII set and is capable of printing at both 40 and 80 characters per line on 8½-inch paper. Operator control includes power, select/deselect, line feed, top of form, and self test. It has the ability to accept data as it is printing and is suited for such applications as CRT dump for printouts and more.

For additional information contact DIP, Inc., 745 Atlantic Ave., Boston, MA 02111, (617) 482-4214. The price of the DIP-81A is $499.

Reader Service ✓ 163

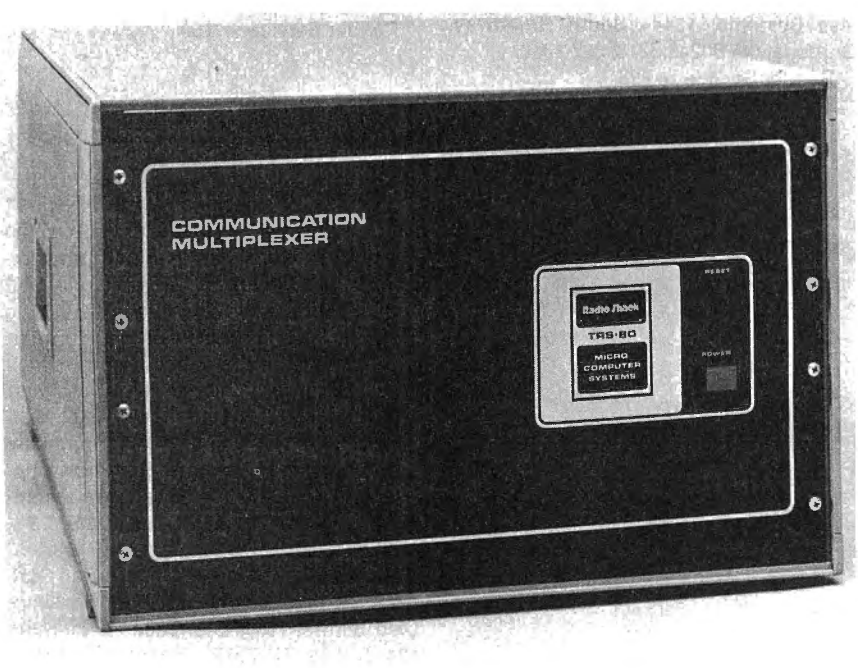
For All Gamesmen

The Voyage of the Valkyrie, a game for the Models I and III, has 10 levels of skill.

As you move through the Island of Fugloy songs of success or failure ring in your ears as each battle with the bird-like creatures rages to an end.

Voyage of the Valkyrie is available on cassette for $34.95, or disk for $39.95, from Advanced Operating Systems, 450 St. John Road, Michigan City, IN 46360, (219) 879-4693.

Reader Service ✓ 348



The Radio Shack Communications Multiplexer

BASF-DPS Cassette Tape

BASF-DPS tape is now available from York 10 Computerware in C-10, C-20, C-45, and C-90 lengths.

BASF has an extra smooth, durable media surface which increases head life as well as permitting higher density recordings. A five-screw shell fits all Phillips type cassette recorders.

One dozen cassettes, bulk pack, cost $8 for the C-10 length, $10 for C-20s, $13 for C-45s, and $21 for C-90s. Contact York 10 Computerware, 24573 Kittridge St., Canoga Park, CA 91307, (213) 710-1430, for additional information.

Reader Service ✓ 166

MicroPlan

The MicroPlan Financial Modeling program, for the Model II, was developed to reduce time-consuming forecast and budgeting functions into a 15-minute task.

MicroPlan combines user-definable formats similar to VisiCalc with Desktop Plan-like consolidation capabilities allowing users to change any aspect of their budgets and compute new totals instantly.

The plan runs under CP/M and is provided with complete documentation. For more information contact Data Technology Industries, 700 Whitney St., San Leadro, CA 94577, (415) 638-1206.

Reader Service ✓ 172

A Communications Multiplexer

A Communications Multiplexer, which allows your Model II to respond to information requests from as many as 16 telephone lines at once, is now available.

This multiplexer makes the Model II suitable for use as a host computer in private Videotex information retrieval networks.

This product is available in two versions: eight-line capability for $6000; or 16-line capability for $8000 from Radio Shack Special Marketing, 1600 One Tandy Center, Fort Worth, TX 76102.

Reader Service ✓ 167

NEW!

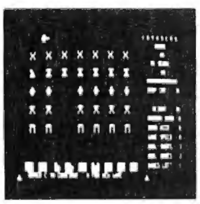
LOST COLONY

By David Feitelberg

The scene is "Warren's World," the world's first attempt at colonizing a planet in deep space. The next support ship from Earth isn't due for about 15 years, and . . . well, let's just say that things aren't going too smoothly. An election was held for an economic manager to straighten things out, and guess who won!

You are presented with all the human, natural, and industrial resources of the planet, with instructions either to make things better or be voted out of office in shame. A remarkable simulation, LOST COLONY arms you with maps and charts as tools for resource management. You assign human and robotic labor forces, explore new land, and allocate production quotas. At the same time you must determine equitable pay scales and taxes that will both optimize productivity and keep the populace happy (or at least quiet and working).

Communication is handled by your TRS-80* model I or III, using sentences like, "ASSIGN 1200 SEMI-LITERATE MEN TO FARMING," or commands as short as "1200 FARM." A challenging game, LOST COLONY might give you insight into real life management as well. Available on 16K tape or 32K disk for only $19.95 each.



INVADERS FROM SPACE

By Carl Miller

A fast machine language approach to this classic (and addictive) space game. The aliens drop bombs, move around, and try to overrun your bases.

You choose the speed, enemy bomb frequency and accuracy, and how many shots and bases you have. Unlike other games of this type, you can move your base and simultaneously fire at the invaders. Fun for all ages and skill levels, it has full sound effects for even more excitement.

Available for TRS-80* model I or III. Only $14.95 on 16K tape version or $20.95 on 16K disk.

ASTROBALL

By John Allen

More features, thrills, and sound than even our famous PINBALL. With ASTROBALL in your TRS-80* model I or III the arrow keys become flipper buttons, the screen becomes the play board, and you become the "Pinball Wizard!"

A flying saucer and moving spaceships add to the fun. Knocking out five meteors earns you an extra ball, while black holes swallow the ball in play.

ASTROBALL's five skill levels will have all your family and friends lining up for the pinball action. Available on 16K tape or disk for just $19.95 each.

By Steven Kearns

Huge antimatter rocks appear on the Tactical Display Screen of your spacecraft. You blast away but they just explode into smaller chunks for you to destroy.

To add to your woes, alien ships and time bombs appear periodically. If the ships hit you or the timers reach zero—BOOM! Maneuver, fire lasers, jump to hyper-space—anything to avoid the onslaught. For one or two players, with nine skill levels.

A realtime arcade game with sound, for TRS-80* model I or III. Versions for 16K tape or 32K disk are $19.95 each.

ENTERTAINMENT SOFTWARE SUPPLIED FOR TRS-80* MODELS I & III

Acorn
Software Products, Inc.

✓34

634 North Carolina Avenue, S.E.
Washington, D.C. 20003
(202) 544-4259

EVEREST EXPLORER

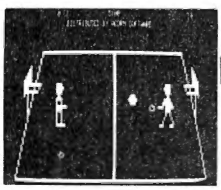
By William Godwin & Don Knowlton

Towering above the clouds, the summit of Mount Everest is a forbidding spot. Only a handful of daring adventurers have made it to this five-mile-high pinnacle where the air is thin, violent storms erupt with little warning, and danger lurks at every footfall.

EVEREST EXPLORER lets you challenge the world's highest mountain without ever leaving home. This logistical simulation goes beyond most adventures, forcing you to make all the life-or-death decisions required at every step of the attempt.

Assigned an expedition budget, you must select the manpower, food, fuel, shelter and oxygen supplies you will need to support your quest. Now the adventure begins as you manipulate your resources to outwit the elements and terrain, establishing ever higher encampments. Weather, route, season, climbers' condition and morale all play key roles in the final question: Will you reach Everest's 29,028 foot summit—and return to tell about it?

Available for TRS-80* model I or III for $19.95. Order on 16K tape, or the 32K disk version which includes a "save game" feature and other enhancements.



BASKET-BALL

By John Allen

You have to be fast to keep up with the action as you try to outscore your opponent in five minutes of one-on-one basketball against a friend or your TRS-80* model I or III.

Steal the ball, duck around your opponent and slant toward the basket for a lay-up! The graphics are based on a 3-dimensional depiction of a basketball court, and ball dribbling sounds add to the realism. It's all there but the cheers in this fast, machine language game!

Available on tape for only $14.95, or disk for $20.95. Requires 16K.

MONEY MANAGER

By Andrew P. Bartorillo

A complete management tool for the home budget, it accurately keeps track of your checkbook and provides an easy method of budget allocation.

You can store information on up to 100 checkbook entries per month (250 with 48K), specify any automatic withdrawals, keep records of tax-deductibles, and record expenses by category. You can even break up charge account payments into the proper categories.

Requires TRS-80* model I or III with a minimum of 32K and one disk drive. Order now for only $39.95.

By Richard Wilkes

Using your SUPERSCRIPT Modified SCRIPSIT* Word Processor and a compatible printer, you can now underline, boldface, insert text during printout, slash zeros, set type pitch, subscript and, of course, superscript! You can even read your directory and kill files without ever leaving SCRIPSIT*.

Includes drivers for popular printers and easy instructions. Just $50.00 for TRS-80* model I (requires 32K, 1 drive, SCRIPSIT/LC*) or model III (requires 32K, 2-drives, model I version of SCRIPSIT/LC*).

ACORN PROGRAMS ARE AVAILABLE AT FINE COMPUTER STORES EVERYWHERE

MAIL ORDERS: Include $2.00 shipping & handling (D.C. residents add 6% sales tax).

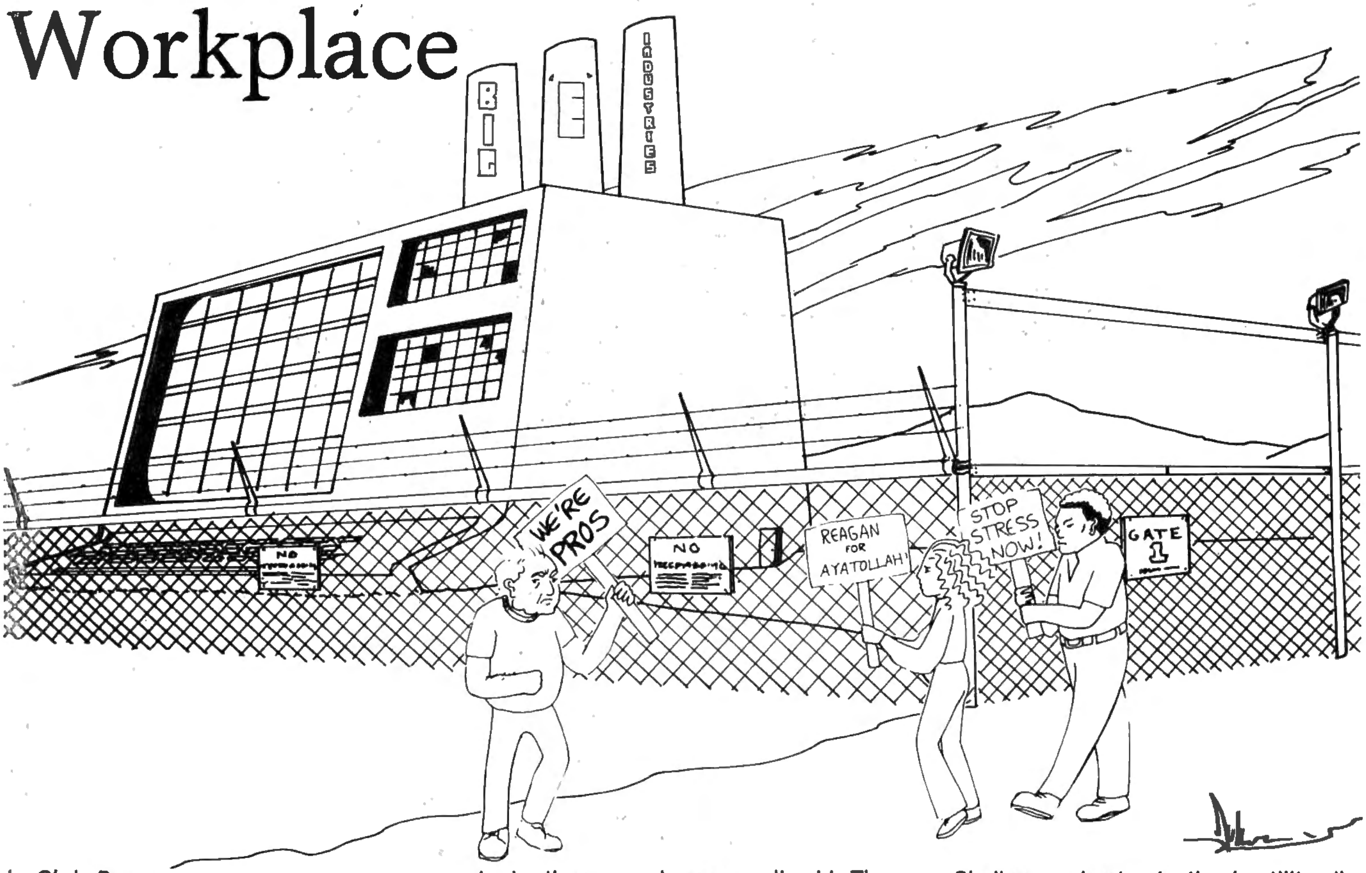
CHARGE CARD CUSTOMERS: Call (202) 544-4259 for fastest service.

*Trademark of Tandy Corp.

Prices Subject to Change

✓See List of Advertisers on page 402

Computerization Of the Workplace



by Chris Brown
Technical Editor

Can it be that we, the true believers in the Computer Revolution, labor under a misapprehension? For years we have told anyone who would listen that the dawn of the computing age would herald a renaissance in the American workplace. Changes for the better were coming.

We saw the machine ending the costly and time-consuming paper chase in the American office. We saw it freeing us from repetitive and demeaning operations on the production floor. We saw it enhancing our lives. Well, for many children of the computer age, the promise has failed.

Early in the summer of 1981 one national group of workers deeply involved in computers walked off their jobs. By most standards, these people were well-paid. They were willing to risk fines, firing and imprisonment for intangibles that had not been issues in the American labor movement before. Professional stature, recognition, and the relief of job-induced stress lead the air traffic controllers to strike. In striking, the controllers were typical of a new breed of American worker—a worker involved with technology and the computer, the victim of problems created by the machines in the American workplace.

"Computer technology creates the false illusion that problems with the American workplace are over," says Harley Shaiken, an MIT researcher of advanced technologies. Shaiken went on to tell *80 Microcomputing* that in his view a new set of labor related problems is spawned by the computer. He said, "Computer technology has always had the potential to create a better workplace. The reality of the situation however, has been the opposite. The computerized workplace has been deteriorating in quality."

Shaiken understands the hostility displayed by many workers involved with computers. He sees three key factors at the root of dissatisfaction of data processing personnel, air traffic controllers, nurses and operators of computerized machine tools. He feels that computerizing the workplace limits worker skills, increases management control and allows constant monitoring of worker performance. While not particularly evil in themselves, these three factors taken together dehumanize the worker.

A similar theme is sounded by Robert Shrank of the Ford Foundation. Shrank, a self-described Program Officer at Ford and an ex-tool maker, sees time as a key issue. "Control of an employee's time is the key to how the computer is affecting the American workplace. Be it a VDT (visual display terminal), machine tool, or what have you, where computers are involved an employee may be measured as to what, when and how much, as never before."

Shrank told *80 Microcomputing* that the ability to control time and employee activity

". . . trouble is on the horizon when American labor meets the computer."

closely is the ultimate strength of the computerized workplace. He indicated that management welcomes this aspect of the new technology but labor unions cannot cope with it. Both Shaiken and Shrank agree that trouble is on the horizon when American labor meets the computer.

"The self-perception of employees involved with computers is changing," says Shrank. "These people realize they have a large amount of power in the society and they see themselves as white collar professionals, even if their employers don't." Power is a real issue in the computerization of the American workplace; as in other aspects of our lives, the computer alters traditional power and political structures of the business world. One hundred people and a few computers in Gander, Newfoundland halted traffic in the busiest air corridor in the world. This scares managers who have contemplated computerizing their plants and offices. So much power in the hands of so few workers means vulnerability for American business.

"Management must decentralize where the computer is involved," says Ford Foundation's Robert Shrank. "By doing so an inordinate amount of power will not be concentrated in a few hands. Gander, Newfoundland taught us that." Shrank feels the air traffic controllers overestimated their clout and were "intoxicated by their perception of their own power," but he does not minimize the real problems associated with who wields the power of the computer. Ironically, the centralized power that allows a few employees to disrupt a large process also makes them vulnerable. They may be replaced by a smaller number of supervisors who can keep a plant from shutting down completely.

MIT's Shaiken told *80 Microcomputing* that American workplace computerization will change labor union operation. "Union power may not be great enough to deal with computerization," he said. "In times of crisis labor unions in high-technology areas may find themselves with only two alternatives: First, to try and widen the scope of their strikes by involving other groups as the controllers unsuccessfully attempted to do with the machinists union, or second, to look internationally for support."

The second alternative carries far-reaching implications. Does computerization of the workplace lead to a world labor movement? For Shaiken, the answer is yes. "A world labor movement could definitely result from the computerization of the workplace," he said. "In fact," he added, "we have already seen the seeds of this movement in the controller's strike when Canadian and Canary Islands controllers came haltingly to the aid of their American counterparts."

Both Shaiken and Shrank believe there are ways to overcome the problems of the computerized workplace short of striking. They agree that a decent salary for technically trained people is mandatory. They also feel other issues must be addressed; stress is one.

"Management has made the mistake of thinking that all productivity problems can be solved by computers," says Robert Shrank. "They feel that using 60 minutes out of each hour of an employee's time is possible where computers are involved. If you believe this, you're nuts." Shrank feels that 50 minutes per hour is a more realistic figure if job stress is to be kept within bounds.

Harley Shaiken suggests, "We should make an effort to see the social implications of the introduction of technology in the workplace *before* we use it, not after." Says Shaiken, "The design and deployment of computers in work environments should be keyed to social goals." Organizing the computerized workplace becomes increasingly important. Organization can heighten the employee's sense of identity and self-respect as he manages a machine that, in reality, controls him.

How workers and management cope with the computer is subject to conjecture. Changes in traditional views and actions are imminent for both management and labor. The golden age of the computer may have arrived finally. Its promise is yet to be fulfilled. ■

✓See List of Advertisers on page 402

Will the postman soon bring bytes?

Digital Delivery

by Bert Latamore
Desktop Computing staff

Long distance communication has been an issue since civilization began. The ancients used drums, mirror reflection and smoke signals. When the Egyptians and Chinese invented writing, they also invented the courier and built complex road networks for their runners to travel. These courier services developed into the modern postal system. Samuel Morse started the electronic communication revolution, but not until the 1920s was the telephone widely accepted and a real competitor to postal systems.

Today the United States Postal Service (USPS) handles more than one billion first-class letters a year. Although the telephone service carries 2.5 calls for each first-class letter mailed, the post office is still a major force. Ironically, this old-fashioned method of moving information survives at the heart of the "information revolution."

The postal revolution may finally be upon us. A new form of message delivery, electronic mail, is riding on the coattails of computer networking. Futurists believe it may replace the postal system and take part of the load now carried by telephone. According to prognosticators, the question in electronic mail is not whether it will be an important force in the future but how much impact it will have and what form it will take.

As always when predicting the future, experts disagree. Some expect all first-class mail to become electronic, with users sending and receiving messages directly from terminals in their homes and offices. Others expect a more moderate use of electronic mail using facsimile equipment in post offices or other central locations.

Network Systems

These two versions of the future reflect two different forms of electronic mail. The first, used today by businesses with local networks, is available to the public from subscription services such as CompuServe, Columbus, OH; The Source, Vienna, VA; Tymnet, Cupertino, CA; and Telenet, Vienna, VA. This kind of electronic mail requires a central computer connected to terminals or microcomputers at remote locations.

Compose and address your message on your terminal, then contact the electronic mail computer you use and send it on its way. The computer stores the message until the intended receiver checks in. The computer informs the contacted party of a message and sends it to his terminal upon request.

If both parties are on line simultaneously, they can communicate in real time by alternately writing messages. The effect is much like a telephone call.

Large businesses with local networking systems use this form of electronic mail for interoffice communications. They are slow to use it for intercompany and personal correspondence, partly because various electronic mail databases do not communicate with each other. These databases use password security systems. Upon subscription to a database you are given a set of passwords; one identifies the service you want and one identifies you as a legitimate subscriber. The database computer recognizes the password by its length and form. Because these forms and lengths are not uniform, computers cannot communicate between databases. If you subscribe to the electronic mail database at CompuServe and I am on Tymnet's OnTym we cannot send letters to each other.

Promoters of this version expect industry standards similar to those allowing the various common carrier packet networks to interconnect worldwide. A standard would enable the services to communicate freely.

Meanwhile, several subscriber-supported electronic mail services are in operation. Rich Baker of CompuServe said their mail database is put to a wide range of uses, from replying to want ads posted on electronic bulletin boards to playing electronic games. Newspapers use electronic mail to transmit stories from remote bureaus to their main offices and traveling salesmen use it to communicate with their offices.

Facsimile Systems

The second picture of the future is based on the presumption that the dominant form of electronic mail will send and receive photographic copies of documents. To use this system you would prepare your letter with a pen or typewriter. A business could prepare advertising packages as it does now. The finished product would be placed in a machine resembling a photocopier. The machine would read the document and send an exact description to the machine at the location you selected. That machine would create an identical copy.

The big advantage of this system is the control the sender has over the information's form. This might not be important in a personal letter or interoffice memo, but it is very important when the information is

". . . this opposition from a politically strong White House probably means ECOM is dead."

a graph or photograph. Unfortunately, the machinery is more expensive than a dumb terminal and less versatile than a microcomputer. It is unlikely that the average home in the year 2001 will have such equipment. This system would probably be used between central locations; individual documents would be mailed from the receiving stations to their ultimate destinations.

The USPS was scheduled to inaugurate such a facsimile system, called Electronic Computer Originated Mail (ECOM), in January. All the necessary equipment to connect several US cities has been purchased. The Reagan administration, however, is opposed to the USPS entering the electronic mail market. The White House and Departments of Justice and Commerce have jointly filed statements with the Postal Rate Commission opposing the ECOM plan and any other move into this area by the postal service. David Foulger, editor of *Electronic Mail and Message Systems*, a biweekly bulletin published by International Resource Development, Norwalk, Ct, said this opposition from a politically strong White House probably means ECOM is dead.

However, he said several other firms are waiting in the wings, ready to start their own services; Federal Express, particularly, seems to be preparing for an overnight facsimile mail delivery in competition with the USPS, and is renting satellite transmission capacity far beyond their internal needs.

In spite of the problems with ECOM and competition with desk-to-desk mail, facsimile apparently has a good future. *The Report on Electronic Mail* for the second quarter of 1981, a publication of The Yankee Group, Cambridge, MA covers the market potential of several facsimile systems designed for office use. They conclude this form of message moving will retain part of the electronic mail market because it eliminates typing information already on paper into the system. They predict facsimile transmission equipment integrated into office work-stations rather than marketed as independent items. The integrated systems would offer word processing, database access, data processing and electronic filing as well as electronic mail.

AT&T Entry

The other big communications monopoly in the United States, American Telephone and Telegraph (AT&T), is also preparing for entry into electronic mail. Presently AT&T is voluntarily going through the throes of partial deregulation.

Foulger expects AT&T will probably meet the deadline Congress has set to create a separate subsidiary operating in electronic information. The subsidiary is required so the telephone giant cannot use revenues from business areas in which it has a legal monopoly to fund adventures in other areas, an unfair advantage over competitors. AT&T will move quickly into electronic mail as well as other electronic data service areas as soon as legal requirements are met.

However, he said, "Deregulation may not be the gift that AT&T thinks it is."

AT&T will be entering a highly competitive market in which several corporations, including GTE Telenet and Tymnet, are already established. Foulger predicts other firms will unveil ambitious systems.

Home Use

Futurists disagree whether electronic mail will be major in the home for the rest of the century; they agree on the definition of electronic mail. Ray Boggs of the Consumer Division of Venture Development Corp., Wellesley, MA, recently finished a study of the electronic home of the future. He concluded electronic mail "doesn't fit in" as a communications device and will not replace the telephone.

On the other hand, he and many others expect electronic money transfer arrangements in our future, allowing the consumer to send orders from home directly to the bank's computer.

"It will be precisely this kind of application that will drive the home computer market," Boggs said. Francis O'Reilly, an independent business analyst for Business Communications Corp., Stamford, CT agrees with this scenario. He points out that financial transfers account for 90 percent of all first-class mail.

"By the end of the century, electronic money transfer is going to eliminate most of this volume," he said, "so I think electronic mail is going to creep in in ways other than the one we think of."

The result would be less paperwork for the customer and less data entry for the bank.

Your bills would come directly to your terminal through the network. Your employer could complete the cycle by having his computer deposit your pay directly into your bank.

Electronic Mail and the Telephone

O'Reilly also believes electronic mail may partially replace the telephone for personal conversations. If we communicated through electronic mail I could send him my interview questions on electronic mail when I wanted to, and he could send his answers and comments back at his convenience.

"Of course we would lose the interactive nature of the conversation," he said. On the other hand, it would save days of telephone work spent alternately calling people and transcribing answers from long lists of questions.

All these applications wait for home computers to become commonplace. It is not practical to offer these services until enough of the public can use them. O'Reilly and Foulger both predict we will reach this point by 1990.

Mail by Cable

A recent issue of *Electronic Mail and Message Systems* reported the present trend in cable television is to offer a huge selection of channels. One cable company now has a license to offer 150 channels in Princeton, NJ, and another has proposed a 200-channel cable in Florida.

Foulger wonders what they are going to put on all those channels. He suggests that some of those extra channels could carry data services into the home. The Source and CompuServe are already experimenting with cable.

Another possibility is electronic data services offered simultaneously with voice telephone transmission. Telephone lines carry several calls at once through multiplexing. According to Lawrence Roberts of GTE, the creator of the first operating packet network, there is no reason data transmissions could not be multiplexed with voice.

The French government is experimenting with this kind of service. At least one French firm manufactures a telephone modem allowing simultaneous voice and digital data transmission, according to Foulger.

Business Mail

Vision is clearer in the business world, where the trend for electronic mail as part of the electronic office is gathering strength. So far, O'Reilly said, electronic mail is "a large corporation phenomenon. The smaller organizations have the communications network to do it, and I think they have the need."

Electronic mail is used in three situations:

- In office complexes, where it is incon-

"The lengthy Canadian mail strike divided the electronic mail sheep from the goats..."

venient for one executive to meet with another, electronic mail replaces telephones and sends graphic data the telephone cannot display.

- Firms with traveling sales staffs equip their salesmen with portable terminals used to send orders and receive memos from the home office.
- Firms with widely separated offices (including overseas branches) use electronic mail databases on the common carrier networks for interoffice communication.

Costs

Electronic mail is still more expensive than the USPS. According to sources for both Tymnet and CompuServe, electronic "postage" for the average message is about 50 cents, as opposed to 18 cents for first class mail. The stamp, however, is the smallest part of a letter's cost. Electronic mail could save time, labor and money composing the letter with word processing.

Electronic mail, however, is not being seen as a stand-alone service in the marketplace. H. Paris Burstyn of The Yankee Group said the current office trend is towards integrated systems which offer word processing, database access, data processing and electronic filing as well as electronic mail.

A similar situation can be envisioned in the home market. The average homeowner may not be interested in investing in a microcomputer solely to have access to electronic mail. However, if he does buy a computer for other reasons electronic mail becomes a useful and easily added utility.

International Mail

The legal situation is not entirely clear, depending in part on the country. Foulger may be correct in his opinion that the Reagan administration will prevent the USPS from entering the electronic mail arena. Furthermore, as Burstyn points out, the department moved too late to establish a legal monopoly on the service. Too many other companies are already in electronic mail in the United States.

In other countries the formula is different: The national telephone monopoly, often a part of the government, also runs the dominant data network. Foulger says Canada sees electronic mail as a direct threat to its postal system; the government is restricting its development.

How long can they delay the inevitable? The lengthy Canadian mail strike divided the electronic mail sheep from the goats, according to Foulger. Firms with electronic mail systems in place were not greatly hurt by the strike, but those without it had much more trouble.

Foulger said a strike alone will not cause businesses to adopt electronic mail, mainly

Relief at last . . .

☐ YES Sign me up as a Charter Subscriber to Desktop Computing. I will get one year of the only computing magazine written in plain English for $17.97 (a savings of 50 % off the newsstand). My subscription will begin with the premiere issue—October 1981.

Bill ☐ Me ☐ MC ☐ VISA ☐ AE

Your first issue will be arriving by mid-October 1981

Canadian 1 yr. only/US funds $19.97

Foreign 1 yr. only/US funds $29.97

Name ______________________

Address ______________________

City ______________ State ________ Zip ________

Signature ______________________ Expire Date ________

Card # ______________ Interbank # ______________

DESKTOP COMPUTING Peterborough, New Hampshire 03458 31NC8B


Relief at last . . .

☐ YES Sign me up as a Charter Subscriber to Desktop Computing. I will get one year of the only computing magazine written in plain English for $17.97 (a savings of 50 % off the newsstand). My subscription will begin with the premiere issue—October 1981.

Bill ☐ Me ☐ MC ☐ VISA ☐ AE

Your first issue will be arriving by mid-October 1981

Canadian 1 yr. only/US funds $19.97

Foreign 1 yr. only/US funds $29.97

Name ______________________

Address ______________________

City ______________ State ________ Zip ________

Signature ______________________ Expire Date ________

Card # ______________ Interbank # ______________

DESKTOP COMPUTING Peterborough, New Hampshire 03458 31NC8B

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 82 PETERBOROUGH, NH 03458

POSTAGE WILL BE PAID BY ADDRESSEE

Desktop Computing

A WAYNE GREEN PUBLICATION ™

80 Pine Street,
Peterborough, N.H. 03458

Attn: Debra Boudrieau

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 82 PETERBOROUGH, NH 03458

POSTAGE WILL BE PAID BY ADDRESSEE

Desktop Computing

A WAYNE GREEN PUBLICATION ™

80 Pine Street,
Peterborough, N.H. 03458

Attn: Debra Boudrieau

"...electronic mail is appropriate for some uses and the telephone for others; they will not really compete."

because of the long lead time necessary to install a network (up to six months). If the business chooses the common carrier system it can get on line as quickly as it can acquire terminals. The Canadian packet network Datapac is not as well developed as packet network systems and does not reach all areas of the country.

The story is entirely different in France, where the new Socialist government is sponsoring a public experiment in home networking in a Paris suburb. At the opening, the new communications minister told reporters France is definitely in the lead in this area and intends to export equipment to the United States.

Impact

Electronic mail's impact on users is hard to assess. Certainly it will not change personal lifestyles as the telephone did in the 1920s. Electronic mail is simply another kind of electronic communications.

The USPS and its equivalents in the other industrialized countries are obvious targets for a major impact. Most first-class mail will disappear if O'Reilly's model of electronic fund transfer comes true. Tymnet spokesman Dick Jokinen said his firm has a prototype service called TymGram allowing a single copy of a document at one end to produce multiple copies at different locations. It is a small step to electronic mass mailing.

Postal officials have read the writing on the wall. According to O'Reilly, the USPS attempt to enter the facsimile market was a bid to preserve the postal system, even at the expense of post office jobs lost to automation. If it is prevented from entering this market or if the facsimile market does not catch on with the public, the post office runs the danger of becoming the system of last resort.

The new service will probably have much less impact on telephone companies. As Baker pointed out, electronic mail is appropriate for some uses and the telephone for others; they will not really compete. Which service you use will be a matter of individual taste.

For the average user, Foulger said, the important fact is that the services are there and growing. If you are interested in learning more about them, subscribe to *The Report on Electronic Mail* and *Electronic Mail and Message Systems*. You might also be interested in The Yankee Group's report, *Home Electronics through 1995, Strategies for Providing Information and Control.*

Whether you decide to jump in today or wait and see what happens, whether you are an executive in a multinational corporation, a professional with a one-person office, or a homebody, electronic mail may be somewhere in your future.■

Not all Spelling Checkers are the same.

MICROPROOF[T.M.] stands out!

EASY TO USE: Prepare your text on any Z-80 based microcomputer, using any of a number of popular word processing programs. When you are finished, enter the appropriate command, and MICROPROOF proofreads your document, displaying misspellings and typos on the screen. Then correcting MICROPROOF can display each error separately, requesting you to enter the correct spelling for each. You are also given the option of displaying errors in context or adding words to MICROPROOF's 50,000 word vocabulary. Finally, MICROPROOF corrects your document. All in less than a minute.

```
SELECT APPROPRIATE RESPONSE:

CORRECT MISSPELLED WORD:   ENTER CORRECT WORD
LEAVE WORD "AS IS":        HIT <ENTER> KEY
DISPLAY WORD IN CONTEXT:   ?
ADD WORD TO DICTIONARY:    +
EXIT:                      !

WORD:          (Your error)
RESPONSE:
```

Correcting MICROPROOF Screen Display

SPEED is the single most important factor in a dictionary program. All dictionary programs will find your potential errors but if the program is too slow, you are not likely to use it. MICROPROOF's speed is outstanding. It can proofread a several page letter in 20 seconds.

LOW PRICES: Standard MICROPROOF is available for either $89.50 (TRS-80® Models I or III) or $149.50 (CP/M®, TRS-80® Model II and all others). The optional correction feature can be added at any time for an additional $60.00. Optional patches to integrate MICROPROOF into your word processing software can also be added at any time for an additional $35.00. (Integration patch not needed for Wordstar®.)

MICROPROOF'S FULL 50,000 WORD VOCABULARY saves you time and allows you greater confidence in the lists of potential errors that MICROPROOF identifies. The mini-dictionary programs, with their 10,000 and 20,000 word vocabularies, have many correctly spelled words omitted from their vocabularies. Consequentially, they identify as potential "errors" many words that are actually spelled correctly; five to ten times as many such words as does MICROPROOF. So, when you use MICROPROOF you will have far fewer extra words to evaluate, a major time savings. There will be less need to look up words in order to verify that they are in fact spelled correctly. The extra 30,000 words in MICROPROOF's vocabulary assures you confidence in the error lists that MICROPROOF generates.

There are other proofreading programs available to choose from. Since MICROPROOF became available in December of 1980, a number of companies have announced programs with small dictionaries. It took us almost two years to develop MICROPROOF. During that time we were able to compress our full 50,000 word dictionary into a manageable size (fits on one single density 5¼ inch disk). And we were able to design a proofing program which operates remarkably fast. The chart below illustrates the comparative advantages of MICROPROOF.

ADVANTAGES OF MICROPROOF

	MICROPROOF DICTIONARY SOFTWARE	OTHERS (Mini-Dictionaries)
DICTIONARY SIZE	50,000 Words	20,000 Words
DISK SPACE REQUIRED FOR DICTIONARY	70,000 BYTES (fits easily on one 5¼" disk)	170,000 BYTES
DICTIONARY ENLARGEMENT	VIRTUALLY UNLIMITED	EXTREMELY LIMITED
SPEED—400 Words	20 Seconds	1 to 5 Minutes
SPEED—3,000 Words	1 Minute	2 to 10 Minutes
CORRECTION FEATURE	Optional	Not Available

See your local microcomputer dealer or write to:

CORNUCOPIA SOFTWARE

Post Office Box 5028, Walnut Creek, California 94596 • (415) 893-0633 ✓378

A Model II brings one school system out from under.

Paper Mountain

Theron Wierenga
Box 2007
Holland, MI 49423

The public school district where I am employed has for years suffered under a paper mountain of personnel records and files. Whenever it was necessary to form a seniority list, mail out letters to teachers, list personnel by various account codes or total salaries, it took weeks to sort through the personnel files for the information.

In April, the school system I worked for was going to furnish the intermediate school district with personnel records written on special transmittal forms. A change-over to place our general ledger and payroll on their large computer was to take place in July. It became apparent that having our personnel records in a small computer of our own would be of great value for printing these transmittal forms as well as numerous other jobs. Someone is always asking for a list of special education teachers in the junior and senior high schools or some other version of the employee list. An in-house computer could also keep track of salary schedules for future projections of wage costs, along with being of real service during contract negotiations.

After much consideration, we decided that Radio Shack's Model II with 64K, a Line Printer III and two additional floppy disk drivers would do the job. The cabinet for the additional drives has an extra slot open for a third drive. With three drives future expansion should not be a problem. About 1,000 employee records needed to be stored, each taking up about 500 characters of data. It was decided that I would author all of the software, as the district's needs were peculiar to its own record keeping system (as is often the case). The major portion of the software was written in about two months, with several months of clean-up and changes after this.

Basically, this system is designed around a set of personnel records that can be manipulated by several utility programs. A single employee's data is stored in two sectors of a disk. The employee number is the same as the record number in which the data is stored. Because a record is 256 bytes long on the TRS-80 Model II system (one sector), each employee's data is divided into two files called P1 and P2. For example, the first half of the data for employee number seven is stored in record seven of file P1 and the second half is in record seven of file P2. This gives a direct relationship between employee number and record number, allowing much faster access to records when editing a file.

The two sectors contain up to 81 different items for each employee. The item descriptors, the number of spaces allocated for each item and the necessary format (if any) can be found in Table 1.

All data that is stored in an employee's record is stored as string (ASCII) data. Numeric values must be extracted from the string data if it is needed, as with monetary amounts. Although the Model II has Basic functions for changing numeric values to strings and back again, these were not used in most of the programs. (As it turns out, in this application it becomes more bother than it's worth.) Special routines are used within the programs to do this whenever it is necessary.

There are a total of five programs used to create and manipulate the file. They are called Builder, Printer, Sorter, TMatrix and Teacher.

Builder

This program builds the employee file and allows the user to update and edit it. There are five sections: Input, where an entire employee record of 81 items can be entered in order; Edit, where individual items in a record can be changed; Update, where the same item in each record can be changed; List, which gives the user a display of the item labels and Delete, where an employee's record is erased.

When entering the Input section, the computer prompts the user to enter the 81 individual items. The employee number entered must be the same as the record number dislayed just before the employee number is asked for. For any item that is to be left blank, Enter can be pressed without typing any characters, and a single dash (-) will be entered for the blank item. After all 81 items are entered, the computer will store the entire record on the disk and then display the transmittal form, as entered on the screen. The user is then asked if he or she wishes to continue with the next employee record. If the answer is no the program will ask what section to enter next. If Edit is then entered, corrections or changes could be made in the record just entered.

When Builder displays a label asking the user to enter an item, a row of asterisks will appear under the label. The length of this row shows the amount of space allowed for the item. If fewer characters are typed in, the computer will fill the rest of the item with blanks. If too many characters are entered, the computer will shorten the item to the maximum length.

The Edit section asks you for the number of the employee whose record you wish to change. The second question is the item number you wish to change. After this is entered, the computer displays the entire employee record and asks for new information for the item specified, while displaying the current entry for that item. The program will continue to ask for different items to be edited until the user types in a zero to quit.

If Enter is pressed without typing a new entry in, no change will be made in the record.

The update section is similar to Edit except the program allows you to edit the same item in every record. The program will display the entire record and the current entry for the item to be updated.

After a new item is entered, the next employee record will be displayed and then it can be updated. If you don't want to make any changes, press Enter and the item will not be changed from the current entry.

The List section of the program does not operate on the file but can be useful before using Edit or Update. When an L is typed, the computer displays all the item labels together with the item number and the number of spaces allowed for each item. The display can be frozen by pressing the Hold key. (Pressing the Hold key a second time continues writing the items on the screen.) The entire list of labels will not fit on the screen at one time.

Typing a Q(uit) will return the user to Basic.

When the Delete section is entered, the program prompts the user for the employee number of the person whose record is to be deleted. After the correction employee number is entered, all items in the record will be erased except the following:

1. Employee Number
2. Position Title
6. Name
7. Social Security Number
8. Address
9. City-State
10. Zip
11. Phone
13. Term. Date
14. Reason
75. Residency

The appropriate reason for deleting the record must be entered in item 14. Active employees will have a dash, i.e., nothing entered, for Reason. The entry is Reason is used by the Printer program to determine whether a person is an active employee or a former employee and, consequently, whether the record should be printed or not. Note: It is important to always exit Builder through the Quit command. If the break key is used instead, some data may not be written to disk and lost.

		Number of Characters	Format, Etc.
	Dist—County and district numbers. The computer fills this with 61010 always.		
1.	Employee Number—We assign a 4 digit number which is the record number in our computer	5	XXXX
2.	Position Title—A narrative description.	20	
3.	Class—Type of employee	2	
4.	Bldg.—Employee's work center	2	
5.	Dept.—Where employee is assigned	2	
6.	Name—Employee's name—last, first, initial with no punctuation	25	
7.	Social Security Number—9 digits, include the two dashes	12	XXX-XX-XXXX
8.	Address—Number and street	20	
9.	City-State—City and State abbreviation, state in last two spaces at right	15	
10.	Zip	5	
11.	Phone—Home phone, include the dash	9	XXX-XXXX

Table 1. A sample of the employee transmittal form. The number of characters cannot be increased or the format changed.

Printer

This program is used to output the employee file, or portions of it, to the line printer. There are three options.

One option prints a complete set of employee transmittal forms. The appropriate forms are placed in the printer, and the print head is centered on the top line. Next, enter the employee number of the first and last employee to be printed. This allows you to start in the middle of the file, if necessary; a useful feature after the printer has run out of paper, for example.

Option number two prints a listing of the file with only a few basic items.

The items printed are:

1. Employee number
2. Position Title
3. Class
4. Building
6. Name
45. Total
58, 63, 68, 73. The 4 Contracts
59, 64, 69, 74. The 4 Accounts

This option also allows one to specify the first and last employee desired.

The third option prints the employee transmittal form as in option one, however, only one employee is done at a time. This allows the user to obtain a single set of forms after a record has been edited. After the transmittal form has been printed, the program asks for another employee number. Additional forms can be obtained, or by typing a zero, the program will quit. If the user wishes to have the printer discontinue immediately, the Break key can be pressed.

Before Printer begins to output the employee records, the user is asked, "Should former employee's records be printed?" If the response is yes, then all records in the file are printed. If no is the response, then only those records that have a "–" for Reason (item 14) will be printed. Item 14 should be changed to the appropriate Reason code whenever an employee's record is deleted in Builder.

Sorter

The program is a general purpose sort and print routine. It has several features: an unlimited number of keys, partial keys, And...Or function, four types of printout, seven items that can be summed, and the printout can be ordered or alphabetized by a given item.

At the beginning of the program you are taken through a series of questions to de-

Photo 1

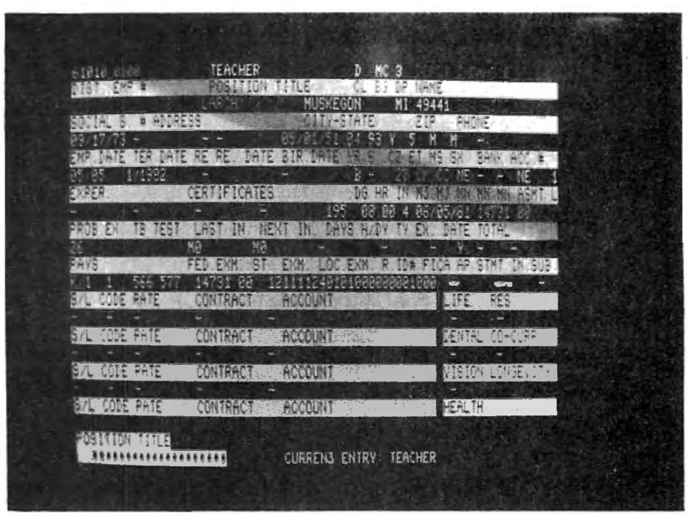


Photo 2

"We want a minimal printout of all teachers who reside in the district."

termine what type of a sort will be done on the file and what kind of printout is desired. As an example, consider the following sort through the file. We want a minimal printout of all teachers that are at the high school who are residents of the district. This will be an And sort because the person must be a teacher *and* a high school employee *and* a resident. We will not sum any items, but we want the list alphabetized by name.

The questions asked are as follows:

- How many keys?—you must respond with the number of different items that will be used to perform the search. In our example this would be three: teachers, high school employee and resident.
- Enter item number for key#—at this point you enter the item number for the three keys. They would be 3, 4 and 75.
- Do you wish a partial key?—In this case no. A partial key is used only when a portion of an item is being searched for. An example of this might be a particular three numbers in the middle of one of the account numbers.
- Enter position of first character, number of characters and the exact Key—this section prompts the user for the partial key if one was chosen. Three values must be typed in, separated by commas. The first value is the numerical position of the first character of the partial key. The second value is the number of characters in the partial key, and the third value is the partial key itself. An example would be 4, 2, AB, meaning the fourth character in the item is the first character in the key. The key is two characters long and the exact key is AB.
- Enter exact key for______—if a partial key is not chosen, this message prompts you for the exact key to be compared with the entire item. An error message will occur if the key typed in is longer than the number of characters used for the item. If the key typed in is shorter than the full number of characters used for the item, the program fills extra spaces with blanks to the right.
- Is this an And or Or sort?—this input allows the user to choose two different types of sort. An And sort is when all of the various keys must match the items in an employee record before that record will be printed. The Or sort will print the record if one or more of the various keys match the items in the employee record.
- What type of printout do you desire?—there are four possible types of printout:

A minimal listing of employees that includes the employee number, position title, class, building, name, total and the four contracts amounts and the account numbers. This printout takes four lines for each employee.

A printout of the entire record in a style very similar to the employee transmittal form, but instead it is printed on regular blank paper.

A mailing label that contains only the name and address of the employee. Single width, pre-gummed labels, with a six line spacing from label to label, must be placed in the printer.

A printout that consists of a list of desired items that the user has specified. If this option is chosen the program asks, "How many items do you want printed for each match?" This can be any value up to 80. The program then asks you to Enter item #—to be printed. The

Now for the TRS-80™ Models I & III

LDOS

VERSION 5.1
THE TRS-80™ OPERATING SYSTEM
MODEL I AND III

* * **DOUBLE** Sided & **DOUBLE** Density support.
* * **AUTOMATIC** Density recognition.
* * 35, 40, 77, 80 and any other track counts are supported.
* * All available drive stepping rates are supported.
* * Hard Drive support, can be **HANDLED AS A SINGLE DRIVE.**
* * Hard drive partitioning, one drive can act as up to six.
* * Intermix 5", 8" and Hard drives, up to a total of 8 drives.
* Compatible with the Model I Radio Shack Expansion Interface.
* Upward compatible with **TRSDOS** (2.3 & 1.3 as documented).
* Fully supports Microsoft language products for **TRS-80.**
* Complete media compatibility Model I to Model III and back.
* Full support for **LOBO's LX-80** interface.
* Full support for **AEROCOMP'S** double density controller**(DDC).**
* Full support for **PERCOM's DOUBLER II.**
* Complete documentation (well over 250 pages).
* Complete technical information.
* A **TOLL-FREE** 800 number for customer service.
* An **LDOS** users bulletin board on MicroNET.
* A Quarterly **LDOS** users magazine (The **LDOS QUARTERLY**).
* A liberal update policy.
* An enhanced **BASIC (LBASIC)** including:
  * > Upward compatible with Microsoft Basic.
  * > High speed **LOAD** and **SAVE.**
  * > Run multiple programs with common variables.
  * > **BLOCKED** (variable length) files are supported.
  * > **DOS** commands may be executed from **LBASIC.**
  * > Built in string array. **SORT.**
  * > Single stepper for debugging.
  * > Several new statements and file modes.
* A compiled **JOB CONTROL LANGUAGE (JCL).**
* **CONV**ert utility to move files from Model III **TRSDOS.**
* An Extended Debugging and Monitor program (with disk access).
* **CMDFILE** for movement of disk and/or tape system **(/CMD)** files.
* Device independent operation.
* Full **LINK**ing, **ROUTE**ing, **FILTER**ing and **SET**ting are supported.
* MiniDOS feature for constant access to certain **DOS** commands.
* **RS-232 DRIVER** for serial support.
* Sophisticated communications software included.
* Wildcard characters and partial Filespecs are supported.
* **DATED FILES,** show when a file was last written to.
* Backup: Mirror, by Class, if Modified, by Date, by Extension, etc.
* Selectable **PURGE** for fast disk "cleanup" of unwanted files.
* Print formatter, for control of printer output.
* Built in printer **SPOOLER,** to both disk and ram.
* Joblog to record all system operations with time stamps.
* **UPPER** and *lower* case support, throughout the system.
* Blinking cursor with selectable cursor character.
* 128 character **TYPE AHEAD** buffering for keyboard input.
* Assign strings to individual keys with Key Stroke Multiply **(KSM).**
* **SUPER FAST** operation with the **SYSRES** feature.
* Extensive user control and system feedback.
* Advanced **PATCH** utility for easy maintenance.
* Complete transportability of software among all **Z-80 LDOS** systems through the use of the **LDOS** high ram supervisory call system.
* Dealers to serve users Nationwide and in the Common Market.
* The only **DOS** with a limited **ONE YEAR WARRANTY!**
* Enjoy a professional operating system on **YOUR TRS-80!**

* Specific hardware is required to use these features.

The Ultimate In Operating Systems For Model I & III

Only $169⁰⁰

Version 5.1

* Model I LDOS provided on 35 track single density media.
* Model III LDOS provided on 40 track double density media.
* LDOS can be provided on special media configurations at an additional charge.
* Prices & Specifications are subject to change without notice.

35

For Further Information Contact The Distributor Or Dealer Nearest You:

(West)	(Central)	(East)	(The Common Market)
LOBO DRIVES INT'L	**GALACTIC SOFTWARE LTD.**	**MISOSYS**	**MOLIMERX LTD.**
354 S. Fairview Ave.	11520 N. Port Washington Rd.	5904 Edgehill Dr.	1 Buckhurst Rd., Bexhill
Goleta, CA 93117	Mequon, WI 53092	Alexandria, VA 22303	Sussex, England
(805) 683-1576	(414) 241-8030	(703) 960-2998	(0424)-220391

DEALER INQUIRIES WELCOME. LDOS is a product of LSI. TRS-80 & Radio Shack are trademarks.

LOGICAL SYSTEMS INC.

Mequon, WI 53092
(414) 241-3066

"The program prints a complete new salary schedule from one to 13 years. . ."

list of item numbers for the desired printout is then entered. Users should always enter several items that will easily identify the employee when a match is found. Examples would be the employee number, name, class, building, etc. If a large number of items are requested the user should keep in mind that they will be printed across the line in the order they were entered. If the length of the requested items exceeds the width as specified by the Forms statement the printer will continue on the next line.

- Do you wish to have any items summed?—there are a total of seven items in the employee's record that may be summed as the individual employees are found in the sort. These items are the Total (item 45), the four Contract amounts (items 58, 63, 68, 73), the C—curricular percentage (item 76) and the Longevity amount (item 81).

If you wish to have items summed, the program will first ask, for "How many items?" and then ask the user to Enter the item number. If more than seven items are asked to be summed, the program will ask, "How many items?" again. The program also checks to see if the items entered are valid ones for summation. If the item is invalid, the program asks for the item number to be entered again.

- Do you want the printout to be alphabetized or ordered?—if no order is needed the program will simply print out the employee records as matches to keys found, and the order will be by employee number. If something like a printout in alphabet order by employee name is desired, Y should be answered. The program then asks the user to Enter the item that the printout will be alphabetized or ordered by. A 6 should be entered if alphabetizing by employee name is desired. Other item numbers will produce an ordered printout based on that particular item. This allows you to print sorted records by building number, class, years of service, etc.

TMatrix

This program must be run to calculate the number of employees on each step and level of the salary schedule. Once run, the program Teacher will use the data file that TMatrix produces and stores on the disk.

TMatrix does a limited sort on the four account codes and also allows the user to choose the class code (item 3). The questions the user is prompted with are as follows:

- Enter the file name for the matrix—the user should enter a name for the matrix that is six characters or less. This same name must be used when running Teacher.
- Is this an And or Or sort?—an And sort is when all keys must match the employee's record and an Or sort is when only one or more keys must match.
- How many keys for the account codes?—enter the number of keys there will be in the search.
- Enter item number, position of first character, number of characters and the exact key for each account number desired—the valid item numbers are 59, 64, 69, and 74. The next three values are the position of the first character of the partial key, the number of characters in the key, and the key itself. The four values are all typed on one line, separated by commas. This is similar to a prompt in Sorter, but the item number for the account code must also be included.

The current record number being read is displayed on the screen as the search continues. When the search is finished the matrix is printed along with a separate matrix for the longevity schedule.

Teacher

This program computes a new teacher salary schedule using the amount of the current base salary and the percent increase for the new schedule. Upon entry into the program the following questions are asked.

- Enter the file name of the matrix you wish to use—a file name that was created by the use of the TMatrix program should be entered. Various matrices can be stored and used depending upon the employee group desired.
- Enter current base and percent increase—the current base salary and the percentage increase (in whole number percents, for example 7.5 for 7.5 percent) should be entered.
- If projections are to be on next year's matrix Enter A 1, if on this year's Enter A 0—if a 1 is entered, each employee will be entered into the matrix on the next year's step. This allows the user to project the cost of next year's salary schedule. The program will print a complete new salary schedule from one to 13 years by half year steps and from 14 to 40 years by yearly steps.

After the new schedule is computed, the program uses the data file produced by TMatrix to find the total amount of teacher salaries for each step and level of the salary schedule. A printout of this follows in which each step is subtotaled and a projected total of all teacher salaries is arrived at. A separate schedule for the longevity totals is also printed in the same format.

Although the programs shown and described here are structured for the employee file at a particular public school system, the basic structure can be used for the employee file of any company or public institution. Actually, very few changes would have to be made in the three main programs Builder, Printer and Sorter. The salary schedule projection programs TMatrix and Teacher would have to be rewritten extensively.

Changes

The number of items (NUM), item labels, and the number of characters in the data statement would need to be changed. The format of the printout on the video display in Builder and printing in Printer and Sorter will need complete changing to reflect the items each employee record has. The Dim statement and the Open and Field statements will need to be changed depending on the number of items and characters in each record. Along with this, a few constants in the program will have to be changed, for example the value 254 (which is a record length) in several formulas. The specific items deleted in Builder and summed in Sorter will have to be changed to meet individual needs.

Very little work was done on these programs in terms of optimizing them for speed. In spite of this, they run fairly quickly for a microcomputer that is reading and writing to disks. The total time to run Sorter given roughly 1,000 employee records, multiple keys and alphabetizing the output is under one hour.■

Program Listing 1

```
100 REM - BUILDER - This program builds the employee file and allows one to
120 REM             to update and edit it. There are five sections: Input,
140 REM             where an entire record is entered, Edit, where items in
160 REM             a record can be changed and Update, where the same item
180 REM             in each record can be changed. There is also a List option
200 REM             where a listing of the labels can be obtained on the
220 REM             screen, and a Delete function that will erase a record.
240 REM             Written by Theron Wierenga during March 1980.
```

Program Listing 1 continues

Program Listing 1 continued

```
260 CLEAR 2000
280 DIM L$(90),L1(90),F$(90)
300 REM - READ IN DATA
320 NUM=81
340 FOR I=1 TO NUM
360 READ L$(I),L1(I)
380 DATA EMPLOYEE #,5,POSITION TITLE,20,CLASS,2,BUILDING,2,DEPARTMENT,2,NAME,25,SOCIAL SECURITY #,12,ADDRESS
400 DATA 20,CITY-STATE,15,ZIP,5,PHONE,9,EMPLOY DATE,8,TERM. DATE,8,REASON,1,RE-EMPLOY DATE,8,BIRTH DATE,8
420 DATA YRS. SERVICE,5,CITIZEN,1,ETHNIC,1,MARITAL STATUS,1,SEX,1,BANK ACCOUNT #,10,EXPERIENCE,5
440 DATA CERTIFICATE 1,6,CERTIFICATE 2,6,CERTIFICATE 3,6,DEGREE,1,HOURS,3,INSTITUTION,2,MAJOR 1,2,MAJOR 2,2
460 DATA MINOR 1,2,MINOR 2,2,MINOR 3,2,ASSIGNMENT,4,LEVEL,1,PROB. EXPR. DATE,8,LAST TB TEST,8
480 DATA LAST INCREASE,8,NEXT INCREASE,8,DAYS,4,HRS./DAY,5,TYPE,1,EXPR. DATE,8
500 DATA TOTAL,9,PAYS,2,FED. EXEMP.,7,STATE EXEMP.,7,LOCAL EXEMP.,7,RET. ID #,5,FICA EX.,1,AUTO PAY,1
520 DATA STMT,1,INS. SUBSIDY,7,STEP/LEVEL 1,3,CODE 1,1,RATE 1,8,CONTRACT 1,9,ACCOUNT 1,24,STEP/LEVEL 2,3,CODE
,1
540 DATA RATE 2,8,CONTRACT 2,9,ACCOUNT 2,24,STEP/LEVEL 3,3,CODE 3,1,RATE 3,8,CONTRACT 3,9,ACCOUNT 3,24,STEP/LE
EL 4,3,CODE 4,1
560 DATA RATE 4,8,CONTRACT 4,9,ACCOUNT 4,24,RESIDENCY,1,CO-CURR. %,5,LIFE INS.,1,DENTAL INS.,1,VISION INS.,1
580 DATA HEALTH INS.,4,LONGEVITY,4
600 NEXT I
620 OPEN "D",1,"P1",254
640 FIELD 1,254 AS A1$
660 OPEN "D",2,"P2",253
680 FIELD 2,253 AS A2$
700 GOSUB 2100
720 IF G$="I" THEN 1260
740 IF G$="U" THEN 1820
760 IF G$="L" THEN 4000
780 IF G$="D" THEN 4220
800 REM**************************************************************************
820 REM - EDIT A RECORD
840 PRINT"INPUT EMPLOYEE NUMBER - ENTER 0 (ZERO) TO QUIT"
860 S=-1
880 INPUT S
900 IF S=0 THEN 700
920 IF S>0 AND S<=LOF(1) THEN 980
940 PRINT"BAD EMPLOYEE NUMBER":PRINT
960 GOTO 840
980 GET 1,S
1000 GET 2,S
1020 PRINT:PRINT
1040 PRINT"WHICH ITEM - ENTER 0 (ZERO) TO QUIT"
1060 I=-1
1080 INPUT I
1100 IF I=0 THEN 700
1120 IF I>0 AND I<=NUM THEN 1180
1140 PRINT"INVALID ITEM NUMBER - TRY AGAIN"
1160 GOTO 1020
1180 GOSUB 3300
1200 GOSUB 2320
1220 GOTO 1020
1240 REM**************************************************************************
1260 REM - INPUT ENTIRE FILES
1280 H$="-"
1300 R=LOF(1)
1320 R=R+1
1340 PRINT"NEXT RECORD NUMBER IS";R
1360 D$="":E$=""
1380 FOR I=1 TO NUM
1400 GOSUB 2840
1420 PRINT
1440 GOSUB 3040
1460 IF I>42 THEN 1520
1480 D$=D$+C$
1500 GOTO 1540
1520 E$=E$+C$
1540 NEXT I
1560 LSET A1$=D$
1580 PUT 1,R
1600 LSET A2$=E$
1620 PUT 2,R
1640 GOSUB 3300
1660 PRINT:PRINT
1680 PRINT"CONTINUE (Y OR N)";
1700 INPUT G$
1720 IF G$="Y" THEN 1320
1740 IF G$="N" THEN 700
1760 GOTO 1680
1780 REM**************************************************************************
1800 REM - UPDATE A SINGLE ITEM IN ALL FILES
1820 PRINT"INPUT ITEM NUMBER TO BE UPDATED IN ALL RECORDS"
1840 I=0
1860 INPUT I
1880 IF I>0 AND I<=NUM THEN 1940
1900 PRINT"INVALID ITEM NUMBER - TRY AGAIN"
1920 GOTO 1820
1940 FOR S=1 TO LOF(1)
1960 GET 1,S
1980 GET 2,S
2000 GOSUB 3300
2020 GOSUB 2320
2040 NEXT S
2060 GOTO 700
2080 REM**************************************************************************
2100 PRINT:PRINT"INPUT WHOLE RECORDS, EDIT, UPDATE, DELETE, LIST LABELS OR QUIT"
```

Program Listing 1 continues

Program Listing 1 continued

```
2120 PRINT"ENTER I, E, U, L, D OR Q"
2140 G$=".→."
2160 INPUT G$
2180 IF G$=".→." THEN 2100
2200 IF G$="I" OR G$="E" OR G$="U" OR G$="L" OR G$="D" THEN 2280
2220 IF G$<>"Q" THEN 2100
2240 CLOSE 1,2
2260 STOP
2280 RETURN
2300 REM***********************************************************************
2320 REM - INPUT AND UPDATE AN ITEM
2340 D$=A1$:E$=A2$
2360 T=0
2380 IF I=1 THEN 2460
2400 FOR J=1 TO I-1
2420 T=T+L1(J)
2440 NEXT J
2460 GOSUB 2840
2480 PRINT TAB(L1(I)+10);"CURRENT ENTRY: ";
2500 IF I>42 THEN 2580
2520 H$=MID$(D$,T+1,L1(I))
2540 PRINT H$
2560 GOTO 2620
2580 H$=MID$(E$,T+1-254,L1(I))
2600 PRINT H$
2620 GOSUB 3040
2640 IF I>42 THEN 2700
2660 MID$(D$,T+1,L1(I))=C$
2680 GOTO 2720
2700 MID$(E$,T+1-254,L1(I))=C$
2720 LSET A1$=D$
2740 LSET A2$=E$
2760 PUT 1,S
2780 PUT 2,S
2800 RETURN
2820 REM***********************************************************************
2840 REM - PRINT OUT LABEL
2860 PRINT:PRINT CHR$(26);
2880 PRINT L$(I)
2900 PRINT"  ";
2920 FOR I2=1 TO L1(I)
2940 PRINT"*";
2960 NEXT I2
2980 PRINT CHR$(25);
3000 RETURN
3020 REM***********************************************************************
3040 REM - FILL IN C$ WITH BLANKS
3060 C$=".→."
3080 INPUT C$
3100 IF C$<>".→." THEN 3140
3120 C$=H$
3140 Z=LEN(C$)
3160 IF Z<=L1(I) THEN 3220
3180 C$=LEFT$(C$,L1(I))
3200 GOTO 3260
3220 IF Z=L1(I) THEN 3260
3240 C$=C$+SPACE$(L1(I)-Z)
3260 RETURN
3280 REM***********************************************************************
3300 REM - PRINT TRANSMITTAL FORM ON VIDEO SCREEN
3320 PRINT:PRINT
3340 T=1
3360 FOR J=1 TO 42
3380 F$(J)=MID$(A1$,T,L1(J))
3400 T=T+L1(J)
3420 NEXT J
3440 T=1
3460 FOR J=43 TO NUM
3480 F$(J)=MID$(A2$,T,L1(J))
3500 T=T+L1(J)
3520 NEXT J
3540 PRINT"61010 ";F$(1);SPACE$(9);F$(2);F$(3);" ";F$(4);" ";F$(5);" ";F$(6)
3560 PRINTCHR$(26);"DIST  EMP #          POSITION TITLE      CL BG DP NAME                      ";CHR$(25)
3580 PRINT F$(7);F$(8);" ";F$(9);" ";F$(10);" ";F$(11)
3600 PRINTCHR$(26);"SOCIAL S. # ADDRESS                  CITY-STATE      ZIP   PHONE          ";CHR$(25)
3620 PRINT F$(12);" ";F$(13);"  ";F$(14);" ";F$(15);" ";F$(16);" ";F$(17);" ";F$(18);"  ";F$(19);"  ";F$(20);"
 ";F$(21);"   ";F$(2
2)
3640 PRINTCHR$(26);"EMP.DATE TER.DATE RE RE. DATE BIR.DATE YR.S  CZ ET MS SX  BANK ACC.#  ";CHR$(25)
3660 PRINT F$(23);"   ";F$(24);"     ";F$(25);"     ";F$(26);"     ";F$(27);" ";F$(28);" ";F$(29);" ";F$(30);"

;F$(31);" ";F$(32);"
 ";F$(33);" ";F$(34);" ";F$(35);" ";F$(36)
3680 PRINTCHR$(26);"EXPER.              CERTIFICATES              DG HR IN MJ MJ MN MN MN ASMT L";CHR$(25)
3700 PRINT F$(37);" ";F$(38);" ";F$(39);" ";F$(40);" ";F$(41);" ";F$(42);" ";F$(43);" ";F$(44);" ";F$(45)
3720 PRINTCHR$(26);"PROB.EX. TB TEST  LAST IN. NEXT IN. DAYS H/DY TY EX. DATE TOTAL      ";CHR$(25)
3740 PRINT F$(46);SPACE$(15);F$(47);"  ";F$(48);"  ";F$(49);"  ";F$(50);"  ";F$(51);"   ";F$(52);"  ";F$(53);"

   ";F$(54)
3760 PRINTCHR$(26);"PAYS                FED.EXM. ST. EXM. LOC.EXM. R.ID# FICA AP STMT IN.SUB.";CHR$(25)
3780 PRINT F$(55);"  ";F$(56);"   ";F$(57);" ";F$(58);" ";F$(59);"  ";F$(77);"      ";F$(75)
3800 PRINTCHR$(26);"S/L CODE RATE     CONTRACT    ACCOUNT                   ";CHR$(25);" ";CHR$(26);"LIFE   RES.

   ";CHR$(25)
3820 PRINT F$(60);"  ";F$(61);"   ";F$(62);" ";F$(63);" ";F$(64);"  ";F$(78);"      ";F$(76)
3840 PRINTCHR$(26);"S/L CODE RATE     CONTRACT    ACCOUNT                   ";CHR$(25);" ";CHR$(26);"DENTAL CO-CUR
```

Program Listing 1 continued

```
.  ";CHR$(25)
3860 PRINT F$(65);"  ";F$(66);"   ";F$(67);" ";F$(68);" ";F$(69);"  ";F$(79);"      ";F$(81)
3880 PRINTCHR$(26);"S/L CODE RATE     CONTRACT    ACCOUNT                     ";CHR$(25);" ";CHR$(26);"VISION LONGEV

TY ";CHR$(25)
3900 PRINT F$(70);"  ";F$(71);"   ";F$(72);" ";F$(73);" ";F$(74);"  ";F$(80)
3920 PRINTCHR$(26);"S/L CODE RATE     CONTRACT    ACCOUNT                     ";CHR$(25);" ";CHR$(26);"HEALTH

   ";CHR$(25)
3940 RETURN
3960 REM*******************************************************************
3980 REM - LIST OUT THE LABELS ON THE SCREEN
4000 J=1
4020 FOR I=1 TO NUM
4040 PRINT TAB(J);I;L$(I);" - ";L1(I);" SPACES";
4060 J=J+40
4080 IF J<50 THEN 4140
4100 PRINT
4120 J=1
4140 NEXT I
4160 PRINT
4180 GOTO 700
4200 REM*******************************************************************
4220 REM - DELETE A RECORD
4240 R=-1
4260 PRINT"ENTER THE NUMBER OF THE EMPLOYEE THAT YOU WISH TO DELETE"
4280 INPUT"ENTER A 0 (ZERO) TO QUIT";R
4300 IF R=0 THEN 700
4320 IF R>0 AND R<=LOF(1) THEN 4380
4340 PRINT"BAD EMPLOYEE NUMBER - TRY AGAIN"
4360 GOTO 4240
4380 D$="":E$="":H$="-":T=0
4400 GET 1,R:GET 2,R
4420 FOR I=1 TO NUM
4440 IF I=1 THEN 4480
4460 T=T+L1(I-1)
4480 GOSUB 3120
4500 IF I=1 OR I=2 OR I=13 OR I=14 THEN GOSUB 4780
4520 IF I>5 AND I<12 THEN GOSUB 4780
4540 IF I=75 THEN GOSUB 4820
4560 IF I>42 THEN 4620
4580 D$=D$+C$
4600 GOTO 4640
4620 E$=E$+C$
4640 NEXT I
4660 LSET A1$=D$
4680 LSET A2$=E$
4700 PUT 1,R:PUT 2,R
4720 GOSUB 3320
4740 PRINT
4760 GOTO 4240
4780 C$=MID$(A1$,T+1,L1(I))
4800 RETURN
4820 C$=MID$(A2$,T+1-254,L1(I))
4840 RETURN
4860 REM**********END OF PROGRAM**********
```

Program Listing 2

```
100 REM - PRINTER - This program prints out the entire file as either a list
120 REM             with basic information or on the Employee Transmittal
140 REM             Forms. Single Transmittal Forms can also be produced.
160 REM             Written by Theron Wierenga during March 1980.
180 CLEAR 2000
200 DIM L1(90),F$(90)
220 OPEN "D",1,"P1",254
240 FIELD 1,254 AS A1$
260 OPEN "D",2,"P2",253
280 FIELD 2,253 AS A2$
300 REM*******************************************************************
320 REM - READ IN DATA
340 NUM=81
360 FOR I=1 TO NUM
380 READ L1(I)
400 DATA5,20,2,2,2,25,12,20,15,5,9,8,8,1,8,8,5,1,1,1,1,10,5,6,6,6,1,3,2,2,2,2,2
420 DATA2,4,1,8,8,8,8,4,5,1,8,9,2,7,7,7,5,1,1,1,7,3,1,8,9,24,3,1,8,9,24,3,1,8,9
440 DATA24,3,1,8,9,24,1,5,1,1,1,4,4
460 NEXT I
480 INPUT"SHOULD FORMER EMPLOYEE'S RECORDS BE PRINTED (Y OR N)";G$
500 IF G$="Y" OR G$="N" THEN 540
520 GOTO 480
540 PRINT"WHAT TYPE OF PRINTOUT IS DESIRED ?"
560 PRINT"     1. Complete set of records on the Employee Transmittal Forms."
580 PRINT"     2. List of employees with basic information."
600 PRINT"     3. A single Employee Transmittal Form at a time."
620 INPUT Z$
640 IF Z$="1" THEN 1540
660 IF Z$="2" THEN 1080
680 IF Z$="3" THEN 760
700 GOTO 540
720 REM*******************************************************************
740 REM - PRINT SINGLE TRANSMITTAL FORM
760 PRINT"Be sure that Transmittal Forms are aligned in the printer."
```

the HOW-TO, SHOW&TELL, STEPbySTEP book of creating DOS RANDOM ACCESS PROGRAMS.

Written for TRS-80 and all computers using Microsoft basic. The breakthrough book for non-programmers ... businessmen and professionals who want to solve in house problems by writing your own customized program ... and the hobbyist with the desire to get beyond the cassette and into disk data storage and file manipulation. **"DOS Random Access & File Handling"** is the plain speaking, easy to read 150 page book for learning **HOW [illegible] WRITE SPECIAL PROGRAMS** for [illegible]ntories, mailing lists, work schedules, record keeping and more. Your acquaintance with Level I and Level II basic and this self-instruction tutorial will assure quick progress.

DSC Publishing
2 Dogwood Drive
PO Box 769
Danbury, CT 06810

Yes, I want to expand my programming knowledge.
Please send me a copy of **"DOS Random Access & File Handling"** for 24.50 + 1.50 shipping ☐
Also send me the compatible program disk for 28.50 + 1.50 shipping. ☐
Overseas add 8.00 per book and disk airmail.

Name ______________________
Address ______________________
City ______________________
State ______________ Zip __________
Master Charge ☐ VISA ☐
Expiration Date ______________________
Card # ______________________
Check ☐ Money Order ☐
COD (2.00 extra) ☐
Signature ______________________
Phone Orders (203) 748-3231
CT. orders add 7.5% sales tax.

Program Listing 2 continued

```
780 PRINT
800 INPUT"Enter Employee Number or 0 (Zero) to quit.";R
820 IF R=0 THEN 1820
840 IF R>0 AND R<=LOF(1) THEN 900
860 PRINT"BAD EMPLOYEE NUMBER - TRY AGAIN"
880 GOTO 800
900 GET 1,R
920 GET 2,R
940 GOSUB 1960
960 IF G$="Y" THEN 1020
980 IF F$(14)<>"-" THEN PRINT"FORMER EMPLOYEE"
1000 IF F$(14)<>"-" THEN 800
1020 GOSUB 2240
1040 GOTO 800
1060 REM**********************************************************************
1080 REM - PRINT OUT LIST WITH BASIC INFORMATION
1100 PRINT"------------------------------------------------------------------------"
1120 PRINT" *** WIDTH MUST BE SET TO 100 CHARACTERS MINIMUM ON THE PRINTER *** "
1140 PRINT"------------------------------------------------------------------------"
1160 PRINT:PRINT"ALIGN PAPER TO TOP OF PAGE --- THEN"
1180 PRINT"INPUT THE FIRST AND LAST EMPLOYEE NUMBER DESIRED"
1200 PRINT"EXAMPLES: 1,999  OR  256,512"
1220 INPUT F1,F2
1240 LPRINT CHR$(31);DATE$;"    ";TIME$;CHR$(30):LPRINT
1260 REM - HEADING HERE
1280 FOR R=F1 TO F2
1300 GET 1,R
1320 GET 2,R
1340 GOSUB 1960
1360 IF G$="Y" THEN 1400
1380 IF F$(14)<>"-" THEN 1480
1400 LPRINT F$(1);" ";F$(2);" ";F$(3);" ";F$(4);" ";F$(6);" ";F$(45);" ";F$(58);" ";F$(59)
1420 LPRINT TAB(69);F$(63);" ";F$(64)
1440 LPRINT TAB(69);F$(68);" ";F$(69)
1460 LPRINT TAB(69);F$(73);" ";F$(74)
1480 NEXT R
1500 GOTO 1820
1520 REM**********************************************************************
1540 REM - PRINT SET OF EMPLOYEE TRANSMITTAL FORMS"
1560 PRINT"ALIGN TRANSMITTAL FORMS IN THE PRINTER --- THEN"
1580 PRINT"INPUT THE FIRST AND LAST EMPLOYEE NUMBER DESIRED"
1600 PRINT"EXAMPLES: 1,999  OR  256,512"
1620 INPUT F1,F2
1640 FOR R=F1 TO F2
1660 GET 1,R
1680 GET 2,R
1700 GOSUB 1960
1720 IF G$="Y" THEN 1760
1740 IF F$(14)<>"-" THEN 1780
1760 GOSUB 2240
1780 NEXT R
1800 REM**********************************************************************
1820 REM - ADVANCE PAPER AND STOP
1840 FOR I=1 TO 20
1860 LPRINT
1880 NEXT I
1900 CLOSE 1,2
1920 STOP
1940 REM**********************************************************************
1960 REM - BREAK DOWN THE RECORD INTO ITS PARTS
1980 T=1
2000 FOR J=1 TO 42
2020 F$(J)=MID$(A1$,T,L1(J))
2040 T=T+L1(J)
2060 NEXT J
2080 T=1
2100 FOR J=43 TO NUM
2120 F$(J)=MID$(A2$,T,L1(J))
2140 T=T+L1(J)
2160 NEXT J
2180 RETURN
2200 REM**********************************************************************
2220 REM - PRINT ENTIRE RECORD ON FORMS
2240 LPRINT"61010 ";F$(1);SPACE$(9);F$(2);F$(3);" ";F$(4);" "F$(5);" ";F$(6)
2260 LPRINT:LPRINT
2280 LPRINT F$(7);" ";F$(8);" ";F$(9);" ";F$(10);"  ";F$(11)
2300 LPRINT:LPRINT
2320 LPRINT F$(12);" ";F$(13);"  ";F$(14);" ";F$(15);" ";F$(16);" ";F$(17);" ";F$(18);"   ";F$(19);"   ";F$(20
;"   ";F$(21);"   ";
F$(22)
2340 LPRINT:LPRINT SPACE$(64);LEFT$(F$(35),2)
2360 LPRINT F$(23);"   ";F$(24);"     ";F$(25);"     ";F$(26);"    ";F$(27);" ";F$(28);" ";F$(29);" ";F$(30);"
";F$(31);" ";F$(32);
" ";F$(33);" ";F$(34);" ";RIGHT$(F$(35),2);" ";F$(36)
2380 LPRINT:LPRINT
2400 LPRINT F$(37);"   ";F$(38);" ";F$(39);" ";F$(40);" ";F$(41);" ";F$(42);" ";F$(43);"  ";F$(44);" ";F$(45)
2420 LPRINT:LPRINT:LPRINT
2440 LPRINT F$(46);SPACE$(16);F$(47);"  ";F$(48);"  ";F$(49);"  ";F$(50);" ";F$(51);"   ";F$(52);"   ";F$(53);
   ";F$(54)
2460 LPRINT:LPRINT
2480 LPRINT SPACE$(40);"RES:";F$(75);"  CO-CUR:";F$(76);" LONG:";F$(81):LPRINT
2500 LPRINT F$(55);"  ";F$(56);"  ";F$(57);"  ";F$(58);" ";F$(59);"      LIFE   ";F$(77)
```

Program Listing 2 continues

✓See List of Advertisers on page 402

Program Listing 2 continued

```
2520 LPRINT
2540 LPRINT F$(60);"  ";F$(61);"  ";F$(62);"  ";F$(63);" ";F$(64);"      DENTAL ";F$(78)
2560 LPRINT
2580 LPRINT F$(65);"  ";F$(66);"  ";F$(67);"  ";F$(68);" ";F$(69);"      VISION ";F$(79)
2600 LPRINT
2620 LPRINT F$(70);"  ";F$(71);"  ";F$(72);"  ";F$(73);" ";F$(74);"      HEALTH ";F$(80)
2640 LPRINT:LPRINT
2660 RETURN
2680 REM**********END OF PROGRAM**********
```

Program Listing 3

```
100 REM - SORTER - This program is a general purpose sort and print routine.
120 REM            Its features are: An unlimited number of keys, partial keys,
140 REM            AND/OR function for the sort, four types of printout, seven
160 REM            items can be summed and the printout can be ordered or
180 REM            alphabetized.
200 REM            Written by Theron Wierenga during March 1980.
220 CLEAR 10000
240 DIM L$(90),L1(90),F$(90),Y(100),E1(100),E2(100),K$(100),W1(1000),W2$(1000)
260 DIM P4(80)
280 REM*****************************************************************
300 REM - READ IN DATA
320 NUM=81
340 FOR I=1 TO NUM
360 READ L$(I),L1(I)
380 DATA EMPLOYEE #,5,POSITION TITLE,20,CLASS,2,BUILDING,2,DEPARTMENT,2,NAME,25,SOCIAL SECURITY #,12,ADDRESS
400 DATA 20,CITY-STATE,15,ZIP,5,PHONE,9,EMPLOY DATE,8,TERM. DATE,8,REASON,1,RE-EMPLOY DATE,8,BIRTH DATE,8
420 DATA YRS. SERVICE,5,CITIZEN,1,ETHNIC,1,MARITAL STATUS,1,SEX,1,BANK ACCOUNT #,10,EXPERIENCE,5
440 DATA CERTIFICATE 1,6,CERTIFICATE 2,6,CERTIFICATE 3,6,DEGREE,1,HOURS,3,INSTITUTION,2,MAJOR 1,2,MAJOR 2,2
460 DATA MINOR 1,2,MINOR 2,2,MINOR 3,2,ASSIGNMENT,4,LEVEL,1,PROB. EXPR. DATE,8,LAST TB TEST,8
480 DATA LAST INCREASE,8,NEXT INCREASE,8,DAYS,4,HRS./DAY,5,TYPE,1,EXPR. DATE,8
500 DATA TOTAL,9,PAYS,2,FED. EXEMP.,7,STATE EXEMP.,7,LOCAL EXEMP.,7,RET. ID #,5,FICA EX.,1,AUTO PAY,1
520 DATA STMT,1,INS. SUBSIDY,7,STEP/LEVEL,3,CODE,1,RATE,8,CONTRACT,9,ACCOUNT,24,STEP/LEVEL,3,CODE,1
540 DATA RATE,8,CONTRACT,9,ACCOUNT,24,STEP/LEVEL,3,CODE,1,RATE,8,CONTRACT,9,ACCOUNT,24,STEP/LEVEL,3,CODE,1
560 DATA RATE,8,CONTRACT,9,ACCOUNT,24,RESIDENCY,1,CO-CURR. %,5,LIFE INS.,1,DENTAL INS.,1,VISION INS.,1
580 DATA HEALTH INS.,4,LONGEVITY,4
600 NEXT I
620 FOR I=1 TO 7
640 READ S2(I)
660 DATA 45,59,64,69,74,76,81
680 NEXT I
700 FOR I=0 TO 9
720 READ S3$(I)
740 DATA0,1,2,3,4,5,6,7,8,9
760 NEXT I
780 OPEN "D",1,"P1",254
800 FIELD 1,254 AS A1$
820 OPEN "D",2,"P2",253
840 FIELD 2,253 AS A2$
860 REM*****************************************************************
880 REM - SET UP THE QUALIFIERS
900 INPUT"HOW MANY KEYS";X
920 IF X<1 OR X>100 THEN 900
940 FOR I=1 TO X
960 PRINT"ENTER ITEM NUMBER FOR KEY #";I;
980 INPUT Y(I)
1000 IF Y(I)>0 AND Y(I)<=NUM THEN 1060
1020 PRINT"INVALID ITEM NUMBER - TRY AGAIN"
1040 GOTO 960
1060 INPUT"DO YOU WISH A PARTIAL KEY (Y OR N)";E$
1080 IF E$="Y" THEN 1140
1100 IF E$="N" THEN 1300
1120 GOTO 1060
1140 PRINT"ENTER POSITION OF 1ST CHARACTER, NUMBER OF CHARACTERS"
1160 PRINT"AND THE EXACT KEY FOR ";L$(Y(I));".  EXAMPLE: 4,2,AB  WHICH MEANS START"
1180 PRINT"LOOKING AT THE 4TH CHARACTER, THE KEY IS 2 CHARACTERS"
1200 PRINT"LONG AND THE KEY IS AB."
1220 K$(I)=".→."
1240 INPUT E1(I),E2(I),K$(I)
1260 IF K$(I)=".→." THEN 1140
1280 GOTO 1500
1300 PRINT"ENTER EXACT KEY FOR ";L$(Y(I));
1320 K$(I)=".→."
1340 INPUT K$(I)
1360 IF K$(I)=".→." THEN 1300
1380 Z=LEN(K$(I))
1400 IF Z<=L1(Y(I)) THEN 1460
1420 PRINT"THE KEY IS TOO LONG"
1440 GOTO 1300
1460 IF Z=L1(Y(I)) THEN 1500
1480 K$(I)=K$(I)+SPACE$(L1(Y(I))-Z)
1500 NEXT I
1520 REM*****************************************************************
1540 REM - IS AN AN AND/OR SORT
1560 INPUT"IS THIS AN 'AND' OR 'OR' SORT";G2$
1580 IF G2$="AND" THEN 1640
1600 IF G2$="OR" THEN 1680
1620 GOTO 1560
1640 P3=1
1660 GOTO 1740
1680 P3=2
1700 REM*****************************************************************
```

Program Listing 3 continues

Program Listing 3 continued

```
1720 REM - DETERMINE THE TYPE OF PRINTOUT
1740 PRINT"WHAT TYPE OF PRINTOUT DO YOU DESIRE ?"
1760 PRINT"     1. Minimal printout for each match."
1780 PRINT"     2. Entire record for each match."
1800 PRINT"     3. A mailing label for each match."
1820 PRINT"     4. Print only specific items requested."
1840 INPUT P1
1860 IF P1=1 OR P1=2 OR P1=3 THEN 2060
1880 IF P1<>4 THEN 1740
1900 INPUT"HOW MANY ITEMS DO YOU WANT PRINTED FOR EACH MATCH";P2
1920 FOR I=1 TO P2
1940 PRINT"ENTER ITEM #";I;"TO BE PRINTED";
1960 INPUT P4(I)
1980 IF P4(I)<1 OR P4(I)>NUM THEN 1940
2000 NEXT I
2020 REM*******************************************************************
2040 REM - ARE ANY ITEMS SUMMED ?
2060 INPUT"DO YOU WISH TO HAVE ANY ITEMS SUMMED (Y OR N)";G$
2080 IF G$="Y" THEN 2140
2100 IF G$="N" THEN 2400
2120 GOTO 2060
2140 INPUT"HOW MANY ITEMS";N
2160 IF N>0 AND N<=7 THEN 2220
2180 PRINT"ONLY SEVEN ITEMS CAN BE SUMMED - ENTER AGAIN"
2200 GOTO 2140
2220 FOR I=1 TO N
2240 INPUT"ENTER THE ITEM NUMBER";S(I)
2260 FOR J=1 TO 7
2280 IF S2(J)=S(I) THEN 2360
2300 NEXT J
2320 PRINT"INVALID ITEM NUMBER FOR SUMMATION  - ENTER AGAIN"
2340 GOTO 2240
2360 NEXT I
2380 REM*******************************************************************
2400 REM - ALPHABETIZE OR ORDER THE PRINTOUT ?
2420 INPUT"DO YOU WANT THE PRINTOUT TO BE ALPHABETIZED OR ORDERED (Y OR N)";O$
2440 IF O$="Y" THEN 2500
2460 IF O$="N" THEN 2620
2480 GOTO 2420
2500 INPUT"ENTER ITEM THAT THE PRINTOUT WILL BE ALPHABETIZED OR ORDERED BY";O1
2520 IF O1>0 AND O1<=NUM THEN 2620
2540 PRINT"INVALID ITEM NUMBER - TRY AGAIN"
2560 GOTO 2500
2580 REM*******************************************************************
2600 REM - START OF THE SEARCH LOOP
2620 GOSUB 5760
2640 PRINT" ***** WIDTH MUST BE SET TO 110 CHARCTERS ON THE PRINTER *****"
2660 GOSUB 5760
2680 INPUT"ALIGN PAPER TO THE TOP OF PAGE - THEN PRESS ENTER";F9$
2700 REM - PRINT HEADING
2720 LPRINT CHR$(31);"SORTER - ";DATE$
2740 LPRINT"*****************************************************"
2760 LPRINT"FOR THE '";G2$;"' SORT THE KEYS ARE:"
2780 FOR I=1 TO X
2800 LPRINT L$(Y(I));" = ";K$(I)
2820 NEXT I
2840 LPRINT CHR$(30):LPRINT
2860 R=1
2880 CLS
2900 PRINT @(10,1),"NOW READING RECORD NUMBER"
2920 FOR I=1 TO LOF(1)
2940 GET 1,I
2960 GET 2,I
2980 PRINT @(10,27),I
3000 GOSUB 4700
3020 FOR J=1 TO X
3040 IF E1(J)=0 THEN 3160'IF NOT PARTIAL SKIP OVER
3060 IF P3=1 THEN 3120  'AND SORT ?
3080 IF MID$(F$(Y(J)),E1(J),E2(J))=K$(J) THEN 3280'DOES PARTIAL KEY MATCH ?
3100 GOTO 3240
3120 IF MID$(F$(Y(J)),E1(J),E2(J))<>K$(J) THEN 3440'PARTIAL KEY NOT MATCH ?
3140 GOTO 3240
3160 IF P3=1 THEN 3220
3180 IF F$(Y(J))=K$(J) THEN 3280'DOES KEY MATCH ?
3200 GOTO 3240
3220 IF F$(Y(J))<>K$(J) THEN 3440'DOES KEY NOT MATCH ?
3240 NEXT J
3260 IF P3=2 THEN 3440          'OR SORT ?
3280 IF G$="N" THEN 3320        'CAN SUMMING BE SKIPPED ?
3300 GOSUB 4140                 'DO THE SUMMING
3320 IF O$="N" THEN 3420        'CAN ALPHABETIZING BE SKIPPED ?
3340 W1(R)=I
3360 W2$(R)=F$(O1)                     'SAVE DATA
3380 R=R+1
3400 GOTO 3440
3420 ON P1 GOSUB 5880,4940,6020,6120
3440 NEXT I
3460 IF O$="Y" THEN 3580        'IS LIST ORDERED ?
3480 IF G$="N" THEN 3540        'CAN SUMMING BE SKIPPED ?
3500 GOSUB 4540
3520 CLOSE 1,2
3540 STOP
3560 REM*******************************************************************
3580 REM - ORDER THE LIST FOUND
3600 PRINT @(10,1),"NOW ORDERING THE LIST OF RECORDS FOUND"
3620 N2=R-1
3640 D=INT(N2/2)
```

Program Listing 3 continues

```
3660 IF D<=0 THEN 3920
3680 L2=N2-D
3700 FOR J=1 TO L2
3720 FOR I=J TO 1 STEP -D
3740 IF W2$(I+D)=>W2$(I) THEN 3820
3760 SWAP W2$(I+D),W2$(I)
3780 SWAP W1(I+D),W1(I)
3800 NEXT I
3820 NEXT J
3840 D=INT(D/2)
3860 GOTO 3660
3880 REM*****************************************************************
3900 REM - PRINT OUT THE ORDERED RECORDS
3920 FOR I=1 TO N2
3940 GET 1,W1(I)
3960 GET 2,W1(I)
3980 GOSUB 4680
4000 ON P1 GOSUB 5880,4940,6020,6120
4020 NEXT I
4040 IF G$="N" THEN 4100
4060 GOSUB 4540
4080 CLOSE 1,2
4100 STOP
4120 REM*****************************************************************
4140 REM - SUM FOR TOTALS
4160 FOR J2=1 TO N
4180 T7#=0:NN#=.01#
4200 N3=9
4220 IF S(J2)=76 THEN N3=5
4240 IF S(J2)=81 THEN N3=4
4260 IF S(J2)=81 THEN NN#=1#
4280 FOR J3=1 TO N3
4300 T1$=MID$(F$(S(J2)),N3+1-J3,1)
4320 FOR J4=0 TO 9
4340 IF S3$(J4)=T1$ THEN 4400
4360 NEXT J4
4380 GOTO 4440
4400 T7#=T7#+NN#*J4
4420 NN#=NN#*10
4440 NEXT J3
4460 T8#(J2)=T8#(J2)+T7#
4480 NEXT J2
4500 RETURN
4520 REM*****************************************************************
4540 REM - PRINT OUT THE TOTALS
4560 LPRINT:LPRINT
4580 FOR I=1 TO N
4600 LPRINT:LPRINT"SUM FOR ";L$(S(I));" = ";T8#(I)
4620 NEXT I
4640 RETURN
4660 REM*****************************************************************
4680 REM - BREAK DOWN RECORD INTO ITS PARTS
4700 T=1
4720 FOR J=1 TO 42
4740 F$(J)=MID$(A1$,T,L1(J))
4760 T=T+L1(J)
4780 NEXT J
4800 T=1
4820 FOR J=43 TO NUM
4840 F$(J)=MID$(A2$,T,L1(J))
4860 T=T+L1(J)
4880 NEXT J
4900 RETURN
4920 REM*****************************************************************
4940 REM - PRINT ENTIRE TRANSMITTAL FORM
4960 LPRINT"61010 ";F$(1);SPACE$(9);F$(2);F$(3);" ";F$(4);" ";F$(5);" ";F$(6)
4980 GOSUB 5660
5000 LPRINT"DIST  EMP #          POSITION TITLE      CL BG DP NAME"
5020 LPRINT:LPRINT F$(7);" ";F$(8);" ";F$(9);" ";F$(10);" ";F$(11)
5040 GOSUB 5660
5060 LPRINT"SOCIAL S. #  ADDRESS                  CITY-STATE      ZIP   PHONE"
5080 LPRINT:LPRINT F$(12);" ";F$(13);"  ";F$(14);" ";F$(15);" ";F$(16);" ";F$(17);" ";F$(18);"  ";F$(19);"  ";

$(20);"  ";F$(21);"
  ";F$(22)
5100 GOSUB 5660
5120 LPRINT"EMP.DATE TER.DATE RE RE. DATE BIR.DATE YR.S  CZ ET MS SX  BANK ACC.#"
5140 LPRINT:LPRINT F$(23);"   ";F$(24);"     ";F$(25);"     ";F$(26);"    ";F$(27);" ";F$(28);" ";F$(29);" ";F

(30);" ";F$(31);" ";
F$(32);" ";F$(33);" ";F$(34);" ";F$(35);" ";F$(36)
5160 GOSUB 5660
5180 LPRINT"EXPER.            CERTIFICATES            DG HR IN MJ MJ MN MN MN ASMT L"
5200 LPRINT:LPRINT F$(37);" ";F$(38);" ";F$(39);" ";F$(40);" ";F$(41);" ";F$(42);" ";F$(43);" ";F$(44);" ";F$(

5)
5220 GOSUB 5660
5240 LPRINT"PROB.EX. TB TEST  LAST IN. NEXT IN. DAYS H/DY TY EX. DATE TOTAL"
5260 LPRINT:LPRINT F$(46);SPACE$(15);F$(47);"  ";F$(48);"  ";F$(49);"  ";F$(50);"  ";F$(51);"   ";F$(52);"   "

F$(53);"   ";F$(54)
5280 GOSUB 5660
5300 LPRINT"PAYS               FED.EXM. ST. EXM. LOC.EXM. R.ID# FICA AP STMT IN.SUB."
5320 LPRINT:LPRINT SPACE$(40);" RES:";F$(75);" CO-CUR:";F$(76);" LONG:";F$(81)
5340 LPRINT:LPRINT F$(55);"  ";F$(56);"   ";F$(57);" ";F$(58);" ";F$(59);"  ";F$(77)
5360 GOSUB 5660
5380 LPRINT"S/L CODE RATE     CONTRACT    ACCOUNT                  LIFE"
5400 LPRINT:LPRINT F$(60);"  ";F$(61);"   ";F$(62);" ";F$(63);" ";F$(64);"  ";F$(78)
```

Program Listing 3 continued

INTRODUCING THE MEMORY BOX
MEMORY THAT NEVER NEEDS PROTECTION

Plugs directly into the CPU or expansion interface.
Includes its own regulated power supply and ribbon cable.
1024 bytes of ram in an unused area on the memory map (3000-33FF HEX).
Solves memory allocation problems when using monitor programs and utility support routines.
Programs stay in the memory box even when the CPU is turned off.

For Model I Level II and all DOS systems (4, 16, 32, 48K)
Price $49.95

Two utility software programs written for the Memory Box are:

1. Micmon–A powerful machine language monitor (available at a keystroke without disturbing main programs).
Price $19.95

2. U3000–A utility including cassette level meter program, adjustable keyboard debounce, screen print driver, page numbering, smart line termination, and more (all available at a keystroke).
Price $12.95
Special offer complete package............... $69.95

DISPLAYED VIDEO IS DRIVING DOWN PRICES ON DISKS

DISPLAYED VIDEO is now offering **TRS-80* MODEL III** with disk drives, one of the most reliable disk drive systems on the market, for incredibly low prices!!

MODEL III with 48K ram, dual 80 track double-density disk drives, complete, *PLUG IT IN AND GO* **$2,275**

MODEL III with 48K ram, dual 40 track double-density disk drives, complete, *PLUG IT IN AND GO* **$1,875**

Disk Drives: Includes case and power supply
- 40 track disk drive **$289**
- 80 track disk drive **$395**
- 80/80 track disk drive.............................. **$520**

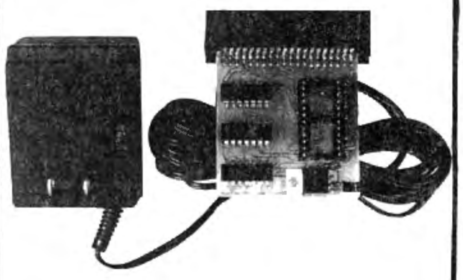
Disk Drives: Not including case and power supply
- 40 track (bare) disk drive **$250**
- 80 track (bare) disk drive **$345**
- 80/80 track (bare) disk drive....................... **$450**

Drive Cases and Power Supplies:
- 5½ inch single drive unit and power supply .... **$ 80**
- 5½ inch dual drive unit and power supply.. **$110**
- 8 inch single drive unit and power supply **$140**

16K RAM Memory Chip (200 NS).. **$1.95 each**

TRS-80* MODEL III 16K **$839**

MODEL III Internal Drive Kit: Includes controller board, dual drive mounting bracket, dual power supply, all hardware, cables and connectors **$410**



THE 2K HIDEAWAY EPROM BOARD

Accepts 2716 compatible memory chips (not included).
Includes its own power supply and connector.
Install a pre-programed eprom or use a ram chip in combination with the powerful monitor and utility programs sold above "Micmon" or "U3000" (not included).
2K Hideaway $49.95

THE EPROM PROGRAMER

Attaches to cassette interface on Model I or III.
Programs 5 volt 1K, 2K, and 4K eproms.
Includes driver program, instructions, cable, and power supply.
Programmer........ $69.95

U.V. Eprom Eraser Lamp $49.95
2716 Eprom Chip............................. $ 8.95
2K Ram Chip $29.95

IMMEDIATE DELIVERY
90 Days Parts and Labor Warranty
Dealer inquiries invited

DISPLAYED VIDEO

Displayed Video
7538 Jackson Road
Ann Arbor, Michigan 48103
(313) 426-5086

✓30

To Order: Call (313) 426-5086 or (313) 482-4424 or write DISPLAYED VIDEO, 7538 Jackson Road, Ann Arbor, Michigan 48103

*TRS-80 is a trademark of the Tandy Corporation

Prices subject to change without notice.

DISPLAYED VIDEO'S CERTIFIED DATA CASSETTES

Made of top grade tape formulated by Agfa-Gevaert of Germany.
Cases are of superior grade five-screw construction.
Guaranteed 100% error-free or your money back.

Length	C-10	C-20
12 pack	.68	.88 ea.
24 pack	.58	.78 ea.

Norelco style hard plastic cassette cases .21 ea.
Special offer with every order of 24 cassettes: a Pilot Razor Point pen will be included.

DISKETTES (BOX OF 10)
Verbatim certified diskettes
Guaranteed 100% error-free or your money back.
5¼ inch MD525-01
10 pack $26.90
100 pack $25.90

Add $1.50 for shipping & handling. To Order: Send check or money order or call (313) 426-5086 or (313) 482-4424.

Program Listing 3 continued

```
5420 GOSUB 5660
5440 LPRINT"S/L CODE RATE     CONTRACT    ACCOUNT                    DENTAL"
5460 LPRINT:LPRINT F$(65);"  ";F$(66);"    ";F$(67);" ";F$(68);" ";F$(69);"   ";F$(79)
5480 GOSUB 5660
5500 LPRINT"S/L CODE RATE     CONTRACT    ACCOUNT                    VISION"
5520 LPRINT:LPRINT F$(70);"  ";F$(71);"    ";F$(72);" ";F$(73);" ";F$(74);"   ";F$(80)
5540 GOSUB 5660
5560 LPRINT"S/L CODE RATE     CONTRACT    ACCOUNT                    HEALTH"
5580 LPRINT:LPRINT:LPRINT
5600 RETURN
5620 REM*********************************************************************
5640 REM - DRAW LINES OF STARS AND DASHES
5660 FOR D7=1 TO 70
5680 LPRINT"*";
5700 NEXT D7
5720 LPRINT
5740 RETURN
5760 FOR D7=1 TO 70
5780 PRINT"-";
5800 NEXT D7
5820 PRINT
5840 RETURN
5860 REM*********************************************************************
5880 REM - PRINT OUT MINIMAL DATA
5900 LPRINT F$(1);" ";F$(2);" ";F$(3);" ";F$(4);" ";F$(6);" ";F$(45);" ";F$(58);" ";F$(59)
5920 LPRINT TAB(69);F$(63);" ";F$(64)
5940 LPRINT TAB(69);F$(68);" ";F$(69)
5960 LPRINT TAB(69);F$(73);" ";F$(74)
5980 RETURN
6000 REM*********************************************************************
6020 REM - PRINT A MAILING LABEL
6040 LPRINT F$(6):LPRINT F$(8):LPRINT F$(9);"  ";F$(10)
6060 LPRINT:LPRINT:LPRINT
6080 RETURN
```

Program Listing 4

```
100 REM - TMATRIX - This program reads the entire personnel file and creates a
120 REM             matrix that represents the number of teachers on each step
140 REM             and level of the salary schedule for the next year. The
160 REM             matrix is stored in a file to be used by a program to do
180 REM             salary schedule modeling.
200 REM             Written by Theron Wierenga during March of 1980.
220 CLEAR 2000
240 DIM A(55,5),S$(55),Y(55),Z(50),E1(50),E2(50),K$(50),B(55,5)
260 INPUT"ENTER THE FILE NAME FOR THE MATRIX";FIL$
280 OPEN "D",1,"P1",254
300 FIELD 1,254 AS A1$
320 OPEN "D",2,"P2",253
340 FIELD 2,253 AS A2$
360 OPEN "D",3,FIL$,20
380 FIELD 3,4 AS T$(1),4 AS T$(2),4 AS T$(3),4 AS T$(4),4 AS T$(5)
400 OPEN "D",4,FIL$+"L",20
420 FIELD 4,4 AS T2$(1),4 AS T2$(2),4 AS T2$(3),4 AS T2$(4),4 AS T2$(5)
440 FOR I=1 TO 52
460 READ Y(I),S$(I)
480 DATA 1,A,1.5,B,2,C,2.5,D,3,E,3.5,F,4,G,4.5,H,5,I,5.5,J,6,K,6.5,L,7,M
500 DATA 7.5,N,8,O,8.5,P,9,Q,9.5,R,10,S,10.5,T,11,U,11.5,V,12,W,12.5,X
520 DATA 13,Y,14,Z,15,a,16,b,17,c,18,d,19,e,20,f,21,g,22,h,23,i,24,j,25
540 DATA k,26,l,27,m,28,n,29,o,30,p,31,q,32,r,33,s,34,t,35,u,36,v,37,w,38
560 DATA x,39,y,40,z
580 NEXT I
600 FOR I=0 TO 9
620 READ L$(I)
640 DATA0,1,2,3,4,5,6,7,8,9
660 NEXT I
680 INPUT"ENTER CLASS WANTED FOR THE SEARCH";CL$
700 INPUT"IS THIS AN 'AND' OR 'OR' SORT";AN$
720 IF AN$="AND" OR AN$="OR" THEN 760
740 GOTO 700
760 INPUT"HOW MANY KEYS FOR THE ACCOUNT CODES";NK
780 PRINT"ENTER ITEM NUMBER, POSITION OF 1ST CHARACTER, NUMBER OF CHARACTERS"
800 PRINT"AND THE EXACT KEY FOR EACH ACCOUNT NUMBER DESIRED"
820 FOR I=1 TO NK
840 K$(I)=".→."
860 INPUT Z(I),E1(I),E2(I),K$(I)
880 IF K$(I)<>".→." THEN 940
900 PRINT"BAD ENTRY - TRY AGAIN"
920 GOTO 860
940 IF Z(I)=59 OR Z(I)=64 OR Z(I)=69 OR Z(I)=74 THEN 980
960 GOTO 900
980 NEXT I
1000 REM********************************************************************
1020 REM - START OF MAIN SEARCH LOOP
1040 CLS
1060 PRINT @(10,1),"NOW READING RECORD NUMBER"
1080 FOR R=1 TO LOF(1)
1100 GET 1,R:GET 2,R
1120 PRINT @(10,27),R
1140 IF MID$(A1$,26,1)<>CL$ THEN 1760
1160 FOR I=1 TO NK
1180 Z2=(Z(I)-54)/5
1200 Z3=77+(45*(Z2-1))+E1(I)
```

Program Listing 4 continues

See List of Advertisers on page 402

Program continued

```
1220 IF AN$="AND" THEN 1280
1240 IF MID$(A2$,Z3,E2(I))=K$(I) THEN 1340
1260 GOTO 1300
1280 IF MID$(A2$,Z3,E2(I))<>K$(I) THEN 1760
1300 NEXT I
1320 IF AN$="OR" THEN 1760
1340 FOR I=1 TO 52
1360 IF S$(I)<>MID$(A2$,57,1) THEN 1740
1380 FOR J=1 TO 5
1400 IF L$(J)<>MID$(A2$,59,1) THEN 1720
1420 T7#=0:NN#=.01#
1440 FOR K=5 TO 1 STEP -1
1460 H$=MID$(A1$,249+K,1)
1480 FOR L=0 TO 9
1500 IF L$(L)=H$ THEN 1560
1520 NEXT L
1540 GOTO 1600
1560 T7#=T7#+NN#*L
1580 NN#=NN#*10
1600 NEXT K
1620 M=T7#/8
1640 A(I,J)=A(I,J)+M
1660 IF I<27 THEN 1760
1680 B(I,J)=B(I,J)+M
1700 GOTO 1760
1720 NEXT J
1740 NEXT I
1760 NEXT R
1780 REM***********************************************************************
1800 REM - WRITE MATRIX TO DISC
1820 FOR R=1 TO 52
1840 FOR I=1 TO 5
1860 LSET T$(I)=MKS$(A(R,I))
1880 LSET T2$(I)=MKS$(B(R,I))
1900 NEXT I
1920 PUT 3,R
1940 PUT 4,R
1960 NEXT R
1980 REM - PRINT OUT THE MATRIX
2000 LPRINT CHR$(31);"EMPLOYEE DISTRIBUTION MATRIX - ";FIL$;"  ";DATE$;CHR$(30):LPRINT
2020 LPRINT TAB(10);"BA";TAB(20);"BA+20";TAB(30);"MA";TAB(40);"MA+30";TAB(50);"MA+60":LPRINT
2040 J=13
2060 FOR R=52 TO 1 STEP -1
2080 IF R<26 THEN 2140
2100 LPRINT R-12;"YRS.";
2120 GOTO 2180
2140 LPRINT J;"YRS.";
2160 J=J-.5
2180 GET 3,R
2200 FOR I=1 TO 5
2220 LPRINT TAB(I*10);CVS(T$(I));
2240 NEXT I
2260 LPRINT
2280 NEXT R
2300 LPRINT:LPRINT CHR$(31);"LONGEVITY MATRIX";CHR$(30):LPRINT
2320 FOR R=52 TO 27 STEP -1
2340 LPRINT R-12;"YRS.";
2360 GET 4,R
2380 FOR I=1 TO 5
2400 LPRINT TAB(I*10);CVS(T2$(I));
2420 NEXT I
2440 LPRINT
2460 NEXT R
2480 REM**********END OF PROGRAM**********
```

Program Listing 5

```
100 REM - TEACHER - This program does salary schedule modeling for teachers.
120 REM             Total costs are computed for various increases in teacher
140 REM             salaries.
160 REM             Written by Theron Wierenga during March 1980.
180 DIM M(13,5),S(55,5),N(55,5),N2(55,5)
200 INPUT"ENTER THE FILE NAME OF THE MATRIX YOU WISH TO USE";FIL$
220 OPEN "D",3,FIL$,20
240 FIELD 3,4 AS T$(1),4 AS T$(2),4 AS T$(3),4 AS T$(4),4 AS T$(5)
260 OPEN "D",4,FIL$+"L",20
280 FIELD 4,4 AS T2$(1),4 AS T2$(2),4 AS T2$(3),4 AS T2$(4),4 AS T2$(5)
300 LPRINT
320 REM***********************************************************************
340 REM - READ THE STEP MULTIPLIERS
360 FOR I=13 TO 1 STEP -1
380 FOR J=1 TO 5
400 READ M(I,J)
420 DATA 163,171,179,182,184,163,164,171,173,175,155,157,164,166,168
440 DATA 148,150,157,159,161,141,143,150,152,154,135,137,143,145,147
460 DATA 129,131,136,138,140,123,125,130,132,134,118,120,124,126,128
480 DATA 113,115,119,121,123,108,110,114,116,118,104,106,110,112,114
500 DATA 100,102,106,108,110
520 NEXT J
540 NEXT I
560 REM***********************************************************************
580 REM - COMPUTE NEW SALARY SCHEDULE FOR THE INCREASE
```

Program Listing 5 continues

Program Listing 5 continued

```
600 PRINT"ENTER CURRENT BASE AND PERCENT INCREASE"
620 PRINT"EXAMPLE: 12777,7.5"
640 INPUT B,K
660 PRINT"IF PROJECTIONS ARE TO BE ON NEXT YEARS MATRIX ENTER A 1, IF ON THIS"
680 PRINT"YEARS ENTER A 0 (ZERO)";
700 INPUT X9
720 IF X9=0 OR X9=1 THEN 740ELSE 660
740 LPRINT CHR$(31);"SALARY SCHEDULE FOR A";K;"% INCREASE"
760 LPRINT"ON A";B;"DOLLAR BASE USING MATRIX ";FIL$;CHR$(30):LPRINT
780 REM - CALCULATE NEW BASE
800 B2=INT(B*(1+K/100)+.5)
820 LPRINT TAB(16);"AB";TAB(30);"AB+20";TAB(44);"MA";TAB(58);"MA+30";TAB(72);"MA+60"
840 GOSUB 2320
860 FOR I=40 TO 14 STEP -1
880 LPRINT I;"YRS.";
900 FOR J=1 TO 5
920 S(I+12,J)=INT(B2*(M(13,J)/100)+.5)+INT((I-10)/5)*300
940 IF I<>40 THEN 980
960 S(I+12,J)=S(I+12,J)-300
980 LPRINT TAB(14*J+1);S(I+12,J);
1000 NEXT J
1020 LPRINT
1040 NEXT I
1060 N=25
1080 FOR I=13 TO 1 STEP -.5
1100 LPRINT I;"YRS.";
1120 FOR J=1 TO 5
1140 Z=INT(I)
1160 IF I=INT(I) THEN 1260
1180 X=(M(Z,J)+M(Z+1,J))/2
1200 S(N,J)=INT(B2*(X/100)+.5)
1220 LPRINT TAB(14*J-4);"      ";S(N,J);
1240 GOTO 1300
1260 S(N,J)=INT(B2*(M(Z,J)/100)+.5)
1280 LPRINT TAB(14*J-4);M(Z,J);S(N,J);
1300 NEXT J
1320 LPRINT
1340 N=N-1
1360 NEXT I
1380 LPRINT:LPRINT:LPRINT
1400 REM*********************************************************************
1420 REM - READ MATRIX SHOWING EMPLOYEE DISTRIBUTION
1440 FOR R=1 TO 52
1460 GET 3,R
1480 GET 4,R
1500 FOR I=1 TO 5
1520 N(R,I)=CVS(T$(I))
1540 N2(R,I)=CVS(T2$(I))
1560 NEXT I
1580 NEXT R
1600 REM - COMPUTE PRODUCT MATRIX
1620 C2=0:I=13
1640 LPRINT CHR$(31);"EMPLOYEE SALARY DISTRIBUTION"
1642 LPRINT "WITH A";X9;"YEAR STEP INCREASE TO CURRENT FILE";CHR$(30):LPRINT
1660 GOSUB 2420
1680 FOR R=52 TO 1 STEP -1
1700 IF R<26 THEN 1760
1720 LPRINT R-12;"YRS.";
1740 GOTO 1800
1760 LPRINT I;"YRS.";
1780 I=I-.5
1800 C1=0
1820 FOR J=1 TO 5
1840 T9=S(R,J)*N(R-X9,J)
1860 LPRINT TAB(12*J);T9;
1880 C1=C1+T9
1900 NEXT J
1920 LPRINT TAB(73);"/";C1
1940 C2=C2+C1
1960 NEXT R
1980 LPRINT:LPRINT"TOTAL OF ALL TEACHER SALARIES =";C2:LPRINT
2000 LPRINT CHR$(31);"LONGEVITY DISTRIBUTION";CHR$(30):LPRINT
2020 GOSUB 2420
2040 C2=0
2060 FOR R=52 TO 26 STEP -1
2080 LPRINT R-12;"YRS.";
2100 C1=0
2120 FOR J=1 TO 5
2140 T9=N2(R-X9,J)*(INT((R-22)/5)*300)
2160 LPRINT TAB(12*J);T9;
2180 C1=C1+T9
2200 NEXT J
2220 LPRINT TAB(73);"/";C1
2240 C2=C2+C1
2260 NEXT R
2280 LPRINT:LPRINT"TOTAL OF ALL LONGEVITY =";C2
2300 STOP
2320 FOR II=1 TO 90
2340 LPRINT"*";
2360 NEXT II
2380 LPRINT
2400 RETURN
2420 LPRINT TAB(13);"AB";TAB(25);"AB+20";TAB(37);"MA";TAB(49);"MA+30";TAB(61);
2440 LPRINT "MA+60";TAB(75);"TOTAL"
2460 GOSUB 2320
2480 RETURN
2500 REM**********END OF PROGRAM**********
```

Computerize your rental records before the first of the month.

The Tenant Tracker

George Kwascha
8007 Mahogany Drive
Charlotte, NC 28212

Tenant	Address	Month	Rent	Date Rec.	Amt. Rec.	Rec. #	Code
Smith Jonathan	1800 Century Av	01-80	250.00	01-02-80	250.00	1234	T
Smith Jonathan	1800 Century Av	12-80	265.00	12-04-80	265.00	1400	T
Smith Jonathan	1800 Century Av	11-80	265.00	11-14-80	275.00	1399	L
Smith Jonathan	1800 Century Av	02-80	250.00	02-01-80	250.00	1245	T
Smith Jonathan	1800 Century Av	03-80	250.00	03-03-80	250.00	1250	T
Smith Jonathan	1800 Century Av	05-80	250.00	05-04-80	250.00	1300	T
Smith Jonathan	1800 Century Av	04-80	250.00	04-01-80	250.00	1405	T
Smith Jonathan	1800 Century Av	06-80	265.00	06-11-80	265.00	1470	N
Smith Jonathan	1800 Century Av	07-80	265.00	07-02-80	265.00	1500	T
Smith Jonathan	1800 Century Av	10-80	265.00	10-01-80	265.00	1801	T
Smith Jonathan	1800 Century Av	09-80	265.00	09-02-80	265.00	1659	T
Smith Jonathan	1800 Century Av	08-80	265.00	08-15-80	275.00	1645	LB
					$3125.00		

Fig. 1. Rental Record Input File Dump.

Month	Tenant Name	Rent	Date Rec.	Amt. Rec.	Rec. #	Code
1-80	Smith Jonathan	$250.00	01-02-80	$250.00	1234	T
2-80	Smith Jonathan	$250.00	02-01-80	$250.00	1245	T
3-80	Smith Jonathan	$250.00	03-03-80	$250.00	1250	T
4-80	Smith Jonathan	$250.00	04-01-80	$250.00	1405	T
5-80	Smith Jonathan	$250.00	05-04-80	$250.00	1300	T
6-80	Smith Jonathan	$265.00	06-11-80	$265.00	1470	N
7-80	Smith Jonathan	$265.00	07-02-80	$265.00	1500	T
8-80	Smith Jonathan	$265.00	08-15-80	$275.00	1645	LB
9-80	Smith Jonathan	$265.00	09-02-80	$265.00	1659	T
10-80	Smith Jonathan	$265.00	10-01-80	$265.00	1801	T
11-80	Smith Jonathan	$265.00	11-14-80	$275.00	1399	L
12-80	Smith Jonathan	$265.00	12-04-80	$265.00	1400	T
				$3125.00		

Fig. 2. Rental Record History for 1980

If you need a method of computerizing records for rental property that you may be managing, I have written the following program for a local real estate management firm. The program is designed to keep a record of every rental unit's monthly rent payments for one year. Admittedly, it is a simple program that can be altered to suit your particular needs. This modified version will run on a TRS-80, Level II 16K with cassette.

How the Program Works

After CLOADing and typing Run, the program prints out a heading on the screen and, after a brief delay, will display a menu of options. Option 1 allows you to enter initial rental information. The program prompts for the following fields:

- Tenant name—Enter last name first, then first name (without commas).
- Address—Enter as you wish without commas, but be consistent in format or you may have difficulty retrieving data. In my sample run I used "#### street name."
- Rental month—Enter two digits for the months, dash, two digits for the year (MM-YY).
- Date payment received—This, of

Variable	Program Function
M$( ),L$( )	Rental Month (MM-YY)
D$( ),O$( )	Date payment was received (MM-DD-YY)
A!( ),P!( )	Rental amount ($$$.00)
R( ),T( )	Receipt number or other control number
A$( ),J$( )	Address of rental property (no commas)
N$( ),K$( )	Name of tenant (no commas)
B!( ),S!( )	Amount received ($$$.00)
BB,SS	Totaled amount received
C$( ),U$( )	Code to describe status of payment
	Codes used in this program:
	T - payment in time
	L - late paymt with late charge
	N - late paymt w/o late chge
	B - bounced check
	D - deposit forfeiture
	O - other

Table 1. Variable Listing

"The program is designed to keep a record of every rental unit's monthly rent payments for one year."

course, is very important to any real estate management firm. Use format "MM-DD-YY."

- Rental amount—Enter rent due as a dollar amount (ex:200.00).
- Amount received—Enter same as above.
- Receipt number—or some other control number, if one is used by your firm.
- Code—Refer to Table 1 for list of codes used in the sample run. You may wish to add your own.

Type Enter after each field entry and End in the tenant name field to close the file and display the option menu.

Option 2 allows you to change data in a file in case an error exists. The program prompts for the tenant name and rental month. Only the first five characters of the name need be entered. However, the entire rental month field (MM-YY) must be entered exactly. When a match occurs in the file, the program will prompt each field once again for the correct data. You need only enter data in the field that needs correcting. In all other fields, type Enter to maintain already existing data. Afterwards, the program will ask whether you want to enter another correction; indicate either yes or no. The Option menu will then be displayed.

Option 3 allows you to inspect the entire file in order of entry. The program will display two months of data per screen. Type Enter to continue scrolling.

Option 4 also permits you to display the file except you must specify the tenant's name. Once again, only five characters of the name are needed.

In Option 5 by specifying the address the file is displayed. You must enter at least nine characters of the address to obtain a match. In my sample run I used "1800 Cent."

Option 6 enables you to save your data on cassette tape. Type 6, press record-play on your recorder and hit Enter. The program will prompt you for a menu when it has completed saving the data on tape.

Option 7 allows you to input data from cassette. Type 7, press play on your recorder and hit Enter. The program will prompt you for a menu when it has completed loading.

In Option 8 typing 8 will display the sorting options available.

In Option 9 typing 9 will terminate the program. It also erases all data that you may have in memory, so be sure to save it on tape before ending the program.

Output Reports and Sorting Options

Requesting Option 8 in the above section will display another menu of sorting options. It needs to be pointed out that Options 2,4,5 and 6 are not included in this program listing. Output reports in the area of real estate management depend on each firm's needs and must be written as such. I have included two output reports as examples. For example:

- Option 1 will print a hard copy report of the entire file. See Fig. 1 for my sample run. This form of output report is not very practical. It is strictly designed as a control report.
- Option 3 will print a hard copy annual report of each requested address. This option will also sort the data in descending numerical order by month. See Fig. 2 for my sample run. The program will first prompt you for an address. Enter the full address. The program then will prompt you for "# line feeds to heading." Enter the number of lines you wish skipped before the heading is printed.

Lines	Program Function
10-20	Initialize program and print heading
25-30	Dimension variables
35-80	Display available options
85-95	Direct program to chosen option
100-155	Option 1 - enter payment to file
160-195	Option 2 - make corrections to file
200-220	Option 3 - display entire file
225-260	Option 4 - display individual tenant file
265-300	Option 5 - display address file
305-330	Option 6 - save current file on tape
335-360	Option 7 - input existing file from tape
365-400	Option 8 - list sorting options for output reports
405-415	Direct program to chosen sort option
420-455	Sort Option 1 - Print dump of entire file
460	Sort Option 2 - (unavailable in this listing)
465-532	Sort Option 3 - Annual report sorted by address
535	Sort Option 4 - (unavailable in this listing)
540	Sort Option 5 - (unavailable in this listing)
545	Sort Option 6 - (unavailable in this listing)
550-555	Terminates program
560-565	Subroutines - formatted print for output in Options 3,4,5
570-610	Subroutine - sorts data in descending order, rental month
615-620	Subroutine - used in options 2,4,5

Table 2. Program Summary

Program Listing

```
10 CLEAR2000:CLS:PRINTSTRING$(64,191);STRING$(64,143)
15 PRINT@142,"* * * RENTAL RECORD PROGRAM * * *":PRINTSTRING$(64
,188);STRING$(64,191)
20 FORXX=1TO999:NEXT
25 DIMM$(50),D$(50),A!(50),R(50),C$(50),A$(50),N$(50),B!(50),L(5
0)
30 DIMJ$(50),K$(50),L$(50),O$(50),P!(50),S!(50),T(50),U$(50)
35 CLS:PRINT@10,"* * OPTIONS AVAILABLE * *":PRINT:PRINT:YY=0
40 PRINT"TO ENTER PAYMENT TO FILE............TYPE 1"
45 PRINT"TO MAKE CORRECTIONS TO FILE.........TYPE 2"
50 PRINT"TO DISPLAY ENTIRE FILE..............TYPE 3"
55 PRINT"TO DISPLAY INDIVIDUAL TENANT FILE...TYPE 4"
60 PRINT"TO DISPLAY ADDRESS FILE.............TYPE 5"
65 PRINT"TO SAVE CURRENT FILE ON TAPE........TYPE 6"
70 PRINT"TO INPUT EXISTING FILE FROM TAPE....TYPE 7"
75 PRINT"TO LIST PROGRAM SORTING OPTIONS.....TYPE 8"
80 PRINT"TO END PROGRAM......................TYPE 9"
85 Q$=INKEY$:IFQ$=""THEN85ELSE90
90 W=VAL(Q$):ONWGOTO100,160,200,225,265,305,335,365,550
95 PRINT"INCORRECT ENTRY...TRY AGAIN":GOTO85
100 CLS:PRINT"TO CLOSE FILE,TYPE...'END' IN NAME FIELD":P1=P1+1
105 FOR I=P1 TO 50:PRINT"ENTER TENANT NAME (LAST FIRST)"
110 INPUTN$(I):IF N$(I)="END" THEN P1=I-1:GOTO155
115 INPUT"ENTER TENANT ADDRESS(#### STREET)";A$(I)
120 INPUT"ENTER RENTAL MONTH (MM-YY)";M$(I)
125 INPUT"ENTER DATE PAYMENT RECEIVED (MM-DD-YY)";D$(I)
130 INPUT"ENTER RENTAL AMOUNT ($$$.00)";A!(I)
```

Program continues

INTERFACE EXPANSION BOARD

Program continued

```
135 INPUT"ENTER AMOUNT RECEIVED ($$$.00)";B!(I)
140 INPUT"ENTER RECEIPT NUMBER (####)";R(I)
145 PRINT"ENTER CODE...T-ON TIME":PRINTTAB(13)"L-LATE W/CHG":PRI
NTTAB(13)"N-LATE W/O CHG":PRINTTAB(13)"B-BOUNCED CHECK"
150 PRINTTAB(13)"D-DEPOSIT FORFEITURE":PRINTTAB(13)"O-OTHER":INP
UTC$(I):NEXT
155 CLS:PRINT@468,"* * * FILE CLOSED * * *":FORXX=1TO999:NEXT:GO
TO35
160 CLS:INPUT"ENTER NAME (LAST FIRST)";N$:INPUT"ENTER RENTAL MON
TH(MM-YY)";M$
165 FORI=1TOP1:IFLEFT$(N$(I),5)=LEFT$(N$,5)ANDM$=M$(I)GOTO175
170 NEXT:PRINT"NAME NOT IN FILE":GOTO195
175 PRINT:PRINT"RE-ENTER INFO WITH CORRECTIONS":INPUT"ENTER NAME
";N$(I):INPUT"ENTER ADDRESS";A$(I)
180 INPUT"ENTER RENTAL MONTH(MM-YY)";M$(I):INPUT"ENTER DATE PAYM
T RECEIVED(MM-DD-YY)";D$(I)
185 INPUT"ENTER RENT AMOUNT";A!(I):INPUT"ENTER AMOUNT REC";B!(I)

190 INPUT"ENTER RECEIPT #";R(I):INPUT"ENTER CODE: ";C$(I)
195 N$="":M$="":GOSUB615:IFX$="Y"THEN160ELSE35
200 CLS:FOR I=1TOP1:GOSUB560:YY=YY+1
205 IFYY<>2THEN215
210 YY=0:PRINT@960,"TYPE ENTER TO CONTINUE LISTING";:INPUT
215 NEXT
220 INPUT"TYPE ENTER TO DISPLAY OPTIONS";X:GOTO35
225 CLS:INPUT"ENTER TENANT NAME (LAST FIRST)";N$:PRINT
230 FOR I=1TOP1:IFLEFT$(N$(I),5)<>LEFT$(N$,5)THEN250
235 GOSUB560:YY=YY+1
240 IFYY<>2THEN250
245 YY=0:PRINT@960,"TYPE ENTER TO CONTINUE LISTING";:INPUT
250 NEXT
255 PRINT:PRINT"END OF FILE SEARCH...":PRINT:YY=0
260 N$="":GOSUB615:IFX$="Y"THEN225ELSE35
265 CLS:INPUT"ENTER RENTED ADDRESS (#### STREET)";A$:PRINT
270 FOR I=1TOP1:IFLEFT$(A$(I),9)<>LEFT$(A$,9)THEN290
275 GOSUB560:YY=YY+1
280 IFYY<>2THEN290
285 YY=0:PRINT@960,"TYPE ENTER TO CONTINUE LISTING";:INPUT
290 NEXT
295 PRINT:PRINT"END OF FILE SEARCH...":PRINT:YY=0
300 A$="":GOSUB615:IFX$="Y"THEN265ELSE35
305 CLS:INPUT"PREPARE CASSETTE TO SAVE DATA FILE,WHEN READY TYPE
 ENTER";X
310 CLS:PRINT@460,"* * * COPYING FILE ONTO TAPE * * *"
315 PRINT #-1,P1
320 FOR I=1TOP1:PRINT #-1,N$(I),A$(I),M$(I),D$(I),A!(I),B!(I),R(
I),C$(I):NEXT
325 CLS:PRINT@455,"...BURP!..* COPYING COMPLETE *...NOTE TAPE LO
CATION"
330 PRINT@896,"TYPE ENTER TO LIST OPTIONS";:INPUTX:GOTO35
335 CLS:INPUT"PREPARE CASSETTE FOR DATA INPUT...WHEN READY TYPE
ENTER";X
340 CLS:PRINT@460,"* * * COPYING FILE FROM TAPE * * *"
345 INPUT #-1,P1
350 FOR I=1TOP1:INPUT #-1,N$(I),A$(I),M$(I),D$(I),A!(I),B!(I),R(
I),C$(I):NEXT
355 CLS:PRINT@466,"* * * INPUTING COMPLETE * * *"
360 PRINT@896,"TYPE ENTER TO LIST OPTIONS";:INPUT:GOTO35
365 CLS:PRINT@10,"* * SORTING OPTIONS AVAILABLE * *":PRINT:PRINT

370 PRINT"TO PRINT A DUMP OF ENTIRE INPUT FILE............TYPE 1
"
375 PRINT"TO PRINT ANNUAL REPORT SORTED BY RECEIPT #......TYPE 2
"
380 PRINT"TO PRINT ANNUAL REPORT SORTED BY ADDRESS........TYPE 3
"
385 PRINT"TO PRINT ANNUAL REPORT SORTED BY TENANT NAME....TYPE 4
"
390 PRINT"TO PRINT MONTHLY REPORT BY ADDRESS..............TYPE 5
"
395 PRINT"TO PRINT MONTHLY REPORT BY TENANT NAME..........TYPE 6
"
400 PRINT"TO LIST ORIGINAL OPTIONS........................TYPE 7
"
405 Q$=INKEY$:IFQ$=""THEN405ELSE410
410 K=VAL(Q$):ONKGOTO420,460,465,535,540,545,35
415 PRINT"INCORRECT ENTRY...TRY AGAIN":GOTO405
420 BB=0:LPRINTTAB(20)"* * * RENTAL RECORD INPUT FILE DUMP * * *
":LPRINT" "
425 LPRINTTAB(3)"TENANT          ADDRESS          MONTH    RENT  D
ATE REC  AMT REC  REC#  CODE"
430 FORI=1TOP1
```

Program continues

✓86

✓See List of Advertisers on page 402

". . . there is no direct method of deleting a record completely."

Program continued

```
435 LPRINTUSING"   %                    % %                % %   % ###.##
%      %   ###.##  ####  %  %";N$(I),A$(I),M$(I),A!(I),D$(I),B!
(I),R(I),C$(I)
440 BB=BB+B!(I)
445 NEXTI
450 LPRINTTAB(59)STRING$(7,45):LPRINTTAB(58);:LPRINTUSING"$####.
##";BB
455 LPRINTCHR$(27);CHR$(11):GOTO365
460 GOTO535
465 J=0:KK=0:PRINT:INPUT"ENTER ADDRESS OF DESIRED REPORT";A$:Z$=
"-":SS=0:INPUT"ENTER # LINE FEEDS TO HEADING";GK
470 FORI=1TOP1:IFLEFT$(A$,9)<>LEFT$(A$(I),9)THEN480
475 J=J+1:J$(J)=A$(I):K$(J)=N$(I):L$(J)=M$(I):O$(J)=D$(I):P!(J)=
A!(I):S!(J)=B!(I):T(J)=R(I):U$(J)=C$(I):KK=1
480 NEXTI
482 IFKK<>1GOTO532
485 J1=J:GOSUB570:G$=RIGHT$(L$(1),2)
490 FORX=1TOGK:LPRINTCHR$(138):NEXT
495 LPRINTTAB(20)"* * * 19";G$;" RENTAL RECORD HISTORY * * *":LP
RINTTAB(30)A$:LPRINT" "
500 LPRINTTAB(5)"MONTH  TENANT NAME              RENT   DATE REC
 AMT REC   REC#   CODE"
505 FORJ=1TOJ1
510 LPRINTUSING"     ##!%%  %                    % $###.##   %
 %   $###.##    ####    % %";L(J),Z$,G$,K$(J),P!(J),O$(J),S!(J),T(
J),U$(J)
515 SS=SS+S!(J)
520 NEXTJ
```

Program continues

". . . consider changing dimension statement values and the string space value to suit your needs."

- Option 7 will return you to the original options menu.

A Few More Pointers

You may have noted that there is no direct method of deleting a record completely. To do this, I recommend that you use Option 2 and enter at least one blank in all fields.

You will need to consider changing dimension statement values and the string space value to suit your needs. One idea that is somewhat useful is to set up an entire year's file with tenant names, addresses, and rental amounts. Then, throughout the remainder of the year, Option 2 can be used to add data each month as the rent is paid.

Sort Option 2 was originally designed to print an annual report sorted by receipt number. Sort Option 4 printed an annual report sorted by name, much like sort Option 3. Sort Options 5 and 6 printed monthly reports sorted by name and address. As you can see, an output report can take many forms. Only you can determine which report best suits your needs. ■

Program continued

```
525 LPRINTTAB(53)STRING$(7,45):LPRINTTAB(52);
530 LPRINTUSING"$####.##";SS:GOTO365
532 CLS:PRINT@465,"REQUESTED ADDRESS NOT IN FILE":FORXX=1TO999:N
EXT:GOTO365
535 CLS:PRINT@465,"OPTION UNAVAILABLE AT THIS TIME":FORXX=1TO999
:NEXT:GOTO365
540 GOTO535
545 GOTO535
550 PRINTSTRING$(64,191);STRING$(64,143):PRINTTAB(15)"* * * PROG
RAM TERMINATED * * *":PRINTSTRING$(64,188);STRING$(64,191)
555 END
560 PRINTA$(I),N$(I):PRINT"RENTAL MONTH: ";M$(I):PRINT"DATE PAYM
ENT RECEIVED: ";D$(I):PRINT"RENTAL AMOUNT: ";
565 PRINTUSING"###.##";A!(I):PRINT"AMOUNT RECEIVED: ";:PRINTUSIN
G"###.##";B!(I);:PRINT"     RECEIPT# ";R(I);"    CODE: ";C$(I):PR
INT:RETURN
570 FORJ=1TOJ1:L(J)=VAL(LEFT$(L$(J),2)):NEXTJ
575 FORI=1TOJ1-1
580 FORJ=I+1TOJ1
585 IFL(I)<=L(J)THEN605
590 L=L(I):K$=K$(I):O$=O$(I):P=P!(I):S=S!(I):T=T(I):U$=U$(I)
595 L(I)=L(J):K$(I)=K$(J):O$(I)=O$(J):P!(I)=P!(J):S!(I)=S!(J):T(
I)=T(J):U$(I)=U$(J)
600 L(J)=L:K$(J)=K$:O$(J)=O$:P!(J)=P:S!(J)=S:T(J)=T:U$(J)=U$
605 NEXTJ
610 NEXTI:RETURN
615 PRINT"FOR ANOTHER NAME OR ADDRESS TYPE Y-YES OR N-NO"
620 X$=INKEY$:IFX$=""THEN620ELSERETURN
```

To landlord or not to landlord: that is the calculation.

Investment Property Analysis

Leslie E. Sparks
1014 Evergreen Drive
Durham, NC 27712

The analysis of a real estate investment involves some rather tedious calculations. For example, mortgage payment schedules must be calculated in order to find loan costs. You also need to figure present values of all the cash flows from investments in order to calculate their present worth.

You can use tables to help with the calculations, but often the tables don't have the values you need. And if you have several investments to analyze, the use of tables soon becomes more work than you bargained for. The obvious thing to do is to let your TRS-80 do the tedious math for you.

The program presented here is designed to do all the needed calculations to evaluate an investment in rental property. It calculates the taxable and spendable rental income expected from the investment, along with the expected capital gain. The program can handle both first and second mortgages, which can have both different interest rates and different lives. The life of the investment can be less than, greater than or equal to the life of either of the loans.

The program presents a summary of the calculations which shows total rental income (after tax), total spendable income (after tax), capital gain profit, total profit and the total spendable profit from the investment. A hard copy option is also available.

Overview of the Program

First, enter the data for the program from the keyboard by filling in a five-page menu. The data you enter will concern expenses, income, and capital costs of the investment. When the menu is full and you are sure that the data are correct, you can initiate calculations by pressing G.

The program calculates the loan payment schedule for the life of the investment (not the life of the loan). It also does a

Program Listing

```
10 REM ********************************************************
20 REM RENTAL PROPERTY INVESTMENT ANALYSIS BY L. E. SPARKS
30 REM VERSION 2.0 19 JULY 1980 32K TRS-80 DISK BASIC
40 REM ********************************************************
50 CLS
60 CLEAR 1000
70 PRINT
80 PRINT STRING$(63,"*")
90 PRINT TAB(15)"RENTAL PROPERTY INVESTMENT ANALYSIS "
100 PRINT TAB(15)"              BY"
110 PRINT TAB(15)"     L. E. SPARKS VERSION 2.0"
120 PRINT STRING$(63,"*")
130 GOSUB 7060
140 PRINT STRING$(64,"=")
150 PRINT"ENTER PROPERTY IDENTIFICATION INFORMATION."
160 LINEINPUT ZZ$
170 CLS
180 PRINT "INITIALIZATION "
190 DEFINT J,K
200 DEFSTR F
210 DIM F(41),E(20),EF(20),FT(12),R(12),RY(12),NU(12),Z(30),EQ(2
0),EY(20),DS(30)
220 REM P1 INTEREST PAID ON 1ST MORTGAGE P2 PRINCIPAL PAID 1ST
P3 INTEREST PAID 2ND MORTGAGE P4 PRINCIPAL PAID 2ND
230 DIM P1(35),P2(35),P3(35),P4(35)
240 REM TX TAX PAID,RA AFTER TAX INCOME,CF CASH FLOW AFTER TAX
250 DIM RX(30),RF(30),RZ(30),RT(30),RL(30),RI(30),RG(30)
260 DIM TX(30),RA(30),CF(30),DF(30),SF(30)
270 GOSUB 7350
280 GOSUB1150
290 JM = +1
300 CLS
310 GOSUB 3390
320 IF JM = -1 THEN 280
330 GOSUB 3860
340 GOSUB 4690
350 PRINT"DO YOU WANT HARD COPY ?"
360 GOSUB 410
370 IF Y$="Y"THEN GOSUB5630
```

Program continues

Program continued

```
380 PRINT"DO YOU WANT TO DO ANOTHER ANALYSIS ?"
390 GOSUB 410
400 IF Y$="Y" THEN 280 ELSE END
410 REM GET CHARACTER
420 Y$=INKEY$:IF Y$=""THEN 420 ELSE IX=ASC(Y$)
430 RETURN
440 REM NUMERIC INPUT
450 IC = JV
460 IF IC = 0 THEN IC =55
470 V$=Y$
480 IF Y$="$" THEN V$=""
490 IF Y$="," THEN V$=""
500 Z$=Y$
510 REM
520 PRINT@JP*64+IC,Z$;CHR$(140);"  ";
530 GOSUB 410 :'GO GET A CHARACTER
540 IF IX=13 RETURN :'RETURN IF ENTER PRESSED
550 IF IX = 8 THEN 560 ELSE 580
560 IF LEN(V$) = 0 THEN 580
570 IF MID$(Z$,LEN(Z$),1) <> "," THEN V$= LEFT$(V$,LEN(V$)-1)
580 IF IX= 8 THEN IF LEN(Z$)>0 THEN Z$=LEFT$(Z$,LEN(Z$)-1)
590 IF IX = 8 THEN 510
600 IF IX>31 THEN Z$=Z$+Y$
610 IF Y$="," THEN 650
620 IF Y$="$" THEN 650
630 IF Y$="%" THEN 650
640 V$=V$+Y$
650 GOTO510
660 STOP
670 REM INSTRUCTION SUBROUTINE
680 PRINT  " MAY BE USED TO CHANGE PAGES.  ";
690 PRINT CHR$(91);"  ";CHR$(92);" ";CHR$(93);" ";CHR$(94);" ARE
 ACTIVE .             ";
700 PRINT"USE G TO START CALCULATIONS.  ";
710 PRINT"ENTER DATA TO CHANGE";
720 RETURN
730 REM UPDATE INSTRUCTIONS
740 PRINT @ 896,STRING$(63,"*");
750 PRINT  @JZ,STRING$(63," ");
760 PRINT@960,"USE ";CHR$(93); " FOR CORRECTIONS ";
770 PRINT"ENTER REST OF ITEM AND <ENTER> TO CHANGE";
780 RETURN
790 REM POINTER CONTROL
800 PRINT@JP*64+JV,CHR$(94);
810 GOSUB410:'GO GET A CHARACTER
820 PRINT@JP*64+JV,"   ";
830 ID=0
840 JD=0
850 IF IX=27 OR IX=91 OR IX=123THEN ID=-1
860 IF IX=26 OR IX=10 OR IX=13 THEN ID=ID+1
870 IF IX =8 THEN JD =-JA ELSE IF IX=9 THEN JD=+JA
880 JV=JV+JD
890 IF JV>JM THEN JV=JN ELSE IF JV<JN THEN JV=JM
900 IP=IP+ID
910 IF IP<1 THEN IP=MX
920 IF IP>MX THEN IP=1
930 JP=IP+IJ
940 RETURN
950 REM POINTER CONTROL FOR INCOME SUB
960 IF IP = 8 THEN JV = 0 ELSE JV = IC
970 PRINT@JP*64+JV,CHR$(94);
980 GOSUB 410
990 PRINT @ JP*64+JV,"  ";
1000 ID = 0 : JD = 0
1010 IF IX = 27 OR IX = 91 OR IX = 123 THEN ID = -1
1020 IF IX = 26 OR IX = 10 OR IX = 13 THEN ID = +1
1030 IF IX = 8 THEN JD =-14
1040 IF IX = 9 THEN JD = +14
1050 IF IP = MX AND ID = -1 THEN IP = IP -2: GOTO1080
1060 IP = IP + ID
1070 IC = IC + JD
1080 IF IP <1 THEN IP = MX
1090 IF IP = 7 THEN IP = MX:JV = 0
1100 IF IP > MX THEN IP = 1
1110 IF IC >30 THEN IC = 16
1120 IF IC < 16 THEN IC = 30
1130 JP = IP + IJ
1140 RETURN
1150 REM PAGE 1
1160 JP=6
1170 IP=1
1180 IJ =5:JV=28
1190 IF JM = -1 THEN 2710
1200 JN=28:JM=48:JA=20
1210 CLS:PRINT:PRINT STRING$(63,"="): PRINT"* PAGE 1 OF 5 PAGES
* FIXED EXPENSES *"
```

Program continues

✓See List of Advertisers on page 402

Program continued

```
1220 PRINT STRING$(63,"=")
1230 PRINT "      ITEM ";TAB(40);"$/MO              $/YR"
1240 PRINT STRING$(63,"-")
1250 FOR J = 1 TO 6
1260 PRINT" ";
1270   PRINT USING F(J);EF(J);:PRINT USING "      $###,###.##";E
Y(J)
1280 NEXT J
1290 JZ = 832
1300 PRINT@832,"> ";
1310 GOSUB670 : 'PRINT INSTRUCIONS
1320 IC =45:MX=5
1330 GOSUB 790
1340 IF ID<>0 THEN 1330 ELSE ID=1
1350 IF IX<32 THEN 1330
1360 IF Y$="G" THEN RETURN
1370 IF Y$=">" THEN 1580
1380 IF Y$ = "<" THEN 1150
1390 GOSUB730
1400 GOSUB 1480
1410 GOSUB 1540
1420 IP=IP+1
1430 IF IP<1 THEN IP=MX
1440 IF IP>MX THEN IP=1
1450 JP=IP+IJ
1460 CLS
1470 GOTO1210
1480 REM UPDATE INFORMATION
1490 GOSUB 440
1500 IF JV=JM THEN EY(IP)=VAL(V$):EF(IP)=EY(IP)/12:RETURN
1510 EF(IP) = VAL(V$)
1520 EY(IP)=12.*EF(IP)
1530 RETURN
1540 REM CALCULATE TOTAL COSTS
1550 EF(6) = EF(1)+EF(2)+EF(3)+EF(4)+EF(5)
1560 EY(6) = EY(1)+EY(2)+EY(3)+EY(4)+EY(5)
1570 RETURN
1580 REM PAGE 2
1590 IP=1:JP=3:IJ=2
1600 JV=28:JN=28:JM=48:JA=20
1610 CLS
1620 PRINT "** PAGE 2 OF 5 PAGES ** OPERATING COSTS **"
1630 PRINT"ITEM";TAB(33);"$/MO                $/YR"
1640 PRINT STRING$(63,"=")
1650 FOR J= 1 TO 12
1660   PRINT USING F(J+6);E(J);:PRINT USING "      $###,###.##";
EQ(J)
1670 NEXT J
1680 JZ = 896 +64
1690 PRINT @ 896,"< OR >";
1700 MX=11: IC =45
1710 GOSUB670
1720 GOSUB790
1730 IF ID<>0 THEN 1720 ELSE ID=1
1740 IF IX<32 THEN 1720
1750 GOSUB 730
1760 IF Y$="G" THEN RETURN
1770 IF Y$="<" THEN 1150
1780 IF Y$=">" THEN 2000
1790 GOSUB 1860
1800 GOSUB1920
1810 IP=IP+1
1820 IF IP<1 THEN IP=MX
1830 IF IP>MX THEN IP=1
1840 JP = IP + IJ
1850 GOTO 1610
1860 REM UPDATE COSTS
1870 GOSUB 440
1880 IF JV=JM THEN EQ(IP)=VAL(V$):E(IP)=EQ(IP)/12.:RETURN
1890 E(IP)=VAL(V$)
1900 EQ(IP)=E(IP)*12.
1910 RETURN
1920 REM SUM TOTAL EXPANSES
1930 E(12)=0
1940 EQ(12)=0
1950 FOR JJ = 1 TO 11
1960    E(12)=E(12)+E(JJ)
1970    EQ(12) = EQ(12)+EQ(JJ)
1980 NEXT JJ
1990 RETURN
2000 REM INCOME ESTIMATES GO HERE
2010 IJ = 2
2020 MX = 8
2030 IP = 1
2040 IF IP = MX THEN 2590
2050 IC = 16
```

Program continues

Program continued

```
2Ø6Ø JP = 3
2Ø7Ø JV=28:JA=2Ø:JN=28:JM=48
2Ø8Ø CLS
2Ø9Ø IF IP = MX THEN JV = Ø ELSE JV = IC
21ØØ PRINT "*** PAGE 3 OF 5 PAGES *** INCOME ESTIMATES ***"
211Ø PRINT STRING$(63,"=")
212Ø PRINT"TYPE OF UNIT ";TAB(18);"# UNITS";TAB(3Ø);"RENT $/MO";
TAB(5Ø);"RENT $ YR"
213Ø FOR J = 1 TO NT
214Ø   PRINT "    ";
215Ø     PRINT FT(J);TAB(2Ø);NU(J);TAB(3Ø);
216Ø     PRINT USING "$###,###.##";R(J);: PRINT TAB(45);
217Ø     PRINT USING F(Ø);RY(J)
218Ø NEXT J
219Ø PRINT "**** TOTAL ";TAB(2Ø);NS;TAB(3Ø);
22ØØ PRINT USING "$#,######.##";RM;:PRINT TAB(45);
221Ø PRINT USING F(Ø);RY
222Ø PRINT"    ";
223Ø PRINT USING F(41);VA
224Ø PRINT"****";:PRINT USING " DOLLAR COST OF VACANCY   $###,##
#.##/YR";VA*RY/1ØØ.
225Ø RG = RY*(1ØØ.-VA)/1ØØ
226Ø PRINT "****";:PRINTUSING" ADJUSTED GROSS RENTS      $#,######
.##/YR";RG:PRINT STRING$(63,"=")
227Ø PRINT @ 896, "< & > MAY BE USED TO CHANGE PAGES ";
228Ø PRINT CHR$(91);" ";CHR$(92);" ";CHR$(93); " ";CHR$(94);" AR
E ACTIVE";
229Ø PRINT "  ENTER NEW DATA TO CHANGE. PRESS G TO START CALCULA
TIONS ";
23ØØ GOSUB 95Ø
231Ø MX = 8:IJ=2
232Ø IF ID <> Ø THEN 23ØØ ELSE ID = 1
233Ø IF IX < 31 THEN 23ØØ
234Ø IF Y$ = "G" THEN RETURN
235Ø IF Y$ ="<" THEN 158Ø
236Ø IF Y$ = ">" THEN 271Ø
237Ø JZ = 832
238Ø GOSUB73Ø
239Ø GOSUB 25ØØ
24ØØ GOSUB 26ØØ
241Ø IF IP = MX THEN 244Ø
242Ø IF IC <3Ø THEN IC = 3Ø :GOTO2Ø8Ø
243Ø IF IC > 16 THEN IC =16
244Ø IP = IP + 1
245Ø IF IP >MX THEN IP =1
246Ø IF IP < 1 THEN IP =MX
247Ø JP = IP+IJ
248Ø IF IP = 7 THEN IP = MX
249Ø GOTO 2Ø8Ø
25ØØ REM DATA ENTRY
251Ø GOSUB 44Ø
252Ø IF IP = MX THEN 259Ø
253Ø IF IC = 3Ø THEN 256Ø : REM TO ENTER RENTS
254Ø NU(IP) = VAL(V$)
255Ø GOTO257Ø
256Ø R(IP)=VAL(V$)
257Ø RY(IP) = R(IP)*NU(IP)*12
258Ø RETURN
259Ø VA = VAL(V$): IC = 16:RETURN
26ØØ REM SUM UP TOTALS
261Ø NS =Ø
262Ø RM = Ø
263Ø RY =Ø
264Ø FOR JJ = 1 TO NT
265Ø     NS = NS+NU(JJ)
266Ø   RM = RM + NU(JJ)*R(JJ)
267Ø     RY = RY + RY (JJ)
268Ø NEXT JJ
269Ø RETURN
27ØØ END
271Ø REM ENTER MORTGAGE COSTS ETC
272Ø CLS
273Ø JV=Ø:JA=Ø:JM=Ø:JN=Ø
274Ø IC = 6Ø
275Ø IP = 1
276Ø MX = 12
277Ø IJ = 1
278Ø JP = 2
279Ø JZ = 896+64
28ØØ CLS
281Ø PRINT "**** PAGE 4 OF 5 **** INVESTMENT DATA ****"
282Ø PRINT STRING$(63,"=")
283Ø FOR JJ= 1 TO 12
284Ø   PRINT "    ";
285Ø   PRINT USING F(JJ+18);Z(JJ)
286Ø NEXT JJ
```

Program continues

✓See List of Advertisers on page 402

"...if you have several investments to analyze, the use of tables soon becomes more work than you bargained for."

Program continued

```
2870 PRINT @ 896, "< OR > ";
2880 MX = 12 : IC = 55
2890 GOSUB 670
2900 GOSUB 790
2910 IF ID <> 0 THEN 2900 ELSE ID = 1
2920 IF IX < 32 THEN 2900
2930 GOSUB 730
2940 IF Y$ = "G" THEN RETURN
2950 IF Y$ = ">" THEN 3070
2960 IF Y$ = "<" THEN 2000
2970 GOSUB 3030
2980 IP = IP + 1
2990 IF IP > MX THEN IP = 1
3000 IF IP < 1 THEN IP = MX
3010 JP = IP + IJ
3020 GOTO 2800
3030 REM UPDATE INFORMATION
3040 GOSUB 440
3050 Z(IP) = VAL(V$)
3060 RETURN
3070 REM MORTGAGE ETC DATA CONT.
3080 IP = 1
3090 MX = 10
3100 JP = 2
3110 JZ = 896+64
3120 CLS
3130 PRINT "***** PAGE 5 OF 5 PAGES ***** INVESTMENT DATA CONT *
****"
3140 PRINT STRING$(63,"=")
3150 FOR J = 1 TO MX
3160   PRINT "    ";
3170   PRINT USING F(J+30);Z(J+12)
3180 NEXT J
3190 PRINT @ 896, "< ";
3200 IC = 55
3210 GOSUB 670
3220 GOSUB 790
3230 IF ID<> 0 THEN 3220 ELSE ID = 1
3240 IF IX < 32 THEN 3220
3250 GOSUB 730
3260 IF Y$ ="G" THEN RETURN
3270 IF Y$ = "<" THEN 2710
3280 IF Y$ =">" THEN 3070
3290 GOSUB 3350
3300 IP = IP+1
3310 IF IP > MX THEN IP = 1
3320 IF IP < 1 THEN IP = MX
3330 JP = IP + IJ
3340 GOTO 3120
3350 REM UPDATE INFORMATION
3360 GOSUB 440
3370 Z(IP+12) = VAL(V$)
3380 RETURN
3390 REM CHECK DATA INPUT
3400 SC=Z(1)-Z(2)-Z(3)-Z(7)+Z(11)+Z(12)
3410 CLS
3420 PRINT STRING$(64,"*")
3430 IF SC <> 0 THEN PRINT "POSSIBLE ERROR IN DATA INPUT"ELSE 35
10
3440 PRINT"THE SUM OF THE DOWN PAYMENT THE FIRST AND SECOND MORT
GAGES"
3450 PRINT"DOES NOT EQUAL THE PURCHASE PRICE PLUS CLOSING COSTS"
3460 PRINT "DO YOU WISH TO REVIEW DATA ?"
3470 GOSUB 410
3480 IF Y$ = "N" THEN 3510
3490 JM = -1
3500 RETURN
3510 REM CHECK TO SEE IF INTEREST RATES ARE NON ZERO
3520 CLS
3530 PRINT STRING$(64,"*")
3540 IF Z(4) = 0 AND Z(3) > 0 THEN 3550 ELSE 3600
3550 PRINT"POSSIBLE ERROR. INTEREST RATE FOR FIRST"
3560 PRINT"MORTGAGE IS ZERO."
3570 PRINT"DO YOU WISH TO REVIEW DATA?"
3580 GOSUB 410
3590 IF Y$="Y" THEN JM = -1:RETURN
3600 IF Z(7)>0 AND Z(8) = 0 THEN 3610 ELSE 3660
```

Program continues

```
-----------------------------------------------
Property tax-----------$4,400/yr
Insurance-----------$100/mo
License-----------$600/yr
Replacement reserves-----------$75/mo
Other fixed exp-----------$30/mo
Janitor-----------$450/mo
Soc sec, unemp-----------$45/mo
Garbage collection-----------$25/mo
Utilities-----------$25/mo
Supplies-----------$25/mo
Maintenance-----------$600/yr
Advertising-----------$40/mo
Management-----------$300/mo
Purchase price-----------$380,000
Down payment-----------$130,000
1st mortgage-----------$200,00
2nd mortgage-----------$50,000
Interest rate-----------12% both
Life 1st-----------25 years
Life 2nd-----------10 years
Payments per year-----------12 for both
Life of investment-----------5 years
Life of bldg for dep-----------33 years
Per cent for land-----------20%
Esclation for investment-----------15% per year
Esclation for rent-----------9% per year
Esclation for expense-----------10% per year
Discount rate-----------9%
Marginal income tax-----------50%
Capital gains rate-----------25%
Cost of selling-----------6% of selling price
-----------------------------------------------
2 one bedroom unfur rent at $400/mo,
2 two bedroom fur at $475/mo
4 three bedroom fur at $635/mo.
Vacancy allowance = 6%
-----------------------------------------------
```

Fig. 1. Data for Example.

year by year calculation of the income and expenses (escalated as specified by the user) for the life of the investment. Straight line depreciation is also used as part of the taxable income calculation. Both taxable and spendable income are calculated.

At the end of the last year of the investment, the program calculates the expected selling price (based on a user-specified escalation factor), capital gains tax, total rental income, total capital gain, and total profit from the investment. The program also calculates the present value of various cash flows from the investment and compares the present value of the

```
-----------------------------------------------
-----------------------------------------------
* Page 1 of 5 pages * Fixed Expenses *
-----------------------------------------------
-----------------------------------------------
            Item                 $/Mo       $/Yr
-----------------------------------------------
Property Tax ------------ →$   0.00    $   0.00
Insurance --------------- $   0.00    $   0.00
License ----------------- $   0.00    $   0.00
Replacement Reserves ---- $   0.00    $   0.00
Other Fixed Expenses ---- $   0.00    $   0.00
Total Fixed Expenses ---- $   0.00    $   0.00

>May Be Used To Change Pages. ↑↓←→ Are Active. Use G
To Start Calculations. Enter Data To change
```

Fig. 1. First Page of Menu, Fixed Expenses.

"This message alerts the user that money must be taken from savings. . ."

```
** Page 2 of 5 pages ** Operating Costs **

Item                                 $/Mo          $/Yr
-------------------------------------------------------
-------------------------------------------------------
Janitor -----------------  →$       0.00    $      0.00
Gardener ----------------   $       0.00    $      0.00
Soc. Security & Unemp ----  $       0.00    $      0.00
Garbage Collection -------  $       0.00    $      0.00
Utilities ----------------  $       0.00    $      0.00
Supplies -----------------  $       0.00    $      0.00
Maintenance -------------   $       0.00    $      0.00
Advertising -------------   $       0.00    $      0.00
Management -------------    $       0.00    $      0.00
Free Rent (Mgr Janitor) ----$       0.00    $      0.00
Other Operating Expenses -  $       0.00    $      0.00

< OR > May Be Used to Change Pages. ↑↓←→ Are Active.
Use G to Start Calculations. Enter Data to Change
```

Fig. 2. Page 2 of Menu, Operating Costs.

total profit with the amount of the down payment. If the total profit from the investment is less than the down payment, the program prints out a message calling attention to that fact. The program also prints out a message if the spendable income from the investment is negative. This message alerts the user that money must be taken from savings or borrowed to keep the investment going.

After the summary results are displayed, you are asked if you wish to see detailed results for each year. If you do, the income, expenses, and tax paid for each year are displayed.

If the spendable income for any year is negative, a message is printed to alert you that the investment is not paying for itself. If the taxable income is negative, a message is printed to inform you that there is a tax loss that can be used to shelter other incomes.

If you do not want to see the yearly details, you will be asked if you want a printout of the calculations. If you do, the input data and all the calculated results are printed on the line printer.

After the results of the last year are

```
-------------------------------------------------------
-------------------------------------------------------
Type of Unit    # Units    Rent $/Mo      Rent $ Yr
  1 Bed Fur →    0         $  0.00   $    0.00
  1 Bed Unfur    0         $  0.00   $    0.00
  2 Bed Fur      0         $  0.00   $    0.00
  2 Bed Unfur    0         $  0.00   $    0.00
  3 Bed Fur      0         $  0.00   $    0.00
  3 Bed Unfur    0         $  0.00   $    0.00
**** Total       0         $  0.00   $    0.00
  Allowance For Vacancies As % Of Rents—0.00%
**** Dollar Cost of Vacancy $ 0.00/Yr
**** Adjusted Gross Rents $ 0.00/Yr

-------------------------------------------------------
-------------------------------------------------------
< & > May Be Used To Change Pages ↑↓←→ Are Active
Enter New Data To Change. Press G To Start Calculations
```

Fig. 3. Page 3 of Menu, Income Estimates.

Program continued

```
3610 PRINT"POSSIBLE ERROR."
3620 PRINT"INTEREST RATE FOR SECOND MORTGAGE IS ZERO."
3630 PRINT"DO YOU WISH TO REVIEW DATA ?"
3640 GOSUB 410
3650 IF Y$ ="Y" THEN JM =-1:RETURN
3660 REM CHECK TO SEE IF NUMBER OF PAYMENTS ENTERED
3670 IF Z(3)>0 AND Z(5) >0 THEN 3710
3680 PRINT"POSSIBLE ERROR IN INPUT DATA"
3690 PRINT "LIFE OF 1ST MORTGAGE IS 0. DO YOU WISH TO REVIEW DAT
A?"
3700 GOSUB410:IF Y$="Y" THEN  JM = -1:RETURN
3710 IF Z(6)>0 THEN 3760
3720 PRINT "POSSIBLE ERROR IN INPUT DATA!"
3730 PRINT "NUMBER OF LOAN PAYMENTS PER YEAR IS 0.
3740 PRINT "DO YOU WISH TO REVIE DATA ?"
3750 GOSUB 410 : IF Y$="Y" THEN JM =-1:RETURN
3760 IF Z(7)>0 AND Z(9)>0 AND Z(10) > 0 THEN RETURN
3770 IF Z(9)> 0 THEN 3810 ELSE PRINT "POSSIBLE ERROR IN INPUT DA
TA"
3780 PRINT "LIFE OF SECOND MORTGAGE IS 0. DO YOU WISH TO REVIEW
DATA?"
3790 GOSUB 410
3800 IF Y$="Y" THEN JM =-1:RETURN
3810 IF Z(10)>0 THEN RETURN ELSE PRINT "POSSIBE ERROR IN INPUT D
ATA."
3820 PRINT"THE NUMBER OF LOAN PAYMENTS PER YEAR IS 0"
3830 PRINT "DO YOU WISH TO REVIEW DATA?":GOSUB 410
3840 IF Y$="Y" THEN JM=-1
3850 RETURN
3860 REM SET UP TO CALCULATE PAYMENT SCHEDULE
3870 REM CHECK TO SEE IF THERE IS FIRST MORTGAGE
3880 IF Z(3)=0 THEN 4270
3890 REM NOW AMORIZE LOAN
3900 CLS
3910 PRINT"PLEASE WAIT IT WILL TAKE 1 TO 2 MINUTES FOR CALCULATI
ONS"
3920 PRINT"* FIRST MORTGAGE CALCULATIONS "
3930 REM X1 IS INTEREST RATE PER PERIOD
3940 X1=Z(4)/Z(6)/100:REM INTEREST PER PERIOD
3950 V=(X1+1.0000)[(Z(6)*Z(5))
3960 P=X1*V*Z(3)/(V-1.)
3970 REM NOW CALCULATE PAYMENT SCHEDULE
3980 S=0
3990 S1=0
4000 N3=1
4010 S2=0
4020 A=Z(3)
4030 J1=1
4040 N2=Z(6)*Z(5)
4050 IF N2>Z(6)*Z(13) THEN N2=Z(6)*Z(13):'LIMIT CALCULATION TO L
IFE OF INVESTMENT
4060 B1=0:' B1=BALANCE ON FIRST MORTGAGE AT END OF N2 PERIODS
4070 P1(J1)=0
4080 P2(J1)=0
4090 J2=1
4100 X2= INT(X1*A*100)/100 : REM INTEREST TO NEAREST CENT
4110 PP=P-X2
4120 P2(J1)=P2(J1)+PP
4130 P1(J1)=P1(J1)+X2
4140 S=S+PP
4150 A=A-PP
4160 IF N3=N2 THEN 4240 :REM HAVE FINISHED PAYMENT SCHEDULE
4170 IF J2=Z(6) THEN  4210 :REM HAVE COMPLETED 1 YEAR OF CALCULA
TION
4180 N3=N3+1
4190 J2=J2+1
4200 GOTO4100
4210 J1=J1+1
4220 N3=N3+1
4230 GOTO 4070
4240 REM FINISHED WITH FIRST MORTGAGE
4250 B1=A:IF B1<0 THEN B1=0
4260 A=0
4270 REM NOW DO SECOND MORTGAGE
4280 IF Z(7)=0 THEN 4670
4290 REM NOW AMORIZE LOAN
4300 CLS
```

Program continues

". . . unlike depreciation of the building, depreciation of the furnishings is a real out-of-pocket expense."

Program continued

```
4310 PRINT"PLEASE WAIT IT WILL TAKE 1 TO 2 MINUTES FOR CALCULATI
ONS"
4320 PRINT "** SECOND MORTGAGE CALCULATIONS"
4330 REM X1 IS INTEREST RATE PER MONTH
4340 X1=Z(8)/Z(10)/100.
4350 V=(X1+1.0000)[(Z(10)*Z(9))
4360 P=X1*V*Z(7)/(V-1.)
4370 REM NOW CALCULATE PAYMENT SCHEDULE
4380 S=0
4390 S1=0
4400 N3=1
4410 S2=0
4420 A=Z(7)
4430 J1=1
4440 FOR J = 1 TO Z(9 ):P3(J)=0:P4(J)=0:NEXT J
4450 N2=Z(9)*Z(10)
4460 IF N2>Z(10)*Z(13) THEN N2=Z(10)*Z(13)
4470 B2=0:'B2=BALANCE ON SECOND MORTGAGE
4480 P3(J1)=0
4490 P4(J1)=0
4500 J2=1
4510 X2= INT(X1*A*100)/100 : REM INTEREST TO NEAREST CENT
4520 PP=P-X2
4530 P4(J1)=P4(J1)+PP
4540 P3(J1)=P3(J1)+X2
4550 S=S+PP
4560 A=A-PP
4570 IF N3=N2 THEN 4650 :REM HAVE FINISHED PAYMENT SCHEDULE
4580 IF J2=Z(10) THEN  4620 :REM HAVE COMPLETED 1 YEAR OF CALCUL
ATION
4590 N3=N3+1
4600 J2=J2+1
4610 GOTO4510
4620 J1=J1+1
4630 N3=N3+1
4640 GOTO 4480
4650 REM FINISHED
4660 B2=A:IF B2<0 THEN B2=0
4670 REM ALL LOANS TAKEN CARE OF
4680 RETURN
4690 REM NOW DO YEAR BY YEAR CALCULATION
4700 DF=0:TR=0:TP=0:PQ=0:DS=0
4710 DR = 1+Z(20)/100.:'DISCOUNT RATE FACTOR
4720 N%=INT(Z(13)):'LIFE OF INVESTMENT
4730 DP=Z(1)*(1-Z(15)/100.)/Z(14):'STRAIGHT LINE DEPRECIATION
4740 EY=1+Z(18)/100.:'ESCLATION FOR EXPENSES
4750 EX='+Z(17)/100:'ESCLATION FOR INCOME
4760 X9=EY(6)
4770 X8=EQ(12)
4780 ZT = Z(21)/100.
4790 EQ=0
4800 TI = 0 :'TOTAL INTEREST PAID
4810 SI = 0 :'TOTAL AFTER TAX SPENDABLE INCOME
4820 FOR J = 1 TO N%
4830    RX(J) = RG*EX[(J-1):'ADJUSTED GROSS INCOME ESCLATION LAG
S
4840    RF(J) = X9*EY[J:'FIXED EXPENSES
4850    RZ(J) = X8*EY[J:'OPERATING EXPENSES
4860    RL(J) = P1(J)+P2(J)+P3(J)+P4(J):'LOAN PAYMENTS PRINCIPAL
 + INTEREST
4870   PQ = PQ+(P2(J)+P4(J))*DR[-J :'PRESENT VALUE OF PRINCIPAL
PAYMENTS
4880    RI(J) = P1(J)+P3(J):'INTEREST PAID FOR YEAR
4890   TI = TI + RI(J) :'TOTAL INTEREST PAID TO DATE
4900    RG(J) = RX(J) - RF(J) - RZ(J) - RL(J) - Z(23):'GROSS INC
OME
4910    RT(J) = RX(J) - RF(J) - RZ(J) - RI(J) - DP - Z(23):'TAXA
BLE INCOME
4920    IF RT(J) < 0 THEN AL = AL + RT(J):GOTO4980
4930    IF AL > RT(J) THEN RT(J) = 0:AL = AL - RT(J) :GOTO4980
4940    IF AL> 0 THEN RT(J) = RT(J)-AL:'ADJUST FOR PREVIOUS LOSS
4950    TX(J) = ZT*RT(J):'TAX PAID
4960    RA(J) = RT(J)-TX(J):'AFTER TAX PROFIT
4970   GOTO 5000
4980    TX(J)=0
4990   RA(J)=RT(J)
5000   CF(J) = RA(J)+DP
```

Program continues

printed or if you decline the printout option, you will be asked if you want to analyze another investment. If you do, the program returns to page one of the input menu. Note that all the previously entered data are still in the program. Thus, if you want to look at the same investment but with a different interest rate, just change the interest rate. You do not need to reenter all the data. You may initiate calculations for the next investment at any time by pressing G.

Input

Data input is handled by filling in a five-page menu. The information required for each of the five pages is shown in Figs. 1 through 5. You can move from page to page using the < and > keys. The > key allows you to go to the next page in the menu while the < key returns you to the previous page. Note that the > page is the only key active for page 1 and the < key is the only key active for page 5.

The particular data item being entered is indicated by the position of the right arrow (which is used as a marker) on the CRT. You can move around the menu pages by using the four arrows on the keyboard. Each arrow moves the marker arrow on the CRT in the same direction that the arrow points. For example, the up arrow moves the marker on the CRT up.

The first page is used to enter data for fixed expenses. Replacement reserves are the monies that you should set aside each month or each year to ensure that there is money to replace such items as furnishings in the rental property. Basically this is the same as an account for depreciation of the furnishings of the property. It is shown as a separate expense because, unlike depreciation of the building, depreciation of the furnishings is a real out-of-pocket expense. It is highly probable you will need to replace some of the furnishings during the life of the investment and money that is set aside can handle this. This handling of the depreciation of the furnishing may not meet IRS standards, but it is realistic and the proper way to handle it for analysis purposes.

The second page of the menu is for operating expenses. These are expenses over which you have some control. You can get rid of the gardener, for example if things get tough. The items here are straightforward and do not require much discussion. The main thing to remember is to include all expenses. If you have unlisted expenses, enter them as other expenses. Be sure to include an expense for management—even if you manage the property yourself.

The third page of the menu is for entering expected income from rentals. You can

✓See List of Advertisers on page 402

“The discount factor is the minimum return that you will accept on your money.”

Program continued

```
5010   DF(J) = CF(J)*DR[-J
5020   TR = TR + CF(J)
5030   DF = DF+DF(J)
5040   EQ=P2(J)+P4(J)+EQ : 'EQUITY ACCUMULATED
5050   SI = SI + RG(J) - TX(J) :'TOTAL SPENDABLE INCOME TO DATE
5060   DS(J) = (RG(J)-TX(J))*DR[-J :'DISCOUNED SPENDABLE CASH
5070   DS = DS + DS(J):'ACCUMULATED DISCOUNTED SPENDABLE CASH
5080 NEXT J
5090 REM NOW CALCULATE SELLING PRICE
5100 SP = Z(1)*(1+Z(16)/100.)[Z(13)
5110 PR=SP*(1-Z(19)/100):REM SELLING PRICE LESS COST OF SELLING
5120 AD = DP*Z(13):'ACCUMULATED DEPRECIATION
5130 PR = PR - Z(1) +AD:'TAXABLE PROFIT ON SALE OF PROPERTY NOTE
 DEPRECIATION ADDED BACK Z(1) PURCHASE PRICE
5140 TX = PR *Z(22)/100.:'TAX PAID ON SALE OF PROPERTY
5150 PZ = PR - TX -AD :'AFTER TAX PROFIT NOTE DEPRECIATION REMOV
ED FROM PROFIT
5160 PD = PZ*DR[-N%:'DISCOUNTED VALUE OF AFTER TAX PROFIT
5170 DZ = PD + DF:'TOTAL PRESENT VALUE OF PROFITS FROM INVESTMEN
T
5180 REM
5190 REM NOW PRINT OUT THE RESULTS
5200 PRINT
5210 PRINT"**** RESULTS OF CALCULATIONS ****"
5220 PRINT STRING$(64,"=")
5230 PRINT "                    SUMMARY RESULTS     "
5240 PRINT "SELLING PRICE OF PROPERTY";TAB(40);:PRINT USING F(0)
;SP
5250 PRINT "OUTSTANDING LOAN BALANCE ";TAB(40);:PRINT USING F(0)
;B1+B2
5260 PRINT"DOWN PAYMENT + PRINCIPAL PAYMENTS " ;TAB(40);:PRINT U
SING F(0);EQ+Z(2)
5270 PRINT "TAXABLE PROFIT FROM SALE OF PROPERTY";TAB(40);:PRINT
 USING F(0);PR
5280 PRINT "TOTAL AFTER TAX PROFIT FROM RENTS";TAB(40);:PRINT US
ING F(0);TR
5290 PRINT "AFTER TAX PROFIT FROM SALE OF PROPERTY";TAB(40);:PRI
NT USING F(0);PZ
5300 PRINT"TOTAL PROFIT FOR INVESTMENT ";TAB(40);:PRINT USING F(
0);PZ + TR
5310 PRINT"PRESENT VALUE OF PROFITS";TAB(40);:PRINT USING F(0);D
Z
5320 IF DZ < Z(2) THEN PRINT "NOTE PRESENT VALUE OF PROFITS IS L
ESS THAN YOUR DOWN PAYMENT" ELSE 5340
5330 PRINT"THIS IS NOT A GOOD INVESTMENT."
5340 IF AL<0 THEN PRINT"NOTE INVESTMENT DID NOT PAY ITS WAY":PRI
NT"YOU WOULD HAVE TO TAKE MONEY FROM SAVINGS":PRINT"OR BORROW MO
RE MONEY TO PAY EXPENSES"
5350 PRINT"DO YOU WANT TO SEE DETAILS?"
5360 GOSUB 410
5370 IF Y$="N" THEN RETURN
5380 IF Y$<>"Y" THEN PRINT"PLEASE ANSWER Y OR N":GOTO5350
5390 REM PRINT DETAILS
5400 CLS
5410 FOR J = 1 TO N%
5420   CLS
5430     PRINT "** RESULTS FOR YEAR ";J;" **"
5440     PRINT STRING$(63,"=")
5450     PRINT USING "RENTAL INCOME -----------------$###,####.##
";RX(J)
5460     PRINT USING "FIXED EXPENSES ----------------$###,####.##
";RF(J)
5470     PRINT USING "OPERATING EXPENSES ------------$###,####.##
";RZ(J)
5480     PRINT USING "INTEREST PAID -----------------$###,####.##
";RI(J)
5490     PRINT USING "PRINCIPAL PAID ----------------$###,####.##
";P2(J)+P4(J)
5500   PRINT USING "DEPRECIATION ALLOWANCE --------$###,####.##"
;DP
5510     PRINT USING "TAXABLE INCOME ----------------$###,####.##
";RT(J)
5520     PRINT USING "TAX PAID ----------------------$###,####.##
";TX(J)
5530 SF(J)=RG(J)-TX(J)
5540     PRINT USING "AFTER TAX PROFIT --------------$###,####.##
";RA(J)
```

Program continues

```
**** Page 4 of 5 **** Investment Data ****
---------------------------------------------
---------------------------------------------
→ Purchase Price of Property ------------$      0.00
  Amount of Down Payment -------------$         0.00
  Amount of First Mortgage -------------$       0.00
  Interest Rate First Mortgage % Per Yr ---     0.00%
  Life of First Mortgage Years -----------         0
  Number of Payments Per Year ----   0
  Amount of Second Mortgage -----------$        0.00
  Interest Rate Second Mortgage % Yr----        0.00%
  Life of Second Mortgage Years ---------          0
  Number of Payments Per Year -----------          0
  Closing Costs------------------------$        0.00
  Other Costs of Buying ----------------$       0.00
< OR > May Be Used to Change Pages ↑↓←→ Are Active.
Use G to Start Calculations. Enter Data to Change.
```

Fig. 4. Page 4 of Menu, Investment Data.

enter information for six different rental units (one, two and three bedroom furnished and unfurnished rental units).

You are asked for two pieces of information for each type of rental unit—the number of units and the monthly rent for that type of unit. The computer calculates the total annual rent expected from all units of a given type and the monthly and yearly income for all rental units.

The program calculates and displays the dollar cost of the vacancy allowance and the expected rental income also, taking into account the vacancy allowance.

The fourth page is used to enter information on the capital costs of the investment. Here, you enter information on the purchase price of the property, the amount of the down payment, the amount of the first and second mortgages, the closing cost and other costs involved in purchasing property.

The fifth and final page is where you enter information on life of the investment, life of the property for depreciation, expected escalation for rents and expenses, expected appreciation of the investment, income tax rates, and the discount factor.

The discount factor is the minimum return that you will accept on your money. What you enter here is up to you. In general, a good value for the discount rate is the tax-free interest you can earn on a risk-free investment.

Output

The first two pages of the hard-copy output are a repeat of the input information in the same format as the menus used to enter the input information. If you don't believe the calculated output, carefully double-check the input data.

The third page of the hard-copy output is a summary of the calculated results. The first item on this page is total spendable income from rents. Total spendable income is the total income available to spend on your computer or other bad

"...it is possible to have a positive taxable income and a negative spendable income..."

habits. Spendable income is the rental receipts minus all the cash outlays you made to operate the property. Cash outlays also include the total loan payment (principal and interest) but do *not* include depreciation of the property. Principal payments are included as pocket expenses because your checkbook sees the total loan cost as the interest payment plus the principal payment. Depreciation is not included as pocket expense because you don't write a check for depreciation. Depreciation is an expense for tax purposes but not an expense for the purpose of calculating spendable income.

If the total spendable income is negative, then the rental income from the property was not enough to cover upkeep. You need to borrow or withdraw money from savings when the spendable income is negative. Note that it is possible to have a positive taxable income and a negative spendable income or vice versa.

(Remember: Spendable income is what goes into or comes out of your pocket.)

The second item shown is the after-tax profit from rents. This is what the IRS says you made, or lost, from renting the property. In this case, depreciation (straight line is used in the program) is an allowable expense but principal payment is not. A negative after-tax profit from rents can be used to shelter income from other sources (check with the IRS to be sure). So a negative after-tax profit from rents is not all bad—provided, of course, that the spendable income is positive.

The next several items in the output show the capital gain realized by selling the property at the end of the investment life. First the selling price, followed by the purchase price are shown. Next, your equity in the property is shown (down payment plus principal payments). Loan balance for the first and second mortgages is shown followed by the balance after the loans are paid off. The accumulated depreciation follows along with the cost basis for tax purposes. The

*****Page 5 of 5 Pages*****Investment Data Cont*****

→ Life of Investment Years ------------------	0
Life of Prop For Depreciation Years ---------	0
Per Cent of Value Due to Land --------------	0.00%
Escalation Rate For Property %/Yr----------	0.00%
Escalation Rate For Rent Increases ---------	0.00%
Escalation Rate For Expense Increases ------	0.00%
Costs of Selling Property % of Price ---------	0.00%
Discount Rate % ----------------------------	0.00%
Marginal Income Tax Rate % --------------	0.00%
Tax Rate for Capital Gains % --------------	0.00%

< May be Used to Change Pages ↑↓←→ Are Active. Use G to Start Calculations. Enter Data to Change

Fig. 5. Page 5 of Menu, Investment Data.

Program continued

```
5550 PRINT USING "SPENDABLE CASH FLOW -------------$###,####.##"
;SF(J)
5560    PRINT USING "CASH FLOW --------------------$###,####.##";
CF(J)
5570   PRINT USING "DISCOUNTED CASH FLOW ----------$###,####.##"
;DF(J)
5580    PRINT STRING$(63,"=")
5590    PRINT@896,"PRESS ANY KEY TO SEE MORE "
5600    GOSUB410
5610 NEXT J
5620 RETURN
5630 REM HARD COPY ROUTINE
5640 CLS
5650 PRINT"**** HARD COPY ROUTINE ****"
5660 PRINT "CHECK TO SEE IF PRINTER IS ON "
5670 PRINT STRING$(64,"=")
5680 LPRINT TAB(5)"REAL ESTATE INVESTMENT ANALYIS"
5690 LPRINT TAB(5) "VERSION 2.0 BY L. E. SPARKS"
5700 LPRINT STRING$(70,"=")
5710 LPRINT TAB(5)"PROPERTY IDENTIFICATION ";ZZ$
5720 LPRINT STRING$(70,"=")
5730 LPRINT TAB(25)"*****";TAB(30);"INPUT DATA ****"
5740 LPRINT STRING$(70,"=")
5750 LPRINT TAB(5) "* PAGE 1 OF 5 PAGES * FIXED EXPENSES "
5760 LPRINT STRING$(70,"=")
5770 LPRINT TAB(5) "   ITEM ";TAB(40);"$/M0          $/YR"
5780 LPRINT STRING$(70,"=")
5790 FOR J = 1 TO 6
5800       LPRINT TAB(5);:LPRINT USING F(J);EF(J);
5810    LPRINT USING "     $###,###.##";EY(J)
5820 NEXT J
5830 LPRINT STRING$(70,"*")
5840 LPRINT TAB(5) "** PAGE 2 OF 2 * OPERATING EXPENSES **"
5850 LPRINT STRING$(70,"=")
5860 LPRINT TAB(5)"ITEM ";TAB(40);"$/M0";TAB(50);"$/YR"
5870 LPRINT STRING$(70,"=")
5880 FOR J = 1 TO 12
5890    LPRINT TAB(5);:LPRINT USING F(J+6);E(J);
5900    LPRINT USING "     $###,###.##";EQ(J)
5910 NEXT J
5920 LPRINT STRING$(70,"*")
5930 LPRINT TAB(5)"*** PAGE 3 OF 5 PAGES * INCOME ESTIMATES ***"
5940 LPRINT STRING$(70,"=")
5950 LPRINT TAB(5)"TYPE OF UNIT";TAB(18);"# OF UNITS";TAB(30);"R
ENT $/MO";TAB(50);"RENT $/YR"
5960 FOR J = 1 TO NT
5970    LPRINT TAB(5);:LPRINT FT(J);TAB(20);NU(J);TAB(30);
5980    LPRINT USING "$#,######.##";R(J);:LPRINT TAB(45);
5990    LPRINT USING F(0);RY(J)
6000 NEXT J
6010 LPRINT TAB(5);:LPRINT USING"VACANCY ALLOWANCE AS % OF RENTS
 = ##.##%";VA
6020 LPRINT TAB(5);:LPRINT USING"DOLLAR COST OF VACANCY = $###,#
##.##/YR";VA*RY/100
6030 LPRINT TAB(5);:LPRINT USING "ADJUSTED GROSS RENTS =    $####
,###.##/YR";RG
6040 LPRINT STRING$(70,"*")
6050 LPRINT CHR$(140):'TOP OF FORM PRINTER CONTROL
6060 LPRINT " ":LPRINT" ":LPRINT" "
6070 LPRINT TAB(5)"PAGE 2 OF OUTPUT INPUT DATA CONT "
6080 LPRINT STRING$(70,"=")
6090 LPRINT " "
6100 LPRINT TAB(5) "**** PAGE 4 OF 5 PAGES * INVESTMENT INFORMAT
ION ****"
6110 LPRINT STRING$(70,"=")
6120 FOR J = 1 TO 12
6130    LPRINT TAB(5);:LPRINT USING F(J+18);Z(J)
6140 NEXT J
6150 LPRINT STRING$(70,"*")
6160 LPRINT TAB(5)"***** PAGE 5 OF 5 PAGES INVESTMENT DATA CONT
*****"
6170 LPRINT STRING$(70,"=")
6180 FOR J = 13 TO 22
6190    LPRINT TAB(5);:LPRINT USING F(J+18);Z(J)
6200 NEXT J
6210 LPRINT STRING$(70,"*")
```

Program continues

"The investment with the maximum present value of spendable income is the best."

Program continued

```
6220 LPRINT CHR$(140):' TOP OF FORM FOR PRINTER
6230 LPRINT " "
6240 LPRINT " "
6250 LPRINT " "
6260 LPRINT TAB(5)"PAGE 3 OF OUTPUT "
6270 LPRINT STRING$(70,"=")
6280 LPRINT " "
6290 REM
6300 REM
6310 REM NOW LPRINT OUT THE RESULTS
6320 LPRINT
6330 LPRINT TAB(5)"**** RESULTS OF CALCULATIONS ****"
6340 LPRINT STRING$(70,"=")
6350 LPRINT " "
6360 LPRINTTAB(9) "              SUMMARY RESULTS     "
6370 LPRINT STRING$(70,"=")
6380 LPRINT TAB(5)"TOTAL SPENDABLE INCOME FORM RENTS ";TAB(44);
6390 LPRINT USING F(0);SI
6400 LPRINT TAB(5) "TOTAL AFTERTAX PROFIT FROM RENTS";TAB(44);:L
PRINT USING F(0);TR
6410 LPRINT TAB(5)"SELLING PRICE ";TAB(44);:LPRINT USING F(0);SP
6420 LPRINT TAB(5)"PURCHASE PRICE";TAB(44);:LPRINT USING F(0);Z(
1)
6430 LPRINT TAB(5)"DOWN PAYMENT + PRINCIPAL PAYMENTS";TAB(44);:L
PRINT USING F(0);EQ+Z(2)
6440 LPRINT TAB(5)"PRESENT VALUE OF PRINCIPAL PAYMENTS ";TAB(44)
;
6450 LPRINT USING F(0);PQ
6460 LPRINT TAB(5)"PRESENT VALUE OF EQUITY ";TAB(44);
6470 LPRINT USING F(0);Z(2)+PQ
6480 LPRINT TAB(5)"BALANCE LEFT ON LOANS";TAB(44);:LPRINT USING
F(0);B1+B2
6490 LPRINT TAB(5)"SELLING PRICE - LOAN BALANCE ";TAB(44);:LPRIN
T USING F(0);SP-B1-B2
6500 LPRINT TAB(5)"ACCUMULATED  DEPRECIATION ";TAB(44):LPRINT US
ING F(0);AD
6510 LPRINT TAB(5)"COST BASIS FOR TAX PURPOSES ";TAB(44):LPRINT
USING F(0);Z(1)-AD
6520 LPRINT TAB(5)"COST BASIS FOR TAX IS PURCHASE PRICE - ACCUMU
LATED DEPRECIATION"
6525 LPRINT TAB(5)"COST OF SELLING ";TAB(44);:LPRINT USING F(0);
SP*Z(19)/100
6530 LPRINT TAB(5) "TAXABLE PROFIT FROM SALE OF PROPERTY";TAB(44
);:LPRINT USING F(0);PR
6540 LPRINT TAB(5)"CAPITAL GAINS TAX ";TAB(44);:LPRINT USING F(0
);TX
6550 LPRINT TAB(5)"AFTER TAX PROFIT ON SALE";TAB(44);:LPRINT USI
NG F(0);PZ
6560 LPRINT TAB(5) "TOTAL AFTER TAX PROFIT FOR INVESTMENT ";TAB(
44);:LPRINT USING F(0);PZ + TR
6570 LPRINT TAB(5)"PRESENT VALUE OF PROFIT ON SALE ";TAB(44);:LP
RINT USING F(0);PZ*(1+Z(20)/100.)[-Z(13)
6580 LPRINT TAB(5) "PRESENT VALUE OF ALL PROFITS";TAB(44);:LPRIN
T USING F(0);DZ
6590 LPRINT TAB(5)"PRESENT VALUE OF ALL SPENDABLE INCOME";TAB(44
):LPRINT USING F(0);DS+PZ*(1+Z(20)/100)[-Z(13)
6600 LPRINT " ":LPRINT STRING$(70,"*")
6610 IF DZ> Z(2) THEN 6630
6620 IF DZ < Z(2) THEN LPRINT TAB( 5)"NOTE THAT PRESENT VALUE OF
 PROFITS IS LESS THAN YOUR DOWN PAYMENT":LPRINT TAB(5)"THIS IS N
OT A GOOD INVESTMENT"
6630 IF SI<0 THEN LPRINT TAB(5)"!!! NOTE !!! THE INVESTMENT DID
NOT PAY ITS WAY":LPRINT TAB(5) "YOU WOULD HAVE TO TAKE MONEY FRO
M SAVINGS":LPRINT TAB(5)"OR BORROW MORE MONEY TO PAY EXPENSES."
6640 LPRINT STRING$(70,"=")
6650 REM LPRINT DETAILS
6660 JP = 3
6670 LPRINT CHR$(140):'TOP OF FORM PRINTER CONTROL
6680 LPRINT" ":LPRINT " ":LPRINT " "
6690 LPRINT TAB(5)"PAGE ";JP+1;" OF OUTPUT DETAILED RESULTS"
6700 JP=JP+2
6710 JZ=1
6720 FOR J = 1 TO N%
6730   JZ=JZ+1
6740     LPRINT  TAB(5) "** RESULTS FOR YEAR ";J;" **"
6750     LPRINT STRING$(70,"=")
```

Program continues

IRS requires that you subtract the accumulated depreciation from the purchase price of the property to calculate the cost of the property for tax purposes. The taxable capital gain followed by the capital gains tax paid are then shown.

The next four items summarize the total profit from the investment. The last items show the present value of the total after-tax profit and total spendable profit expected from the investment. These three items are very important. If you are evaluating more than one investment, the investment with the maximum present value of spendable income is the best.

If you are evaluating one investment, the investment is attractive if the present value of the spendable income is greater than your down payment and equity. Any investment that gives a present value less than your down payment and equity is a poor one. A present value less than your down payment and equity means that you could increase your spendable income by putting money in a risk-free investment represented by the discount rate rather than from the investment you're analyzing.

The next several pages of the printed output are the year-by-year details of income and expenses for the rental property. The main thing to look at here is the spendable income—if it is negative, you need to borrow or take money from savings to keep the investment going. Also note the taxable income. If it is negative, you have a tax loss that can be used to shelter other incomes (check with the IRS for specifics). As long as the spendable income is positive, a negative taxable income is fine—especially if you're in a high tax bracket.

In the program, tax losses in one year are carried forward to offset the next year's taxable rental income.

The output displayed on the CRT is essentially the same as the printed output. Some of the items are not displayed, but all of the important information is shown both on the CRT and the hard-copy output.

Calculations

The calculations performed by the program are straightforward. The program calculates the loan payment requirements of the first and second mortgages for the life of the investment (not the life of the loan). The interest and principal amounts are kept separate for later use.

After the mortgage calculations are taken care of, the program does a year-by-year calculation of the income and the expenses of the investment. The expenses are escalated using the escalation factor and entered as part of the input data. The rental income is also escalated. However, the program assumes that rental price in-

✓See List of Advertisers on page 402

"The present value of all cash flow is calculated using the discount rate entered during data entry."

Program continued

```
6760    LPRINT TAB(5);:LPRINT USING "RENTAL INCOME -------------
----$###,####.##";RX(J)
6770    LPRINT TAB(5);:LPRINT USING "FIXED EXPENSES ------------
----$###,####.##";RF(J)
6780    LPRINT TAB(5);:LPRINT USING "OPERATING EXPENSES --------
----$###,####.##";RZ(J)
6790    LPRINT TAB(5);:LPRINT USING "INTEREST PAID -------------
----$###,####.##";RI(J)
6800    LPRINT TAB(5);:LPRINT USING "PRINCIPAL PAID ------------
----$###,####.##";P2(J)+P4(J)
6810    LPRINT TAB(5);:LPRINT USING "SPENDABLE GROSS INCOME ----
----$###,####.##";RG(J)
6820   IF RG(J)<0 THEN LPRINTTAB(5)"*** NOTE THAT GROSS INCOME I
S NEGATIVE. YOU NEED A LOAN TO KEEP GOING "
6830    LPRINT TAB(5);:LPRINT USING "TAXABLE INCOME ------------
----$###,####.##";RT(J)
6840 IF RT(J) < 0 THEN LPRINT TAB(5)" *** NOTE YOU HAVE A LOSS F
OR TAX PURPOSES WHICH CAN SHELTER OTHER INCOME."
6850 IF RG(J) <0 THEN LPRINT TAB(5)"*** YOU ALSO HAVE A REAL OUT
 OF POCKET LOSS AS NOTED ABOVE!!"
6860 LPRINT TAB(5);:LPRINT USING "DEPRECIATION -----------------
---$###,####.##";DP
6870    LPRINT TAB(5);:LPRINT USING "TAX PAID ------------------
----$###,####.##";TX(J)
6880    LPRINT TAB(5);:LPRINT USING "AFTER TAX PROFIT ----------
----$###,####.##";RA(J)
6890    LPRINT TAB(5);:LPRINT USING "TAXABLE CASH FLOW ---------
----$###,####.##";CF(J)
6900   LPRINT TAB(5);:LPRINT USING "SPENDABLE CASH FLOW --------
---$###,####.##";RG(J)-TX(J)
6910   LPRINT TAB(5);:LPRINT USING "DISCOUNTED TAXABLE CASH FLOW
 --$###,####.##";DF(J)
6920   LPRINT TAB(5);:LPRINT USING "DISCOUNTED SPENDABLE CASH FL
OW $###,####.##";DS(J)
6930    LPRINT STRING$(70,"=")
6940   IF JZ < 3 THEN 6980 ELSE LPRINT CHR$(140)
6950   LPRINT " ":LPRINT " ":LPRINT "PAGE ";JP;" OF OUTPUT. DETA
ILED RESULTS CONT "
6960   LPRINT STRING$(70,"=")
6970   JZ=1:JP=JP+1
6980 NEXT J
6990 LPRINT TAB(5)"NOTE.SPENDABLE CASH FLOW IS TAKEN AS THE TOTA
L INCOME":LPRINT TAB(5)"MINUS FIXED & OPERATING EXPENSES";
7000 LPRINT "MINUS INTEREST & PRINCIPAL PAYMENTS MINUS TAXES."
7010 LPRINT TAB(5)" CASH  FLOW IS TOTAL INCOME MINUS FIXED & OPE
RATING EXPENSES MINUS INTEREST"
7020 LPRINT TAB(5)"MINUS TAXES. NOTE PRINCIPAL PAYMENT IS NOT TA
X DEDUCTABLE.
7030 LPRINT STRING$(70,"=")
7040 RETURN
7050 RETURN
7060 REM INSTRUCTION ROUTINE
7070 PRINT "THIS PROGRAM IS WRITTEN FOR A TRS-80 MODEL 1 WITH"
7080 PRINT "LEVEL II BASIC AND 32 K RAM."
7090 PRINT"THIS PROGRAM IS DESIGNED TO CALCULATE THE PRESENT VAL
UE"
7100 PRINT"OF THE CASH FLOWS FROM A RENTAL PROPERTY INVESTMENT."
7110 PRINT"DATA ARE ENTERED BY FILLING IN THE MENUS THAT FOLLOW.
"
7120 PRINT"USE > TO GO TO THE NEXT MENU PAGE. USE < TO GO TO THE
"
7130 PRINT"PREVIOUS MENU PAGE. USE THE ARROWS TO MOVE AROUND THE
 MENU.
7140 PRINT"MENU INSTRUCTIONS ARE PRINTED AT THE BOTTOM OF EACH P
AGE."
7150 PRINT STRING$(63,"=")
7160 PRINT" ***** PRESS ANY KEY TO CONTINUE *****"
7170 Y$=INKEY$:IF Y$="" THEN 7170 ELSE CLS
7180 PRINT" ****** INSTRUCTION CONTINUED ******"
7190 PRINT STRING$(63,"=")
7200 PRINT"FOR SOME ITEMS YOU HAVE THE OPTION OF ENTERING COSTS

7210 PRINT"ON A MONTHLY OR YEARLY BASIS DEPENDING ON THE DATA "
7220 PRINT"YOU HAVE. MOVE THE ARROW TO THE APPROPIATE COLUMN"
7230 PRINT"AND ENTER THE DATA. THE COMPUTER WILL CALCULATE THE"
7240 PRINT"COST THAT YOU DID NOT ENTER. FOR EXAMPLE IF YOU ENTER
"
```

Program continues

creases are on a yearly basis since rents are often set by leases which run for a year or more.

Depreciation is calculated for the building only. Straight line depreciation is used.

The present value of all cash flow is calculated using the discount rate entered during data entry. The present value of all principal payments is also calculated.

The selling price is calculated by escalating the purchase price which is specified by the input data. The taxable capital gain is calculated by subtracting the accumulated depreciation from the purchase price which gives you the cost basis. This is then subtracted from the selling price, less the selling costs from the basis. The present value of the net, after tax profit of the sale, is then calculated. Finally the present values, after taxes, of cash flow and spendable cash flow are calculated.

User Instructions

In this section, user instructions followed by an example run are presented. After you have run the example, you should not have any trouble running the program.

Before showing you the input data menu, the program asks you to provide identification for the investment you are analyzing. You can use any identification scheme you want. The program uses a Line Input (Disk Basic) so you can enter any set of characters you wish for the identification. If you do not have Disk Basic, change Line Input to Input.

After you have provided the program identification, program operation is a matter of filling in the menu pages and then pressing G when you are ready to initiate calculations. You select the data item you want to enter by using the > and < keys to find the appropriate page and then by using the up, down, left, or right arrow to move the marker on the CRT to the appropriate entry point. When you enter numbers from the keyboard, you may use commas to separate thousands. For example, the number 1000 may be entered as 1,000 or 1000. It is easier to keep track of things if I enter the commas.

The key to running the program is filling out the menus. (See Fig. 1, page 1 of the menu.) There are three columns for the page—Item, $/MO, and $/YR. The Item column is the name of the expense, the $/MO column is where you enter the monthly cost of the expense (if you know it) and the $/YR is where you enter the yearly cost of the expense (if you know it). Also note the right arrow (→) in the figure after property tax. The right arrow indicates what data

". . . this program only considers the time and money factors of the investment."

will be entered if you want to enter data. The right arrow can be moved around the menu by using the up arrow (↑), the down arrow (↓), the left arrow (←) and the right arrow (→) keys on the keyboard. Thus, if you want to enter cost of license, press the down arrow key until the right arrow is to the right of license on the CRT. If you wish to enter the monthly cost of licenses, simply enter the appropriate amount and press Enter when you are done. Note that when you enter the amount, you can use commas if you want. When you have entered the last digit of the amount, press Enter. Now note that the menu is reprinted with the amount you entered in the $/MO column and also note that the computer has calculated the $/YR and filled in that column. Also observe that the computer has added the $/MO and $/YR amounts to the total fixed expenses.

If you know the $/YR instead of the $/MO, press the right arrow key (→) to move the arrow to the $/YR column. Then enter the cost per year. The computer will calculate the monthly cost for you.

Data entered on page two are the same as for page one.

Page three requires that you enter information on the number of units and the monthly rent per unit. The computer calculates the yearly rent for you. You can not enter the yearly rent. You must enter the monthly rent. Be sure to enter the allowance for vacancies. Note that this is entered as a percent of the rental income and not a dollar amount. The computer will calculate the dollar amount for you. Pages four and five have one column for data.

As soon as you press G, the computer starts the calculations. The first thing it does is check key input data to see if they are reasonable. If the data are not reasonable, the computer prints an error message and asks if you want to review the input information. Return to the appropriate page of the menu, review the data and make necessary corrections and then press G to begin calculations again.

If you do not wish to review the input data, press N and the computer will do the calculations. Sample results for the data given in Fig. 1 appear in Fig. 2.

When you use the program to analyze your own investments, remember that the computer calculations are based on the data you provided. If those data are bad, so is the calculated result.

Finally, remember that no matter what the computer results say, you are the one that has to decide if the investment is good or bad. Also remember that this program only considers the time and money factors of the investment. Consideration of risk and other factors is totally up to you. ■

Program continued

```
7250 PRINT"MONTHLY COST, THE COMPUTER WILL CALCULATE THE YEARLY
COST."
7260 PRINT"WHEN YOU HAVE ENTERED ALL THE DATA,  PRESS G AND THE"

7270 PRINT"COMPUTER WILL BEGIN THE CALCULATIONS."
7280 PRINT"AFTER ALL CALCULATIONS ARE COMPLETED, THE COMPUTER WI
LL"
7290 PRINT"DISPLAY A SUMMARY OF THE RESULTS AND ASK IF YOU WANT"
7300 PRINT"TO SEE DETAILS. A HARD COPY OPTION IS ALSO AVAILABLE.
"
7310 PRINT STRING$(63,"=")
7320 PRINT"**** PRESS ANY KEY TO BEGIN ****"
7330 Y$=INKEY$:IF Y$=""THEN 7330 ELSE CLS
7340 RETURN
7350 REM INITIALIZE
7360  F(0)="$###,######.##"
7370 F(1)="PROPERTY TAX ------------     $###,######.##"
7380 F(2)="INSURANCE ----------------    $###,######.##"
7390 F(3)="LICENSE   ----------------    $###,######.##"
7400 F(4) ="REPLACEMENT RESERVES -----    $###,######.##"
7410 F(5)="OTHER FIXED EXPENSES -----    $###,######.##"
7420 F(6)="TOTAL FIXED EXPENSES =========$###,######.##"
7430 F(7)="JANITOR -----------------      $###,####.##"
7440 F(8)="GARDNER -----------------      $###,####.##"
7450 F(9)="SOC. SECURITY & UNEMP ---      $###,####.##"
7460 F(10)="GARBAGE COLLECTION ------      $###,####.##"
7470 F(11)="UTILITIES ---------------      $###,####.##"
7480 F(12)="SUPPLIES ----------------      $###,####.##"
7490 F(13)="MAINTENANCE -------------      $###,####.##"
7500 F(14)="ADVERTISING -------------      $###,####.##"
7510 F(15)="MANAGEMENT --------------      $###,####.##"
7520 F(16)="FREE RENT (MGR JANITOR)--      $###,####.##"
7530 F(17)="OTHER OPERATING EXPENSES --    $###,####.##"
7540 F(18)="TOTAL OPERATING EXPENSES =======$###,######.##"
7550 REM LABELS FOR TYPE OF UNITS
7560 REM IF YOU HAVE DIFFERENT TYPES OF UINTS CHANGE THE
7570 REM DATA STATEMENTS BEGINNING AT LINE 7930
7580 REM BE SURE TO TYPE END FOR THE LAST ITEM IN THE DATA LIST
7590 REM THE DATA LIST WILL HANDLE UP TO 12 DIFFERENT TYPE
7600 REM OF UNITS
7610 REM A TYPE OF UNIT IS FOR EXAMPE 1 BEDROOM FURNISHED
7620 REM LISTED AS 1 BED FUR
7630 FOR J = 1 TO 12
7640   READ FT(J)
7650      IF FT(J) = "END" THEN 7670
7660 NEXT J
7670 NT = J -1
7680 F(19) = "PURCHASE PRICE OF PROPERTY -----$###,######.##"
7690 F(20) = "AMOUNT OF DOWN PAYMENT ---------$###,######.##"
7700 F(21) = "AMOUNT OF FIRST MORTGAGE -------$###,######.##"
7710 F(22) = "INTEREST RATE FIRST MORTGAGE % PER YR ---##.##%"
7720 F(23) = "LIFE OF FIRST MORTGAGE YEARS ---------------##"
7730 F(24) = "NUMBER OF PAYMENTS PER YEAR ----------------##"
7740 F(25) = "AMOUNT OF SECOND MORTGAGE ------$###,######.##"
7750 F(26) = "INTEREST RATE SECOND MORTGAGE % YR ------##.##%"
7760 F(27) = "LIFE OF SECOND MORTGAGE YEARS --------------##"
7770 F(28) = "NUMBER OF PAYMENTS PER YEAR ----------------##"
7780 F(29) = "CLOSING COSTS ------------------$###,######.##"
7790 F(30) = "OTHER COSTS OF BUYING ----------$###,######.##"
7800 F(31) = "LIFE OF INVESTMENT  YEARS ------------------##"
7810 F(32) = "LIFE OF PROP FOR DEPRECIATION YEARS --------##"
7820 F(33)=  "PER CENT OF VALUE DUE TO LAND ----------##.##%"
7830 F(34) = "ESCLATION RATE FOR PROPERTY  %/YR--------##.##%"
7840 F(35) = "ESCLATION RATE FOR RENT INCREASES ------ ##.##%"
7850 F(36) = "ESCLATION RATE FOR EXPENSE INCREASES --- ##.##%"
7860 F(37) = "COSTS OF SELLING PROPERTY % OF PRICE ----##.##%"
7870 F(38) = "DISCOUNT RATE  % -----------------------##.##%"
7880 F(39) = "MARGINAL INCOME TAX RATE % --------------##.##%"
7890 F(40) = "TAX RATE FOR CAPITAL GAINS %-------------##.##%
7900 F(41) = "ALLOWANCE FOR VACANCIES AS % OF RENTS --##.##%"
7910 RETURN
7920 REM ***********************************************
7930 REM DATA STATEMENTS FOR TYPE OF APARTMENTS
7940 DATA 1 BED FUR, 1 BED UNFUR, 2 BED FUR, 2 BED UNFUR
7950 DATA 3 BED FUR, 3 BED UNFUR, END
```

DISK III

100% Compatible

Model III Disks

Complete Business System includes: 48K TRS-80™ Model III, Disk III™ 2 Drive System, TRSDOS and Manual.

$1882

[1] DISK III single drive assembly includes: one 40 track 5¼" double density drive, power supply, controller, mounting hardware, and applicable cables.

DISK III Single drive assy[1]	**$599.00**
DISK III Two drive assy	**864.00**
DISK III Assy w/out drives	**435.00**
TRSDOS™ & Manual	**21.90**
External drives (3 & 4)	**299.00**

IMMEDIATE DELIVERY - COMPARE AND SAVE

VR Data's DISK III features:

- Completely compatible with TRSDOS
- State-of-the-art circuitry
- Fully tested, calibrated & burned-in
- Up to 4 drive configuration
- Heavy duty linear power supply
- Warranty 120 days - 100% parts and labor
- Installation with simple hand tools
- Optional 80 tk and 2 sided drives
- **READ 40** - Pgm. to read 40 tk. Diskettes on 80 tk. Drives $25.00

Peripherals

Epson MX-80	500.00
Epson MX-80 FT	615.00
Epson MX-100	800.00
Centronics 739	700.00
Starwriter 25 (P)	1395.00
RS-232	95.00
Lexicon modem	105.00

MODEL I/III EXTERNAL DRIVE W/ PS & ENC **$275**

Fully Compatible
120 day warranty
Easy installation

80 tk or 2 sided	$419.00
80 tk & 2 sided	549.00

OTHER PRODUCTS

SUPERBRAIN 64K	2990.00
PARALLEL PORT — SUPERBRAIN	99.95
DISK & MYSTERIES	22.50
BASIC & MYSTERIES	29.95
NEC Ribbons (min. 6)	5.95
Epson Ribbons	12.50
NEW—	
LDOS Operating System	149.00

COMING SOON!!
Internal MODEM for MOD III

IF YOU DON'T SEE IT ADVERTISED CALL US AND ASK FOR IT.
PUBLISHED PRICES REFLECT CASH DISCOUNT.
ALL PRICES ARE SUBJECT TO CHANGE WITHOUT NOTICE.
TRS-80 and TRSDOS are trademarks of Tandy Corp.
DISK III is a trademark of VR Data Corp. Dealership available.

CABLE "VRDATA" • TELEX

777 Henderson Boulevard • Folcroft, PA 19032

A software real estate information library.

The House Detective

John Cominio
626 Tortoise Way
Satellite Beach, FL 32937

If you work at a real estate office you probably know what a hassle it is to dig through page after page of listings to find a house for a customer. I have written a program to store, edit and search files containing specific information on houses.

When you finish entering the program and run it, the computer will display a menu showing all options available to you:

- Create files. This allows you to enter information about a house. You can store the address, price, waterfront (ocean, river, canal or not waterfront), pool (yes or no), lot size, number of bedrooms and the number of bathrooms (see Table 1). Each block of information is considered one file and with 16K, 100 files can be maintained in memory.
- Search files. This allows you to enter the specifications of a house you want the computer to search for. The specifications are as general or specific as you care to make them. You are asked for a price range, waterfront (yes or no), pool (yes or no), number of bedrooms, and number of bathrooms. After you have entered the desired information, the computer will search for all houses in memory which have those qualifications. And if you have a printer it will print them out (provided you tell it to).
- View stored files. This option allows you to look at all the houses currently in memory. To advance the listing, press Enter, or to exit, press the # key. This option is designed as a check to let you see if you have correctly entered the information.

Program Listing 1

```
10 'PROGRAM BY JOHN COMINIO
20 'VERSION 1.5 JUNE 17,1981  ** CASSETTE VERSION **
30 DEFINTA,C,E-O,Q-Z
40 CLEAR6800
50 X=100:DIMH$(X),P(X),W$(X),PO$(X),S$(X),B(X),BA(X),T(X):H1=X
60 ONERRORGOTO1410
70 B=1:L=1:V=1:X=0
80 J=0:CLS:PRINTTAB(12)"REAL ESTATE INFORMATION LIBRARY":PRINT
90 PRINTTAB(5)"(1) -- CREATE FILES":PRINTTAB(5)"(2) -- SEARCH FI
LES":PRINTTAB(5)"(3) -- VIEW STORED FILES":PRINTTAB(5)"(4) -- SA
VE ON CASSETTE":PRINTTAB(5)"(5) -- LOAD FROM CASSETTE":PRINTTAB(
5)"(6) -- CHANGE FILE":PRINTTAB(5)"(7) -- DELETE FILE"
100 PRINTTAB(5)"(8) -- INSERT FILE":PRINTTAB(5)"(9) -- CLEAR FIL
ES IN MEMORY"
110 R$="":PRINT@768,"";:PRINTCHR$(30);:A=0:INPUT"COMMAND";R$:A=V
AL(R$):IFA=0THEN110
120 IFA<1ORA>9THEN110
130 ONAGOTO140,890,300,460,540,630,820,1310,1590
140 CLS:PRINTTAB(10)"FILE NUMBER";B;" == TO EXIT TYPE '##' ==":P
RINT
150 PRINT@128,"";:PRINTCHR$(30);:H$(B)="":INPUT"ADDRESS? ";H$(B)
160 IFLEN(H$(B))>55THENH$(B)=LEFT$(H$(B),55):PRINT@128,"";:PRINT
CHR$(30);:PRINT" -- TO LONG --  SHORTENED AS FOLLOWS ":FORR=1TO5
00:NEXTR:PRINT@128,"";:PRINTCHR$(30);:PRINT"ADDRESS";:PRINT@136,
H$(B);
170 IFH$(B)="##"ANDFL=1THEN150
180 IFH$(B)=""THEN150
190 IFH$(B)="##"THENH$(B)="":X=1:GOTO80
200 R$="":PRINT@192,"";:PRINTCHR$(30);:P(B)=0:INPUT"PRICE";R$:P(
B)=VAL(R$):IFP(B)=0THEN200
210 PRINT@256,"";:PRINTCHR$(30);:W$(B)="":INPUT"WATERFRONT (O/R/
C/N)";W$(B)
220 IFW$(B)="O"ORW$(B)="R"ORW$(B)="C"ORW$(B)="N"THEN230ELSE210
230 PRINT@320,"";:PRINTCHR$(30);:PO$(B)="":INPUT"POOL (Y/N)";PO$
(B)
240 IFPO$(B)="Y"ORPO$(B)="N"THEN250ELSE230
250 S$(B)="":INPUT"LOT SIZE";S$(B)
260 R$="":PRINT@448,"";:PRINTCHR$(30);:B(B)=0:INPUT"NUMBER OF BE
DROOMS";R$:B(B)=VAL(R$):IFB(B)=0THEN260
270 R$="":PRINT@512,"";:PRINTCHR$(30);:BA(B)=0:INPUT"NUMBER OF B
ATHROOMS";R$:BA(B)=VAL(R$):IFBA(B)=0THEN270
280 IFFL=1THENRETURNELSEFL=0
290 B=B+1:GOTO140
300 IFX=0THEN80ELSEX=1
310 IFH$(X)=""THENX=X+1
320 IFX>B-1THEN430
330 CLS:PRINTTAB(10)"FILE NUMBER";X;"== TO EXIT TYPE '#' ==":PRI
NTTAB(24)"== HIT 'ENTER' TO CONTINUE =="
340 PRINT"ADDRESS -- ";H$(X)
350 PRINT"PRICE -- ";P(X)
360 IFLEFT$(W$(X),1)="O"THENY$="OCEAN"ELSEIFLEFT$(W$(X),1)="C"TH
ENY$="CANAL"ELSEIFLEFT$(W$(X),1)="R"THENY$="RIVER"ELSEY$="NO"
370 PRINT"WATERFRONT -- ";Y$
380 IFLEFT$(PO$(X),1)="Y"THENY$="YES"ELSEY$="NO"
390 PRINT"POOL -- ";Y$
400 PRINT"LOT SIZE -- ";S$(X)
410 PRINT"NUMBER OF BEDROOMS -- ";B(X)
420 PRINT"NUMBER OF BATHROOMS -- ";BA(X)
430 A$=INKEY$:IFA$="#"THEN80
440 IFA$=CHR$(13)ANDX<B-1THENX=X+1:GOTO310
```

Program continues

"A compress program removes all spaces so it uses less memory and it runs faster."

H$(x)—Address
P(x)—Price
W$(x)—Waterfront (o/r/c/n)
PO$(x)—Pool (y/n)
S$(x)—Lot size
B(x)—Number of bedrooms
BA(x)—Number of bathrooms
B—Current file number
H1—Maximum number of files
Q—District number

Table 1. Important Variables.

• Save on cassette. This will save all files onto a cassette. You are prompted to enter a district number corresponding to the houses you have stored in memory.

• Load from cassette. This will read in files from a cassette that were previously saved. It will print the district number for that tape on its initial read of data.

• Change file. If you have made a mistake while entering information, this option will let you correct it. You are asked for the file number you want to change and once this is entered, you may begin to correct it.

• Delete file. It does just that: You enter a file number and it erases that file. However, if you are viewing files and it jumps from file six to file eight, it is because you have deleted file seven.

• Insert file. This will allow you to insert a file only if you have previously deleted that file. This was designed to be used if you sell a house and erase its file. Then you can insert a new file in its place. If any files are open the program will tell you.

• Clear all files in memory. This erases all information that is in memory. When you enter this command you will be asked if you are sure, a precaution against accidentally wiping out all of your files. If you answer N you will return to the menu with all your files intact.

The actual program is about 7.5K of code, although it uses all 16K of memory. The additional 9.5K is cleared and dimensioned for use as file space. If you own a 32K or 48K system, change lines 30 and 50 to increase the number of files that can be handled in memory at one time.

You may have noticed there are no spaces in the program. I used a compress program to remove them all. By doing this the program uses less memory and it runs faster. I would suggest that you type it in without any spaces also.

There are many error trapping routines used throughout the program. If you try to view or search the files while none are in memory, you will be returned to the menu. ■

Program continued

```
450 GOTO430
460 IFB=1THEN80
470 R$="":INPUT"DISTRICT NUMBER";Q
480 PRINT"SAVING";
490 PRINT#-1,Q,B
500 FORT=1TOB+2STEP3
510 PRINT#-1,H$(T);",";H$(T+1);",";H$(T+2);",";P(T);",";P(T+1);"
,";P(T+2);",";W$(T);",";W$(T+1);",";W$(T+2);",";PO$(T);",";PO$(T
+1);",";PO$(T+2);",";S$(T);",";S$(T+1);",";S$(T+2);",";B(T);",";
B(T+1);",";B(T+2);",";BA(T);",";BA(T+1);",";BA(T+2)
520 NEXTT
530 GOTO80
540
550 A1=B:INPUT#-1,Q,B
560 PRINT"DISK FOR DISTRICT";Q
570 R$="":PRINT@896,"";:PRINTCHR$(30);:INPUT"CONTINUE";R$:IFLEFT
$(R$,1)="N"THENB=A1:GOTO80ELSEIFLEFT$(R$,1)="Y"THEN580ELSE570
580 PRINT"LOADING";
590 FORT=1TOBSTEP3
600 INPUT#-1,H$(T),H$(T+1),H$(T+2),P(T),P(T+1),P(T+2),W$(T),W$(T
+1),W$(T+2),PO$(T),PO$(T+1),PO$(T+2),S$(T),S$(T+1),S$(T+2),B(T),
B(T+1),B(T+2),BA(T),BA(T+1),BA(T+2)
610 NEXTT
620 X=1:GOTO80
630 CLS:P=0:PRINTTAB(25)"CHANGE FILE":PRINT
640 R$="":PRINT@128,"";:PRINTCHR$(30);:INPUT"WHICH FILE DO YOU W
ANT CHANGED (-1 TO EXIT)";R$:IFR$=""THEN640ELSEP=VAL(R$)
650 IFP<1ORP>H1THEN80
660 IFH$(P)=""THEN80
670 IFP>B-1THEN80
680 GOTO1520
690 ONRGOTO700,730,740,760,780,790,800
700 R$="":PRINT@896,"";:PRINTCHR$(30);:INPUT"NEW ADDRESS? ";R$:I
FR$=""THEN700ELSEH$(P)=R$
710 IFLEN(H$(P))>55THENH$(P)=LEFT$(H$(P),55):PRINT@896,"";:PRINT
CHR$(30);:PRINT" -- LINE TO LONG  SHORTENED AS FOLLOWS --":FORR=
1TO500:NEXTR:GOTO1520
720 F2=1:GOTO1520
730 R$="":PRINT@896,"";:PRINTCHR$(30);:INPUT"NEW PRICE";R$:P(P)=
VAL(R$):IFP(P)=0THEN730ELSEF2=1:GOTO1520
740 PRINT@896,"";:PRINTCHR$(30);:INPUT"WATERFRONT (O/R/C/N)";W$(
P)
750 IFW$(P)="O"ORW$(P)="R"ORW$(P)="C"ORW$(P)="N"THENF2=1:GOTO152
0ELSE740
760 PRINT@896,"";:PRINTCHR$(30);:INPUT"POOL (Y/N)";PO$(P)
770 IFPO$(P)="Y"ORPO$(P)="N"THENF2=1:GOTO1520ELSE760
780 PRINT@896,"";:PRINTCHR$(30);:INPUT"NEW LOT SIZE";S$(P):F2=1:
GOTO1520
790 R1$="":PRINT@896,"";:PRINTCHR$(30);:INPUT"NEW NUMBER OF BEDR
OOMS";R1$:B(P)=VAL(R1$):IFB(P)=0THEN790ELSEF2=1:GOTO1520
800 R2$="":PRINT@896,"";:PRINTCHR$(30);:INPUT"NEW NUMBER OF BATH
ROOMS";R2$:BA(P)=VAL(R2$):IFBA(P)=0THEN800ELSEF2=1:GOTO1520
810 GOTO80
820 CLS:PRINTTAB(15)"DELETE A FILE"
830 INPUT"WHICH FILE DO YOU WANT DELETED (TYPE -1 TO EXIT)";P
840 IFP<1ORP>H1THEN80
850 IFP>B-1THENPRINT"NO SUCH FILE":FORS=1TO1000:NEXT:GOTO80
860 IFP=B-1THENB=B-1
870 H$(P)="":P(P)=0:W$(P)="":PO$(P)="":S$(P)="":B(P)=0:BA(P)=0
880 PRINT"FILE";P;"DELETED":FORS=1TO1000:NEXT:GOTO80
890 IFH$(1)=""THEN80ELSEV=1:L=1:W=B:DH=0:DL=0:CLS:PRINT"SEARCH T
HROUGH FILES":PRINT
900 FORK=1TOW:T(K)=0:NEXT
910 PRINT"PRICE RANGE:"
920 PRINT@192,"";:PRINTCHR$(30);:INPUT"FROM";R$:DL=VAL(R$):IFDL=
0THEN920
930 PRINT@256,"";:PRINTCHR$(30);:INPUT" TO ";R$:DH=VAL(R$):IFDH=
0THEN930
940 IFDL>DHTHEN920
950 PRINT@320,"";:PRINTCHR$(30);:INPUT"WATERFRONT (Y/N)";Y$
960 IFY$="Y"ORY$="N"THEN970ELSE950
970 PRINT@384,"";:PRINTCHR$(30);:INPUT"POOL (Y/N)";Y1$
980 IFY1$="Y"ORY1$="N"THEN990ELSE970
990 PRINT@448,"";:PRINTCHR$(30);:PRINT"NUMBER OF BEDROOMS: ":INP
UT"FROM";R$:B1=VAL(R$):IFB1=0THEN990
1000 PRINT@576,"";:PRINTCHR$(30);:INPUT"  TO";R$:B2=VAL(R$):IFB2
=0THEN1000
1010 IFB1>B2THEN990
1020 PRINT@640,"";:PRINTCHR$(30);:PRINT"NUMBER OF BATHROOMS:":IN
```

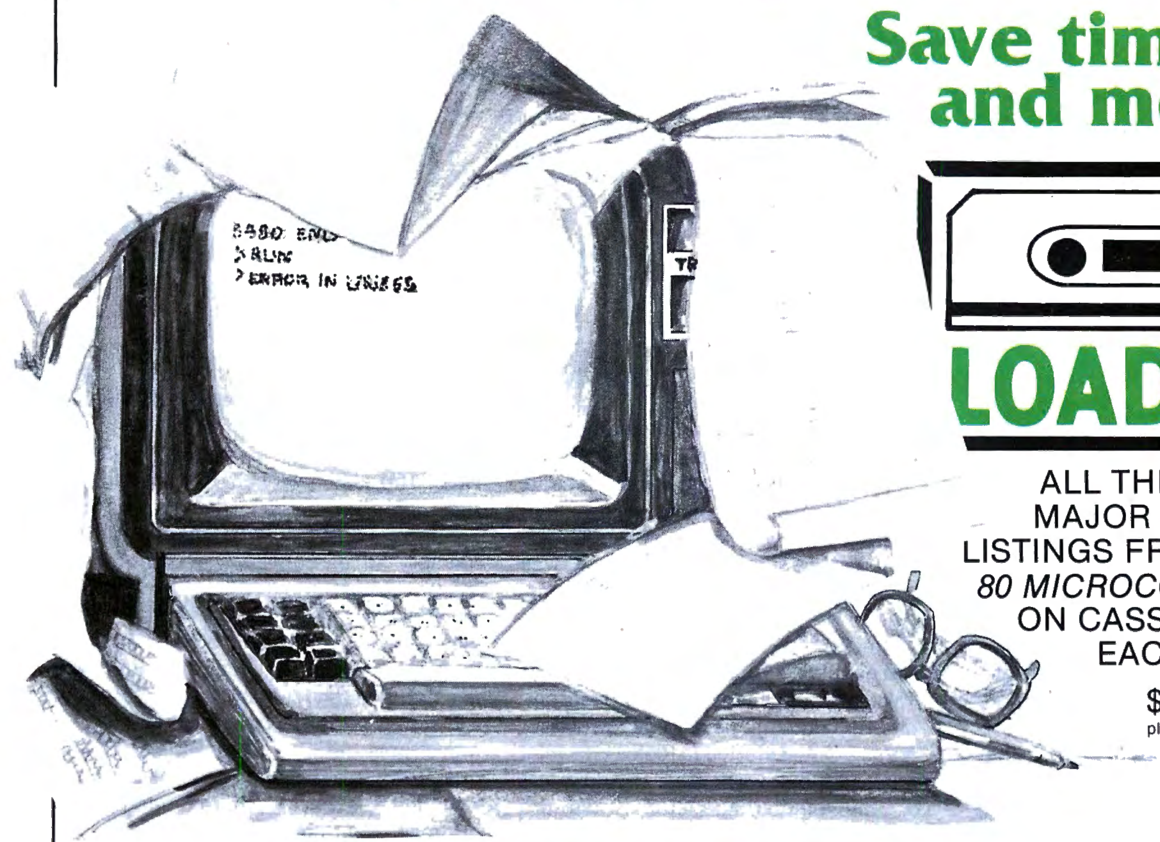
Program continues

Program continued

```
PUT"FROM";R$:BA=VAL(R$):IFBA=0THEN1020
1030 PRINT@768,"";:PRINTCHR$(30);:INPUT"   TO";R$:BB=VAL(R$):IFBB
=0THEN1030
1040 IFBA>BBTHEN1020
1050 IFW$(L)="O"THENZZ$="Y"ELSEIFW$(L)="R"THENZZ$="Y"ELSEIFW$(L)
="C"THENZZ$="Y"ELSEZZ$="N"
1060 IFP(L)>=DLANDP(L)<=DHANDZZ$=Y$ANDPO$(L)=Y1$ANDB(L)>=B1ANDB(
L)<=B2ANDBA(L)>=BAANDBA(L)<=BBTHEN1080
1070 L=L+1:IFL>WTHEN1090ELSE1050
1080 T(V)=L:V=V+1:L=L+1:GOTO1050
1090 CLS:PRINT"SEARCH COMPLETED":IF(V-1)=0THENII$="ARE":IH$="HOU
SES"
1100 IF(V-1)>1THENII$="ARE":IH$="HOUSES"
1110 IF(V-1)=1THENII$="IS":IH$="HOUSE"
1120 PRINT"THERE ";II$;(V-1);IH$;" WITH THOSE SPECIFICATIONS"
1130 IFT(1)=0THENFORM=1TO4000:NEXT:GOTO80ELSE1140
1140 R$="":PRINT@128,"";:PRINTCHR$(30);:INPUT"READY (Y/N)";R$:R$
=LEFT$(R$,1):IFR$="N"THEN80ELSEIFR$="Y"THEN1150ELSE1140
1150 CLS
1160 FORU=1TOV-1
1170 PRINT"HOUSE";U;"OUT OF";(V-1):PRINT"PRESS 'ENTER' TO CONTIN
UE":PRINT
1180 PRINT"ADDRESS -- ";H$(T(U))
1190 PRINT"PRICE --";P(T(U))
1200 IFW$(T(U))="O"THENY$="OCEAN"ELSEIFW$(T(U))="R"THENY$="RIVER
"ELSEIFW$(T(U))="C"THENY$="CANAL"ELSEY$="NO"
1210 PRINT"WATERFRONT POPERTY -- ";Y$
1220 IFPO$(T(U))="Y"THENY$="YES"ELSEY$="NO"
1230 PRINT"POOL -- ";Y$
1240 PRINT"LOT SIZE -- ";S$(T(U))
1250 PRINT"NUMBER OF BEDROOMS -- ";B(T(U))
1260 PRINT"NUMBER OF BATHROOMS -- ";BA(T(U))
1270 A$=INKEY$:IFA$=""THEN1270
1280 IFA$=CHR$(13)THENCLS:NEXTUELSE1270
1290 PRINT@0,"";:PRINTCHR$(30);:INPUT"DO YOU WANT A PRINTOUT (Y/
N) ";P1$
1300 IFLEFT$(P1$,1)="N"THEN80ELSEIFLEFT$(P1$,1)="Y"THEN1420ELSE1
290
1310 IFB-1=0THEN80ELSECLS:PRINTTAB(15)"INSERT FILE":PRINT
1320 PRINT"OPEN FILES: ";:FORC=1TOB-1
1330 IFH$(C)=""THENPRINTC;:J=1
1340 NEXTC
1350 IFC=BANDJ=0THENPRINT"NONE OPEN":FORC=1TO1500:NEXTC:GOTO80
1360 PRINT
1370 INPUT"WHICH FILE NUMBER (-1 TO EXIT)";P
1380 IFP<1ORP>H1THEN80
1390 IFP>B-1THEN80
1400 IFH$(P)=""THENPRINT@192,CHR$(30);:FL=1:Q=B:B=P:GOSUB150:B=Q
:FL=0:GOTO80ELSE80
1410 PRINT@832,CHR$(30);:FORC=1TO5:PRINT@832,"** ERROR **":FORC1
=1TO150:NEXTC1:PRINT@832,CHR$(30);:FORC1=1TO150:NEXTC1:NEXTC:RES
UME80
1420 IFPEEK(14312)>127THENPRINT"PRINTER NOT READY":GOTO1290ELSEP
RINTTAB(25)"PRINTING"
1430 LPRINT"FORMAT: (ADDRESS)  (PRICE)  (WATERFRONT)  (POOL)  (L
OT SIZE)  (# OF BEDROOMS)  (# OF BATHROOMS)"
1440 LPRINTSTRING$(64,"-")
1450 FORU=1TOV-1
1460 IFW$(T(U))="O"THENY$="OCEAN"ELSEIFW$(T(U))="R"THENY$="RIVER
"ELSEIFW$(T(U))="C"THENY$="CANAL"ELSEY$="NOT WATERFRONT"
1470 IFPO$(T(U))="Y"THENY1$="YES POOL"ELSEY1$="NO POOL"
1480 LPRINT"(";H$(T(U));")  ";:LPRINT"(";P(T(U));")  ";:LPRINT"(
";Y$;")  ";:LPRINT"(";Y1$;")  ";:LPRINT"(";S$(T(U));")  ";:LPRIN
T"(";B(T(U));")  ";:LPRINT"(";BA(T(U));")  "
1490 LPRINTSTRING$(64,"-")
1500 NEXTU
1510 GOTO80
1520 CLS:PRINTTAB(20)"CHANGE FILE":PRINT:IFPO$(P)="Y"THENQ1$="YE
S"ELSEQ1$="NO"
1530 IFW$(P)="O"THENQ$="OCEAN"ELSEIFW$(P)="R"THENQ$="RIVER"ELSEI
FW$(P)="C"THENQ$="CANAL"ELSEQ$="NO"
1540 R$=CHR$(94)+" ":PRINT"(1) -- ADDRESS ";R$;H$(P):PRINT"(2) -
- PRICE ";R$;P(P):PRINT"(3) -- WATERFRONT ";R$;Q$:PRINT"(4) -- P
OOL ";R$;Q1$:PRINT"(5) -- LOT SIZE ";R$;S$(P):PRINT"(6) -- NUMBE
R OF BEDROOMS ";R$;B(P):PRINT"(7) -- NUMBER OF BATHROOMS ";R$;BA
(P)
1550 IFF2=1THEN1570
1560 A$="":PRINT@896,"";:PRINTCHR$(30);:INPUT"WHICH NUMBER (1-7)
";A$:IFA$=""THEN1560ELSER=VAL(A$):IFR<1ORR>7THEN1560ELSE690
1570 F2=0:A$=" ":PRINT@896,"";:PRINTCHR$(30);:INPUT"MORE CORRECT
IONS (Y/N)";A$:A$=LEFT$(A$,1):IFA$=" "THEN1570
1580 IFA$="Y"THEN1560ELSEIFA$="N"THEN80ELSE1570
1590 A$="":PRINT@768,"";:PRINTCHR$(30);:INPUT"ARE YOU SURE (Y/N)
";A$:A$=LEFT$(A$,1)
1600 IFA$="N"THEN80ELSEIFA$="Y"THEN1610ELSE1590
1610 FORR=1TOX
1620 H$(R)="":P(R)=0:W$(R)="":PO$(R)="":S$(R)="":B(R)=0:BA(R)=0
1630 NEXTR:B=1:L=1:V=1:X=0:GOTO80
```

Tired of typing in all your programs?

Save time and money!



LOAD 80

ALL THE MAJOR PROGRAM LISTINGS FROM *80 MICROCOMPUTING* ON CASSETTE EACH MONTH

$9.95

plus $1.50 postage & handling

SAVE HOURS OF TYPING & AGGRAVATION

"Frankly, after hundreds of hours of frustration, I seldom ever try to keyboard a published program. Inevitably, I mess it up when I keyboard it. Who needs the aggravation?

"This is why I started a new series of cassettes called **LOAD-80**. Each cassette will have program dumps of the listings in an issue of *80 Microcomputing*. These listings are direct from the authors and tested by the *80 Microcomputing* staff. All but the very short program listings will be on the **LOAD-80** cassettes. Thus, you will be able to save hours of inputting programs and even more of debugging your keyboard errors."

Wayne Green

Wayne Green
Publisher, *80 Microcomputing*

The **LOAD-80** cassette is simply the program listings that appear in the articles in *80 Microcomputing*. It was created to save you the time involved in typing the listings yourself. Successful loading of the programs depends on reading the documentation in the articles. If you have your current magazine at hand when you load the cassette, you should have no difficulty. If you still have problems, please return the tape for replacement.

LOAD-80 began with the April 1981 issue. To order back issues, look for the back issue advertisement in this magazine or ask your local dealer.

To order **LOAD-80**, fill in the attached card and we will send you your **LOAD-80** cassette for the major programs in this issue. If the card is missing, fill in the coupon below (a photocopy is acceptable) and mail to: **LOAD-80**, 80 Pine Street, Peterborough, NH 03458.

11/81

Yes, I want the November LOAD-80

☐ Check/Money Order ($9.95 plus $1.50 p&h) ☐ MasterCard ☐ VISA ☐ Am. Ex.

Name: ____________________

Address: ____________________

City: ____________________ State: ________ Zip: ________

Signature: ____________________ Card # ____________________ Exp. Date: ____________

LOAD-80 is manufactured by Instant Software, a division of WAYNE GREEN INC., 80 Pine Street, Peterborough, NH 03458. There is no warranty expressed or implied that **LOAD-80** will do anything other than save you typing.

Centronics . . . shake hands with the Shack.

A Driven Printer

Wayne Mueller
130 Sunset Court
Roswell, GA 30075

Recently my Selectric printer was stolen. Since I need a letter-quality printer, I thought about another Selectric. But then I thought about how slow, noisy, and how much the old Selectric needed to be readjusted. I finally settled on an NEC 5530 Spinwriter. I knew it was faster, quieter and more reliable. And since it had a Centronics interface, I could "plug in and print," and forget about those nuisance driver routines I had to use with the Selectric.

But after loading Basic and typing in a few lines of LPRINTs, all the lines printed on top of each other! When all else fails, read the manual. Sure enough, the manual says that when a carriage return is received, the carriage returns. . .nothing about a trailing line feed. Well, so much for my belief that no drivers were needed.

There are actually some advantages to a nonautomatic line feed following a carriage return. For example, Software Concepts has a slick mail processing system that supports both underscores and bold-face characters on such printers, and I was tempted to just leave well enough alone. But then I realized that Scripsit's line feed and paragraph spacing was off by one, that I could not get a printout from the editor/assembler, and that nearly all of my Basic programs were going to have to be reworked.

At this point, it started to look a lot easier to write a little driver routine to add a line feed after each carriage return. It did turn out that I needed more than one driver. This article may benefit a few others who find themselves in the same fix.

Look at Program Listing 1. The TRS-80 has several device control blocks (DCBs). The DCB which starts at 4025H is normally used for the line printer. So, what we are doing at lines 130–150 is telling the machine that we have a line printer (line 140), and the address of the driver routine is labeled Busy. The assembler can determine the correct numeric address from the label. At lines 160–170, we are protecting some high memory for this driver. The TRS-80 stores the end of unprotected memory in 4049H. What we told the assembler here was that we wanted unprotected memory to end two words less than Busy. Again, the assembler is smart enough to do the arithmetic to find the actual address. Notice that we have not moved any data yet. If just this part were loaded, that would be sufficient to tell the machine to expect a different printer-driver address, and to protect high memory.

```
                  00100 ;GENERAL HI-MEM NEC SPINWRITER DRIVER
                  00110 ;WITH SCREEN DISPLAY INCLUDED
4467              00120 DISPLA  EQU     4467H   ;DOS DISPLAY CALL
4025              00130         ORG     4025H   ;FIX PRINTER DCB TO:
4025 02           00140         DEFB    2       ; -LOOK LIKE A PRINTER
4026 68FF         00150         DEFW    BUSY    ; -HOLD NEW DRIVER ADDRESS
4049              00160         ORG     4049H   ;SET MEMORY PROTECT
4049 66FF         00170         DEFW    BUSY-2  ;  TO JUST BELOW THIS DRIVER
FF50              00180         ORG     0FF50H  ;** OR USE WHATEVER OTHER START
                  00190                         ; LOCATION IS RIGHT
                  00200                         ; FOR YOUR SIZE MACHINE **
FF50 E5           00210 START   PUSH    HL      ;SAVE HL
FF51 2159FF       00220         LD      HL,MSG  ;POINT TO START OF MESSAGE
FF54 CD6744       00230         CALL    DISPLA  ;CALL DOS TO DISPLAY MESSAGE
FF57 E1           00240         POP     HL      ;RESTORE HL
FF58 C9           00250         RET             ;AND RETURN
                  00260 ;OR JUMP TO DOS (402DH) IF YOU PREFER
FF59 4E           00270 MSG     DEFM    'NEC DRIVER ON' ; TEXT OF DISPLAY
FF66 0D           00280         DEFB    0DH     ;CARRIAGE RETURN AFTER MSG
FF67 00           00290         DEFB    0       ;END OF MESSAGE
FF68 3AE837       00300 BUSY    LD      A,(37E8H)       ;GET PRINTER STATUS
FF6B E6F0         00310         AND     0F0H    ;GET UPPER
FF6D FE30         00320         CP      30H     ; PART OF STATUS WORD
FF6F 20F7         00330         JR      NZ,BUSY ;  KEEP TRYING IF NOT READY
FF71 79           00340         LD      A,C     ;GET CHARACTER TO PRINT
FF72 32E837       00350         LD      (37E8H),A       ;CHARACTER TO PRINTER
FF75 79           00360         LD      A,C     ;TAKE ANOTHER LOOK AT CHARACTER
FF76 FE0D         00370         CP      0DH     ;IS IT CARRIAGE RETURN?
FF78 C0           00380         RET     NZ      ;IF NOT, WE'RE DONE
FF79 0E0A         00390         LD      C,0AH   ;  IF SO, LOAD LINE FEED,
FF7B 18EB         00400         JR      BUSY    ;    AND OUTPUT IT
FF50              00410         END     START   ;AUTO START
```

Program Listing 1

The Driver

The ORG at line 180 says we want the driver to start at FF50H, or 65360 decimal. If you want it somewhere else, or have a smaller machine, use some other value. For example, 7F50H (32592 decimal) would work well on a 16K machine. The printer port on the TRS-80 is mapped to 37E8H. Line 300 gives us the printer status. The printer returns to its state of readiness to receive characters to this address. Between lines 300–330, we are looking for a printer ready signal. If it is not ready, we keep looping back to Busy at line 300. When we get to line 340, the printer is ready to accept a character.

The next character comes to us in register C, just because that is where the machine puts it. So we load it into register A at 340, and send it to the printer port at line 350.

Up to here we have not done anything differently from the standard ROM driver. But now is when we take care of the needed line feed. The last character is still in register C, so we load it again at line 360, and see if it was a carriage return at line 370. If not, we are done, and we return control to whatever routine called us. If it was a carriage return, we load a line feed into register C at line 390, and jump back to the start of the driver (Busy) at line 400. This driver takes care of

"I find it disquieting when I run a program, and nothing happens . . ."

almost all of our printing needs.

Program Listing 2

This is a bare bones, minimum routine. Specifically, I find it somewhat disquieting when I run a program, and nothing happens on the screen to confirm that it is working. Program Listing 2, is similar to Listing 1 and also displays a message to let us know it is working. The line numbers are the same as Listing 1, but we have added a few things in the empty spaces.

Notice line 120. There is a label called Display, which refers to a DOS routine at 4467H that displays text. Also notice that lines 210–290 are added. Lines 270–290 define the message, while 210–250 do the work. The text of the message "NEC Driver On" is at line 270, a following carriage return is at line 280, and the zero at line 290 simply tells DOS that this is the end of the message.

Now let's see how it is displayed. The HL register pair must point to the start of the message text, but HL may be in use for something else. So, we save HL at line 210 by PUSHing it onto the stack, point it to the beginning of the message at line 220, and call Display at line 230 to display the text on the screen. Finally, we restore HL to its previous condition by POPping it off the stack at line 240, and return to the calling routine. The only other change is at line 410. We have added the label Start after the end to tell the loader that we want to jump to Start automatically after loading the routine. These changes do not affect the printer driver, but now I can see something happening on the screen when I load the routine.

Other Annoyances

This takes care of almost all my printer driver needs but there are still a few annoyances that need attention. The first is Scripsit's behavior. Scripsit works fine until nearly all of the text buffer is full. Then it proceeds to overwrite my driver with text. Worse, it does not tell me until I try to print, which results in a spectacular crash, and a machine lockup.

The second item is those Basic programs that already protect high memory, and use the protected space for some other machine-language routine. Some of these other routines behave nicely if you just load the printer driver first, and the other routine just above it. But some of them use all of protected memory for their own purposes, and again overwrite my driver with unpleasant results.

Fortunately, Radio Shack offers some patch code that modifies Scripsit to work with a variety of non-Radio Shack printers.

Program Listing 3 is mostly Radio Shack's, with my driver embedded. Not all of the Radio Shack code was used, because not all was needed. A few things in the patch code are worth noting. We see at lines 310–320, that Scripsit does not use the value stored at 4049H for memory protect. Rather, whatever you want to declare as protected memory must be stored in 5268H. This is why Scripsit used to overwrite my driver.

My driver runs from lines 560–680, and it looks a bit different from what we have discussed so far. The first thing we do at line 560 is output a character from the A register. This is because Scripsit has already put the character in A, and checked to make sure the printer is ready. From here on, things start to look familiar. That is, we check to see if the last character was a carriage return, and output a line feed if it was. At line 620, the printer status checks again for line feed, so that Scripsit knows we want to print an extra character, and will check to see if the printer is ready.

Notice line 710. If you do not have lower-

```
                 00100 ;GENERAL HI-MEM NEC SPINWRITER DRIVER
4025             00130           ORG     4025H    ;FIX PRINTER DCB TO:
4025 02          00140           DEFB    2        ; -LOOK LIKE A PRINTER
4026 50FF        00150           DEFW    BUSY     ; -HOLD NEW DRIVER ADDRESS
4049             00160           ORG     4049H    ;SET MEMORY PROTECT
4049 4EFF        00170           DEFW    BUSY-2   ;  TO JUST BELOW THIS DRIVER
FF50             00180           ORG     0FF50H   ;** OR USE WHATEVER OTHER START
                 00190                            ; LOCATION IS RIGHT
                 00200                            ; FOR YOUR SIZE MACHINE **
FF50 3AE837      00300 BUSY      LD      A,(37E8H)       ;GET PRINTER STATUS
FF53 E6F0        00310           AND     0F0H     ;GET UPPER
FF55 FE30        00320           CP      30H      ; PART OF STATUS WORD
FF57 20F7        00330           JR      NZ,BUSY  ;  KEEP TRYING IF NOT READY
FF59 79          00340           LD      A,C      ;GET CHARACTER TO PRINT
FF5A 32E837      00350           LD      (37E8H),A       ;CHARACTER TO PRINTER
FF5D 79          00360           LD      A,C      ;TAKE ANOTHER LOOK AT CHARACTER
FF5E FE0D        00370           CP      0DH      ;IS IT CARRIAGE RETURN?
FF60 C0          00380           RET     NZ       ;IF NOT, WE'RE DONE
FF61 0E0A        00390           LD      C,0AH    ;  IF SO, LOAD LINE FEED,
FF63 18EB        00400           JR      BUSY     ;    AND OUTPUT IT
0000             00410           END
```

Program Listing 2

Program Listing 3

```
                 00100 ;THIS ROUTINE LOADS AND MODIFIES SCRIPSIT TO PERMIT
                 00110 ;USE OF YOUR OWN PRINTER DRIVER
                 00120 ;
                 00130 ;IT SHOULD BE LOCATED IN HIGH RAM.THIS EXAMPLE IS
                 00140 ;LOCATED AT 0FF60H FOR A 65 K MACHINE
                 00150 ;
FF60             00160           ORG     0FF60H
FF60             00170 PARLEL    EQU     0FF60H           ;ENTRY ADDRESS
FF60 E5          00180           PUSH    HL
FF61 21E3FF      00190           LD      HL,SWITCH        ;TEST FOR 1ST CALL
FF64 34          00200           INC     (HL)
FF65 35          00210           DEC     (HL)
FF66 E1          00220           POP     HL
FF67 2038        00230           JR      NZ,PATCH         ;GO IF NOT 1ST CALL
FF69 11E4FF      00240           LD      DE,DCBADR        ;DE=> FILE SPEC
FF6C CD3044      00250           CALL    LOAD             ;OPEN AND LOAD SCRIPSIT
FF6F CD2844      00260           CALL    CLOSE            ;CLOSE COMMAND FILE
FF72 3E01        00270           LD      A,1              ;SET SWITCH TO SKIP THIS
FF74 32E3FF      00280           LD      (SWITCH),A       ;CODE ON FUTURE CALLS
FF77 3E21        00290           LD      A,21H            ;PATCH SCRIPSIT
FF79 326752      00300           LD      (5267H),A        ; TO PROTECT
FF7C 215FFF      00310           LD      HL,PARLEL-1      ; THIS DRIVER
FF7F 226852      00320           LD      (5268H),HL
FF81 323F66      00340           LD      (663FH),A        ;YOUR INIT. ROUTINE
FF84 3ECD        00350           LD      A,0CDH           ;PATCH SCRIPSIT TO RUN
FF86 329E7A      00360           LD      (7A9EH),A        ; YOUR DRIVER ROUTINE
FF89 32977A      00370           LD      (7A97H),A
FF8C 21DFFF      00380           LD      HL,INIT          ;ADD OF INITIALIZATION
FF8F 224066      00390           LD      (6640H),HL       ; ROUTINE
FF92 2160FF      00400           LD      HL,PARLEL        ;ADDRESS OF YOUR DRIVER
FF95 229F7A      00410           LD      (7A9FH),HL       ;    ROUTINE
FF98 22987A      00420           LD      (7A98H),HL
FF9B C30052      00430           JP      5200H            ; ENTER SCRIPSIT
FF9E ED43EDFF    00440 PATCH     LD      (SAV1),BC        ; SAVE THE REGISTERS
FFA2 ED53EFFF    00450           LD      (SAV2),DE
FFA6 22F1FF      00460           LD      (SAV3),HL
FFA9 ED73F3FF    00470           LD      (SAV4),SP
FFAD 32F5FF      00480           LD      (SAV5),A
```

Program continued

"This means that we have to change a few pointers so List and Edit know where to look. . ."

Program continued

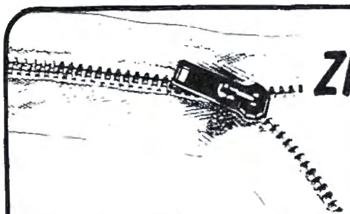
```
FFBØ CDC6FF   ØØ49Ø          CALL    OUTPUT
FFB3 ED4BEDFF ØØ5ØØ          LD      BC,(SAV1)        ;  REPLACE THE REGISTERS
FFB7 ED5BEFFF ØØ51Ø          LD      DE,(SAV2)
FFBB 2AF1FF   ØØ52Ø          LD      HL,(SAV3)
FFBE ED7BF3FF ØØ53Ø          LD      SP,(SAV4)
FFC2 3AF5FF   ØØ54Ø          LD      A,(SAV5)
FFC5 C9       ØØ55Ø          RET
FFC6 32E837   ØØ56Ø OUTPUT   LD      (37E8H),A        ;OUTPUT CHARACTER
              ØØ57Ø ;PRINTER STATUS  IS OK--SCRIPSIT CHECKED FIRST
FFC9 3AF5FF   ØØ58Ø          LD      A,(SAV5)         ;TAKE ANOTHER LOOK
FFCC FEØD     ØØ59Ø          CP      ØDH              ;IS IT CARRIAGE RETURN?
FFCE CØ       ØØ6ØØ          RET     NZ               ;IF NOT WE'RE DONE
FFCF ØEØA     ØØ61Ø          LD      C,ØAH    ;PUT LINE FEED IN C,
FFD1 3AE837   ØØ62Ø BUSY1    LD      A,(37E8H)        ;GET PRINTER STATUS
FFD4 E6FØ     ØØ63Ø          AND     ØFØH     ;CHECK UPPER
FFD6 FE3Ø     ØØ64Ø          CP      3ØH      ;  PART OF STATUS BYTE
FFD8 2ØF7     ØØ65Ø          JR      NZ,BUSY1         ;IF BUSY KEEP TRYING
FFDA 79       ØØ66Ø          LD      A,C      ;PRINTER READY-GET LF IN A,
FFDB 32E837   ØØ67Ø          LD      (37E8H),A        ;AND OUTPUT IT.
FFDE C9       ØØ68Ø          RET              ;ALL DONE--BACK TO SCRIPSIT
FFDF C9       ØØ69Ø INIT     RET                      ;RETURN
FFEØ ØØ       ØØ7ØØ SWITCH   DEFB    Ø        ;BUCKET TO STASH SWITCH
FFE1 53       ØØ71Ø DCBADR   DEFM    'SCRIPSIT/LC'    ; OR USE 'SCRIPSIT/UC'
FFEC Ø3       ØØ72Ø          DEFB    3                ;MARK END OF FILESPEC
              ØØ73Ø ;
443Ø          ØØ74Ø LOAD     EQU     443ØH            ;CALL DOS TO LOAD FILE
4428          ØØ75Ø CLOSE    EQU     4428H            ;CALL DOS TO CLOSE FILE
              ØØ76Ø ;
FFED ØØØØ     ØØ77Ø SAV1     DEFW    ØØØØ             ;REGISTER SAVE AREA
FFEF ØØØØ     ØØ78Ø SAV2     DEFW    ØØØØ
FFF1 ØØØØ     ØØ79Ø SAV3     DEFW    ØØØØ
FFF3 ØØØØ     ØØ8ØØ SAV4     DEFW    ØØØØ
FFF5 ØØ       ØØ81Ø SAV5     DEFB    ØØ
FF6Ø          ØØ82Ø          END     PARLEL
```

case, you would substitute Scripsit/UC. Also notice lines 250–260 and 740–750. These refer to DOS calls which bring Scripsit into memory, and close the file. This program loads Scripsit; you don't have to.

The Last Problem Solved

There are some machine language routines you might want to use from Basic that treat all of protected memory as their exclusive domain. Sometimes another routine in high memory works, sometimes not. My approach was to relocate the Basic program text area upwards, and run the driver from where the Basic text used to be. This means that we have to change a few pointers, so routines like List and Edit know where to start looking, and CLOAD knows where to put what it is reading. The hardest part was finding the right pointers.

Program Listing 4 is similar to Listing 2. At lines 160–170, the assembler was instructed to determine the address two bytes beyond the label Last, and store that in 40A4H, which is where Load and CLOAD can determine where to start putting the



ZIP UP YOUR MODEL I

SPEED-UP UNIT - an easy to install electronic device that enables programs to run 2 to 3 times faster. It's the fastest and finest quality speed-up unit on the market.

- guaranteed to double processing speed - no additional purchases required.
- programs will run up to 3 times faster (5.3 mhz) by adding a Z80B microprocessor and delay line - not supplied.
- returns automatically to normal speed during disk and cassette operation - no software patches required.
- keyboard power-on light changes color to indicate operating speed.

$45.00

VIDEO I. Reverses the video display to provide black characters and graphics on an all white screen for a much easier to read presentation. Software controlled. Cures pulling and distortion problems commonly experienced with TRS-80 monitors. For use with TRS-80 monitors only. Assembled.

$24.00

"SATISFACTION GUARANTEED"

Add $2.00 postage & handling - California residents add 6% sales tax -Foreign orders add 15%.

ARCHBOLD ELECTRONICS

10708 Segovia Way Rancho Cordova, CA 95670
(916) 635-5408
Dealer inquiries invited

TRS-80 CASE

ATTACHE STYLE CASES FOR CARRYING AND PROTECTING A COMPLETE COMPUTER SET-UP. CONSTRUCTED OF THE HIGHEST QUALITY LUGGAGE MATERIAL WITH SADDLE STITCHING WILL ACCOMMODATE EQUIPMENT IN A FULLY OPERATIONAL CONFIGURATION ALONG WITH MANUALS, WORKING PAPERS AND DISKS. NEVER A NEED TO REMOVE EQUIPMENT FROM CASE. SIMPLY REMOVE LID, CONNECT POWER AND OPERATE. LID CAN BE REPLACED AND LOCKED FOR SECURITY AND PROTECTION WITHOUT DISCONNECTING CABLES. FULLY TESTED.

- AP101S Apple and Single Disk Drive $109
- AP102D Apple and Double Disk Drive 119
- AP103M Apple, 9 inch Monitor and Double Drive 129
- RS201 TRS-80, Expansion Unit and Double Drive 109
- RS202 TRS-80 Monitor and Accessories 84
- P401 Paper Tiger Printer 99
- P402 Line Printer II-Centronics 730 89
- CC90 Matching Attache Case 75

master charge VISA®

✓199 COMPUTER CASE COMPANY
5650 INDIANA MOUND CT COLUMBUS OHIO 43213
(614) 868-9464

Make More Money!

MSI

Midwest Scientific Instruments

Profitability, when selecting a business computer, means buying a system that will save you enough money to pay for itself in a reasonable amount of time. *Midwest Scientific Instruments* has the most advanced microcomputer in the marketplace, excelling in hardware design and software support. Our computers will handle a larger business, faster, yielding you more results for your investment.

Let *Midwest Scientific Instruments* help you make more money. We have systems for your business, scientific, educational, professional, industrial control, or personal applications. Inquire today.

For more information about MSI systems, products, and OEM components, fill out the coupon below (a photocopy is acceptable) and mail to:

✓309

Midwest Scientific Instruments
220(A) W. Cedar, Olathe, Kansas 66061

YES, I want to make more money!

Name ______________________
Address ______________________
City ________ State ________ Zip ________

My application is: ☐ Personal ☐ Business ☐ Scientific ☐ Educational ☐ Professional ☐ Industrial Control ☐ Other (specify) ________

✓See List of Advertisers on page 402

"We are not protecting memory here because this routine is already protected."

text. At lines 180–190, we are changing the next available byte pointer to contain the address four bytes beyond Last. Line 200 instructs this routine to start where Disk Basic text used to start. If you want to use this for Level II Basic, set the ORG at 42E9H (17129 decimal). About the only other differences are the jump to Basic at line 250, and the zeros at Lines 400–420. Basic text is always terminated by a series of zeros, so that List and LLIST know where to stop.

There is one omission from Listing 2. We are not protecting memory here. That is because this routine is already protected by being in a place where other routines are not likely to interfere. You can protect high memory when you enter Basic and use all of it for whatever other machine language routines you might need.

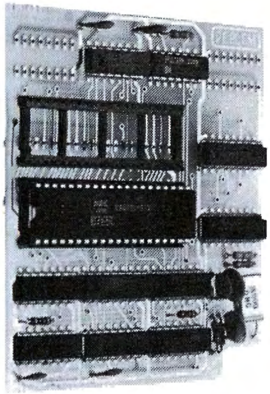
The Spinwriter is a nice piece of equipment, and I'm thoroughly pleased with it, now that I have the software to drive it properly. ■

Cassette and disk versions of the listings in this article are available from the author.

```
                00100 ;ROUTINE RELOCATES START OF BASIC PROGRAM TEXT, SO
                00110 ;NEC SPINWRITER DRIVER CAN RESIDE IN LOW MEMORY.
4467            00120 DISPLA  EQU     4467H    ;DOS DISPLAY ROUTINE
4025            00130         ORG     4025H    ;PRINTER DEVICE CONTROL BLOCK
4025 02         00140         DEFB    2        ; DESCRIBE AS PRINTER
4026 606A       00150         DEFW    BUSY     ; ADDR OF DRIVER==>4026
40A4            00160         ORG     40A4H    ;CLOAD START POINTER
40A4 776A       00170         DEFW    LAST+2   ; STASH END DRIVER +2
40F9            00180         ORG     40F9H    ;NEXT FREE TEXT BYTE POINTER
40F9 796A       00190         DEFW    LAST+4   ; STASH END DRIVER +4
6A46            00200         ORG     6A46H    ;WHERE BASIC TEXT USED TO BE
6A46 E5         00210 START   PUSH    HL       ;SAVE HL
6A47 21516A     00220         LD      HL,MSG   ;POINT TO MESSAGE
6A4A CD6744     00230         CALL    DISPLA   ;CALL DOS TO DISPLAY MESSAGE
6A4D E1         00240         POP     HL       ;RESTORE HL
6A4E C3191A     00250         JP      1A19H    ;JUMP TO BASIC
6A51 4E         00260 MSG     DEFM    'NEC DRIVER ON' ;MSG FOR DISPLAY
6A5E 0D         00270         DEFB    0DH      ;CARRIAGE RETURN AFTER MSG
6A5F 00         00280         DEFB    0        ;END OF MESSAGE
6A60 3AE837     00290 BUSY    LD      A,(37E8H)        ;GET PRINTER STATUS
6A63 E6F0       00300         AND     0F0H     ; AND SEE
6A65 FE30       00310         CP      30H      ;  IF READY BIT IS OK
6A67 20F7       00320         JR      NZ,BUSY  ;IF NOT, KEEP CHECKING
6A69 79         00330         LD      A,C      ; ELSE GET CHARACTER TO PRINT
6A6A 32E837     00340         LD      (37E8H),A        ;AND OUTPUT TO PRINTER
6A6D 79         00350         LD      A,C      ;TAKE ANOTHER LOOK AT CHARACTER
6A6E FE0D       00360         CP      0DH      ; IS IT CARRIAGE RETURN?
6A70 C0         00370         RET     NZ       ;  IF NOT, WE'RE DONE
6A71 0E0A       00380         LD      C,0AH    ;  IF SO, PUT LINE FEED IN C,
6A73 18EB       00390         JR      BUSY     ;    AND KEEP GOING.
6A75 0000       00400 LAST    DEFW    0000     ;TELL LIST, ETC WHERE TO QUIT
6A77 0000       00410         DEFW    0000     ; LOOKING FOR MORE
6A79 0000       00420         DEFW    0000
6A46            00430         END     START    ;AUTO START
```

Program Listing 4

Same Old Savings Galore.



Upgrade Your Model I to Double Density

only $159.95

And not with some unproven hacker's kit, either. But with an assembled and fully tested *Percom DOUBLER-II*†. The DOUBLER-II† simply plugs into the disk drive controller socket of your Expansion Interface. Then it lets you store over four times more on one side of a diskette, depending on the drive, than you can store on standard Tandy Model I drives. Other features: • Reads, writes and formats *either* single- or double-density • Includes DBLDOS™, a TRSDOS* compatible disk-operating system • Runs TRSDOS*, NEWDOS††, Percom OS-80† and other single-density software immediately. Change to double-density operation when convenient • Lets Model III software be read on your Model I • Includes on-card high-performance data separation and write precompensation circuitry.

Microline-80 Printer

Grand-Opening Special $399.95 (REGULAR $446.00)

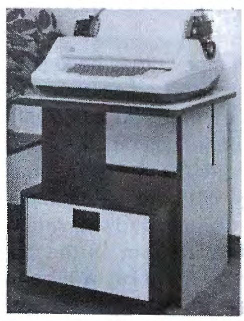
Compare the Microline-80 feature-for-feature with *any* other competitively priced printer. • Uses standard roll paper, fan-fold forms or individual sheets. • Prints expanded characters and *graphics*. • Prints 96 ASCII characters (9x6 dot format), 64 graphic characters (12x6 dot format). • Throughput is 80-characters per second, standard-width characters. • Print head is rated at 200 million characters. • Compact, lightweight, quiet.

Low-Cost System Furniture

Desks from $115.

Printer Stand only $144.

Modular design accommodates customized arrangements • Furniture quality styling and construction • TRS-80* compatible colors • Units knockdown for easy transportation.



Save! Make ACCESS your one-call shopping center for TRS-80* hardware and software. Call our toll-free order number for free product literature and latest prices.

AUTHORIZED PERCOM SALES AND SERVICE CENTER

Toll-Free Order No.

1-800-527-3475◇

(orders & literature only)

ACCESS UNLIMITED

401 N. Central Expressway #600
Richardson, Texas 75080
(214) 690-0206

™Trademark of Access Unlimited
†Trademark of Percom Data Company, Inc.

229

*Trademark of Tandy Radio Shack Corporation.
††Trademark of Apparat Corporation.

Now for some fancy schmancy printing. . .

Scripsit—Sans Serif

Gary D. Cochrell
1617 Sagebrush Ct. SE
Albuquerque, NM 87123

How many times have you had troubles using the "special print" functions on your line printer from Scripsit? If you have ever experienced difficulties, I think you will appreciate the following modification to Scripsit. With this patch in place all you do when you are ready to print is press two keys. The first key is the print command "P". The second key will be for the special print feature you want. There's no need to reserve high memory and no special program to load except Scripsit.

Before you start this project, however, make sure you have the following:

- A backup copy of Scripsit in its original form. A copy with a block move routine appended to it will not work.
- A utility program that can access disk sectors and modify them like Apparat's Superzap or Small Software Systems' RSM-2D.
- An ability to write Z80 machine language instructions.

I am assuming that a backup of the original Scripsit disk will be the same regardless of when or where the program was purchased. Radio Shack should not have moved the program around on the disk from one production run to the next. So the first precaution to take before you make any changes is to check your sector dumps with the ones I show and make sure they match.

Printer Commands

Before you make any modifications to the disk, customize the Program Listing to fit your printer characteristics. That is why you must be familiar with the Z80 instruction set. The listing is the program that I use with my printer, the Epson MX-80. The following information will help you modify the program to accommodate your printer.

First decide what features in your printer you would like to access from Scripsit. I had four types of printing I wanted to use, but I was only able to incorporate two. Here is why: When the Epson printer is programmed for expanded printing, 40 characters per line, it will default to its normal 80 characters per line every time a line feed/carriage return is sent to the printer. To continue printing in the expanded mode the appropriate command must be sent preceding every line. This trick was a little more than I could handle inside a machine language program like Scripsit, so I dropped this mode from my list. Once you have decided on the commands you want, check to make sure they will not default when a line feed/carriage return is transmitted to the printer.

Program Modifications

Having made the decision about which printer commands you want, convert their decimal values to hex numbers (all references to hex numbers in this article will be followed by a capital H). For instance, if you want to print in the compressed mode, 132 characters per line, send a CHR$(15) to the printer. Decimal 15 converts to 0FH. If you look at line 7AD0H of the listing, you will see that 0FH has been loaded into register C. The other mode included in the program is the double pass mode which causes the print head to print a line twice for a much darker appearance. This is accomplished by sending two commands to the printer, 1BH and 47H. Lines 7AD7H and 7ADCH are where these commands are loaded.

If you have a one-command feature in your printer that you want to use, then insert that command in line 7AD0H. Likewise, any two-command modes can be placed into lines 7AD7H and 7ADCH. If you have more commands to add, then you will have to modify the program. You can add up to 100 bytes before you run out of room.

You should be aware of the following steps if you decide to modify the program. Load all printer commands into register C before calling the subroutine which sends the command to the printer. When you add lines to the program, most if not all of the relative jumps will change. At the end of the program be sure to include the lines:

```
CD  C9  70    CALL  70C9
C3  64  70    JP    7064
```

You may also want to change the prompts that the program places on your monitor. The program as it stands will place the letters C, D and N on the lower right side of the monitor. These letters stand for compressed, dark and normal. Lines 7AADH, 7AB2H and 7AB7H change these letters.

Disk Modifications

There will be four sectors on the disk that will be altered: two for Scripsit/UC and two for Scripsit/LC. I will start with the modifications on Scripsit/LC. If you are using Super-

"Now if this explanation has made any sense at all, the instructions to follow will be a snap."

zap, display track 0BH, sector 0. If you are using RSM-2D, load sector 110 to a convenient place in memory like 6000H. This is a good time for me to explain a term I will be using frequently—relative byte.

For RSM-2D users I am referring to the last two digits of a memory address. For instance, if you have just loaded the sector to location 6000H, and I mention relative byte DDH, you will be looking at 60DDH. If you have loaded the sector to another location, such as 4300H as an example, then you would be looking at 43DDH. In either case, relative byte DDH contains the same information.

With a Superzap dump, relative rows 00H through F0H are shown in the last two digits of the first column. For example, the first column of relative row D0H from Fig. 1 looks like this: 00B0D0.

To find relative byte DDH drop down to relative row D0H and count over, in hex, 14 places. Now if this explanation has made any sense at all, the instructions to follow will be a snap.

If track 0BH, sector 0 is displayed, compare your display with Fig. 1. Except for relative bytes DDH, DEH, and DFH, the underlined bytes, your display should look like Fig. 1. Relative bytes DDH, DEH, and DFH should be: CD C9 70. Modify these three bytes to read: C3 A5 7A.

This change causes Scripsit to jump to location 7AA5H—the location for our "Program the Printer" routine. The changes to this sector are complete. Save the sector back at its original location on the disk.

Next load track 0CH, sector 1. This is sector 121 for users of RSM-2D. Compare your dump with Fig. 2. Be very careful—a lot of changes take place in this sector. In Fig. 2 relative bytes 49H, 4AH, 4BH, and 4CH are underlined. All the hex code that follows relative byte 4CH to the end of the sector is just filler, it is not part of Scripsit. The underlined bytes tell the disk program loader that the end of Scripsit/LC has been reached (0202) and the next two bytes are the start of the program (0052), which is 5200H. These four bytes will be moved to the end of the sector. Modify the four bytes at relative positions FCH, FDH, FEH and FFH to read: 02 02 00 52. The lower right corner of Fig. 3 shows this change.

Next we want to tell the disk program loader that the end of the Scripsit program has been moved to the end of the sector. If you look at Fig. 3 you will see that relative bytes 49H, 4AH, 4BH and 4CH are underlined again but they contain a new code. The 01 tells the disk loader that there is more program to be loaded. The B1 tells the loader that there are 177 more bytes to be loaded, and A5 7A will direct these bytes to memory location 7AA5H. Insert the following bytes at relative positions 49H, 4AH, 4BH and 4CH: 01 B1 A5 7A.

An additional 177 bytes within Scripsit is now available in which to insert our own program. Don't worry; the disk directory has always had this entire sector reserved for Scripsit. The additional 177 bytes at the end of Scripsit are protected when loaded into memory because the program scratch pad memory starts at 7C00H.

All that remains now is to modify (or edit) the relative bytes beginning with 4DH with the program listing or one you have developed for your printer. After I inserted my program, I went ahead and zeroed out the remainder of the sector to make it easier to see the changes made to the sector. Once you have inserted your program, you can

```
00B000 C3C8 6BCD E96B ED5B 437C AF32 3A60 3256  ........C..2:..2V
00B010 603A 327C B728 07FD 3603 00CD 966D D5CD  .:2..(..6.......
00B020 3A6E CD7C 53D1 FDCB 348E CDA2 61C3 C052  :...S...4......R
00B030 1054 4142 2043 4F4D 4D41 4E44 204D 4F44  .TAB.COMMAND.MOD
00B040 4513 5749 4E44 4F57 2043 4F4D 4D41 4E44  E.WINDOW.COMMAND
00B050 204D 4F44 4514 454E 5445 5220 5245 5045  .MODE.ENTER.REPE
00B060 4154 2043 4F4D 4D41 4E44 1752 4550 4541  AT.COMMAND.REPEA
00B070 5420 484F 5720 4D41 0102 0070 4E59 2054  T.HOW.MA....NY.T
00B080 494D 4553 3F20 1B44 454C 4554 4520 5041  IMES?..DELETE.PA
00B090 5241 4752 4150 4820 2859 204F 5220 4E29  RAGRAPH.(Y.OR.N)
00B0A0 3F20 2044 454C 4554 4520 544F 2045 4E44  ?..DELETE.TO.END
00B0B0 204F 4620 5445 5854 202B 5920 4F52 204E  .OF.TEXT.(Y.OR.N
00B0C0 293F 2021 5274 CDC8 6BFD 3634 C0CD 8B70  )?.!R.....64....
00B0D0 CDC9 70FD CB34 B6CD E96B CD8B 70C3 A57A  .....4..........
00B0E0 1B0E FD21 2F7C ED53 437C CD7C 6F31 FA41  ...!/..SC....1.A
00B0F0 FD36 3400 218B 6F22 297C 3A39 7C32 647C  .64.!..").:9.2..
```

Fig. 1

```
00C100 CD24 6ED1 CDFA 52C3 706E 0000 0000 0000  .$....R.........
00C110 0000 3AB9 7CB7 2801 05C9 EB09 79B7 C825  ..:...(........%
00C120 C9FE 8CCA 8371 FE8D C289 71C3 8371 2812  ..............(.
00C130 FE0D 280E FE0A 280A 083E 2032 E837 08C3  ..(...(..>.2.7..
00C140 745F 32E8 37C3 745F 3B02 0200 525D 6E24  ..2.7...;...R..$
00C150 6E5F 6B50 6E18 6E64 7937 6FF3 6E26 6F18  ...P.....7...&..
00C160 5583 5878 56FB 5442 6FB5 586A 58AD 6D46  U.X.V.TB..X.X..F
00C170 6426 6F18 55FB 545D 6E24 6EB9 542D 5505  .&..U.T..$..T-U.
00C180 0618 031C 1A6E 6D1D 6D55 6D2F 6D96 6D09  .........U./....
00C190 0601 9C9B 9E9D 1B8E 8D0E 6EFE 6D56 6EC7  .............V..
00C1A0 01A7 007A 6DE5 6D11 5463 54E5 6DC7 6D03  ........T.T.....
00C1B0 111A 034D 590E 59D2 5A3C 0C48 0159 4E4E  ...MY.Y.Z<.H.YNN
00C1C0 5959 0106 3C4E 4280 8034 1278 5621 ED5C  YY..<NB..4..V!..
00C1D0 FD36 078F C9CD E96B C35D 5ACD FA52 2141  .6........Z..R!A
00C1E0 7A22 297C C931 FA41 218B 6F22 297C CDE9  .")..1.A!..")...
00C1F0 6BED 5B43 7CFD CB34 C6D5 CD24 6ECD 846E  ...C...4...$....
```

Fig. 2

```
00C100 CD24 6ED1 CDFA 52C3 706E 0000 0000 0000  .$....R.........
00C110 0000 3AB9 7CB7 2801 05C9 EB09 79B7 C825  ..:...(........%
00C120 C9FE 8CCA 8371 FE8D C289 71C3 8371 2812  ..............(.
00C130 FE0D 280E FE0A 280A 083E 2032 E837 08C3  ..(...(..>.2.7..
00C140 745F 32E8 37C3 745F 3B01 B1A5 7ADD E5FD  ..2.7...;.......
00C150 E5E5 C5D5 F53E 4332 E23F 3E44 32E5 3F3E  .....>C2.?>D2.?>
00C160 4E32 E83F CD2B 00B7 28FA FE43 280A FE44  N2.?.+..(..C(..D
00C170 280D FE4E 2813 18EC 0E0F CD8D 0518 0A0E  (..N(...........
00C180 1BCD 8D05 0E47 CD8D 05F1 D1C1 E1FD E1DD  .....G..........
00C190 E1CD C970 C364 7000 0000 0000 0000 0000  ................
00C1A0 0000 0000 0000 0000 0000 0000 0000 0000  ................
00C1B0 0000 0000 0000 0000 0000 0000 0000 0000  ................
00C1C0 0000 0000 0000 0000 0000 0000 0000 0000  ................
00C1D0 0000 0000 0000 0000 0000 0000 0000 0000  ................
00C1E0 0000 0000 0000 0000 0000 0000 0000 0000  ................
00C1F0 0000 0000 0000 0000 0000 0000 0202 0052  ...............R
```

Fig. 3

"If you ever change printers, hang on to this article . . ."

save this sector to disk. Make sure you double check everything before you save it, though.

Check the operation of Scripsit/LC with the new modification. After loading Scripsit/LC, either load text that you have already written or write something into the text buffer. Hit Break, then P for print. The message "Testing for Errors" will appear and then clear from the screen. Your prompts will now display. Select one of the letters and press the corresponding key. Your text should start printing immediately in the form you have programmed. Check each of the special commands if you have more than one. Do not forget to turn off the printer to cancel your last command before you check the next one.

Scripsit/UC modifications will be easy since the work in the lowercase version is already complete. All that is necessary is to copy the lowercase sectors to the appropriate uppercase sectors. If you are using Superzap, copy track 0BH, sector 0, to track 0CH, sector 5 and track 0CH, sector 1, to track 0DH, sector 6. If you are using RSM-2D, load sector 110 and save it on sector 125. Also load sector 121 to save on sector 136.

It would be a good idea to check track 0CH sector 5 with Fig. 1 and track 0DH sector 6 with Fig. 2 before you make any modifications to make sure they are the correct sectors.

Some Final Thoughts

Depending on how your printer handles its programming commands, it may be possible to obtain some additional commands without having to include them in the patch. For instance, if I send the Compress command and hit the Clear to escape the print routine, I can again enter the print routine and respond with the Darken Print mode. If you want to switch printing modes, you will have to turn the printer off and then back on to cancel the last command.

One final word: If you ever change printers, hang on to this article so you will be able to modify Scripsit to accommodate your new printer. ■

```
7AA5   DD E5      PUSH IX         ;
7AA7   FD E5      PUSH IY         ;
7AA9   E5         PUSH HL         ; SAVE ALL OF THE
7AAA   C5         PUSH BC         ; REGISTERS.
7AAB   D5         PUSH DE         ;
7AAC   F5         PUSH AF         ;

7AAD   3E 43      LD   A,43       ;
7AAF   32 E2 3F   LD   (3FE2),A   ; DISPLAY THE LETTERS
7AB2   3E 44      LD   A,44       ; C, D, AND N ON THE
7AB4   32 E5 3F   LD   (3FE5),A   ; VIDEO MONITOR.
7AB7   3E 4E      LD   A,4E       ;
7AB9   32 E8 3F   LD   (3FE8),A   ;

7ABC   CD 2B 00   CALL 002B       ; SCAN THE KEYBOARD
7ABF   B7         OR   A          ; FOR AN INPUT
7AC0   28 FA      JR   Z,7ABC     ;

7AC6   FE 43      CP   43         ; IS IT A "C"?
7AC8   28 0A      JR   Z,7AD0     ; IF YES, JUMP FWD 11
7ACA   FE 44      CP   44         ; IS IT A "D"?
7AC8   28 0D      JR   Z,7AD7     ; IF YES, JUMP FWD 14
7ACA   FE 4E      CP   4E         ; IS IT AN "N"?
7ACC   28 13      JR   Z,7AE1     ; IF YES, JUMP FWD 20
7ACE   18 EC      JR   7ABC       ; GO BACK TO KEYBD

7AD0   0E 0F      LD   C,0F       ; COMPRESSED PRINT
7AD2   CD 8D 05   CALL 058D       ; SEND "0F" TO PRINTER
7AD5   18 0A      JR   7AE1       ; JUMP AHEAD 11
7AD7   0E 1B      LD   C,1B       ; DARKEN PRINT, SEND
7AD9   CD 8D 05   CALL 058D       ; "1B" AND "47" TO
7ADC   0E 47      LD   C,47       ; THE PRINTER
7ADE   CD 8D 05   CALL 058D       ;

7AE1   F1         POP  AF         ;
7AE2   D1         POP  DE         ;
7AE3   C1         POP  BC         ; RESTORE ALL THE
7AE4   E1         POP  HL         ; REGISTERS
7AE5   FD E1      POP  IY         ;
7AE7   DD E1      POP  IX         ;

7AE9   CD C9 70   CALL 70C9       ; A SCRIPSIT ROUTINE
7AEC   C3 64 70   JP   7064       ; JMP BACK TO SCRIPSIT
```

Program Listing


Invasion Orion: Can You Defeat The Klaatu and Your Computer?
INVASION ORION
COMMAND A STELLAR NAVY! STOP THE ALIEN INVASION FORCES AND SAVE THE GALAXY FROM DESTRUCTION!
EPYX
INVASION ORION
EPYX
Look your computer straight in the eye, load in the new Invasion Orion and suddenly you are the Fleet Admiral, the Commander-In-Chief. And only you can stop the alien forces: the robotic Klaatu who have just invaded your Stellar Union space.
You command as many as nine starships; each ship spends energy on moving, on shielding itself, on firing its three weapon systems—destructor beams, missiles, torpedoes. There are 30 different types of starships—all armed with such a fantastic array as to intimidate the Klaatu.
But it isn't as easy as it seems. Your ships have only a limited amount of energy and you must decide how to allocate that energy to destroy the enemy.
Will your ship's armor be enough to stop the enemy's torpedo? Or should you divert energy from your beam to your shield? Move in for the kill on your weakened opponent and risk a beam attack? Or outrun the enemy? With each turn you take, your energy is replenished. Can you defeat the Klaatu? Or is your computer smarter than you are?
Invasion Orion is an EPYX game. Like all EPYX games, you will never get bored playing. Not in your or your computer's lifetime. Every game is different and fresh. Choose from three levels of skill: beginner, intermediate, expert. Ten fully tested scenarios, from one-on-one starship combat to full scale battle. Two programs: the first uses your pre-created scenarios to play the game; the second lets you create your own scenarios and design your own ships. A game that is infinitely expandable.
Yet so very easy to learn. With any of the ten scenarios, the computer takes care of all the details; no complex rules to remember. The screen shows prompts for your battle orders. Just concentrate on your strategy for victory.
Complete with superb graphics (if you have either an Apple or an Atari, you can enjoy color and sound!) and with battle manual, game program, scenario creation program, data files for your computer.
Invasion Orion. Another bug-free, easy-loading lifetime computer game from EPYX. With the unique EPYX lifetime warranty: If anything happens to your cassette or disk at any time and for any reason, send it back with just $5.00 for shipping and handling and we will send you a brand new one.
(Of course, there is also our 30-day unconditional guarantee: if your EPYX game has any defect whatsoever within 30 days of purchase, return it to us or your dealer and we will replace it free. No questions asked.)
Visit your dealer now and pick up Invasion Orion in its good-looking, protective box with the best instruction book you've ever read. Now available on disk for the Apple II (48K RAM with Applesoft) and Radio Shack's TRS-80 (32K RAM). Or on cassette for the Atari (32K), TRS-80 (Level II, 16K), Apple (16K and Applesoft) and Pet (16K). Only $24.95, disk or cassette.
If your dealer is out of stock and you can't wait, order directly from Automated Simulations. $24.95 plus $2.00 for shipping and handling (and sales tax if you are in California).
We guarantee you will be delighted with Invasion Orion. (If it is not exactly what you want, return it to us for full refund. Again, no questions asked.)
Enclose your check. Or if you order by Visa or Master-Card, use our toll-free phones: In the United States: operator 861 (800) 824-7888; In California: operator 861 (800) 852-7777; In Hawaii and Alaska: operator 861 (800) 824-7919.
Order today. You and your computer deserve the fun.
48
AUTOMATED SIMULATIONS, INC.
Dept. 101, P.O. Box 4247
1988 Leghorn Street
Mountain View, CA 94040
EPYX

See if you can follow the plot of this article.

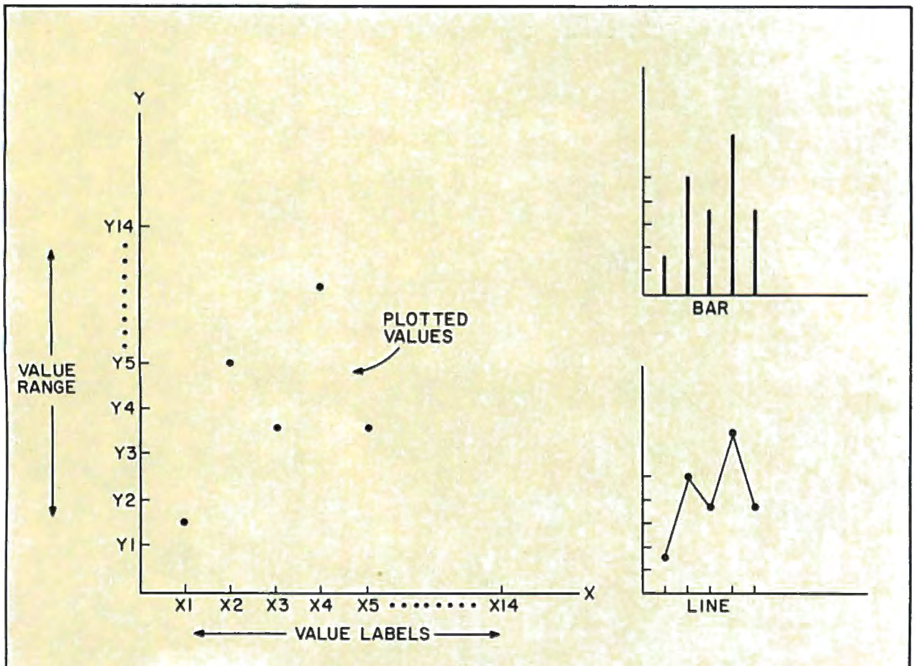
The Ups and Downs of Graphs

Richard Foley
14629 N. 61st Drive
Phoenix, AZ 85306

Line and bar graphs are, in most cases, the best method of interpreting statistical or numerical data. With just one glance at a graph you get an overall picture of the data presented and can easily spot dips, peaks and trends. In the business world graphs are used in such areas as sales, profits, earnings, production and others. On a graph you can *see* an upward trend in sales or a downward trend in earnings, instead of deducing it from a list of figures.

At home, graphs can also be used in many areas. People interested in the weather can graph high and low temperatures, rainfall and barometric pressure. Those on a diet can watch the up and down trends of their weight. Monthly utility bills, miles jogged, and a child's growth are a few of the many other applications that generate informative graphs.

However, actually sitting down with graph paper, rulers, pencils, and so forth to draw a graph is not a task many of us like to undertake. The TRS-80, with its built-in graphics and Level II Basic, is ideally suited to computerize graph building and storing. The program described in this article makes it possible to build and maintain line and bar graphs using the TRS-80.



GENERAL LAYOUT OF GRAPHS
FIGURE 1

Graph Description

The general layout of the graphs this program generates is shown in Fig. 1. It is a standard X, Y graph with the value range running up the Y axis, and the value labels along the X axis. The actual values are plotted as points in the area defined by the X and Y axis. There is room for 14 values to be plotted along the X axis and for 14 values of the value range to be displayed on the Y axis. The actual number of Y values that can be plotted is 40. The plotted points either can be connected together by a line, or connected to the X axis by bars. Photos 1 and 2 show how these graphs appear on the TRS-80's screen. The program can store and retrieve up to 50 different graphs in memory.

Program Functions

The four major functions the program performs are:

- Read Data Tape—previously saved graphs are read from tape and loaded into memory.
- Build a New Graph—the program asks for the information needed to plot a new graph, and draws it on the screen.

"With just one glance at a graph you get an overall picture of the data presented. . ."

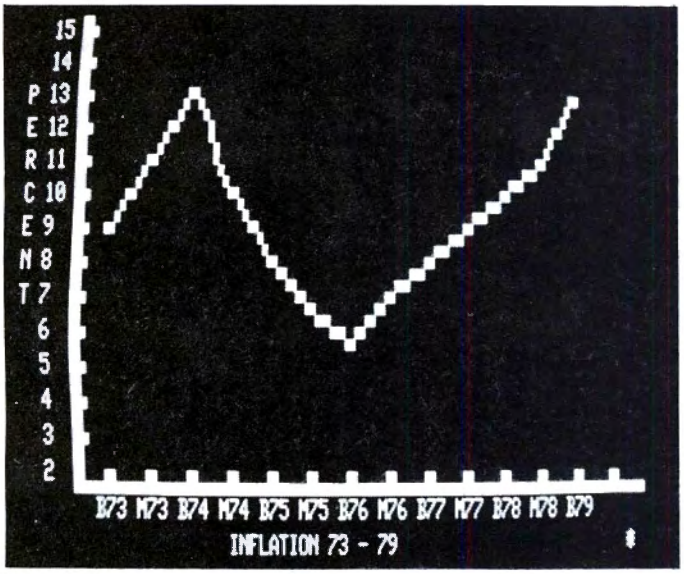


Photo 1

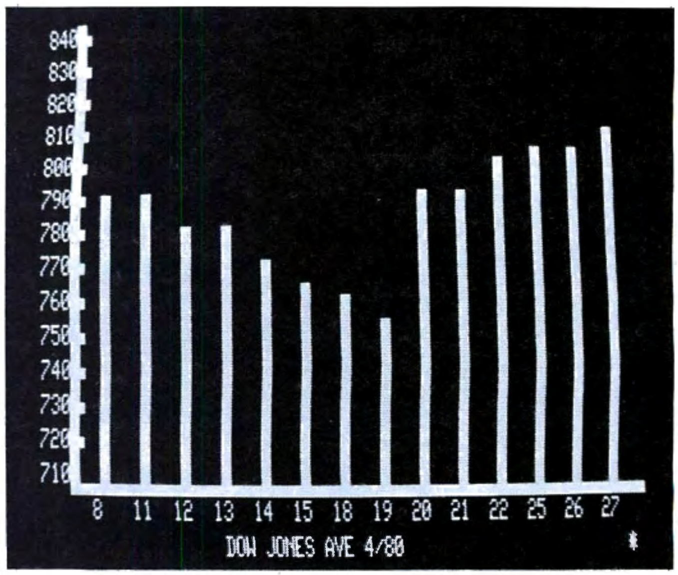


Photo 2

• Retrieve a Graph—a list of all the graphs in memory is displayed and a selection is asked for; the selected graph is retrieved and drawn on the screen.

• Write Data Tape—all graphs in memory are written to tape. The above functions are selected from the menu screen by pressing the corresponding numeric key.

Once a graph is fully displayed on the screen, either a new graph or one retrieved from memory, there are six additional functions available:

• Return to Menu—the screen is cleared and the menu screen is displayed; if the displayed graph had not been previously saved it will be lost.

• Save Graph—if the graph is new the program asks for a one-to-eight character name and inserts the graph data sequentially into memory; if the graph is old it is replaced in memory.

• Switch Type—if the graph is a line graph it will be redrawn as a bar graph and vice versa.

• Add a Value—the label and value for the next open spot on the X axis are asked for and then plotted.

• Change a Value—the program asks for the label of the value to be changed and then asks for the new value; the graph is redrawn with the new value.

• Delete—if the graph is in memory, it is deleted.

All of the above functions are selected by pressing the corresponding numeric key, when the asterisk is displayed in the bottom right corner of the graph screen.

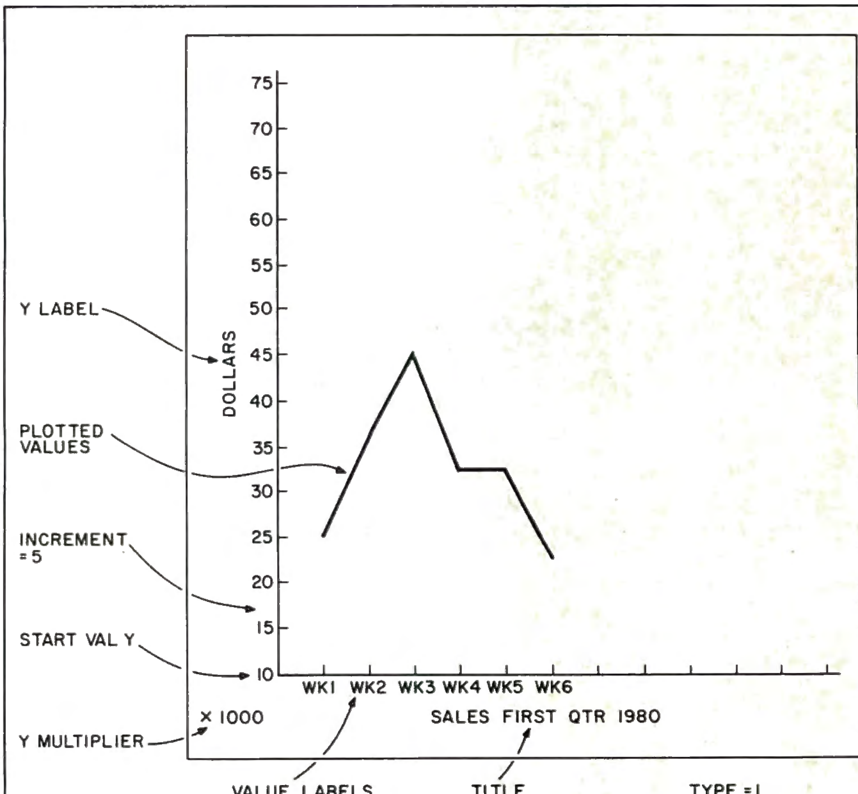
Drawing a Graph

The primary function of the program is, of course, to draw graphs. After pressing key two from the menu screen, the program asks for the following information to label and plot a graph (Fig. 2 shows how this information is used in the graph):

• Start Value Y—this is the integer value with which the Y axis starts. It can range from minus-999 to approximately plus-950.

• Increment Y (1–99)—this is the integer value by which the Y value will be incremented, going up the Y axis.

• Label Y (one-eight characters)—this is the name of what the values represent,



GRAPH INFORMATION
FIGURE 2

Program Listing

```
10 REM  ***  SET UP  ***
15 CLEAR6750:DEFINTC,V,Y:DEFSTRI,S,M,L,P,T,N,B:DEFSNGF,D:DIMM(50
),LV(14),Y(14):BL=STRING$(55," ")
50 REM  ***  TITLE PAGE  ***
55 CLS:PRINTCHR$(23):PRINT:PRINT:PRINT"         *** GRAPH-IT ***":
PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT"       BY RICHARD FOLE
Y":PRINT"          11 PERKINS AVE.":PRINT"          WILMINGTON D
E 19809":FORCT=1TO3000:NEXT
100 REM  ***  MENU  ***
150 CLS:PRINT@88,"*** GRAPH-IT ***";:PRINT@210,"PRESS NUMBER TO
SELECT -";:PRINT@345,"1 - LOAD DATA TAPE";:PRINT@409,"2 - BUILD
NEW GRAPH";:PRINT@473,"3 - RETRIEVE GRAPH";:PRINT@537,"4 - WRITE
 DATA TAPE";:PRINT@601,"5 - END";
200 I=INKEY$:IFI=""THEN200
250 IFI>"5"ORI<"1"THEN200ELSEC=VAL(I)
300 IFC=5THEN450
350 ONCGOSUB1000,1400,2500,4000
400 GOTO150
450 IFSW<>"1"THEN650
475 IFCM=0THEN650
500 PRINT@717,"GRAPHS IN MEMORY HAVE BEEN UPDATED.";:PRINT@777,"
DO YOU WANT TO SAVE MEMORY ON TAPE (Y OR N)?";
510 I=INKEY$:IFI=""THEN510
515 IFI="N"THEN650
550 IFI="Y"THENGOSUB4000:GOTO150
600 GOTO510
650 END
1000 REM  ***  LOAD DATA TAPE  ***
1050 CLS:PRINT@213,"1. MOUNT TAPE";:PRINT@341,"2. SET TO PLAY";:
PRINT@469,"3. PRESS ENTER WHEN READY";:CI=495:GOSUB14000
1100 PRINT@602,"* LOADING *";:INPUT#-1,C:CM=C:CT=0
1150 INPUT#-1,I:CT=CT+1:M(CT)=LEFT$(I,122):IFCT=CMTHEN1250
1200 CT=CT+1:M(CT)=MID$(I,123,122):IFCT<>CMTHEN1150
1250 PRINT@602,"*  DONE   *";:PRINT@725,"PRESS ENTER TO CONTINUE
...";:CI=752:GOSUB14000:RETURN
1400 REM  ***  BUILD NEW GRAPH  ***
1450 GOSUB7000
1500 PRINT@968,BL;:PRINT@968,"START VALUE Y =";:CI=983:GOSUB1400
0:VY=INT(VAL(I))
1510 IFVY<-999THENPRINT@968,BL;:PRINT@968,"Y MUST BE GREATER THA
N -999, REENTER!";:FORCT=1TO1500:NEXT:GOTO1500
1550 PRINT@968,BL;:PRINT@968,"INCREMENT (1-99) =";:CI=986:GOSUB1
4000:C=INT(VAL(I)):IFC<1ORC>99THEN1550
1600 CN=C:IF((CN*13)+VY)>999THENPRINT@968,BL;:PRINT@968,"Y OR IN
C TOO LARGE, REENTER!";:FORCT=1TO1500:NEXT:GOTO1500
1650 PRINT@968,BL;:PRINT@968,"LABEL Y (1-8 CHR) >";:CI=987:GOSUB
14000:IFLEN(I)>8THEN1650
1700 IFLEN(I)<8THENI=I+STRING$((8-LEN(I))," ")
1750 LY=I
1800 PRINT@968,BL;:PRINT@968,"Y MULTIPLIER (1-5 CHR) OR -NONE- >
";:CI=1002:GOSUB14000:IFLEN(I)>5THEN1800
1850 IFLEN(I)<5THENI=I+STRING$((5-LEN(I))," ")
1900 MY=I:GOSUB10500:PW="0":CT=1
2000 GOSUB11300:IFPW="1"THEN2100
2050 CT=CT+1:IFCT<15THEN2000
2100 PRINT@968,BL;:PRINT@968,"TITLE (1-24 CHR) >";:CI=986:GOSUB1
4000:IFLEN(I)>24THEN2100
2150 IFLEN(I)<24THENI=I+STRING$((24-LEN(I))," ")
2200 TI=I
2250 PRINT@968,BL;:PRINT@968,"TYPE (L OR B) >";:CI=983:GOSUB1400
0:IFI<>"L"ANDI<>"B"THEN2250
2300 TY=I:IFTY="L"THENGOSUB5000ELSEGOSUB11000
2350 NS="1":GOSUB7500
2400 RETURN
2500 REM  ***  RETRIEVE GRAPH  ***
2510 IFCM=0GOSUB13000:GOTO3500
2550 CLS:PRINT@17,"***";CM;" GRAPHS IN MEMORY ***";:CP=66:C2=1:C
T=0
2600 PRINT@CP,LEFT$(M(C2),8):C2=C2+1:IFC2>CMTHEN2650
2625 CP=CP+10:CT=CT+1:IFCT=6CP=CP+4:CT=0
2630 GOTO2600
2650 PRINT@982,"ENTER CHOICE >";:CI=996:GOSUB14000
2700 IFLEN(I)>8THENI=LEFT$(I,8)
2750 IFLEN(I)<8THENI=I+STRING$((8-LEN(I))," ")
2800 NA=I:D=INT(.5*CM):D1=1:D2=CM
2850 IF(D2-D1)<3THEN3100
2900 IFNA<LEFT$(M(D),8)THEN3050
3000 D1=D:D=INT(.5*(D2-D))+D:GOTO2850
3050 D2=D:D=INT(.5*(D-D1))+D1:GOTO2850
3100 CL=0:CT=D1
3110 IFNA=LEFT$(M(CT),8)CL=CT:GOTO3200
3125 CT=CT+1:IFCT>D2THEN3150ELSE3110
3150 PRINT@968,"NOT FOUND, ENTER CHOICE OR SPACE TO EXIT >";:CI=
1009:GOSUB14000:IFI=" "THEN3500ELSEPRINT@968,BL;:GOTO2700
3200 NG=MID$(M(CL),9,1):VY=VAL(MID$(M(CL),10,3)):CN=VAL(MID$(M(C
L),13,2)):LY=MID$(M(CL),15,8):MY=MID$(M(CL),23,5):CP=28:IFNG="-"
VY=VY*-1
3250 FORCT=1TO14:LV(CT)=MID$(M(CL),CP,3):Y(CT)=VAL(MID$(M(CL),CP
+3,2)):CP=CP+5:NEXT
3300 TI=MID$(M(CL),98,24):TY=RIGHT$(M(CL),1)
3350 GOSUB7000:GOSUB10500:FORCT=1TO14:GOSUB10700:NEXT
3400 IFTY="L"THENGOSUB5000ELSEGOSUB11000
3450 NS="0":GOSUB7500
3500 RETURN
4000 REM  ***  WRITE DATA TAPE  ***
4010 IFCM=0GOSUB13000:GOTO4400
4050 CLS:PRINT@213,"1. MOUNT TAPE";:PRINT@341,"2. SET TO RECORD"
;:PRINT@469,"3. PRESS ENTER WHEN READY";:CI=495:GOSUB14000
4100 PRINT@602,"* WRITING *";:PRINT#-1,CM:CT=0
4150 CT=CT+1:I=M(CT):IFCT=CMTHEN4250
4200 CT=CT+1:I=I+M(CT)
4250 PRINT#-1,I
4300 IFCT<>CMTHEN4150
4350 SW="0":PRINT@602,"*  DONE   *";:PRINT@725,"PRESS ENTER TO C
ONTINUE...";:CI=752:GOSUB14000
4400 RETURN
5000 REM  ***  DRAW LINE BETWEEN POINTS  ***
5010 YI=1:IFLV(YI+1)="   "THEN6400
5015 CX=9+(YI*8):CY=Y(YI)
5020 IFY(YI)>Y(YI+1)DI=-1
5050 IFY(YI)<Y(YI+1)DI=1
5100 IFY(YI)=Y(YI+1)DI=0:GOTO5300
5150 DF=(Y(YI)-Y(YI+1)):DF=ABS(DF)-1
```

Program continues

Program continued

```
5200 IFDF=1CY=CY+DI:DI=0:GOTO5300
5225 IFDF=0GOSUB6500:CY=CY+DI:GOSUB6500:GOTO6300
5250 IFDF>6THEN5600
5275 IFDF<6THEN5350
5300 FORCT=1TO6:CX=CX+1:CY=CY+DI:SET(CX,CY):NEXT:GOTO6300
5350 IFDF=2CY=CY+DI:GOSUB6500:CY=CY+DI:GOSUB6500:GOTO6300
5400 D2=6-DF:D3=DF-D2:D4=D3:IFD3<1THEN5500
5450 GOSUB6600:D3=D3-1:IFD3>(.5*D4)THEN5450
5500 FORCT=1TOD2:GOSUB6600:CX=CX+1:SET(CX,CY):NEXT
5550 IFD3=0THEN6300
5570 GOSUB6600:D3=D3-1:GOTO5550
5600 D2=DF/6:D2=CINT(D2):D3=6*D2:D3=DF-D3:D4=D2:D2=D2+1:D5=6-D3:
D6=D5:IFD5=0THEN5900
5700 CX=CX+1
5750 FORCT=1TOD4:CY=CY+DI:SET(CX,CY):NEXT
5800 D5=D5-1:IFD5>(.5*D6)THEN5700
5850 IFD3=0THEN6050
5900 CX=CX+1
5950 FORCT=1TOD2:CY=CY+DI:SET(CX,CY):NEXT
6000 D3=D3-1:IFD3<0THEN6050
6010 IFD3=0THEN6050ELSE5900
6050 CT=0:IFD5=0THEN6300
6100 CX=CX+1
6150 CY=CY+DI:SET(CX,CY):CT=CT+1:IFCT=D4THEN6200ELSE6150
6200 D5=D5-1:GOTO6050
6300 YI=YI+1:IFYI=14THEN6400
6350 IFLV(YI+1)="   "THEN6400ELSE5015
6400 RETURN
6500 FORCT=1TO3:CX=CX+1:SET(CX,CY):NEXT:RETURN
6600 CX=CX+1:CY=CY+DI:SET(CX,CY):RETURN
7000 REM  ***  LAYOUT X AND Y AXIS  ***
7020 CLS:FORCY=0TO41:SET(10,CY):SET(11,CY):NEXT
7050 CY=41:FORCX=12TO127:SET(CX,CY):NEXT
7100 CX=12:FORCY=1TO37STEP3:SET(CX,CY):NEXT
7150 CY=40:FORCX=16TO120STEP8:SET(CX,CY):SET(CX+1,CY):NEXT
7200 RETURN
7500 REM  ***  GRAPH FUNCTIONS  ***
7550 PRINT@968,BL;:PRINT@980,TI;:PRINT@1022,"*";
7600 I=INKEY$:IFI<"1"ORI>"6"THEN7600
7625 IFI="1"THEN7675
7650 PRINT@1022," ";:C=VAL(I):ONCGOSUB,7700,8700,9100,9500,10000

7655 GOTO7500
7675 RETURN
7700 REM  ***  SAVE  ***
7725 SW="1"
7750 IFNS="0"THEN7925
7760 IFCM=50PRINT@968,BL;:PRINT@968,"MEMORY FULL, CANNOT ADD!";:
FORCT=1TO2000:NEXT:GOTO7550
7800 PRINT@968,BL;:PRINT@968,"ENTER NAME (1-8 CHR) >";:CI=990:GO
SUB14000:IFLEN(I)>8THEN7800
7850 IFLEN(I)<8THENI=I+STRING$((8-LEN(I))," ")
7900 NA=I
7925 I=NA:I3=" ":IFVY<0THENNG="-":VY=VY*-1ELSENG=" "
7950 I=I+NG:C1=VY:C3=3:GOSUB8550:C1=CN:C3=2:GOSUB8550:I=I+LY:I=I
+MY:CT=1:C3=2
8000 IFLV(CT)="   "I=I+"   00":GOTO8100
8050 I=I+LV(CT):C1=Y(CT):GOSUB8550
8100 CT=CT+1:IFCT<15THEN8000
8150 I=I+TI:I=I+TY:IFNS="0"M(CL)=I:GOTO8400
8175 IFCM=0CL=1:M(CL)=I:CT=0:GOTO8350
8200 CT=CM
8210 IFNA=LEFT$(M(CT),8)PRINT@968,BL;:PRINT@968,"NAME ALREADY IN
 MEMORY, REENTER!";:CT=CT+2:IFCT<CMTHENFORC1=CTTOCM:M(C1)=M(C1+1
):NEXT:FORC1=1TO1500:NEXT:GOTO7800ELSEFORC1=1TO1500:NEXT:GOTO780
0
8250 IFNA>LEFT$(M(CT),8)THENM(CT+1)=I:GOTO8350
8300 M(CT+1)=M(CT):CT=CT-1:IFCT=0THENM(1)=I:GOTO8350ELSE8210
8350 CM=CM+1
8400 PRINT@968,BL;:PRINT@968,"* SAVED *";:FORC1=1TO1500:NEXT
8450 IFNS="1"NS="0":CL=CT+1
8500 RETURN
8550 REM  ***  ENCODE NUMERICS  ***
8575 I2=STR$(C1):IFLEN(I2)>C3THENI2=RIGHT$(I2,C3):GOTO8650
8600 IFLEN(I2)=C3THEN8650ELSEI2=I3+I2:GOTO8600
8650 I=I+I2:RETURN
8700 REM  ***  SWITCH  ***
8750 GOSUB8900:IFTY="B"THENTY="L"ELSETY="B"
8800 IFTY="L"THENGOSUB5000ELSEGOSUB11000
8850 RETURN
8900 REM  *** CLEAR GRAPH AREA  ***
8925 B2=STRING$(56," "):FORCT=7TO775STEP64:PRINT@CT,B2;:NEXT
8950 FORCX=16TO125:RESET(CX,39):RESET(CX,40):NEXT:CY=40:FORCX=16
TO120STEP8:SET(CX,CY):SET(CX+1,CY):NEXT:CT=1
9000 GOSUB10700:CT=CT+1:IFCT>14THEN9050
9025 IFLV(CT)<>"   "THEN9000
9050 RETURN
9100 REM  ***  ADD VALUE  ***
9150 IFLV(14)<>"   "PRINT@968,BL;:PRINT@968,"NO ROOM LEFT TO ADD
";:FORCT=1TO1500:NEXT:GOTO9400
9200 CT=0
9250 CT=CT+1:IFLV(CT)<>"   "THEN9250
9300 PW=" ":GOSUB11300:IFPW="1"THEN9400
9350 IFTY="L"THENGOSUB5000ELSEGOSUB11000
9400 RETURN
9500 REM  ***  CHANGE VALUE  ***
9550 PRINT@968,BL;:PRINT@968,"ENTER NAME OF VAL >";:CI=988:GOSUB
14000:CT=0
9560 IFLEN(I)>3THENI=LEFT$(I,3)
9565 IFLEN(I)<3THENI=I+STRING$((3-LEN(I))," ")
9600 CT=CT+1:IFCT>14THEN9900
9650 IFLV(CT)<>ITHEN9600
9700 PRINT@968,BL;:PRINT@968,"NEW VALUE =";:CI=980:GOSUB14000:F=
VAL(I)
9750 IFF<VYORF>((CN*13)+VY)THENPRINT@968,BL;:PRINT@968,"VAL OUT
OF RANGE, REENTER!";:FORC1=1TO1500:NEXT:GOTO9700
9800 GOSUB12000:GOSUB8900:IFTY="L"THENGOSUB5000ELSEGOSUB11000
9850 RETURN
9900 PRINT@968,BL;:PRINT@968,"NAME NOT FOUND, REENTER!";:FORCT=1
TO1500:NEXT:GOTO9550
10000 REM  ***  DELETE  ***
10050 IFNS="1"PRINT@968,BL;:PRINT@968,"UNABLE TO DELETE, NOT SAV
ED";:FORCT=1TO1500:NEXT:GOTO10300
10100 IFCL=CMGOTO10200
10150 M(CL)=M(CL+1):CL=CL+1:IFCL<>CMTHEN10150
```

Program continues

"After you enter each value, its point is plotted on the screen."

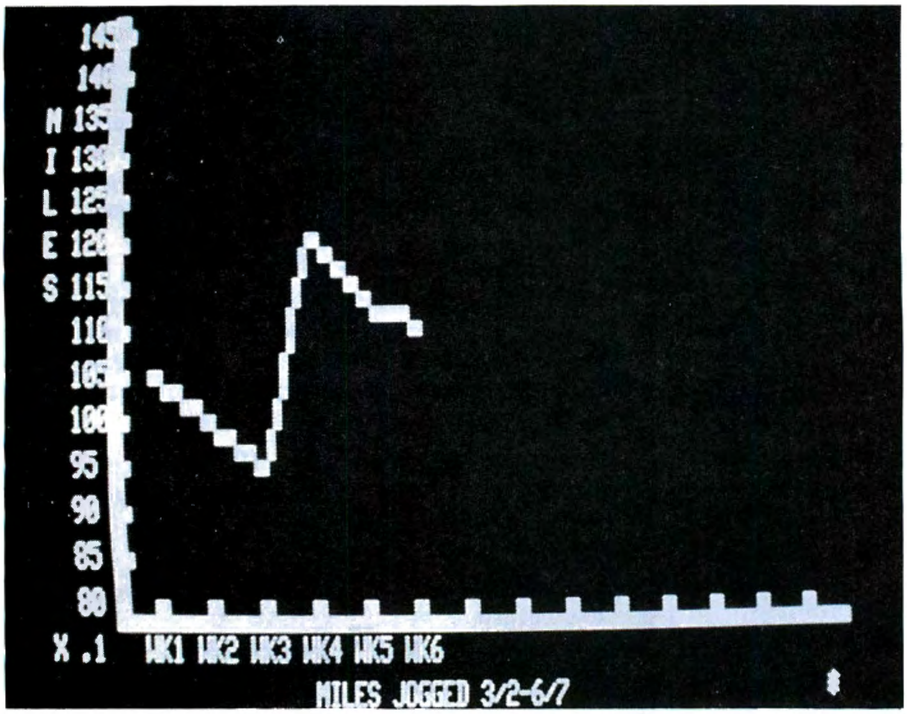


Photo 3

such as dollars, pounds, temperature, miles, and so forth.

- Y Multiplier (one-five characters)—this is what the Y values are to be multiplied by to handle decimal numbers and large and small Y values. Examples are .1, .001, 1000, 100, MIL and so on. If no multiplier is needed, NONE is entered.
- Label Value *n* (one-three characters)—*n* = 1 to 14; this is the name of the value you are going to enter, such as WK1, Sun, May, USA, Bob and so on. A space is entered to stop inputting values.
- Value *n* – *n* = 1 to 14—this is the actual value to be plotted; it has to be within the range displayed on the Y axis, but it does not have to be an integer.
- Title (one-24 characters)—this is the title of the graph.
- Type (L or B)—L to select a line graph or B to select a bar graph.

For example, if you wanted to graph how many miles you jog each week, you could proceed as follows: When the menu screen is displayed, you press two. The X and Y axes are drawn and the program asks for the start Y value. We'll assume you jog between eight and 15 miles a week. You know you can display 14 numbers on the Y axis, so this means you can display every mile and half-mile between eight and 14.5. Only integers are allowed on the Y axis so you enter 80. The program then asks for the increment and you enter five. For the Y label you enter Miles and then for the Y multiplier you enter .1. This shows that the values represent 8.0 to 14.5 miles. All the Y axis information would then be displayed on the screen and the first value label would be asked for.

You could then enter labels and values such as WK1 and 106 (for 10.6 miles). If you had been jogging for six weeks you would enter six labels and values and when asked for the seventh label, you would enter a space to stop entering values. After you enter each value, its point is plotted on the screen. You would then enter a title such as Miles Jogged 3/2-6/7.

Finally you enter L for line graph and the line is drawn on the screen. You could then save the graph and write it to tape with the rest of the graphs in memory. At a later date, you could load the tape, retrieve the graph and add values to it. Photo 3 shows the graph our running example would generate.

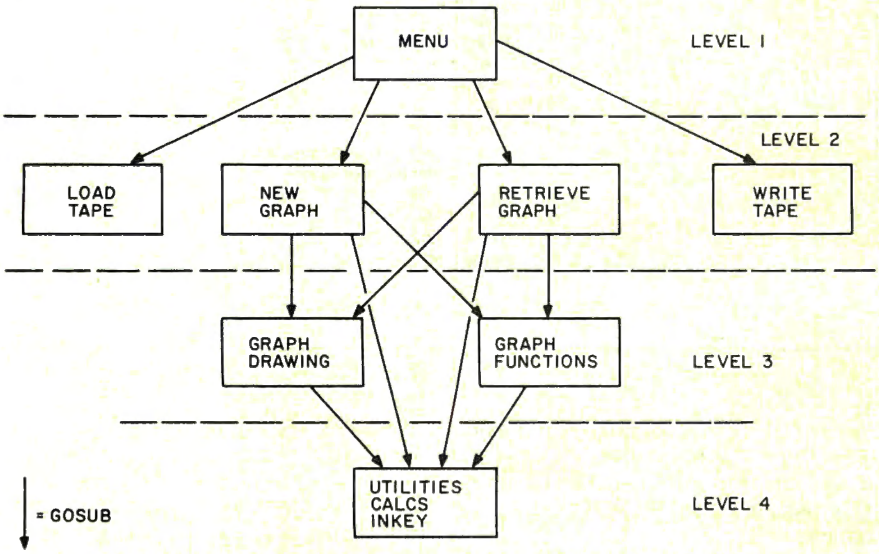
Program Description

The program is written in Level II Basic for 16K memory. It takes 9K to load and run, leaving 7K to store up to 50 graphs. It is written in a structured or modular manner, meaning that each routine or module is a

Program continued

```
10200 CM=CM-1:SW="1"
10250 PRINT@968,BL;:PRINT@968,"* DELETED *";:FORCT=1TO1500:NEXT
10300 RETURN
10500 REM  ***  Y AXIS  ***
10550 C1=VY:FORCT=833TO1STEP-64:PRINT@CT,STR$(C1);:C1=C1+CN:NEXT

10600 FORCT=1TO8:PRINT@(64+(CT*64)),MID$(LY,CT,1);:NEXT
10630 IFMY="NONE "THEN10670
10650 PRINT@896,"X";:PRINT@897,MY;
10670 RETURN
10700 REM  ***  LABEL & PLOT VALUE  ***
10725 IFLV(CT)="   "THEN10850
10750 PRINT@(899+(CT*4)),LV(CT);
10800 SET((8+(CT*8)),Y(CT)):SET((9+(CT*8)),Y(CT))
10850 RETURN
11000 REM  ***  BAR GRAPH  ***
11025 CT=1
11050 FORCY=40TOY(CT)STEP-1:SET((8+(CT*8)),CY):SET((9+(CT*8)),CY
):NEXT
11100 CT=CT+1:IFCT>14THEN11200
11150 IFLV(CT)<>"   "THEN11050
11200 RETURN
11300 REM  *** ASSIGN AND LABEL VALUE  ***
11550 PRINT@(899+(CT*4)),CT;:PRINT@968,BL;:PRINT@968,"LABEL VALU
E";CT;" (3 CHR) OR SPACE TO EXIT >";:CI=1009:GOSUB14000:IFI=" "T
HENPW="1":PRINT@(899+(CT*4)),"   ";:FORC1=CTTO14:LV(C1)="   ":NE
XT:GOTO11900
11600 IFLEN(I)>3THEN11550
11650 IFLEN(I)<3THENI=I+STRING$((3-LEN(I))," ")
11700 LV(CT)=I
11750 PRINT@968,BL;:PRINT@968,"VALUE";CT;"=";:CI=977:GOSUB14000:
F=VAL(I)
11800 IFF<VYORF>((CN*13)+VY)THENPRINT@968,BL;:PRINT@968,"VAL OUT
 OF RANGE, REENTER!";:FORC1=1TO1500:NEXT:GOTO11750
11850 GOSUB12000:GOSUB10700
11900 RETURN
12000 REM  ***  Y CALC  ***
12050 IFCN>2F=INT(F)
12100 F3=F-FIX(F):DF=INT(F-VY):F2=DF/CN:D4=INT(F2*3):IFF3=0THEN1
2200
12150 IFF3<.5THEND4=D4+1ELSED4=D4+2
12200 Y(CT)=40-D4:RETURN
13000 CLS:PRINT@78,"NO GRAPHS IN MEMORY!";:PRINT@142,"PRESS ENTE
R TO RETURN TO MENU -";:CI=175:GOSUB14000:RETURN
14000 REM  ***  INKEY ROUTINE  ***
14010 I2=""
14020 I=INKEY$:IFI=""THEN14020
14030 IFASC(I)=8THEN14110
14060 IFASC(I)=13THEN14150
14100 CI=CI+1:PRINT@CI,I;:I2=I2+I:GOTO14020
14110 IFLEN(I2)=0THEN14020ELSEPRINT@CI," ";:CI=CI-1:I2=LEFT$(I2,
(LEN(I2)-1)):GOTO14020
14150 I=I2:RETURN
```

"Each graph is stored in memory and on tape as one 122-character string."



PROGRAM STRUCTURE
FIGURE 3

GOSUB from a higher level. The top level, the menu, is the only routine that is not a GOSUB. Fig. 3 shows how the levels are arranged in the program. For example, the calculated Y coordinate routine (level 4) is a GOSUB from the assign value routine (level 3), which is a GOSUB from the new graph routine (level 2) that is a GOSUB from the menu routine (level 1).

All of the routines in the program are independent modules. A GOSUB is the only way into the module and the Return statement is the only way out. All GOTOs in a module are to statements within that module or to the Return statement. The variables used by the program are listed in Fig. 4.

Each graph is stored in memory and on tape as one 122-character string. When a graph is saved, all numeric information is converted to string values and is combined with the graph string information and the name string assigned to the graph. When retrieving a graph, the numeric string information is converted back to numeric. To save time when reading and writing to tape, each read/write is of two graphs combined into one 244-character string. The layout of the 122-character graph string is shown in Fig. 5.

The menu routine starts on line 100. This is the main driver of the program. It displays the menu and then, depending on the key pressed, issues a GOSUB to perform one of the four main functions. If the End option is selected, it checks to see if the graphs in memory have been updated, and if they have, it gives you a chance to save memory on tape before ending.

The load data tape routine starts on line 1000. The first record read is the number of graphs on the tape. The data is then read in as 244-character strings, two graphs to a string, and stored in memory.

The write data tape routine starts on line 4000. It does the opposite of the read tape routine, writing out the number of graphs and then writing out the contents of the graph table, combining every two graphs into one 244-character string.

The build new graph routine starts on line 1400. It prints the graph-building questions and accepts the information entered. Because the questions are printed on the bottom line and inputting data on the bottom line causes line feed, the input statement cannot be used. All the data is entered using the INKEY$ subroutine that starts on line 14000. It is described below. All of the graph drawing is done by GOSUBs to graph drawing routines. The inputting of values is also done by subroutine.

The retrieve graph routine starts on line 2500. It first displays the names of all the graphs in memory and then, using the INKEY$ routine, accepts the name chosen. Lines 2800–3100 are a binary search of the graph table for the chosen name. If the name is not found, an error message is displayed and another choice is asked for. When the graph data is found, it is decoded and the graph variables are set up. GOSUBs are issued to draw the graph.

At the end of both the new graph and retrieve graph routines is a GOSUB 7500. Starting in line 7500 is the graph function routine. It displays the graph title and an asterisk and then goes into an INKEY$ loop to accept commands. Depending on the key hit, it issues GOSUBs to perform the various graph functions or returns to the calling routine. When it is about to issue a GOSUB, it blanks out the asterisk.

The save routine starts on line 7700. If the graph is new, it asks for a name for the graph and uses the INKEY$ routine to accept the input. For both old and new graphs, the graph data is then encoded and stored in memory. An old graph is stored in the same location it was retrieved from. For a new graph the graph table is stepped through, each entry being shifted down one position, and the new graph is inserted right before the entry it is greater than. If the new name is already in memory, all the down-shifted table entries are shifted back and an error message is displayed.

The switch routine starts on line 8700. It first issues a GOSUB to the clear graph routine to blank out the graph area and replot the points. It then switches the graph type and issues the GOSUBs to draw the new typed graph.

The add value routine starts on line 9100. It checks the value label table and finds the first empty value. It then issues a GOSUB to the enter label and value routine to accept the value and label being added and then issues a GOSUB to draw the graph with the added value.

The change value routine starts on line 9500. It asks for the label of the value to be changed and then searches the label table for it. If the label is not found, an error message is displayed. When the label is found, it issues a GOSUB to the Enter value routine, replaces the value in the value table, and then issues GOSUBs to the clear graph routine and the graph drawing routines to draw the graph with the new value.

The delete routine starts on line 10000. It checks the new graph switch and if it is a new graph, it displays an error message. If it is an old graph, it is deleted by shifting all the table entries below it up one position.

The draw line graph routine starts on line 5000. This routine determines the length and number of small line segments needed to draw a balanced, proportional line between two points on the graph, and then draws it. The routine is divided into three separate areas. Line 5300 handles point differences of zero, one, two and six. Lines 5350–5550 handle differences of three, four and five and lines 5600–6200 handle differences over six. The six and under differ-

". . . to incorporate graphing into an existing program . . . add the modules you need . . ."

ences are drawn as unique situations. The over six differences are each calculated separately. The over six routine calculates how many middle line segments are needed and how long they will be. Any leftover amounts are divided between the top and bottom of the line. Line 5600 performs the calculations that result in the following variables:

D2 = Length of middle line segments;
D3 = Number of middle line segments;
D4 = Length of end line segments;
D5 = Number of end segments (to be decremented);
D6 = Number of end segments (not to be decremented).

POS	Length	Field
1-8	8	Graph name
9	1	Negative start value Y indicator
10-12	3	Start value Y
13-14	2	Increment
15-22	8	Label Y
23-27	5	Y multiplier
28-97	70	14-5 character label and value combinations 3 character value label + 2 character Y co-or
98-121	24	Graph title
122	1	Graph type

Fig. 5. Graph Data String Layout

The routine then uses these variables to draw the line, splitting the end segments between the top and bottom of the line. In all cases D' is the direction of the line, minus one for up or plus one for down.

The bar graph routine starts on line 11000. It is a straightforward Set loop, Setting all coordinates between the point and the X axis. The layout X and Y axis routine starts on line 7000. It draws the X and Y axis on the screen with Set loops. The Y axis routine starts on line 10500. It displays the Y range numbers, the Y label and the Y multiplier on the screen. The label and plot value routine starts on line 10700. It displays the value label and performs the Sets to plot the point on the screen.

The assign label and value routine starts on line 11300. It asks for the value label and the value. It issues a GOSUB to the Y calculate routine and then stores the label and value coordinate in their respective tables. It then issues a GOSUB to the label and plot routine.

The Y calculate routine starts on line 12000. It calculates the Y coordinate value that is closest to the Y range value that was entered.

The clear graph routine starts on line 8900. It prints long space strings to blank out the graph area, and does Sets and Resets to clean up the X axis. It then issues GOSUBs to replot the points.

The encode numerics routine starts on line 8550. It converts a numeric graph variable into a specified length string and then combines the string with the total graph data string. C3 is the length of the string, which is set up by the calling routine.

The INKEY$ routine starts on line 14000. It is an INKEY$ loop that builds a string of characters of the keys hit. It also prints each character as its key is hit. CI is the print location, which is initialized by the calling routine. If a backspace (ASC 8) is detected, the last character is deleted from the string, the last character printed on the screen is spaced out and CI is decremented by one. If an Enter (ASC 13) is detected, the subroutine stops and returns to the calling routine.

Final Comments

This program has been designed as a general-purpose graph utility program able to draw and maintain a wide variety of different graphs. However, if you want to incorporate graphing into an existing program or one you are writing, you can also use this program. Because each routine is a separate module, you can just add the modules you need to your program to generate your specific graph. If your program generates the values to be graphed, you can have it load the value table and then issue the GOSUBs to draw the graph.

The program should also be very useful in helping to teach children about graphs. For example, the change value function lets them see how different values change the appearance of the graph and the switch function shows them how a bar and line graph show the same information in different ways.

If you want to store your graphs on disk, just rewrite the load data and write data routines to read and write to disk. No other changes should be needed.

A final note: Don't use Break to end the program. Option five on the menu will end the program normally and if you have saved graphs in memory, it will remind you to save them on tape. If you use Break, you could lose the graphs you've stored in memory. ■

Variable	Type	Description
I	String	Primary string I/O
C	Integer	Primary numeric I/O
I1-In	String	Intermediate strings
C1-Cn	Integer	Intermediate integer values
CT	Integer	Counter
SW	String	Graphs updated switch
M(50)	String	Graph data table
CM	Integer	Total graphs in memory
VY	Integer	Start value Y
CN	Integer	Increment value
LY	String	Label Y
MY	String	Y multiplier
PW	String	End inputting values switch
TI	String	Graph title
TY	String	Graph type
NS	String	New graph switch
CP	Integer	Graph name print position
NA	String	Graph name
D1-Dn	Sing-pre	Calculation variables
LV(14)	String	Value label table
Y(14)	Integer	Value Y co-or table
DF	Integer	Point to point Y difference
DI	Integer	Line direction
CX	Integer	Set X co-or
CY	Integer	Set Y co-or
CL	Integer	Current graph table location
F	Sing-pre	Single precision input
YI	Integer	Subscript for draw line routine
BL	String	Blank line

Fig. 4. Program Variables

CONVERT YOUR TRS-80 MODEL-I INTO A

DEVELOPMENT SYSTEM

Now you can develop Z-80 based, *stand-alone* devices such as games, robots, instruments and peripheral controllers, by using your TRS-80 as a development system. The DEVELOPMATE plugs into the expansion connector of your TRS-80 and adds **PROM PROGRAMMING** and **IN-CIRCUIT-EMULATION** capabilities to your system (with or without expansion interface).

Complete instructions and sample schematics are included to help you design your own simple stand-alone microcomputer systems. THESE SYSTEMS CAN BE AS SIMPLE AS FOUR ICs: one TTL circuit for clock and reset, a Z-80, an EPROM, and one peripheral interface chip.

When the In-Circuit-Emulation cable is plugged into the Z-80 socket of your stand-alone system, the system becomes a part of your TRS-80: You can use the full power of your editor/assembler's debug and trace programs to check out both the *hardware and the software*. Simple test loops can be used to check out the hardware, then the system program can be run to debug the logic of your stand-alone device.

Since the program is kept in TRS-80 RAM, changes can be made quickly and easily. When your stand-alone device works as desired you use the Developmate's PROM PROGRAMMER to copy the program into a PROM. With this PROM, and a Z-80 in place of the emulation cable, your stand-alone device will work by itself.

The DEVELOPMATE is extremely compact. Both the PROM programmer and the In-Circuit-Emulator are in one small plastic box only 3.2" · 5.4". A line-plug mounted power supply is included. The PROM programmer has a "personality module" which defines the voltages and connections of the PROM so that future devices can be accommodated. However, the system comes with a "universal" personality module which handles 2758, 2508(8K), 2716, 2516(16K), 2532(32K), as well as the new electrically alterable 2816 and 48016(16K EEPROMs).

The COMPLETE DEVELOPMATE 81, with software, power supply, emulation cable, TRS-80 cable, and "universal" personality module, is ONLY $329!
The PROM PROGRAMMER is available *separately* for ONLY $239.

✓376

VISA

ORION INSTRUMENTS
172 Otis Avenue, Dept. M, Woodside, CA 94062
(415) 851-1172
Master Charge and Visa phone orders accepted
California residents please add 6% sales tax

master charge

GO PIGGYBACK!

CENTRONICS

ADD lowercase with our PLUG-IN piggyback board!

9WX7H Dot Matrix $135
5WX7H Dot Matrix $ 95

TWO complete character sets on board:
96 character ASCII **PLUS** choice of
128 character APL, TRS-80/H-19 Graphics or Scientific. (Customer defined: add $50/set).

Most printers convertible: specify logic board #

SYDNEY SEZ:
Shop Comparatively!

DSE ✓368

Digital Systems Engineering

12503 King's Lake Drive, Reston VA 22091 (703) 620-2994

ALSO AVAILABLE: New Centronics and Integral Data Systems printers at 15-20% below list; also, used printers of several makes. **Call for information!**

MasterCard, VISA, Check, MO, PO — All products waranteed 90 days

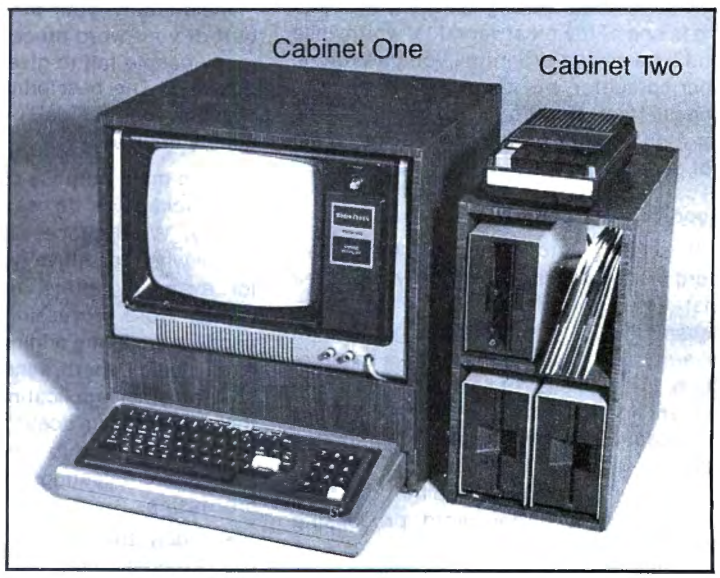
Untangle with the Organizer!

Now you can place your TRS-80 Model 1 components into one convenient desk top package. The ORGANIZER puts everything within reach, with room for Monitor, Expansion Interface, four Disk Drives, Diskettes, Cassettes and tape recorder. The ORGANIZER eliminates messy wires, reduces eye and body strain, provides room for ventilation, and works as a dust shield.

The ORGANIZER consists of two cabinets:

CABINET ONE holds Monitor, Expansion Interface, Power Supplies, Line filters, Cables, Power strip, Fan and has an easily removable front panel for re-booting.

CABINET TWO can hold up to four Disk Drives, Tape Recorder, Fan, Diskettes and cassettes. This cabinet is usable with any manufacturer's 5¼" disk drives.



BUY BOTH—$87.50 + $8.75 ship & handle

OR:
CABINET 1—$64.00 + $5.75 ship & handle
CABINET 2—$44.00 + $4.50 ship & handle

For VISA, MASTERCARD, or C.O.D.

CALL NOW TOLL FREE

Continental U.S....800-824-7888 ask for operator 617
California...800-852-7777 ask for operator 617
Alaska and Hawaii...800-824-7919 ask for operator 617

Or send check or money order to:

DESIGNATRON ✓583

2794 Hume Rd, Malibu, California 90265
For information call (213) 456-9023

✓See List of Advertisers on page 402

Corral the features you want before you put your brand on a word processing system.

A Fast Round-Up

Gordon McComb
410 Escondido Avenue
Vista, CA 92083

Word processing programs for the TRS-80 Models I and III are about a dime a dozen. Well, actually, they cost up to a hundred dollars apiece and that's where the problem begins. Word Processing is one of the most versatile, worthwhile and time saving programs you can run on your computer, but there are so many to choose from! How do you go about looking for a good word processing package and not go broke in the process? Read on!

Good Versus Good

It is difficult to define a good or bad word processing program. Many will agree that when a program is full of bugs, doesn't live up to its advertising and is an all-around letdown—yes, that's a bad program. But, like a lot of software, good is in the application of the beholder. A word processing program may be perfect for one user and lousy for another.

Here are five simple steps that can help you choose a better word processing package:

• Always select a word processing program that will fit your needs—both present and future. You may not always be able to find an off-the-shelf program that will do everything you want it to do, but you should be able to come close. The bottom line: Choose a word processing program that will get all of your jobs done.

• Your program should be flexible enough so that it can grow as your applications grow. If you think that in the future you'll need to create more than just form letters with your system, make sure your software can complement your needs.

• The word processing program you choose should fit the capabilities of your present inventory of hardware. Ideally, your program should grow as you improve your system. If the software includes fancy graphics, along with underlining, boldface and justified margins, it won't do you much good if you own a modified IBM Selectric.

Incidentally, your printer is an integral part of your word processing system, but many people fail to give it the attention it deserves. The best printers for word processors are the daisy-wheel or a high-quality dot matrix.

• The main purpose of a word processing package is to save you time and money. Word processing is particularly useful when repetitive typing is necessary; for example, letters, reports and documents. Word processing software is also good for preparing original manuscripts (no more Liquid Paper!), sending out bills and a host of other applications. It's important that your program doesn't slow you down. If it takes less time to type it out manually, then you're defeating the purpose of word processing.

• Finally, the program should be easy to use. A complicated program may be fine for someone well versed in computers, but it's out of place for the average user. Some programs require a certain knowledge of computers to operate properly. Make sure you (and your operator) can use the program to its fullest.

However, a complicated program doesn't always mean a complicated operation. Many programs jammed with goodies are far simpler to use than some that do nothing more than print out form letters.

The documentation supplied with the software is another important consideration. Some are written with technical jargon understood only by computer engineers; others are written for the rest of us. Tutorial aids, such as audio cassettes, can also be a real help when learning to use the program.

Inside a Word Processor

In order to choose the best word processing program for your particular needs, it is important that you understand the features common to most word processing software. You may find you don't need some features; others you might not be able to do without. Following are several important features that you should consider in your quest for the perfect word processor.

User-defined Video and Printing

You should be able to change the way a document will appear on the page when it is printed. In addition, altering the appearance of the text in the video monitor helps in preparing and editing non-standard documents.

Normally, the computer will set up all screen and printer formats when you start up the system. It will assume you want to use the program in the normal fashion. These are called *defaults*, and unless you tell the computer otherwise, it will perform all of the functions as originally designed.

A word processing program capable of printing documents only on standard 8½ by 11 paper will be of little use to an attorney or others who use a larger paper size.

Cursor Control

The ease with which you can position the cursor for writing and editing the docu-

MDBS...
true data base management for your TRS-80™

Your Model II is not a toy... and neither is MDBS. MDBS is the serious tool for applications development.

Harvey M. Weiss, an independent software reviewer:

"MDBS is truly a data base management system..., not a file management system."

"MDBS... offers the data manipulation, query, and report-writer capabilities of a DBMS found on much larger and more complex computer systems."

"MDBS has come to the rescue with...products whose ability to reduce development time and implement processes not available elsewhere make them invaluable to integrators and developers of μc-based systems. [With MDBS,]...program development and data base description is simple."

For the complete Weiss review, see "Downscaling DBMS to the Microworld," *Mini-Micro Systems*, April, 1981, pp. 187-195.

ONLY MDBS has all these features!

Flexible, powerful data structuring capabilities: network and hierarchical data structures, including direct many-to-many relationships (not available elsewhere).

Genuine data base management capabilities: far surpasses file management in data independence, automatic maintenance of relationships, and control over data redundancy.

English-like, nonprocedural query language interface: ad hoc queries result in the automatic generation of desired reports, tables, files, and relations.

Host language interfaces, including various BASICs, COBOLs, PASCALs, FORTRAN, PL/I, C, and several assemblers.

Extensive built-in data access security (256 levels).

Automatic transaction logging and data base recovery.

Dynamic data base restructuring...and many other advanced data base management features.

Is MDBS right for your system?

MDBS is written in Z-80 machine language and runs under TRSDOS or NEWDOS with TRS DISK BASIC; under CP/M® (and similar derivatives) with CBASIC2, PL/1, PASCAL Z, PASCAL M, PASCAL MT+, BDS C, CIS COBOL, and Microsoft COBOL, FORTRAN, or BASIC.

MDBS code size is only 18K to 20K bytes for Models I and II.

MDBS is independent of the types and sizes of disk drives. Supports data base spread over as many as eight disk drives.

To learn more

Free pamphlet

- For free literature about serious data base management on Models I and II, call, write, or TWX us. We'll include a pamphlet on features to consider when choosing a software tool for application system development.
- For a practical introduction to true data base management for small computers, please send \$20 (Indiana residents, add 80¢ tax) for PRIMER/GUIDE booklets.
- For a comprehensive DOCUMENTATION PACKAGE consisting of THE PRIMER plus user manuals covering the MDBS, QRS, RTL, and DRS, please forward \$60 (+ \$2.40 Ind. tax).

Setting standards of excellence for data base software...worldwide

Micro Data Base Systems, inc.

Box 248
Lafayette, Indiana 47902
317-448-1616/TWX 810-342-1881

Dealer/distributor/OEM inquiries invited.

☐ Please send me free literature on the MDBS.

☐ \$20 (Ind. tax 80¢) enclosed for MDBS PRIMER and GUIDE (total 97 pages).

☐ \$60 (Ind. tax \$2.40) enclosed for Complete MDBS DOCUMENTATION PACKAGE.

Name ______________ Title ______
(Please print)

Company ______________

Address ______________

City ______________ (State) (Zip)

Phone ______________

MAIL TO: Micro Data Base Systems, Inc. ✓315
Box 248-M
Lafayette, IN 47902

"Global editing is a boon for indexing, form letter preparation, specification typing and many other documents."

ment is a feature often overlooked. The longer the document, the more tedious it will be to position the cursor. Some programs move the cursor only one character and one line at a time. Others flip through the document from beginning to end, go back and forth from the start to the end of a line, and so forth. If you need to work with lengthy documents, it might be a wise idea to stay away from a program that has limited cursor control.

Wrap-around

You may notice that on most word processing programs you don't need to push Enter when you finish a line. If a word exceeds the maximum screen width, it will wrap around onto the next line.

Local Edit

Local editing refers to one-at-a-time changes. The simplest form of editing is the strikeover. If you make a typo, you merely position the cursor over the wrong letter and retype the word correctly.

Other local editing features include single word delete or insert and line delete and insert. If you need to take out a word, position the cursor on the first character and push the Delete key. The word is erased and all of the text closes up to fill the gap. The words re-wrap as needed to make it tidy and neat.

Global Editing

Let's say you must change a person's name throughout a lengthy document. Instead of tediously looking for each occurrence and making the changes manually one at a time, you can instruct the computer to perform a global replace function. You tell the computer what name to find and what to exchange it with.

Other global functions are search (or find) and delete. Global editing is a boon for indexing, form letter preparation, specification typing and many other documents.

	Letter Writer Astro Star Enterprises	Write-Righter Tritter Software	Word Processing In Basic Jerry Lee Software	Word Processor Program Blanton Software Service	CPT-80 Bits & Bytes Software	The Word Worker Design Enterprises	Scripsit Radio Shack	The Electric Pencil Michael Shrayer Software
Line Oriented		●	●	●	●	●	●	●
Page Oriented	●				●			
Expanded Cursor Con.	●	●		●	●		●	●
User Defined Video	●		●	●			●	●
User Defined Print	●	●	●	●			●	
Underlining					●	●		●
Boldface		●				●		
Character I/D	●	●	●	●	●	●	●	●
Word I/D	●	●		●	●	●		
Line I/D	●	●	●	●	●	●	●	●
Global Replace					●	●	●	●
Global Search					●	●	●	●
Global Delete					●	●	●	
Headers & Footers	●					●	●	●
Auto Pagination				●		●	●	●
Hyphenation						●	●	
Selectable Tabs		●	●	●	●	●	●	
Block Manipulation	●	●	●	●	●	●	●	●
Flush Right Margin		●		●		●	●	
Justification		●	●	●	●	●	●	●
Centering	●	●	●	●		●	●	
Vertical Centering			●				●	
Merging	●	●	●		●			
Menu Driven	●	●	●	●		●		
Tutorial Aides	●						●	
Free Updates		●	●	●	●	●	●	●
Modification Req'd.								●

Table 1

Header and Footer

There may be times when you need to include a title or your name on each page. A header will place this information on the top of the page; a footer on the bottom. You can also select this function to occur only on odd or even-numbered pages (when a document is printed on both sides of the paper, for example).

Auto Pagination

Since it is often difficult to determine when one page will stop and another will begin, a program that automatically numbers the pages can be very handy. Many programs enable you to begin numbering at any point in the document and place it anywhere you wish in the top or bottom margin.

Margins

In addition to being able to select the amount of margin space on the top, bottom, left and right sides of the paper, your word processing program might also be capable of:

- Flush right margins—where the standard even margin is on the right side, not the left.
- Justified—both side margins are even. Space is added between words to spread out the lines when necessary.
- Centered—each line is centered on the page.
- Vertical centered—where the entire document is placed neatly in the middle of the page. No more top-heavy letters!

Page Scroll and Line Scroll

Depending on how your program "sees" the text, you'll be able to go through the

"Merging, in this case, has nothing to do with Wall Street or big business."

document either one line at a time, or entire pages at once (screen pages; usually only one third or one fourth the amount of copy of a printed page).

Tabs and Indenting

Many programs enable you to select a tab stop at every character position. Others have pre-selected tabs, usually at five or 10 spaces apart. Many programs will indent at the beginning of each new paragraph. Few programs will allow you to have each line indented, as when preparing outlines and specs.

Block Manipulation and Merging

A block is any large amount of text. Whenever you need to alter entire segments, block manipulation will come in handy because you won't have to retype the text. If you need to change the position of a particular paragraph from the beginning to the end of a document, you begin by placing block markers around the paragraph, then tell the computer where you want the text to be moved.

Merging, in this case, has nothing to do with Wall Street or big business. It is a highly useful feature that enables the operator to send out items such as personally addressed form letters in a minimum amount of time.

First, you create a primary document and insert variable commands wherever information is to be changed. On the secondary document, you include names, addresses, account numbers and balances (or whatever). When you merge the two, the information from the second file will be inserted into the proper slots in the text of the primary document.

Merging is a feature many programs don't have, but it is something you can buy as an add-on to your present word processing program.

Modification Requirements

A few of the programs recommend certain hardware modifications (in addition to the standard upper/lowercase modification for the Model I). You'll need to consider this very carefully, especially if you have a new machine which is still under warranty. Opening up the keyboard and adding hardware will definitely void the warranty.

Priorities

How do you go about deciding what you'll need in a word processing program? First, examine the type of documents you'll be preparing. Cross reference these with features that you must have to accomplish the work. If you want, put a point value on the importance of each feature.

Then compile a similar list, this time jotting down features you'd like to have, but which are not absolutely necessary to carry out what you need. Assign a point value to these also.

Compare your lists with the programs available on the market. Those most needed features will be on the top of the list, and software that fails to pass should be dropped from the race.

After this process of elimination, you should have one or two programs that will suit your needs. Contact the nearest supplier of the program and see if you can arrange a demonstration. Since software is something you usually can't return, it's a good idea to be absolutely sure you know what you're getting.

To make your search a little easier, we've compiled a list of several word processing programs available for the Models I and III. We sent out a detailed questionnaire to about two dozen software houses and those that responded have been included here. Undoubtedly, there are many other programs available, but this cross-reference chart should steer you in the right direction in determining which software will be best for your particular application.

To some, word processing seems like a luxurious fad. People have been banging out letters on typewriters for almost 100 years. Why not 100 more? But we guarantee, if you try out a good word processing program, you'll wonder how you ever got along without it! ■

COMPUTEX

321 El Dorado • Webster Texas 77598 • (713) 488-8022 ✓392

...for the most satisfying gifts all year

FROM COMPUTEX

MODEL III SYSTEMS

COMPLETE BUSINESS SYSTEMS

• Astec switching power supply for cooler, more efficient operation • goldplated edge card connections for improved reliability • floppy disk storage to 2 Mbytes • disk controller handles 40/80-track single and double-headed drives • fully compatible with TRSDOS, LDOS, DOS+ and NEWDOS 80 • fully tested and burned in • 90-day warranty on parts and labor.

Model 321 - 350 Kbyte **$1,895.00**
• 48K Model III with two Tandon 40-track single-headed double density disk drives • system includes TRSDOS, dust cover and one box of Verbatim diskettes

Model 322/3 - 700 Kbyte **$2,295.00**
• 48K Model III with your choice of either two Tandon double-headed 40-track disk drives, or two single-headed 80-track drives • system includes LDOS, dust cover, and one box Verbatim diskettes

Model 324 - over 1.4 Mbyte **$2,595.00**
• our largest all-in-one Model III provides over 1.4 Mbytes of floppy disk storage with two Tandon double-headed 80-track drives • 48K Model III system includes LDOS, dust cover, box Verbatim diskettes

We now carry complete business software from SBSG, Galactic, PROSOFT, Radio Shack, many others

TANDON 8" THINLINE DISK DRIVES

- Exactly half the height of any other model
- Proprietary, high-resolution, read-write heads
- DC operation - no AC required
- Industry standard interface
- Same as Shugart SA901, pinout
- 3-ms track-to-track access time
- Single or double headed
- Single or double density

TM848-1 **$495.00**
Single-sided 77-track - Storage: 600K single density, 1.2 Mbyte double density

TM848-2 **$595.00**
Double-sided 77-track - Storage: 1.2 Mbyte single density, 2.4 Mbyte double density

Complete Model II expansion bay system will be announced soon.

TANDON TM600 SERIES 5¼" MINI-WINCHESTER DISK DRIVES

- Highest storage capacity of any 5¼" high-speed random access drive
- Up to 46 Mbytes online storage
- Up to four TM600's can be daisy-chained on single bus
- S-version compatible - Shugurt SA1000 and Seagate ST506

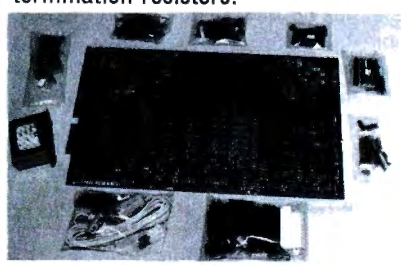
TM602S **$1,195.00**
3-ms access time - 612 tracks - 5 Mbytes/s data transfer rate - 6.38 Mbyte RPM capacity (unformatted)

TM603S **$1,295.00**
3-ms access time - 918 tracks - Mbytes/s data transfer rate - 9.57 Mbyte RPM capacity (unformatted)

Complete Model I and III systems will be announced soon.

TRS 80 MDX-2 SYSTEM EXPANSION

The new MDX-2 from Computex provides more capabilities than any other system expansion currently available. • build only what you need • silk-screened and solder-masked PCB • dip switch instead of hardware jumpers • no messy vertically-mounted termination resistors.



complete kit
$269.00

• MDX02 printed circuit board	$74.95
• Onboard direct connect modem kit	39.95
• 2K/4K Eprom kit	17.95
• RS232 mA serial interface	17.95
• Real-time clock kit	17.95
• Floppy disk controller kit	31.95
• 32K RAM expansion kit (less RAM)	11.95
• Cassette port	3.95
• Analog power supply	29.95
• Hardware and socket kit	19.95
• Centronic line printer port	8.95
• 16K Ram memory	14.95

COMPUTEX

Comprehensive Test Program $21.95

As a leading supplier of disk drives and TRS80 systems, we had to have a system test program - and one that we could trust. Since none existed, we wrote our own. Now we make it available to you. A must for disk drive users - no library should be without one.

• disk drive speed test • disk drive alignment program (requires alignment disk - Dysan Alignment Disk $39.95) **• expansion interface RAM test • keyboard RAM TEST • video RAM test • keyboard bounce test • ROM checksum program • full screen RAM display program**

Check the menu for the business program you want.

Investment du Jour

C. Brian Honess
22 Shaftesbury Lane
Columbia, SC 29209

Groups of programs are often combined into one larger program using the menu technique. The individual programs can be of fairly diverse applications, but usually have a central theme. For example, in a statistics package you might find programs ranging from simple descriptive statistics to the most complex multi-variate analysis programs.

Menu

Let's assume a brief package of three financial routines. First, clear the screen. Then a series of statements print the menu (See Table 1). Put the name and number of more routines starting in line 19 of Program Listing 1.

Number the three routines at lines 100, 200, and 300 of Program Listing 2. Use On. . . Goto for routine selection.

Compound Interest

When only the principal earns interest for the entire life of a deposit or loan, the interest due at the end of the time is simple interest. Invest $100.00 at five percent per year simple interest. At the end of the year you have $105.00. At the end of the second year you have $110.00; the $5.00 interest in the first year is not added to the principal, and draws no interest. Minor problems are caused by different numbers of days in the months and years, but the principal earning interest remains unchanged throughout the life of the loan or deposit.

Compounding (interest due is added to the principal) results in different interest payments. Different principal amounts are used to calculate the interest.

The formula for finding the value of a deposit of P dollars at an interest rate of i percent (expressed as a decimal) for n periods is fairly simple:

$$V = P(1 + i)^n$$

With compound interest, the value of the $100.00 invested for one period at six percent is the same as when using simple interest—$106.00. During the next period the entire $106.00 earns interest at six percent, making the amount at the end of two periods $112.36 instead of the $112.00 with simple interest.

Our simple formula does not apply to quarterly or daily compounding. The formula for this type of problem is:

$$V = P(1 + \frac{i}{c})^{nc}$$

```
10 PRINT
11 PRINT "* * * HONESS FINANCIAL PACKAGE * * *"
12 PRINT
13 PRINT "  PROGRAM                        ENTER  "
14 PRINT "  -------                        -----  "
15 PRINT
16 PRINT "COMPOUND INTEREST                  1    "
17 PRINT "DISCOUNTED PRESENT VALUE           2    "
18 PRINT "DEPRECIATION                       3    "
30 PRINT
31 INPUT "WHICH DO YOU WANT"; N
```

Program Listing 1

```
        * * * HONESS FINANCIAL PACKAGE * * *
PROGRAM                         ENTER
-------                         -----
COMPOUND INTEREST                 1
DISCOUNTED PRESENT VALUE          2
DEPRECIATION                      3
WHICH DO YOU WANT?
```

Table 1

"When only the principal earns interest for the entire life of a deposit or loan, the interest due. . . is simple interest."

where: c is the number of times compounding is to take place in a period
n is the number of periods
i is the interest rate per period n
P is the Principal, and
V is the value at the end of the n periods

Return to the $100.00 at six percent for 12 years problem and assume quarterly compounding:

$$V = 100.00(1.0 = \frac{0.06}{4})^{(4 * 12)}$$

Quarterly compounding yields $204.33 compared with $201.22 under yearly compounding.

Use an Input statement with a literal prompt for each of the values needed:

```
10 INPUT "WHAT IS THE PRINCIPAL";P
```

or combine them all like Program Listing 3.

Run the program for the same principal at the same interest rate, with differing rates of compoundings.

The value of principal subjected to continuous compounding is given by:

$$V = Pe^{in}$$

where: V is the value
P is the principal
e is the base of the common log system and is equal to 2.71828
i is the interest rate
n is the number of periods

The value of $100.00 compounded continuously at six percent for 12 years is $205.443, compared to $204.33 for quarterly compounding and $201.22 for yearly.

Use the EXP function supplied with most Basic versions to code a program:

```
10 INPUT P,I,N
20 V = P * EXP (I * N)
30 PRINT V
```

AGE	DEPR. CHG.	ACCUMULATION DEPRECIATION	BOOK VALUE AT END OF YEAR
0	0.00	0.00	3600.00
1	525.00	525.00	3075.00
2	525.00	1050.00	2550.00
3	525.00	1575.00	2025.00
4	525.00	2100.00	1500.00
5	525.00	2625.00	975.00
6	525.00	3150.00	450.00

Table 2

The effective annual rate compares the interest you get at different rates, compounded a different number of times per period. For example, suppose you invest $100.00 at six percent compounded quarterly for 1 year:

$$V = 100\ (1 + 0.06/4)\uparrow(1 * 4) = \$106.136$$

and also suppose you invest $100 at 6.136 percent compounded annually:

$$V = 100(1 + 0.06136) = 106.136$$

Six percent is the annual rate and 6.136 is the effective annual rate.

This type of calculation is easily coded:

```
10 INPUT "WHAT IS THE NOMINAL INTEREST RATE"; R
20 INPUT "HOW MANY TIMES IS IT COMPOUNDED/YR"; N
30 I = (1 + R/N)↑N - 1
40 PRINT I
```

Present Value

Present value is the value of a debt on a

```
10
 .
 .      headings, menu, etc.
 .
30
31 INPUT "WHICH DO YOU WANT"; N
40 ON N GOTO 100, 200, 300
50 GOTO 31
100
 .
 .      compound interest routine
 .
199 GOTO 900
200
 .
 .      discounted present value routine
 .
299 GOTO 900
300
 .
 .      depreciation routine
 .
399 GOTO 900
 .
 .
 .      any other routines you add go here. (add to line 40 also)
 .
 .
 .
900 INPUT "WANT TO RUN AGAIN—(1 = YES, 0 = NO)"; X
910 IF X=1 GOTO  10
920 IF X=0 GOTO 999
930 GOTO 900
999 END
```

Program Listing 2

```
10 PRINT  "      ENTER PRINCIPAL, INTEREST (DECIMAL),
11 PRINT  "      NO. OF COMPOUNDINGS PER PERIOD, AND       "
12 PRINT  "      NO. OF PERIODS"
13 PRINT
14 PRINT  "      EXAMPLE:"
15PRINT   "       $100.00 INVESTED FOR 12 YEARS AT         "
16 PRINT  "       6% PER YEAR, COMPOUNDED QUARTERLY,       "
17 PRINT  "       WOULD BE ENTERED AS:"
18 PRINT
19 PRINT  "                 100.00,0.06,4,12               "
20 INPUT  P,I,C,N
```

Program Listing 3

✓See List of Advertisers on page 402

"The amount of depreciation each year reduces the taxable income, as you recover your cost. . ."

date earlier than the due date.

Suppose I agree to pay you $1000.00 a year from now. Money is now earning six percent. What is the present value of the debt? You should settle for less than the $1000.00 you will get a year from now. You can invest the amount today at six percent and have your $1000.00 at the end of the year. This is almost the same problem as compound interest; now we know V and solve for P.

The formula:

$$P = V(1+I)^{-n}$$

```
10 INPUT "WHAT IS THE COST OF THE ASSET  "; C
20 INPUT "WHAT IS THE SALVAGE VALUE      "; SV
30 INPUT "HOW MANY YEARS DEPRECIATION"; N
40 PRINT
50 PRINT "   print your headings here   "
60 PRINT
70 A=0
80 DC=0
90 DF=0
100 BV=C
110 PRINT A, DC, DF, BV
120 DC=(C-SV)/N
130 FOR A =1 TO N
140 DF=DF+DC
150 BV=BV-DC
160 PRINT A, DC, DF, BV
170 NEXT A
```

Program Listing 4

is only slightly different than the compound interest formula. Recall that:

x^{-3} is the same as $\frac{1}{X^3}$

When you compound at more than one time per period, the formula reflects this:

$$P = V(1 + \frac{I}{c})^{-nc}$$

where c is the number of times compounding takes place per period.

What amount must be invested now to yield $1000.00 at six percent compounded quarterly, 12 years from now?

$$P = 1000.00\,(1 + \frac{0.06}{4})^{-(4*12)}$$

$$= 1000.00\,\left(\frac{1}{(1+\frac{0.06}{4})^{(4*12)}}\right)$$

$$= 1000.00\,\left(\frac{1}{1.015^{48}}\right)$$

$$= 1000.00 * (1/2.04348)$$

$$= \$489.36$$

Check this result with the compound interest program and formula. The value 489.36 invested for 12 years at six percent compounded quarterly yields $1000.00. The present value program follows:

```
20 INPUT V,I,C,N
30 P=V*(1/((1+I/C)↑(C*N))
40 PRINT V
```

Depreciation

As an asset is used it loses value. The asset has to be replaced at the end of its useful life. Consider a company setting aside a portion of its earnings in a depreciation fund. Actually, the amount of depreciation each year reduces the taxable income, as you recover your cost during the asset's useful life. The straight-line method for depreciating an asset is simplest. Suppose a machine costs $3600.00 and has a useful life of six years. At the end of the six

```
10 INPUT "SCRAP VALUE OF ASSET          ="; SV
20 INPUT "COST OF ASSET                 ="; C
30 INPUT "NO. OF YEARS TO DEPRECIATE    ="; N
40 A=SV/C
50 D=0.001
60 T=1 - D
70 B=T↑N
80 IF A> =B THEN 110
90 D=D+0.001
100 GOTO 60
110 PRINT "RATE OF DEPRECIATION ="; D
```

Program Listing 5

✓See List of Advertisers on page 402

"Seldom is an asset depreciated the same amount in its first year as in one of the later years of its life."

years, the machine has a scrap value of $450.00.

$$\text{Average yearly depreciation} = \frac{3600 - 450}{6} = \$525.00$$

Table 2 shows the yearly depreciation charge, the amount accumulated in the depreciation fund, and the book value of the machine at the end of each year:

The book value drops to the salvage value at the end of six years. The accumulated depreciation plus the book value at the end of the year always equals the cost.

Program Listing 4 prints this table. Code the headings as you like; commas between the variables space the printing in the first four fields.

There are some serious problems with the straight-line depreciation method. Seldom is an asset depreciated the same amount in its first year as in one of the later years of its life. Consider your automobile, for example. You would like to depreciate it more the first year than the second, more the second than the third, and so on. (Tie this to the concept of Present Value. The more you depreciate the car early in the useful life, the more tax savings you get and also the more money you can earn on that depreciation fund money at compound interest.) You want to keep the book value in line with the real value of the car at the end of each year. Overcome this problem by calculating the depreciation charge as a fixed percentage of the book value each year, known as the constant-percentage (or declining-balance) depreciation method.

In the constant-percentage depreciation method, assume an original cost C, and a fixed depreciation percentage d, so that the depreciation charge is Cd at the end of the first year, and the book value is: C – Cd, which equals C (1 – d). At the end of the second year, the depreciation charge is C(1 – d)d and the book value is:

$$C(1-d)^2$$

In other words, the book values for the first n years are:

$$C(1-d),\ C(1-d)^2,\ C(1-d)^3,\ \ldots,\ C(1-d)^n$$

Since the book value at the end of n years is the scrap value, we can say:

$$SV = C\,(1-d)^n$$

Consider the same problem: a $3600.00 machine with a useful life of six years and a scrap value of $450.00. If we substitute what we know into the equation, we have:

$$450 = 3600\,(1-d)^6$$

or:

$$(1-d)^6 = 450/3600 = 0.125$$

solve for d, use logs:

$$6 \log(1-d) = \log 0.125$$

We run into trouble in the Basic program on several different fronts. If your computer has only base e logs, convert from base 10 with:

$$\log_b(X) = \log_e(X)/\log_e(b)$$

To find the base 10 log of 0.125 with a base e log function, you could say:

```
LG = LOG(0.125)/LOG(10)
```

When we start working with:

$$6 \log(1-d) = \log 0.125$$

we eventually must find an anti-log. I use the iteration technique.

Suppose we just set d equal to 0.01 and try it in our formula:

$$(1 - 0.01)^6 = 0.94$$

Because 0.94 is not the 0.125 we would like, 0.01 must be wrong. Let's try 0.02:

$$(1 - 0.02)^6 = 0.88$$

A little closer, but not 0.125. As the value of d goes up, the answer goes down. In the iteration technique, start at some very small value (0.001) and compare the answer to what it should be (in this case, 0.125). If our guess is too low, the answer is more than 0.125. Add a small number to d and try again until the answer equals 0.125. At that time we have a good approximation of d (See Program Listing 5).

Lines 10 through 30 allow us to input the values for the calculations. Line 40 calculates the number we want to approximate in the iteration technique (0.125 with the data from the example problem). In line 50 we start with a d value of 0.1 percent. You can make this and the incremental value in line 90 smaller if you like. These values allow reasonable running times, and produce rates of depreciation accurate to the third decimal place. Line 60 calculates T; (which I named "T" to stand for "Term") line 70 raises this term to the nth power and calls it B. Line 80 compares B to A; if A is greater than or equal to B,D is printed. If A is less than B the program adds 0.001 to D and tries again.

The book value is 41¢ off, as is the amount in the accumulated depreciation at the end of the six years (See Table 3). The value of D is slightly inaccurate; the iteration technique only found a good approximation. If you want greater accuracy (and are willing to wait for it while your program finds d) increment d by 0.0001 or 0.00001 in line 90.

Program Listing 6 makes the table. Insert your own headings.

Here we calculated the book value first and subtracted the new book value from the

AGE	BOOK VALUE AT END OF YEAR	DEPRECIATION CHARGE	ACCUMULATED DEPRECIATION
0	3600.00	0.00	0.00
1	2545.20	1054.80	1054.80
2	1799.46	745.74	1800.54
3	1272.22	527.24	2327.78
4	899.46	372.76	2700.54
5	635.92	263.54	1964.08
6	449.59	186.33	3150.41

Table 3

```
120 AG=0
130 DC=0
140 DF=0
150 BV=C
160 PRINT AG, BV, DC, DF
170 FOR AG=1 TO N
180 X=C * (1-D) ↑ AG
190 DC=BV-X
200 DF=DF+DC
210 BV=X
220 PRINT AG, BV, DC, DF
230 NEXT AG
```

Program Listing 6

AGE	DEPRECIATION CHARGE	ACCUMULATED DEPRECIATION	BOOK VALUE AT END OF YEAR
0	0.00	0.00	3600.00
1	900.00	900.00	2700.00
2	750.00	1650.00	1950.00
3	600.00	2250.00	1350.00
4	450.00	2700.00	900.00
5	300.00	3000.00	600.00
6	150.00	3150.00	450.00

Table 4

"This formula never allows you to reduce the asset balance to zero."

old book value to get the depreciation charge. This is added to the amount already in the accumulated depreciation.

The Sum-of-years-digits method increases the present value of the depreciation tax shield. Assume the same problem: a $3600.00 machine with a $450.00 scrap value and a useful life of six years. Depreciate it 6/21 the first year, 5/21 the second year, 4/21 the third year, 3/21 the fourth year, 2/21 the fifth year, and 1/21 the last year. Add all the numerators:

$$\frac{1+2+3+4+5+6}{21} = \frac{21}{21} = 100\%$$

The sum of the "n" integers is given by:

$$\frac{n(n+1)}{2}$$

In our example (where n = 6):

$$\frac{6(6+1)}{2} = \frac{6*7}{2} = \frac{42}{2} = 21$$

Table 4 is produced.

The scrap value is the book value at the end of six years. The depreciation charge in the first year was found by multiplying 6/21 by the amount to be depreciated (cost-scrap value) = 6/21*(3600-450) = 900.00. All values in this equation stay the same each year except the numerator over the 21, which decreases by 1.

Program Listing 7 produces this table. Add headings to suit your self.

The program follows the methods used before, except for line 110 in the loop. The age (AG) must go from 1 to n but the numerator of the multiplication factor must go from n down to 1. The program calculates a value of UD for each value of AG.

The Double-Declining Balance method of depreciation applies a uniform rate to the remaining book value. Assume that for tax purposes the rate cannot exceed twice the straight-line rate (hence, the "double" in the name). Consider the $3600.00 machine with the $450.00 salvage value and the six year life (See Table 5).

The straight-line rate would be:

$$\frac{100\%}{6 \text{ yrs.}} = 16.6667\% \text{ per year}$$

This method allows us to double that, so we will use 33.3333% per year. The $1200.00

```
10 INPUT "COST OF ASSET     = "; C
20 INPUT "SCRAP VALUE IS    = "; SV
30 INPUT "NO. OF YEARS      = "; N
40 D = 2 * (1/N)
50 AG = 0
60 DC = 0
70 DF = 0
80 BV = 0
90 PRINT AG, DC, DF, BV
100 FOR AG = 1 TO N
110 CD = BV * D
120 DF = DF + DC
130 BV = BV - DC
140 PRINT AG, DC, DF, BV
150 NEXT AG
```

Program Listing 7

AGE	DEPRECIATION CHARGE	ACCUMULATED DEPRECIATION	BOOK VALUE AT END OF YEAR
0	0.00	0.00	3600.00
1	1200.00	1200.00	2400.00
2	800.00	2000.00	1600.00
3	533.33	2533.33	1066.67
4	355.56	2888.89	711.11
5	237.04	3125.93	474.07
6	158.02	3283.95	316.05

Table 5

". . . the loan is amortized when you pay the principal and interest by a sequence of payments at equal intervals of time."

depreciation in the first year was the result of multiplying the book value of $36.00 by 0.333333 = 1200.00. The $800.00 in the second year is 33.333% of 2400.00.

```
10 INPUT "COST OF ASSET    ="; C
20 INPUT "SCRAP VALUE IS   ="; SV
30 INPUT "NO. OF YEARS     ="; N
40 D=2 * (1/N)
50 AG=0
60 DC=0
70 DF=0
80 BV=0
90 PRINT AG, DC, DF, BV
100 FOR AG=1 TO N
110 CD=BV * D
120 DF=BV-DC
130 BV=BVF-DC
140 PRINT AG, DC, DF, BV
150 NEXT AG
```

Program Listing 8

This formula never allows you to reduce the asset balance to zero. The book value cannot fall below the scrap value of the asset before the six years are up. Accountants and tax-payers who favor the high initial write-off in the double-declining-balance method have to convert to some other method (often straight-line) towards the end

10	81	1	75.00	825.00	9175.00
11	81	2	68.81	831.19	8343.81
12	81	3	62.58	837.42	7506.39
1	82	4	.		
			.		
			.		
			.		
7	82	10	17.61	882.39	1456.92
8	82	11	10.99	889.01	576.91
9	82	12	4.33	581.24	0.00

Table 6

of the asset depreciation. The IRS allows you to change from the declining-balance method to the straight-line method any time during the useful life of the asset. The unrecovered cost of the asset, less the estimated salvage value, must be spread over the estimated remaining useful life determined at the time of the change. In many circumstances you may also change from Declining Balance to Sum-of-years-digits, and from Sum-of-years-digits to straight-line depreciation. (See IRS Publication 534.)

Program Listing 8 produces the table. Add your own headings and input prompts.

Loan Amortization Table

If you take out a loan with interest, the loan is amortized when you pay the principal and interest by a sequence of payments at equal intervals of time. The payments are almost always equal, and the time interval is usually one month (see Program Listing 9).

Assume you make the following entries in lines 15 through 35:

10000.00, 0.09, 900.00, 10, 81

You borrow $10,000.00 at nine percent per year. The payments are $900.00 per month, and the first payment will be made in October (month number 10) of 1981 (81). The body of the table is in Table 6.

Column 1 is the month number and column 2 is the year. Column 3 is the payment number. Columns 4 and 5 are the portions of the payment applied to the reduction of the principal and the interest on the principal. The first month interest was $75.00 because $10000 * (0.09/12) = 75. If the interest is $75 and you made a $900 payment, then $825 is available to reduce the principal. In line 11 of the table the remaining balance is $576.91, less than a typical $900.00 payment. In this case pay only $576.91 plus the interest on the $576.91, instead of a full $900.00.

Lines 15 through 35 of Program Listing 8 are the prompted variables input and lines 36 through 41 print the column headings. The Print Using statement lines up the numbers nicely under the headings and prints dollars and cents. Line 42 sets the payment number counter to one. Line 45 finds the interest for this month by multiplying the loan balance by a 12th of the interest rate (assuming monthly payments). Line 50 calculates how much of the payment is applied to the principal reduction by subtracting the calculated interest from the amount of the payment. Line 55 calculates the new out-

"The unrecovered cost of the asset, less the estimated salvage value, must be spread over the estimated remaining useful life. . . ."

standing balance. Line 65 checks for an outstanding balance less than the payment. If this is the case, line 400 calculates the values to be printed for the last payment. If this is not the case lines 70 through 97 add one to the payment number and one to the month number. If the month number is 13 it is set back to one (January) and the year number is incremented by one.■

```
10 CLS:PRINT:PRINT TAB(20) "LOAN AMORTIZATION TABLE":PRINT
15 INPUT "LOAN PRINCIPAL  = "; P
20 INPUT "INTEREST RATE   = "; R
25 INPUT "TOTAL PAYMENT   = "; M
30 INPUT "STARTING MONTH = "; MO
35 INPUT "STARTING YEAR   = "; YR
36 PRINT
38 A$ = " ## #### ### #####.## #####.## #####.##"
40 PRINT"MO  YEAR   N   INTEREST  TO PRIN.  REMAIN."
41 PRINT
42 N = 1
45 I = P * (R/12)
50 PR = M - 1
55 PR = M - 1
60 PRIN USING A$; MO, YR, N, I, PR, P
65 IF P + (P * (R/12)) < = M THEN 400
70 N = N + 1
75 MO = MO + 1
80 IF MO = 13 THEN 90
85 GOTO 45
90 MO = 1
95 YR = YR +
96 GOTO 45
400 N = N + 1
410 NO = MO + 1
420 IF MO < = 12 THEN 430
422 MO = 1
423 YR = YR + 1
430 I = P * (R/12)
435 PR = P + I
440 P = 0
450 PRINT USING A$; MO, YR, N, I, PR, P
455 PRINT
999 END
```

Program Listing 9

ABM Commander

The United States is under attack! You control the master ABM targetry and must defend your six cities from the onslaught of ICBM's, MIRV's, smart bombs, strategic bombers, and killer satellites. Hundreds of millions of lives rest on your reflexes and your supply of anti-ballistic missiles.

Will you still own this game in 1990? Then switch to **Laser Commander**! You control the sites of a land-based anti-ballistic laser. Blast the enemy as he descends, but be prudent! Your energy depletes with each blast, and your power supply depends on your remaining cities.

Features include:

- One or two players
- Multiple difficulty levels
- Fast machine-language graphics
- A guarantee of satisfaction
- Two complete games in one
- Sound effects

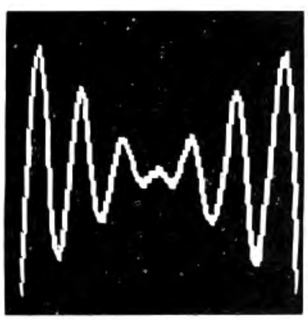
ABM/Laser Commander is the best game of its type on the TRS-80. Other versions don't even come close. We guarantee you will agree!

Model I or Model III 16K Level II only
Cassette . . . $14.95 Diskette . . . $17.95

Laser Commander

To Order Call Med Systems ✓128
1-800-334-5470

GRBASIC 2.0



Our best selling **GRBASIC** has been upgraded. **GRBASIC** adds a set of commands to BASIC that allow the user to produce machine language speed graphics directly from BASIC. This is ***not*** a string packer or a USR function. New BASIC commands are added ***directly*** to BASIC. All existing programs will still run.

Trade-In Policy for GRBASIC 1.0

Send us the *original* **GRBASIC** disk or cassette with $5.00 and we will send you **GRBASIC 2.0.** We must have the original media returned. *No exceptions.*

GRBASIC 2.0 *Model I and Model III 16K and up*
Cassette . . . $19.95
GRBASIC 2.0 *Model I and Model III* *Diskette . . . $24.95*
Please Specify DOS System When Ordering!

Some Features of GRBASIC 2.0

Draws or erases lines between any set of points on the screen. Coordinates may be chained. A corner to corner line is drawn in under 6/100ths of a second!

Draws a circle of any given radius at a given point. Circles may be drawn off the edge of the screen.

Draws a shape at any given point, using the last specified rotation and magnification. Any shape of a multiple set can be specified. Shapes can be drawn off the screen in extended space, and are user-definable with no limitations except the TRS-80's resolution.

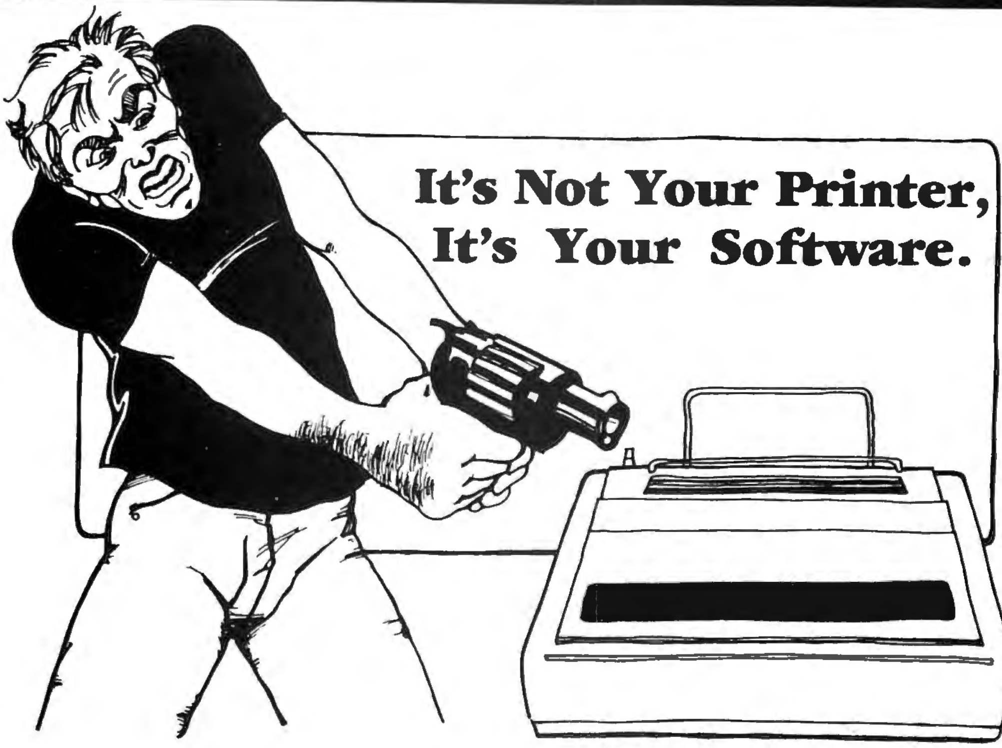
Magnifies a defined shape from 1-255. This changes the size of a previously defined shape.

Rotates a shape in 45 degree increments.

Sends sound to the cassette port. User specifies range of frequency and duration of each frequency. Repeat factor is available.

Extensive, newly re-written manual.

MED SYSTEMS
p.o. box 2674
chapel hill, nc 27514
1·800·334·5470



Some Features of Qwerty 3.0

1. Automatically prints in proportional print, with a suitable format. Transitions between the three print styles are easy, including all expanded print modes.
2. **Qwerty 3.0** adds 75 new symbols, including upper and lower case Greek letters, mathematical symbols such as integrals and summations, arrows, brackets, and probability symbols.
3. Any character can be used as a subscript or superscript, even simultaneously. Carats, bars, and tildes can be placed over any character, with precise position control.
4. Underlining, with or without underlining of spaces, including long ratios and mathematical expressions.
5. **Footnotes** can be placed on any page so that they remain on the desired page, even if text is inserted later.
6. **Table** commands enable positioning of the print head anywhere on a line. Invaluable in printing neat mathematical layouts, tables, columnar material, etc., in proportional print.
7. **Pretty** commands allow printing of repetitions of a chosen character. When combined with **Table**, decorative borders can be produced with ease.
8. **Folio** format produces output in two or three columns per page, in either proportional or 16.7 cpi mode. Ideal for newsletters.
9. Supplies a third output mode, in which only **Scripsit** commands are obeyed. Allows printing of special **Qwerty** commands for future reference.
10. **Page End** indicates where pages will end, and the page number, without printing the text. One can prepare an almost error-free document without ever using paper.
11. Correction of **Scripsit**'s errors and inconveniences, extensive documentation, and much more!

Since we introduced **Qwerty 3.0** in September, people have been calling to ask if we were making ludicrous claims. *The answer is* ***NO!*** **Qwerty 3.0** does everything we claim and more! Like Greek letters. Simultaneous subscripts and superscripts. Even footnotes and two or three column folio formats. See the list of features in the box. No other piece of software of this type can match **Qwerty 3.0**.

Qwerty 3.0 is more than "fully tested." It has seen thousands of hours of use in a university environment. A master's thesis and a statistical doctoral dissertation were produced and accepted right off a Centronics 737 using this package.

Qwerty 3.0 is the finest **Scripsit** addition available. All you need is **Scripsit**, at least one disk drive, and one of the following printers: Centronics 737, Centronics 739, or Lineprinter IV. It is the best. Period. ***We Guarantee You Will Agree!*** If for some reason, you find that this program does not meet your needs, return the entire package within 14 days for a prompt and cheerful refund.

For cautious buyers, we offer the manual (almost 70 pages) for $10.00. When you decide to buy **Qwerty 3.0**, we will credit the full manual price.

Centronics 737 printers were still available at this writing. *Med Systems* sells them. Please call for price and availability.

Qwerty 3.0 Disk, Manual, Reference Card, and Printer Table Rule — $49.95

Qwerty 3.0 Manual Alone — $10.00

MED SYSTEMS

128

You use them in prose, so use them in code.

In Praise of Outlines

Dennis Drew
908 Byers
Joplin, Missouri 64801

I teach programming classes using the TRS-80, and there is something I notice about all new programmers: they do things the hard way. For example, if I ask them to explain the following lines: INPUT "ENTER YOUR NAME ":A$. The explanation will sound like: "This line assigns a literal value into the string variable A$ using input from the keyboard."

What's wrong with saying, "It lets you enter your name into the computer." The use of "twelve large words when one small one will do" is not confined to newcomers either. I consider myself a fair programmer, yet I have to sweat through many articles I read in computer magazines.

STRING VARIABLES:

Variable name/purpose	Location (lines)
A$ Holds list data	10,50,1100–1500, 2040,2060,2080, 2100,2120,3010–3030,4030–4050, 5010,6020,6040–6080
B$ Decision	20,30,2010–2120,4010
C$ Small Sort marker	6010,6020
D$ Sort saver	6040–6080
NUMERIC VARIABLES:	
A For/next counter	4050,2020,3000, 4000,5000,5020, 6020
B,C,D Tape counters	40,50,5000,5010
E Choice	60–80
F Number of items	40,100,1100–1150,2020,3000, 4000,5000,6020, 6090
G For. . .Next counter	4060
H Sort marker	6020
I Sort marker	6020,6040–6100

Program Variables

A Question of Difficulty

How hard is it to write a program? In my early programming days I wrote a game called PHASE VII (available from Cheshire Software, P.O. Box 1295, Joplin, MO 64802). It was a difficult job and, because I knew nothing about program preparation, it took two years to write the game. It should have taken two or three months; but I had to restart three times.

Imagine the following: You have worked several hours on a program. There are about thirty major routines, a hundred minor ones, dozens of variables and everything has to fit together just right. Your brain is weary. Suddenly, you cannot remember the purpose of a certain variable. Now you have big trouble. Like a line of dominoes, the program starts falling, and you can do nothing to stop it. *Crash!* What a mess!

A Writer Doesn't Just Write

When a professional writer begins a novel he starts with a theme, which he writes down, putting the general plot on paper. After this, he writes an outline, a simple explanation of the story, piece by piece. He then writes a rough draft which follows the outline closely. Once the rough draft is finished, he re-reads to correct and revise. Using this rough draft, he writes the finished version. Then he lets a reliable friend read it and comment on it before showing it to anyone else. To do otherwise would be suicide in the writing market.

A Programmer Doesn't Just Code

Coding is actually writing the program in

"I consider myself a fair programmer, yet I have to sweat through many articles I read in computer magazines."

computer language. If a programmer tries to write a program starting with coding he will die a slow, painful death, with bytes all over his body (sorry).

There is a set of easy-to-follow steps that I always use:

1. Define the task
2. Outline the procedure
3. Code
4. Enter
5. Run
6. Debug
7. GOTO 5

These steps can save hours of programming time and effort. Let's look at each step individually, while writing a small sample program. Then we will apply the principles to a larger program.

Define the Task

Write a mailing list program which accepts 100 names and addresses of three lines each, allows corrections, prints addresses on envelopes, prints names on the screen, stores names on tape, retrieves names from tape and sorts by zip codes (we use third class mail).

Outline the Procedure

There are different ways to outline procedure. Some folks diagram with flowcharts, drawings of computer programs. Flowcharts have their use, but I never flowchart. Why? Because in my mind it is too close to coding, and is no substitute for outlining. Let me give an example by writing a standard flowchart for the definition (see Fig. 1).

That flowchart is not simple. I use another method that fits my natural work-saving (lazy) attitudes. I outline in simple English.

If you can't explain a program in simple English, you sure can't write it in Basic. Your mother taught you English when you were a child; you should know the language far better than Basic. Use the power of your native language. This is my outline of the task:

1. Start with a zero.
2. Add five if the number is less than 100.
3. If the number equals 100 end the program.
4. Print the number and its square root in columns.
5. Return to step two.

The outline is easier to understand and faster to write than a flowchart.

Code the Program

Of course, to get the program to work on the computer, it has to be entered in Basic. Let's code it now following the outline:

- Start with zero.

  10 X = 0 (this line should be eliminated on the TRS-80.)
- Add five if the number is less than 100. If the number equals 100 end the program.

  20 IF X<100 X = X + 5 ELSE END
- Print the number and its square root in columns.

  30 PRINT X, SQR(X),
- Continue.

  40 GOTO 20

Of course you now Enter and Run the program. Despite having taken the correct steps so far, the whole thing is not worth a bit (sorry again) unless it's correct.

Debug

Too often we see programs laced with bugs; I am tired of entering 16K adventure and Star Trek games that don't work.

Debugging a program means correcting errors; It sometimes takes as much (if not more) time than writing the program in the first place. Do not run a program once and decide it works. I put all of my programs through *hours* of actual execution before I

```
10 CLS:CLEAR 10800:DIMA$(101,5)
20 INPUT"DO YOU WISH TO PULL DATA FROM TAPE ";B
$
30 IF LEFT$(B$,1)="N"60
40 INPUT"PREPARE TAPE TO PLAY. PRESS ENTER.";F:
INPUT#-1,F:FORA=1TOFSTEP4:PRINTA:B=A+1:C=A+2:D=
A+3
50 INPUT#-1,A$(A,1),A$(A,2),A$(A,3),A$(A,4),A$(
A,5),A$(B,1),A$(B,2),A$(B,3),A$(B,4),A$(B,5),A$
(C,1),A$(C,2),A$(C,3),A$(C,4),A$(C,5),A$(D,1),A
$(D,2),A$(D,3),A$(D,4),A$(D,5):NEXT
60 CLS:PRINT"DO YOU WISH TO  1-ENTER  2-CHANGE
 3-LIST  4-PRINT ENVELOPES    5-STORE  6 SORT "
;
70 INPUTE
80 IFE<1ORE>6THEN60
90 ONEGOTO1000,2000,3000,4000,5000,6000
1000 PRINT:F=F+1:PRINTF
1100 INPUT"NAME ";A$(F,1)
1200 INPUT"ADD: ";A$(F,2)
1300 INPUT"CITY: ";A$(F,3)
1400 INPUT"STATE: ";A$(F,4)
1500 INPUT"ZIP: ";A$(F,5)
1600 GOTO60
2000 PRINT:PRINT"WHICH NAME DO YOU WISH TO CHAN
GE?"
2010 INPUTB$
2020 FORA=1TOF:IFA$(A,1)=B$THEN2030ELSENEXT
2030 INPUT"NAME: ";B$
2040 IFB$=""THEN2050ELSEA$(A,1)=B$
2050 INPUT"ADD: ";B$
2060 IFB$=""THEN2070ELSEA$(A,2)=B$
2070 INPUT"CITY: ";B$
2080 IFB$=""THEN2090ELSEA$(A,3)=B$
2090 INPUT"STATE: ";B$
2100 IFB$=""THEN2110ELSEA$(A,4)=B$
2110 INPUT"ZIP: ";B$
2120 IFB$=""THEN2130ELSEA$(A,5)=B$
2130 GOTO60
3000 CLS:FORA=1TOF:PRINTA
3010 PRINTA$(A,1)
3020 PRINTA$(A,2)
3030 PRINTA$(A,3);", ";A$(A,4);"  ";A$(A,5)
3040 PRINT:NEXT:FORA=1TO500:NEXT:GOTO60
4000 CLS:FORA=1TOF
4010 INPUT"INSERT ENVELOPE. PRESS ENTER ";B$
4020 CLS:PRINT"PLEASE WAIT
4030 LPRINTA$(A,1)
4040 LPRINTA$(A,2)
4050 LPRINTA$(A,3);", ";A$(A,4);"  ";A$(A,5)
4060 FORG=1TO12:LPRINT" ":NEXT
4070 NEXT:GOTO60
5000 CLS:PRINT"PREPARE TO RECORD. ENTER":INPUTA
:PRINT#-1,F:FORA=1TOFSTEP4:B=A+1:C=A+2:D=A+3
5010 PRINT#-1,A$(A,1),A$(A,2),A$(A,3),A$(A,4),A
$(A,5),A$(B,1),A$(B,2),A$(B,3),A$(B,4),A$(B,5),
A$(C,1),A$(C,2),A$(C,3),A$(C,4),A$(C,5),A$(D,1)
,A$(D,2),A$(D,3),A$(D,4),A$(D,5)
5020 PRINTA:NEXT
5030 GOTO60
6000 CLS:PRINT"SORTING. PLEASE WAIT
6010 C$="99999"
6020 I=I+1:FORA=ITOF:IFA$(A,5)<C$THENC$=A$(A,5)
:H=A
6030 NEXT
6040 D$=A$(I,1):A$(I,1)=A$(H,1):A$(H,1)=D$
6050 D$=A$(I,2):A$(I,2)=A$(H,2):A$(H,2)=D$
6060 D$=A$(I,3):A$(I,3)=A$(H,3):A$(H,3)=D$
6070 D$=A$(I,4):A$(I,4)=A$(H,4):A$(H,4)=D$
6080 D$=A$(I,5):A$(I,5)=A$(H,5):A$(H,5)=D$
6090 IFI=F+1I=0:GOTO60
6100 PRINTI:GOTO6010
```

Program Listing 1.

"Any child can write a program; a good programmer writes bug-free programs."

release them for publication. Your programs are no less important. Any child can write a program; a good programmer writes bug-free programs.

Outline the Procedure

1. Clear screen. Reserve string space (100 times 3 times 36 equals 10,800). Set dimensions (name, address, city, state, zip code times 100 plus 1 for extra element in For. . . Next loops equals 101 by 5).
2. Allow pulling of data from tape if desired.
   Pull total number of names.
   Loop through total number of names four at a time to save tape time.
3. Give program options: Clear screen; Correct or change; List on screen; Print envelopes; Store names on tape; Sort by zip; Go to the proper routine.
4. Entry routines: Add 1 to total number of names. Print total. Enter name. Enter address. Enter city, state, zip (in separate variables). Go to Item 3.
5. Change routine: Allow entry of name. Find correct name in list by using a comparing loop. Allow corrections. Go to item 3.
6. List routine: Loop 1 through the total number of names. Display name, address, city, state, zip. Print a blank line. Finish loop. Go to Item 3.
7. Print routine: Loop 1 through total number of names. Prompt to insert envelope. Print an envelope. Feed envelope to operator. Finish loop. Go to Item 3.
8. Store routine: Loop 1 through total number four at a time. Store four on tape. Finish loop. Go to Item 3.
9. Sort routine: Start main loop to go through number of names. Start minor loop 1 through number of names. Find smallest item in list. Exchange with item at top of list. Finish minor loop. Finish main loop. Go to item 3.

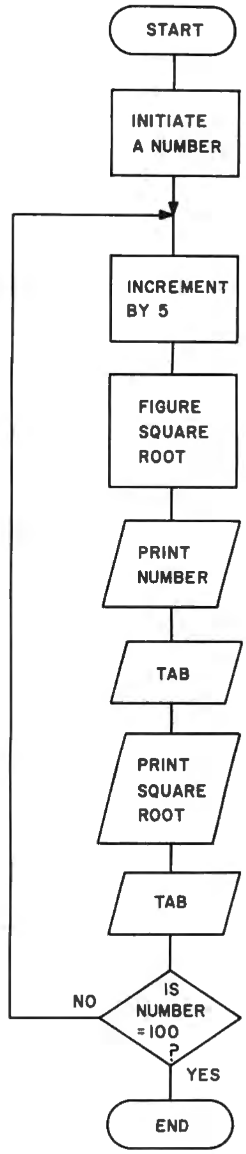


Fig. 1. Flowchart.

End of outline

Code: You can follow the program by comparing it to the outline (see Program Listing 1).

The last thing we need to cover is documentation. I do not use REM statements, because they take up computer space and slow down programs. For example, I clocked the following program at 10.82 seconds.

```
10 FOR X = 1 TO 5000
20 NEXT X
```

After adding the following line: 15 REM the program took 14.68 seconds. A few REM statements outside of loops will not make a lot of difference, but REM statements also consume memory. If you write a program and run out of space the first thing you do is cut REM statements. Why put them there in the first place? RAM is valuable, so do not clog it up with REM.

The solution is to use pencil and paper. On paper you have all the room for REM statements and variable notation you need. Do not skimp on documentation! This is very important if you expect to program in a professional manner. If you look at an undocumented program six months from now you probably will not understand it.

It is necessary to write down REM-type statements in a non-menu driven program to document programming routines. In the mailing list program, the menu in line 60 and 80 will serve this purpose.

This demonstrates the art of programming simply and effectively. If you follow the seven basic steps every time you write a program, you will save time and memory, and avoid extra unnecessary work.■

Color computer owners,

32K PLUS DISKS* $298.00

Yes, that's right - for as little as $298.00 you can add 32K of dynamic RAM, and a disk interface, to your TRS-80 Color Computer! If you just want the extra memory it's only $199.00, and you can add the disk interface later for $99.00.

Just plug the *Color Computer Interface (CCI)*, from Exatron, into your expansion socket and "Hey Presto!" - an extra 32K of memory. No modifications are needed to your computer, so you don't void your Radio Shack warranty, and Exatron give both a 30 day money-back guarantee and full 1 year repair warranty on their interface.

The *CCI* also contains a 2K machine-language monitor, with which you can examine (and change) memory, set break-points, set memory to a constant and block-move memory.

So what about the *CCI Disk Card*? Well as we said it's only an extra $99.00, but you'll probably want Exatron's *CCDOS* which is only $29.95 – unless you want to write your own operating system. The *CCI Disk Card* uses normal TRS-80 Model I type disk drives, and *CCDOS* will even load Model I TRSDOS disks into your color computer – so you can adapt existing TRS-80 BASIC programs.

As a further plus, with the optional *ROM Backup* adaptor, you can dump game cartridges to cassette or disk. Once the ROM cartridge is on cassette, or disk, you can reload, examine and modify the software. The *ROM Backup* adaptor is only $19.95.

For more information, or to place an order, phone Exatron on their Hot Line *800-538 8559* (inside California 408-737 7111), or clip the coupon.

excellence in electronics

exatron

DEALER ENQUIRIES INVITED

Exatron, ✓500
181 Commercial Street,
Sunnyvale, CA 94086

☐ Please send a 32K Color Computer Interface for $199.00

☐ Please send a CCI Disk Card for $99.00

☐ Please include CCDOS and manual for $29.95

☐ Also include a ROM Backup adaptor for $19.95

Please add $5.00 for shipping to all orders, and 6 percent sales tax in California.

Name ....

Address ....

City ....

State .... Zip ....

Charge my:

☐ MasterCard Interbank Code ....

☐ Visa Expiration Date ....

Card ....

☐ Check enclosed for ....

☐ Ship COD ($2.00 extra)

Signature ....

*32K RAM plus Disk Interface TRS-80 is a trademark of Tandy

A to D Alchemy.

Real World Interface—Part II

Elliott K. Rand
P.O. Box 552
Melbourne, FL 32901

There is a need for direct communication between personal computers and the real world. Without it the entry of sophisticated data is limited to input through a keyboard. This precludes real time applications, requires at least one additional level of abstraction, and introduces human error into a system.

The computer counts in whole numbers which change abruptly. Since the computer counts in jumps while the real world is continuous, converting one to the other means some small error must be tolerated (Fig. 1). Each increment of an eight-bit word is .4 percent of the full amount. That would be like measuring a foot and having a maximum error of less than 1/20 inch. If you need greater accuracy and you can cope with their complexity and cost, you can use a 10-bit (.1 percent) or 12-bit .025 percent) system. Most requirements are satisfied by an eight-bit system. Small errors like two plus two equals 3.984 are not a problem, but big errors can result if you assume a computer means 4.000000 when it shows you a 4.

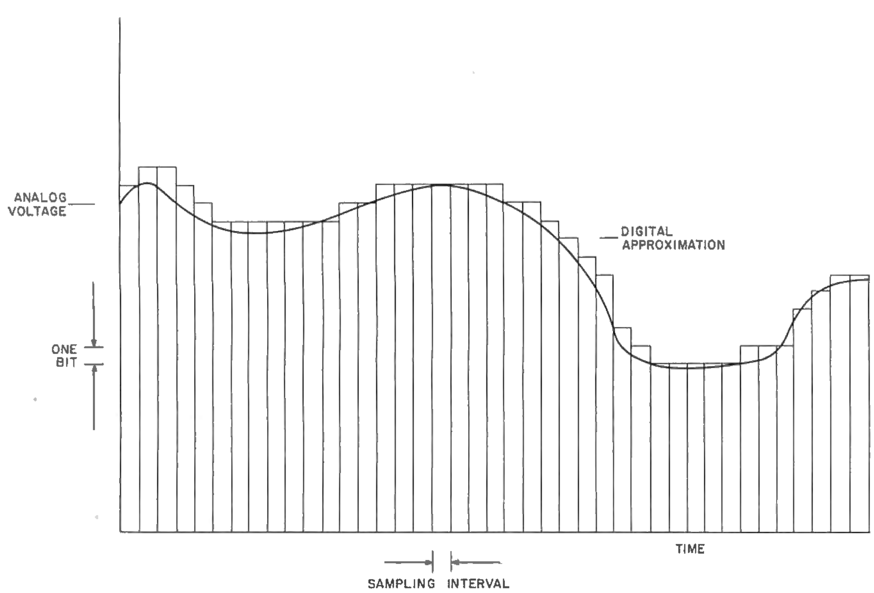


Fig. 1. Digitized Analog Signal.

Computer Talks to Real World

You need a means to convert the digital word into the power to drive a motor, vary an audio tone, or control a light source. A digital-to-analog converter can make that conversion. It can do it with either of two types of resistance networks (Fig. 2). With the supply voltages shown in Fig. 2, pegging the digital switches at the same settings in both systems will give identical current flow through the ammeter.

Converter Resistances

Binary-weighted resistances in a ratio of 2^n (1, 2, 4, 8, etc.) are connected as shown in Fig. 2a. In a four switch system 15 steps of 1/8 ampere from zero amperes to 15/8 amperes provide the 16 levels of a 4-bit system according to the closing of the appropriate switches.

This system found its widest use when complete digital-analog converters were not available on a chip. In this type of conversion, accuracy suffers if many bits are used. The range of resistor values in an eight-bit system would be from R to 256R. Resistor tolerances of .1 percent or better of non-standard higher values are required. It is difficult to maintain the system's linear properties after four or five bits because of its unpredictable drift with variations in temperature and other elements.

Climbing the R-2R-Ladder

Resistances R and 2R are connected as shown in Fig. 2b. With all switches grounded, the resistance from any node is 2R in any direction. Closing any switch divides the current at each node. Connecting S1—the farthest switch from the ammeter—to the minus 4V source makes the current through the ammeter 1/16 of the current through S1; connecting S4—the switch closest to the ammeter—makes the current through the ammeter 1/2 the current through S4. Thevenin's theorem or the principle of superposition will show the current flow through the ammeter is an analog equivalent of the digital value represented by the switch settings.

The major advantage of this system is the ease of fabricating extremely closely-matched resistances right on the chip with values of R. The requirement for twice the number of resistors is not a disadvantage considering microcircuit dimensions.

Affordable Chip

Current-to-voltage conversion is easily accomplished through the use of an operational amplifier. (Fig. 3.) By selecting the re-

"These transistors are inserted within the feedback loop to provide up to 75 mA . . ."

quired value for R_F we can achieve any desired multiplication. The mechanical switches may be replaced with FET switches accessed by the data bus.

A widely used integrated circuit, DAC-08, available from Precision Monolithics and National Semiconductor, provides R-2R capabilities on a single chip. Wide use has made the chip very affordable. A block diagram and an equivalent circuit of the DAC-08 are shown in Figs. 8a and b.

Inside the Beast

A schematic of the digital-to-analog section of the real world interface is shown in Fig. 4. A reference voltage (V_r) is generated across VR1 for use within the digital-analog integrated circuit. Maximum voltage output of the system is 255/256 times VrRm/Rf, available at pin six of 741 operational amplifier AR1. This voltage is used to drive transistors Q1 and Q2. Those transistors are inserted within the feedback loop to provide up to 75 mA output current from the digital-analog converter. This low impedance output voltage is returned to the negative input of AR1. This ensures that the output voltage will remain constant as long as the load does not draw more than 75 mA (the limit established by the 2W 220-ohm collector resistors).

Getting Your Feet Wet

To perform these experiments, all integrated circuits except IC7 must be installed. If IC7 is already installed it does not need to be removed. Both the plus five volt and the plus-minus 15 volt direct current supplies are also required.

Before attempting these experiments be sure to remove power from the TRS-80 and all power supplies. Never leave the power on while setting up or reconfiguring test hardware.

If the interconnect cable is not already installed between the TRS-80 and the real world interface connect the card-edge connector to the expansion port with the ribbon-cable exiting upward. Install the 40-pin DIP plug into SO1. Pin one of the 40-pin DIP plug goes to pin one of SO1.

Connect the plus five volt and the plus-minus 15 volt direct current supplies to the real world interface. Connect a direct current voltmeter capable of reading 15 volts across the digital-analog output and ground.

Turn on the plus five volt supply. All light emitting diodes (LEDs) should light. Turn on the plus-minus 15 volt supply. The voltmeter should indicate approximately 10.25 volts. Turn on the TRS-80.

A memory size start-up routine should occur. No memory need be reserved for these experiments, so press Enter to get into Level II Basic.

Enter and run:

```
10 OUT 127,137 'Configures 8255
20 OUT 124,128 'Loads 80H into Port A
```

You should observe only the most-significant-bit (MSB) (left-most) LED lit. The meter should read exactly one-half the meter reading on power-up.

Run the program after changing line 20 to 20 OUT 124,64 'Loads 40H into Port A. You should observe only the second MSB LED lit. The meter should read exactly one-quarter of the meter reading on power-up.

You may want to try other values in line 20. Note that an error message occurs if a value outside the range of 0-255 is attempted, and quantities to the right of the decimal point are ignored.

Enter and run:

```
10 OUT 127,137
20 OUT 124,064
30 OUT 124,128
40 GOTO 20
```

You should observe the two MSB LEDs appear lit while the meter reads somewhere between one-quarter and one-half of the meter reading on power-up. The system is switching between the two voltages too fast for the meter or eye to follow it.

Run the program after adding 25 FOR P = 1 TO 100:NEXT P and 35 FOR P = 1 TO 100:NEXT P. Now the LEDs toggle visibly and the meter needle swings between the two values.

Change lines 25 and 35 to 25 FOR P = 1 TO 10:NEXT P and 35 FOR P = 1 TO 10:NEXT P and run the program. The LEDs will toggle rapidly while the meter needle

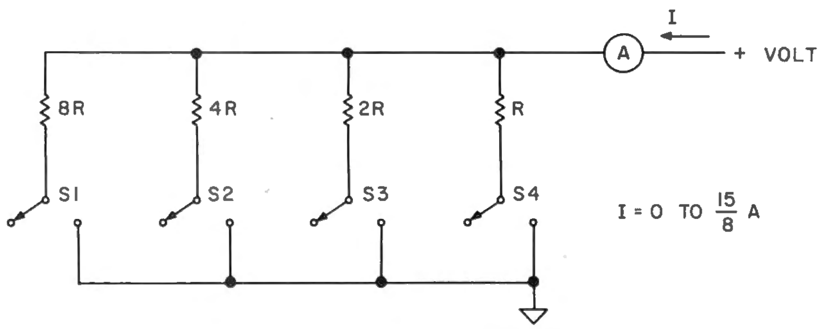


Fig. 2a. Binary weighted D/A converter.

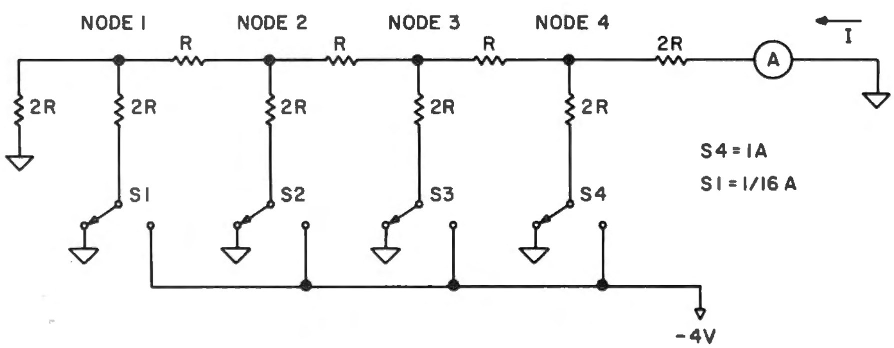


Fig. 2b. R, 2R Ladder D/A converter.

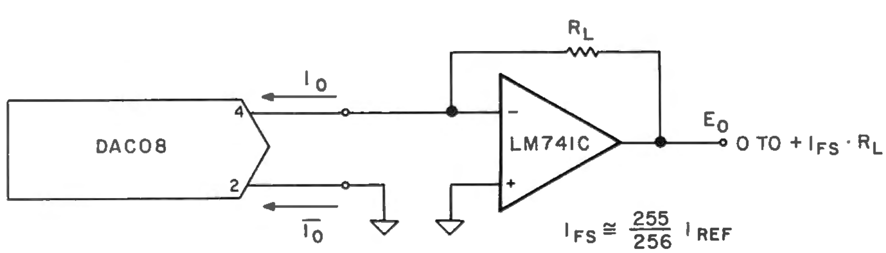
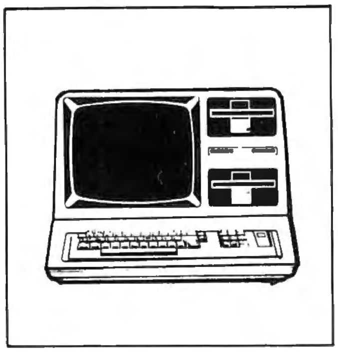


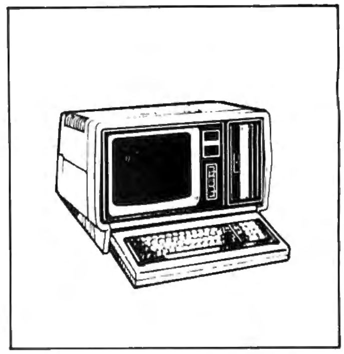
Fig. 3. Positive Low-Impedance Output Operation

LEVEL IV PRODUCTS, INC.

– IS NOW OFFERING –

The Full TRS-80* Line at Discount Prices, with Extended Warranty Coverage





COLOR COMPUTER	
16K (R)	$440
16K (E)	$359
16K w/ext basic (R)	$529
16K w/ext basic (E)	$479

MOD III COMPUTER	
16K (R)	$859
48K 2-40 drives (R)	$2010
48K 2-80 drives (E)	$1829
48K 2-80 drives (E)	$2119
48K 2-80/80 drives (E)	$2419
RS232 interface add	$89

MOD II COMPUTER	
32K (R)	$2950
64K (R)	$3375
Expansion Dr#1 (R)	$999
Expansion Dr#1-2-3 (E)	$695

NOTE: THE MOD III CAN BE ASSEMBLED WITH UP TO 4 MEG UNFORMATTED FLOPPY DISK STORAGE CAPACITY (4-80 TRK DOUBLE SIDED DRIVES) – CALL FOR A QUOTE ON THE SYSTEM TAILORED TO YOUR EXACT NEEDS.

PRINTERS	
Line Printer IV (R)	$849
Line Printer VI (R)	$986
Daisy Wheel II (R)	$1679
C. ITOH Starwriters	Call
Centronics All	Call
Epson All	All
Anadex All	Call

MPI DRIVES FOR: MOD I, MOD III, & COLOR

MODEL	RAW	SINGLE	DUAL
40 TRK (E)	$240	$298	$576
40/40 TRK (E)	$360	$419	$818
80 TRK (E)	$360	$419	$818
80/80 TRK (E)	$502	$549	$1078

PERIPHERALS	
E.I. OK (R)	$249
E.I. 16K (R)	$359
E.I. 32K (R)	$469
E.I. 16K (E)	$305
E.I. 32K (E)	$362
16K RAM (200MS)	$ 39

DEALER'S INQUIRIES INVITED, CALL FOR OUR WHOLESALE DEPARTMENT (313) 525-6201

WARRANTY POLICY

Items including non-Radio Shack components (E) are covered by an extended six-month parts/labor warranty in our tech service center.

Items constructed from Radio Shack approved components (R) are covered by a 90-day parts/labor warranty at any Radio Shack store, plus an additional 90 days in our tech service center.

TERMS OF SALE

2% Discount allowed on cash pre-paid orders. Personal checks require two weeks for bank clearance.

Visa, Master Charge and C.O.D. orders accepted. Add $1.50 for C.O.D.'s

Shipping/handling/insurance add 1.5% or $2.50 minimum.

Prompt shipments from stock.

LEVEL IV PRODUCTS, INC.
32461 SCHOOLCRAFT ROAD
LIVONIA, MICHIGAN 48150

MICHIGAN – (313) 525-6200
TOLL FREE – 800-521-3305

*TRADEMARK OF TANDY CORP.

"No memory need be reserved for these dithers between limits of its earlier swings."

dithers between the limits of its earlier swings. An oscilloscope can be connected to the digital-analog output terminals for a more informative examination of the various signals developed.

Next enter and Run100:

```
100 OUT 127,137
110 FOR N = 1 TO 255
120 OUT 124,N
130 NEXT N
140 GOTO 110
```

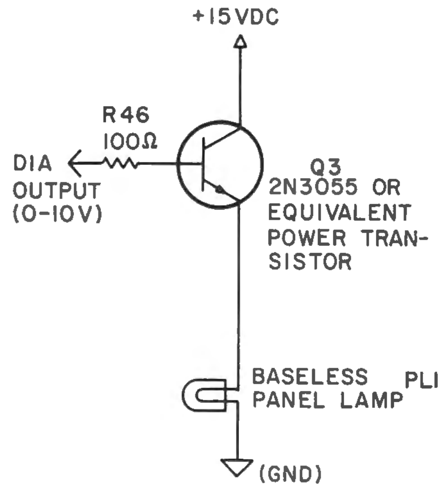


Fig. 5. Lamp Driver Circuitry.

You should observe the LEDs continuously counting from 0–255. The meter swings from zero to maximum in about two seconds and quickly returns to zero at the end of each count. An oscilloscope will show a staircase-sawtooth waveform.

Change line 140 and add the lines:

```
140 FOR N = 255 TO 0 STEP -1
150 OUT 124, N
160 NEXT N
170 GOTO 110
```

Run the program again. Now the voltmeter swings from zero to maximum and back to zero, describing a triangular waveform.

Add 125 FOR P = 1 TO 10:NEXT P and 155 FOR P = 1 TO 10:NEXT P and Run100 the program. Now the ramp may be more closely studied.

Enter and Run200:

```
200 OUT 127,137
210 INPUT "ENTER DECIMAL NUMBER 0-255";M
220 IF M<0 OR M>255 GOTO 210
230 OUT 124, M
240 GOTO 210
```

The voltmeter will indicate a voltage pro-

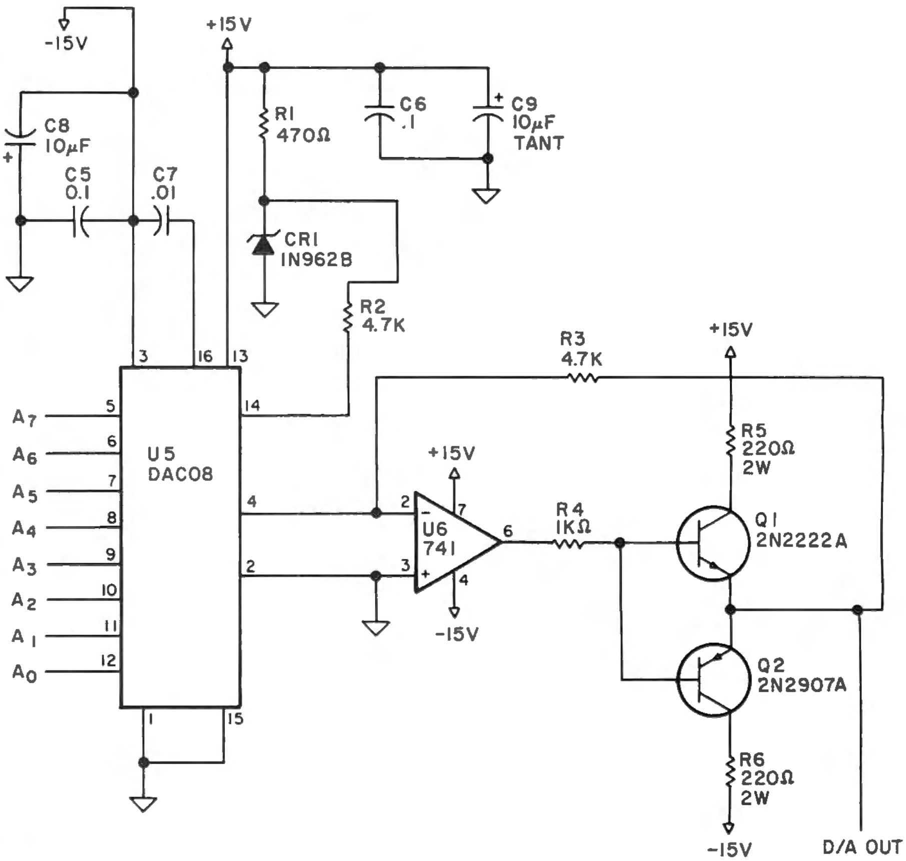


Fig. 4. Digital-to-Analog Converter.

✓See List of Advertisers on page 402

"Without the spacer the heat-sink will be at the same voltage. . ."

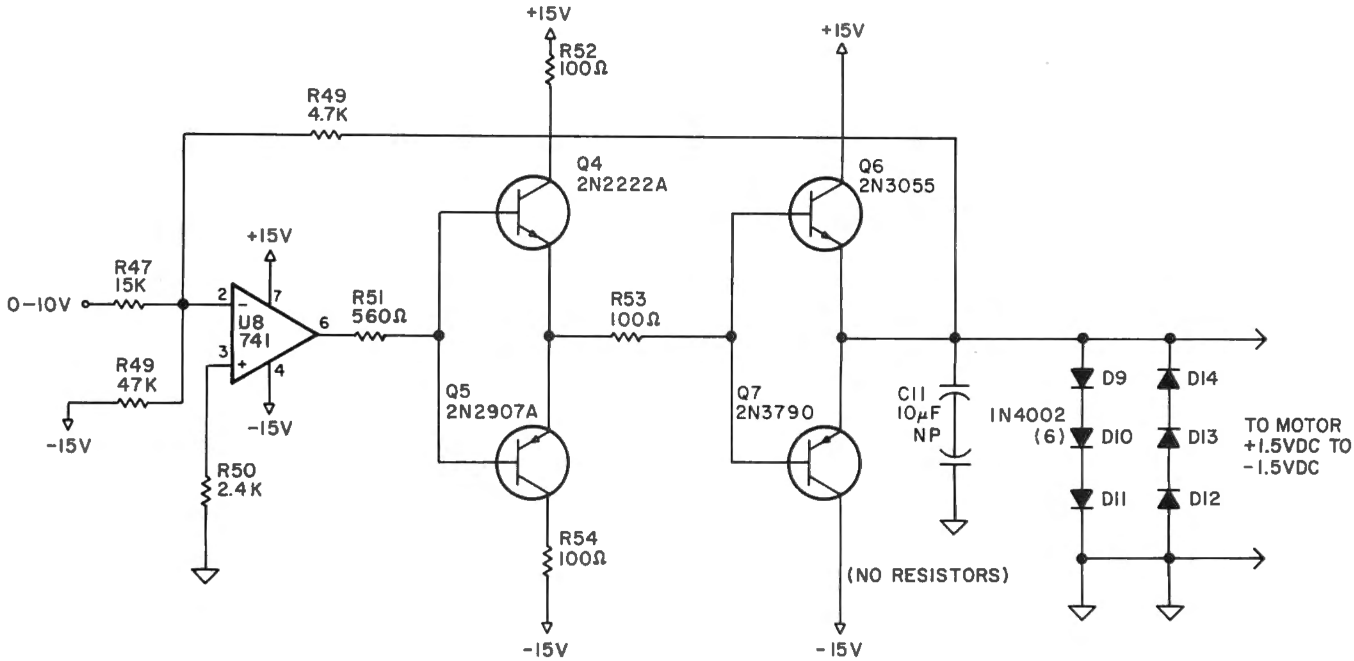


Fig. 6. Bi-directional, Variable-Speed, Permanent-Magnet Motor Driver.

portional to the decimal input (10X/255 volts). The LEDs will indicate the binary equivalent of the decimal input. Both the voltmeter reading and the LEDs still only indicate an abstraction of a physical condition and have no real meaning to the physical world.

The Reality Connection

Connect the lamp driver circuit (Fig. 5) to the output of the digital-analog converter. Use a 2N3055 NPN power transistor or equivalent with a heat-sink. Be sure to apply some silicone grease between the transistor and heat-sink to maximize the transfer of heat and put an insulating spacer between them. Without the spacer the heat-sink will be at the same voltage as the transistor collector. Avoid inadvertent contact with other circuitry.

This circuit provides adequate current to power a small panel lamp. A baseless 12-volt bulb, like the kind used in automobile instrument panels, is convenient to use with clip leads, or inserted directly into a solderless breadboard or the space provided on the printed circuit board, avoiding the requirement for sockets and other items.

Run all the programs I've described with the panel lamp circuitry and the meter across the digital-analog output. Program operation will control the light bulb's brilliance. Note the voltmeter indicates several volts above zero before the lamp glows.

The 0–10 volt direct current output from the digital-analog converter can be translated into a proportional voltage of about minus 1.5 vdc-plus 1.5 vdc (see Fig. 6).

Operational amplifier U8 sums the minus 15 vdc and the 0–10 vdc inputs. With the values shown in Fig. 6, an output of minus 1.5-1.5 vdc results from converter output voltages of 0–10 vdc. Transistors Q4–Q7 increase the circuit current-driving capability from a few mA from U8 to one Ampere.

Connect the motor-driver to the appropriate power supplies. Do not connect the permanent-magnet motor to the motor-driver output circuitry yet.

Type and enter: OUT 127,137: OUT124,128. This sets the motor-drive output to zero volts. Now connect the permanent-magnet motor to the motor drive output circuitry.

This motor is a 1.5 volt direct current motor used in small tape-recorders and battery-operated toys. (Radio Shack #273-208 or equivalent.) Clamp the motor in a small vise to prevent it from jumping around.

Run200 the program previously loaded. For numbers lower than 105 rotation of the motor shaft will be in one direction. Speed of rotation will increase as the numbers approach zero. For numbers higher than 150 rotation of the motor shaft will be in the opposite direction, increasing as the number approaches 255. Painting a spiral on the motor shaft with nail polish or white correction fluid makes the direction of rotation of the shaft more visible.

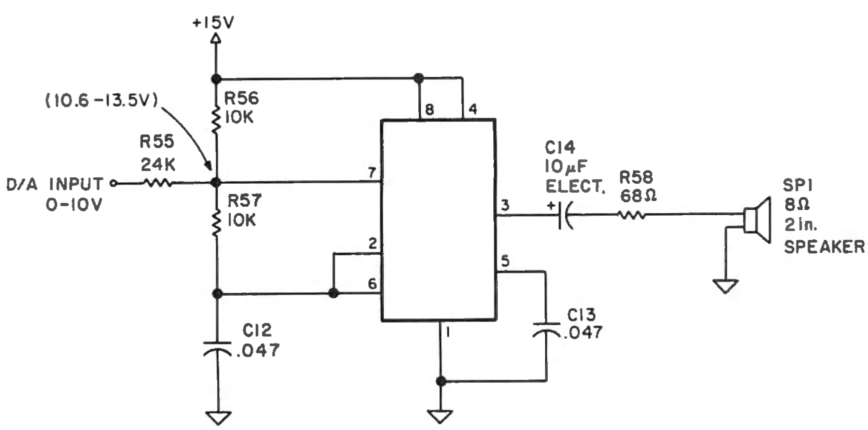


Fig. 7. Voltage-Controlled Audio Oscillator.

✓See List of Advertisers on page 402

"The computer can also convert analog signals into audio tones."

Enter 128 in response to the program prompt to stop the permanent-magnet motor, then Enter and Run300:

```
300 OUT 127,137:OUT 124,128: FOR N = 128 TO 0
  STEP -1:OUT 124, N:NEXT N
310 FOR N = 0 TO 255
320 OUT 124, N
330 FOR M = 1 TO 10:NEXT M
340 NEXT N
350 FOR N = 255 TO 0 STEP -1
360 OUT 124, N
370 FOR M = 1 TO 10:NEXT M
380 NEXT N
390 GOTO 310
```

The permanent-magnet motor gradually increases its speed in one direction, then slows to a stop, reverses direction, and gradually increases to maximum speed in the opposite direction. You may want to increase or decrease the delay loops in line 330 and 370.

When the motor is stopped, depress Break, enter and Run400:

```
400 OUT 127,137:OUT 124,128
410 FOR N = 1 TO 500:NEXT N
420 OUT 124,105
430 FOR N = 1 TO 500:NEXT N
440 OUT 124,70
450 FOR N = 1 TO 500:NEXT N
460 OUT 124,35
470 FOR N = 1 TO 500:NEXT N
480 OUT 124,0
490 FOR N = 1 TO 500:NEXT N
500 OUT 124,255
510 FOR N = 1 TO 500:NEXT N
520 OUT 124,220
530 FOR N = 1 TO 500:NEXT N
540 OUT 124,185
550 FOR N = 1 TO 500:NEXT N
560 OUT 124,150
570 GOTO 410
```

The permanent-magnet motor suddenly changes speed and direction at one-second intervals. Hold the motor in your hand. Notice that large speed changes or reversals cause the motor to torque (twist) opposite to the shaft rotation. These speed and direction changes are sudden but not instantaneous.

The computer can also convert analog signals into audio tones (see Fig. 7). A number of schemes have been devised to produce musical or audio tones from the TRS-80, using the system as a very expensive square-wave generator. The major disadvantage with internally generated audio is it ties up the whole system which could be doing other things.

This is an example when the hardware/software trade-off—always an engineering consideration—favors hardware.

For our application, the integrated circuit for the 555 timer is configured as a voltage-to-frequency converter. Varying the input to R55 (24K) to between zero and 10 volts direct current results in the charging voltage—appearing at the junction of R55 (24K), R56 (10K), and R57 (10K)—varying between 10.6 volts and 13.5 volts dc. To allow the 555 to function as an oscillator, C_t must charge to 2/3 of Vcc (10 volts). This requirement is met with the value given for R55 (24K). Lowering the value of R55 to 20K causes the 555 to stop oscillating as the input to R55 approaches zero volts. This permits silencing the oscillator if desired.

Because of the capacitor involved, linear

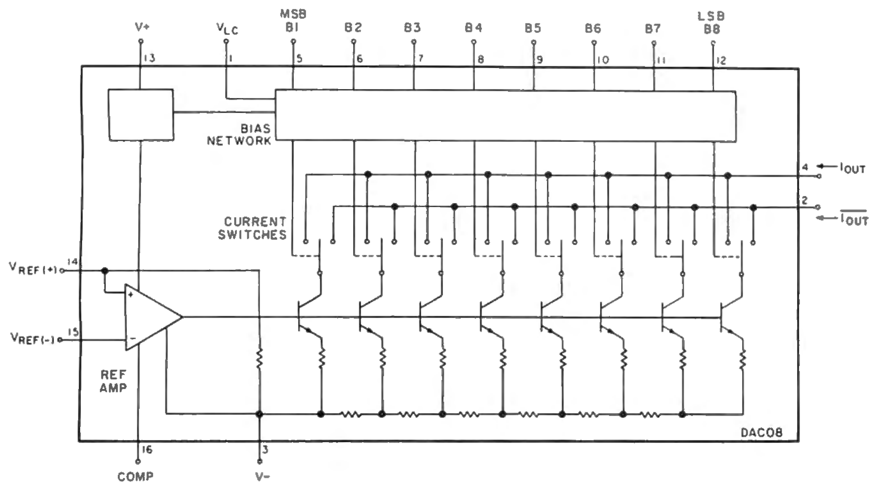


Fig. 8a. DAC-08 Block Diagram.

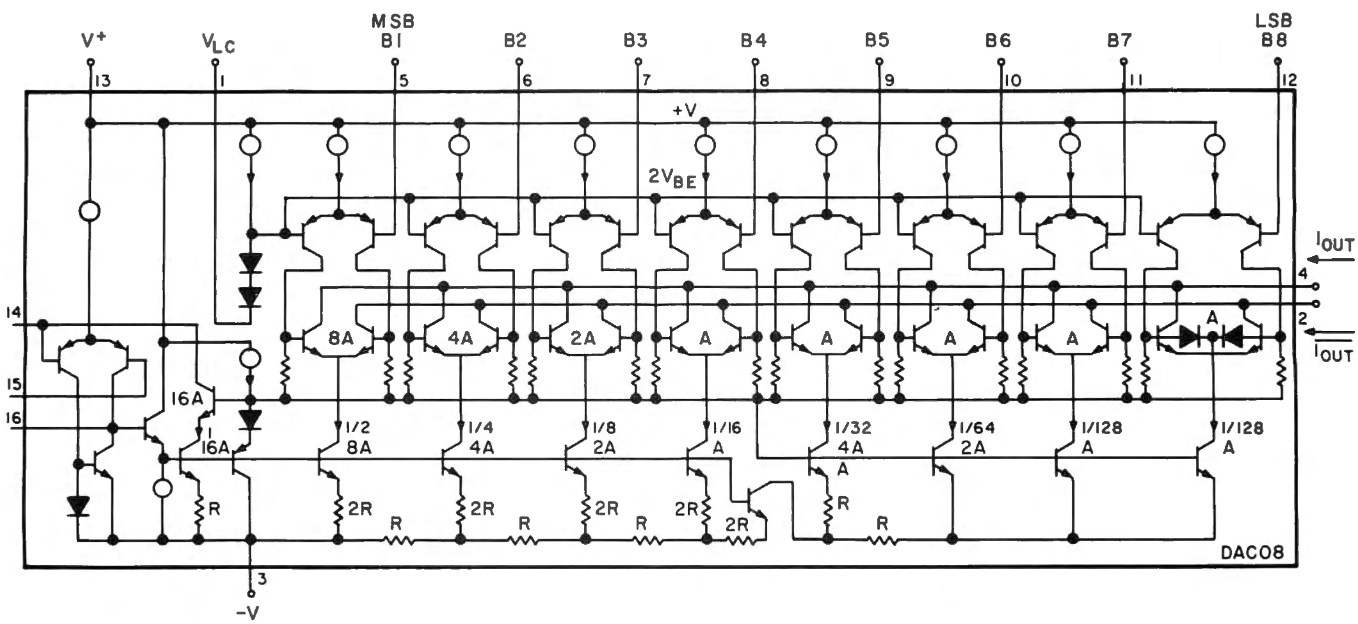


Fig. 8b. DAC-08 Equivalent Circuit.

(straight-line) variations in voltage result in logarithmic changes in frequency. Happily the human ear and brain respond logarithmically to sound, and the result is surprisingly pleasant.

Next Run100 the program previously loaded. The light and motor may remain connected if desired.

You should hear a realistic siren sound from the loudspeaker.

Run200. For each number entered, a unique tone is heard.

Next, enter and Run600:

```
600 OUT 127,137
610 A = RND(256) - 1
620 OUT 124, A
630 GOTO 610
```

You should hear a jangled cacophony of tones. If the motor is connected, it will go bananas trying to follow the signal. The light will flash in disco style.

Enter and Run700:

```
700 OUT 127,137:CLS
710 A = RND(256) - 1
720 PRINT A;
730 OUT 124,A
740 FOR N = 1 TO 50:NEXT N
750 GOTO 710
```

Now the tones are distinct and separated. Some of you may be reminded of the music of J. S. Bach. The duration of each note is equal.

To vary the duration of each note enter and Run800:

```
800 OUT 127,137:CLS
810 A = RND(256) - 1
820 PRINT A;
830 OUT 124,A
840 B = RND(100) + 10
850 FOR N = 1 TO B:NEXT N
860 GOTO 810
```

The tones heard are reminiscent of bagpipes. You may want to experiment with additional program variations. For instance, a third random element may be introduced to control the number of times a note is repeated.

The Real World is Ours

We are now communicating with the physical world using our TRS-80 as sophisticated controllers. We have demonstrated by simple experiments the basic principles of controlling the real world, affecting the senses of sight, sound, and touch.

Control is still through the keyboard requiring human intervention to change program effects. Next month we will explore the methods and applications of analog-to-digital conversion as a sophisticated response to the physical world and decision-making without human intervention. ■

✓See List of Advertisers on page 402

An assembly time-cost analysis program with heart.

The Pacemaker

David Tinis
National Controls Inc.
4500 John Young Pkwy.
Orlando, FL 32804

A modern piece of electronic equipment consists of a multitude of parts. In fact, a printed circuit board assembly may include hundreds of resistors, capacitors, transistors, diodes and integrated circuits. Have you ever wondered how a finished assembly was created from a pile of parts?

It should be immediately obvious a worker does not sit down with all the components to build the finished product. Instead, assembly lines are created in which the product has parts added to it until a completed unit is produced.

Pacing a Line

This brings us to the concept of a "paced" assembly line. Imagine an assembly line consisting of four workstations whose purpose is to build a printed circuit assembly. A bare board enters the first station and has parts mounted on it. Then it's passed to the next three for additional parts, until the final assembly emerges at the last station. All seems to be working well. But suppose it takes the first station five minutes to complete the required task, the second three minutes, the third eight and the fourth four. Some workers will be sitting for long stretches with nothing to do while others must work feverishly to keep up.

Looking at the total time of our hypothetical four-station line, we find it should take 20 minutes of assembly time to build a single unit. But is that true?

Even though station two can finish its job in three minutes, it must wait two more minutes for the next piece from station one. Station three in turn requires eight minutes for its task. A backlog starts to build as the two faster stations pass on more boards than it can handle. Station four, like station two, is limited by its preceding station. It can finish a job in four minutes, but must wait four more minutes for station three to finish.

This results in an assembly time of 26 minutes instead of 20 minutes and an efficiency of only 78 percent. The solution is to redistribute the work among the workstations so each will take equal time.

An actual production line will have these problems multiplied manyfold. How do you set up an efficient assembly line from scratch?

You could go on past experience or a "gut feeling" and later adjust the line as required. But this wastes time and manpower resources. Good practice dictates establishing standard times for every operation. Then, by adding all the standard times, a total time for the finished assembly can be established.

Working backward, you can divide the total time by the number of work stations to arrive at a time per work station. Knowing this, you can return to the per component standard times to determine what combinations of parts should be assigned to each work station. This step is usually the most difficult one, involving the manipulation of lots of numbers, and can turn into a major job when applied to an assembly with a large number of parts.

The Computer Steps In

Here is where an assembly time/cost analysis program is useful. Such a program allows the engineer to easily rearrange work station compositions until a balanced condition is achieved.

Assembly Time/Cost Analysis allows a person to create a paced assembly line with a minimum of aggravation. Through a master menu, the user can create new files or load saved files containing assembly/work station data. Parts can be changed, added, deleted or moved from one station to another. Station composition can be viewed either for a single station or all stations sequentially. Each station may contain up to 20 discrete components, and the number of stations is limited only by your computer's memory size. Finally, cost per work station can be automatically computed, a necessary detail for a profit making business.

Glossary

The program is essentially self-prompting; however, a glossary of terms used as they are encountered in the program is included for maximum understanding.

• Performance Factor-This is an adjustment number used for uncontrollable variables—fatigue, housekeeping, personal needs, etc. It is generally between one and two although it could be higher for really inefficient operations.

To illustrate, it may take 0.5 minutes to bend the leads on a resistor, insert it in a printed circuit board, solder it and clip the leads. Based on this standard time you might expect 120 resistors an hour to be handled. But try doing it yourself sometime; before long you get tired and bored. You stretch, go get a drink, go to the bathroom, talk to a fellow worker, etc. At the end of an hour you may have done only 100 resistors. To allow for this, an adjustment is made to the standard time.

• Raw Minutes-This is the ideal time to perform the required operations for a single part. When displayed in the station breakdown, this will be multiplied by the quantity of the particular part.

• Factored Minutes-This is the adjusted allowable time for installation of a line item. This number represents Raw Minutes times Quantity times Performance Factor.

• Factored Hours-The adjusted allowable time for a line item converted to hours or fractions of hours.

• Direct Labor Hourly Rate-The average hourly wage paid to a direct labor

"Have you ever wondered how a finished assembly was created from a pile of parts?"

worker—those individuals who actually build the product.

• Overhead Percentage-A factor used to add in the cost of doing business other than raw materials and direct labor. This includes rental of facilities, maintenance, management salaries, office workers, utilities, etc. It even extends to the cost of the company's softball team, company picnic and greens fees for the golf league. Anything that costs money adds to the expense of operation and is included in overhead.

Initial entry into the program results in a two choice mini-menu. A new file can be created or an old one recalled. Entering a new file entails inserting file name, performance factor and number of assembly line stations information. If you are unsure about the actual number of stations needed, it causes no harm to specify some extras and then leave them blank. The array scanner used by various parts of the program detects these empty stations and notifies you of their availability. When reading in an old file you don't get initialization prompts since the required data is stored as part of the file.

Both selections eventually get you to the Main Menu. This gives you 10 commands plus an exit from the program:

• Totals By Station-Use this command to view a work station component breakdown. Either a single station or all stations in sequence can be viewed. Output can be either video or printer.

• Factored Hours By Station-This is a summary of all stations, used as a quick check to see how well the line balancing is proceeding. Output can be to either video or printer.

• Reassign Parts-Parts can be moved from one station to another. Requests for movement to or from nonexistent stations or to full stations are trapped and rejected.

• Enter Additional Parts-This is a two-part routine. Parts can be added to an existing station if it is not full or an empty station can be assigned a parts group. The parts entry portion requires a yes/no after each part is entered. The station entry portion uses the "easy entry" mode first used when creating a new file.

• Delete Parts-As the command states, you tell it what item at what station to delete and it does. First, however, it displays the item and part numbers and asks for confirmation to delete it.

• Assembly Costs-The Direct Labor Hourly Rate and Overhead Percentage are entered and a cost per station and total cost are calculated. Output can be video or printer.

• Save Current File-Saves the work station file on disk.

• Restart-Provides a clean way to finish operations on one file and start on a different one.

• Change Performance Factor-Allows instant changes to be made to the factored calculations.

• Edit Current File-Single line items can be changed. A small menu allows selection of specifications to change.

Line List

I tried to write the program in a modular fashion so it could be modified or expanded as the need arose. With that in mind, I've provided a line number list:

• Lines 10-90: Initialization. The user has the option of opening a new file or retrieving an old one.

• Lines 100-110: Main menu.

• Lines 990-1999: Work station composition breakdown by single station or all stations.

• Lines 2000-2999: Factored hours per work station and total for all stations.

• Lines 3000-3999: Parts transfer from one station to another.

• Lines 4000-4999: Addition of parts to a station.

• Lines 5000-5999: Deletion of parts from a station.

• Lines 6000-6999: Cost analysis by work station and total for all stations.

• Lines 7000-7999: Menu driven editor.

• Lines 8000-8999: Saves the current file to disk.

• Lines 9000-9999: Loads a requested file from disk.

• Lines 10000-14040: Various subroutines. ■

Program Listing

```
10 CLS:CLEAR1500:P$="##.####!":C$="$$###.##"
15 PRINT CHR$(23):PRINT@128,"ASSEMBLY TIME/COST ANALYSIS":FORX=1
TO300:NEXT
16 PRINT@262,"(1) CREATE NEW FILE"
17 PRINT@326,"(2) LOAD EXISTING FILE"
18 PRINT@390,"";:INPUT A
19 IF A<1 OR A>2 THEN 16 ELSE IF A=2 THEN 9000
20 INPUT "ASSEMBLY (FILE) NAME ";F$
25 INPUT"PERFORMANCE FACTOR ";PF
30 INPUT "NUMBER OF WORKSTATIONS ";NS
40 GOSUB 14000
50 JO=0:FORX=1TONS
55 CLS:S=1:PRINT"STATION NUMBER :";X:PRINT"ENTRY IN ITEM FIELD:
'@' ENDS STATION ENTRY, '/' ENDS ALL ENTRY"
60 PRINT@25,"PARTS THIS STATION :";S;:PRINT@192,"ITEM NUMBER :
                 ";:PRINT@320,"PART NUMBER :
   ";:PRINT@448,"QUANTITY :       ";:PRINT@576,"RAW MINUTES FOR S
INGLE ITEM :         ";
70 PRINT@205,"";:INPUT X$:IFX$="@"THEN90ELSEIFX$="/"THEN100ELSE
IN(X,S)=VAL(X$):PRINT@333,"";:INPUTPN$(X,S):PRINT@458,"";:INPUT
QT(X,S):PRINT@606,"";:INPUTRM(X,S)
75 PS(X)=S
80 RM(X,S)=RM(X,S)*QT(X,S):FM(X,S)=RM(X,S)*PF:FH#(X,S)=FM(X,S)/6
0:S=S+1:IFS>20THENCLS:PRINT"NO FURTHER ITEMS ALLOWED FOR THIS ST
ATION";:FORA=1TO500:NEXTA:CLS:GOTO90ELSEGOTO60
90 IFJO=1THEN4170ELSENEXTX
100 CLS:PRINTTAB(15);"ENTER OPTION DESIRED":PRINTTAB(15);"(1) TO
TALS BY STATION":PRINTTAB(15);"(2) FACTORED HOURS BY STATION":PR
INTTAB(15);"(3) REASSIGN PARTS":PRINTTAB(15);"(4) ENTER ADDITION
AL PARTS":PRINTTAB(15);"(5) DELETE PARTS"
105 PRINTTAB(15);"(6) ASSEMBLY COSTS":PRINTTAB(15);"(7) SAVE CUR
RENT FILE":PRINTTAB(15);"(8) RESTART":PRINTTAB(15);"(9) CHANGE P
ERFORMANCE FACTOR":PRINTTAB(15);"(10) EDIT CURRENT FILE":PRINTTA
B(15);"(11) END JOB":PRINTTAB(15);"";:INPUT O
110 IF O<1 OR O>11 THEN 100 ELSE ON O GOTO 990,2000,3000,4000,50
00,6000,8000,10,13600,7000,10000
990 CLS:INPUT"(1) ALL STATIONS OR (2) SINGLE STATION ";A:IF A<1
OR A>2 THEN 990
1000 OS=0:INPUT"(S) SCREEN OR (P) PRINTER ";A$:IFA$="P"THENOS=1:
GOSUB13000
1001 IF A=2 AND OS=0 INPUT"WHICH STATION ";X:GOSUB1010:GOTO1999
1002 IF A=2 AND OS=1 INPUT"WHICH STATION ";X:GOSUB1070:GOTO1999
1003 IF A=1 AND OS=0 GOTO 1005
1004 IF A=1 AND OS=1 GOTO 1060
1005 FOR X=1TONS:CLS:GOSUB1010:NEXTX:GOTO1999
1010 S=1:TF#(X)=0:CLS
1020 PRINT"STATION NUMBER:";X:PRINT"ITEM #      PART #      QUAN.
  RAW MIN.  FACT. MIN.  FACT. HRS.":PRINT
1030 IFS>20THEN1050ELSEIFIN(X,S)=0 THEN1050ELSEPRINTTAB(2);IN(X,
S);:PRINTTAB(9);PN$(X,S);:PRINTTAB(23);QT(X,S);:PRINTTAB(31);"";
:PRINTUSING P$;RM(X,S);:PRINTTAB(42);"";:PRINTUSING P$;FM(X,S);:
PRINTTAB(53);"";:FH#(X,S)=FM(X,S)/60:PRINTUSING P$;FH#(X,S)
1035 TF#(X)=TF#(X)+FH#(X,S)
```

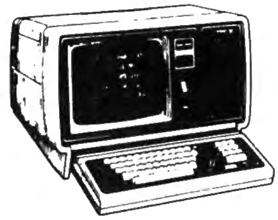
Program continues

Program continued

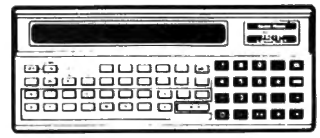
```
1040 IFS=10THENPRINT:INPUT"PRESS <ENTER> FOR MORE";A$:CLS:S=S+1:
GOTO1020:ELSES=S+1:GOTO1030
1050 PRINT:PRINTTAB(16);"TOTAL FACTORED HOURS FOR STATION";X;":
";:PRINTUSING P$;TF#(X):PRINT:INPUT"PRESS <ENTER> TO CONTINUE";A
$:RETURN
1060 LPRINT CHR$(14);:LPRINTTAB(10);"ASSEMBLY:";F$:LPRINT:LPRINT
TAB(18);"PERFORMANCE FACTOR:";PF:LPRINT CHR$(13):FOR X=1TONS:GOS
UB1070:NEXTX:GOTO1999
1080 LPRINT"ITEM #      PART #      QUAN.    RAW MIN.   FACT. MIN.
FACT. HRS.":LPRINT CHR$(13)
1090 IFS>20THEN1110ELSEIFIN(X,S)=0THEN1110ELSELPRINTTAB(2);IN(X,
S);:LPRINTTAB(9);PN$(X,S);:LPRINTTAB(23);QT(X,S);:LPRINTTAB(31);
"";:LPRINTUSING P$;RM(X,S);:LPRINTTAB(42);"";:LPRINTUSING P$;FM(
X,S);:LPRINTTAB(53);"";:FH#(X,S)=FM(X,S)/60:LPRINTUSING P$;FH#(X
,S
1100 TF#(X)=TF#(X)+FH#(X,S):S=S+1:GOTO1090
1110 LPRINT CHR$(13):LPRINTTAB(16);"TOTAL FACTORED HOURS FOR STA
TION";X;": ";:LPRINTUSING P$;TF#(X):FORZ=1TO2:LPRINT CHR$(13):NE
XTZ:RETURN
1999 GOTO100
2000 OS=0:AS#=0:CLS:INPUT"(S) SCREEN OR (P) PRINTER ";A$:IFA$="P
"THEN OS=1:GOSUB13000
2010 HD$="STATION NUMBER            FACTORED HOURS":IF OS=1 THEN L
PRINT"FACTORED HOURS BY STATION":LPRINT:LPRINT HD$ ELSE CLS:PRIN
T HD$
2020 FOR X=1TONS
2030 S=1:TF#(X)=0
2040 IFS>20THEN2050ELSEIF IN(X,S)=0THEN 2050 ELSE TF#(X)=TF#(X)+
FH#(X,S):S=S+1:GOTO2040
2050 AS#=AS#+TF#(X):IF OS=1 THEN LPRINTTAB(6);X;:LPRINTTAB(27);"
";:LPRINTUSING P$;TF#(X) ELSE PRINTTAB(6);X;:PRINTTAB(27);"";:PR
INTUSING P$;TF#(X)
2055 IF OS=0 AND X/10=INT(X/10) THEN INPUT"PRESS <ENTER> TO CONT
INUE";A$:CLS:PRINT HD$
2060 NEXT X
2070 H2$="TOTAL FACTORED HOURS FOR ALL STATIONS: ":IF OS=1 THEN
LPRINT H2$;:LPRINTUSING P$;AS#:LPRINT:LPRINT: ELSE PRINT H2$;:PR
INTUSING P$;AS#
2998 INPUT"PRESS <ENTER> TO CONTINUE";A$
2999 GOTO100
3000 IFNS=1THENPRINT"ONLY ONE STATION. UNABLE TO TRANSFER PARTS.
":GOTO3998ELSECLS:INPUT"REMOVE PART FROM WHICH STATION ";RS:IFRS
<1 OR RS>NS THEN3000
3010 PRINT@128,"";:INPUT"ADD PART TO WHICH STATION ";DS:IFDS<1 O
R DS>NS OR DS=RS THEN3010
3020 INPUT"ENTER ITEM NUMBER ";PA
3030 S=1
3040 IF IN(RS,S)=PA THEN3060ELSEIF IN(RS,S)=0 OR S=20 THEN3050EL
SE S=S+1:GOTO3040
3050 PRINT"ITEM ";PA;" NOT FOUND.":GOTO3998
3060 PRINT"ITEM ";IN(RS,S);" PART NUMBER ";PN$(RS,S)
3070 INPUT"IS THIS THE ITEM TO BE MOVED (Y/N) ";A$:IFA$<>"Y"THEN
3998
3080 D=1
3090 IFD>20PRINT"DESTINATION STATION FULL.":GOTO3998
3100 IF IN(DS,D)<>0THEN D=D+1:GOTO3090
3110 IN(DS,D)=IN(RS,S):PN$(DS,D)=PN$(RS,S):QT(DS,D)=QT(RS,S):RM(
3113 PS(DS)=D:PS(RS)=PS(RS)-1
DS,D)=RM(RS,S):FM(DS,D)=FM(RS,S):FH#(DS,D)=FH#(RS,S)
3115 GOSUB3120:GOTO3998
3120 GOSUB13700:IFS=20THEN3150
3130 IN(RS,S)=IN(RS,S+1):PN$(RS,S)=PN$(RS,S+1):QT(RS,S)=QT(RS,S+
1):RM(RS,S)=RM(RS,S+1):FM(RS,S)=FM(RS,S+1):FH#(RS,S)=FH#(RS,S+1)
3140 S=S+1:IFS=20THEN3150ELSE3130
3150 IN(RS,S)=0:PN$(RS,S)=" ":QT(RS,S)=0:RM(RS,S)=0:FM(RS,S)=0:F
H#(RS,S)=0:RETURN
3998 GOSUB11000
3999 GOTO100
4000 CLS:INPUT"(1) ADD PARTS OR (2) ADD STATION";A
4001 IF A<1 OR A>2 THEN4000ELSEIF A=2 THEN4100
4002 INPUT"ADD TO WHICH STATION ";DS
4005 IF DS<1 OR DS>NS THEN 4000
4010 D=1
4020 IF IN(DS,D)=0 THEN4030ELSEIF D=20 THENPRINT"STATION FULL":G
OTO4060:ELSE D=D+1:GOTO4020
4030 CLS:PRINT"STATION NUMBER :";DS;:PRINT@25,"PARTS THIS STATIO
N :";D:PRINT:INPUT"ITEM NUMBER : ";IN(DS,D):PRINT:INPUT"PART NUM
BER : ";PN$(DS,D):PRINT:INPUT"QUANTITY : ";QT(DS,D):PRINT:INPUT"
RAW MINUTES : ";RM(DS,D)
4040 RM(DS,D)=RM(DS,D)*QT(DS,D):FM(DS,D)=RM(DS,D)*PF:FH#(DS,D)=F
M(DS,D)/60
4045 PS(DS)=D
4050 PRINT:D=D+1:IFD>20THENPRINT"THIS STATION NOW FULL":GOTO4060
:ELSE INPUT"ADD ANOTHER PART TO THIS STATION (Y/N)";A$:IFA$="Y"T
HEN CLS:GOTO4030
4060 INPUT"ADD PARTS TO ANOTHER STATION (Y/N)";A$:IFA$="Y"THEN40
00ELSE4999
4100 ZS=0:GOSUB13500:PRINT"AVAILABLE EMPTY STATIONS:";
4110 FORX=1TONS:IFPS(X)=0PRINT"-";X;:ZS=1
4120 NEXTX
4130 IFZS=0THENPRINT" NONE";:PRINT:GOSUB11000:GOTO4999:ELSEPRINT
4150 INPUT"STATION NUMBER TO BE ADDED";A
4160 IF A<1 OR A>NSTHENPRINT"THAT STATION IS UNDEFINED":GOTO4150
:ELSEIFPS(A)<>0THENPRINT"THAT STATION IS ALREADY IN USE":GOTO415
0ELSEX=A:JO=1:GOTO55
4170 CLS:INPUT"ANYTHING ELSE TO ADD (Y/N)";A$:IFA$="Y"THEN4000EL
SE100
4999 GOTO100
5000 CLS:INPUT"DELETE FROM WHICH STATION ";RS
5005 IFRS<1 OR RS>NS THEN 5000
5010 PRINT:INPUT"DELETE ITEM NUMBER ";PA
5020 S=1
5030 IF IN(RS,S)=PA THEN5050ELSEIF IN(RS,S)=0 OR S=20 THEN5040EL
SE S=S+1:GOTO5030
5040 PRINT"ITEM ";PA;" NOT FOUND.":GOTO5080
5050 PRINT"ITEM ";IN(RS,S);"   PART NUMBER ";PN$(RS,S)
5060 INPUT"IS THIS THE ITEM TO BE DELETED (Y/N)";A$:IFA$<>"Y"THE
N5080
5070 GOSUB 3120
5075 PS(RS)=PS(RS)-1
5080 INPUT"DELETE ANOTHER PART FROM THIS STATION (Y/N)";A$:IFA$=
"Y"THENCLS:GOTO5010
5998 INPUT"DELETE A PART FROM ANOTHER STATION (Y/N)";A$:IFA$="Y"
THEN5000
```

Program continues

Here's a REAL PLUS

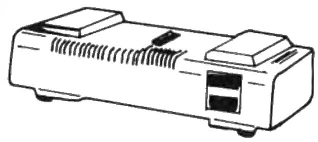


Model II 64K $3300

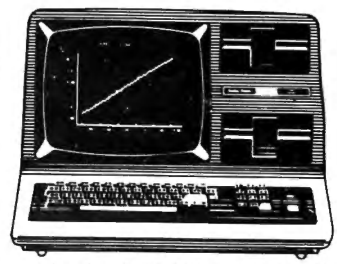


Pocket Computer $189
P. C. Printer $130

Color Computer 4K $310
w/16K Ext. Basic $459



32K Exp. Int. $334



Model III 16K $839
2 DR + RS232C $2100

LOOK Here are just a few of our fine offers . . . call toll-free for full information.

COMPUTERS

Model II 64K	$3300
Model III 4K LEV I	599
MODEL III 16K	839
MODEL III 32K	945.50
*MODEL III 32K	881.50
MODEL III 48K	1052
*MODEL III 48K	924
Model III 48K 2 Disc & RS232 c	2100
Color Computer 4K	310
Color Computer 16K	416.50
*Color Computer 16K	352.50
Color Computer 16K w/extended basic	459
Pocket Computer	189
VIDEOTEX	310
APPLE 48K only	1279
ATARI 800 32K	789

*Computer Plus New Equipment.
180 Day Extended Warranty

PERIPHERALS

Expansion Interface OK	$249
Expansion Interface 16K	355.50
*Expansion Interface 16K	291.50
Expansion Interface 32K	462
*Expansion Interface 32K	334
16K RAM N.E.C. 200 N.S. chips	25

MODEMS

Lynx Direct Connect MI/MIII Auto Ans./Dial	269
Telephone Interface II	169
R.S. Modem I D.C.	130
Atari 830 Modem	159
Atari 850 Interface	183

PRINTERS

Daisy Wheel II	1695
Line Printer VI	999
Epson MX80	499
Epson MX80 FT	599
Epson MX100	799
Line Printer VII	315
Line Printer V	1610
Microline 80	345
Microline 82A	515
Microline 83A	799
Pocket Computer Printer	130

DISK DRIVES

R.S. Model III 1ST-Drive	712
TEAC 40 Track MI	329
R.S. 1 Drive Exp MII	999
R.S. 2 Drive Exp MII	1518
R.S. 3 Drive Exp MII	2040
Seimans 8" MII	799
Atari 810 Disk	499

SOFTWARE

R.S. Software 10% off list	
Atari Software 10% off list	
Newdos + (40) track MI	88
Newdos 80 MIII	149
ST80III	149

ETC.

Verbatum 5" Double Density	32
Verbatum 8" Data Life	49.95
Ctr-80A recorder	52
C. C Joysticks	22

PLUS real back-up warrantees —

Factory warrantees on Apple and Atari equipment. Other equipment carries manufacturer's warranty or Computer Plus 180 day extended warranty. Combined warrantees carry Computer Plus 180 day warranty or original manufacturer's warranty.

DEALER INQUIRIES ARE INVITED

Prices subject to change without notice.
Not responsible for typographical errors.

TRS-80 is a registered trademark of Tandy Corp.

call TOLL FREE 1-800-343-8124

computer plus

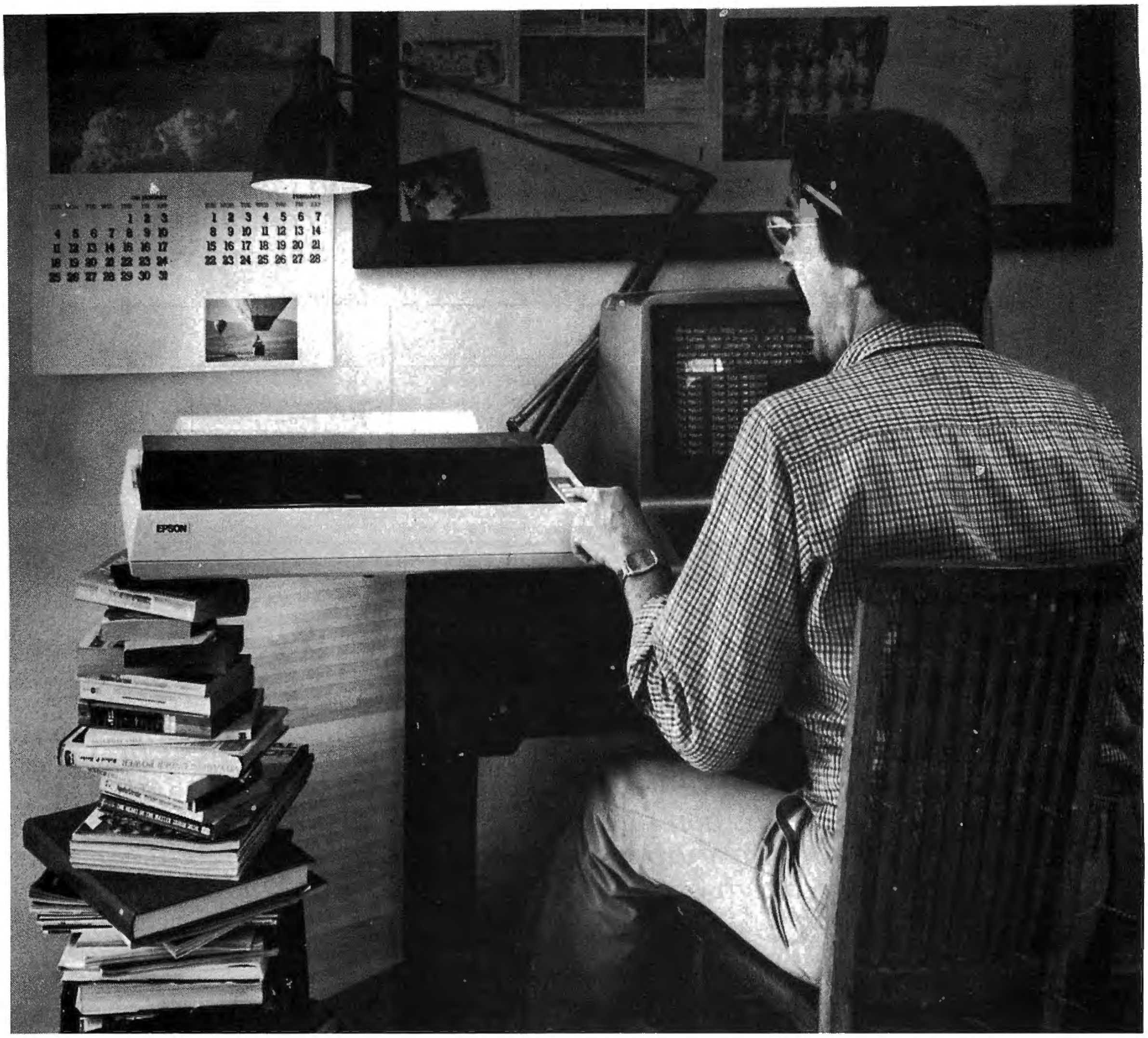
Dept. D
245A Great Road
Littleton, MA 01460
617-486-3193

Write for your free catalog

✓130

Program continued

```
5999 GOTO 100
6000 CLS
6001 INPUT"ENTER DIRECT LABOR HOURLY RATE";DL
6010 INPUT"ENTER OVERHEAD PERCENTAGE";OH:OH=OH/100
6015 OS=0
6020 INPUT"(S) SCREEN OR (P) PRINTER";A$:IFA$="P"THENOS=1:GOSUB1
3000
6030 BC$="STATION--FACTORED HOURS--DIRECT LABOR--OVERHEAD--BURDE
NED COST"
6035 TD=0:TA=0:TB=0
6040 IFOS=1THENLPRINT;"DIRECT LABOR HOURLY RATE:";:LPRINTUSING C
$;DL:LPRINT"OVERHEAD:";OH*100;"%":LPRINT:LPRINTBC$ELSECLS:PRINTB
C$
6050 FORX=1TONS
6060 S=1:TF#(X)=0
6070 IFS>20THEN6080ELSEIF IN(X,S)=0THEN6080ELSE TF#(X)=TF#(X)+FH
#(X,S):S=S+1:GOTO6070
6080 DL(X)=TF#(X)*DL:OH(X)=DL(X)*OH:BL(X)=DL(X)+OH(X):TD=TD+DL(X
):TA=TA+OH(X):TB=TB+BL(X)
6090 IF OS=1 THEN LPRINTTAB(3);X;:LPRINTTAB(12);"";:LPRINTUSING
P$;TF#(X);:LPRINTTAB(26);"";:LPRINTUSING C$;DL(X);:LPRINTTAB(38)
;"";:LPRINTUSING C$;OH(X);:LPRINTTAB(50);"";:LPRINTUSING C$;BL(X
):GOTO6120
6100 PRINTTAB(3);X;:PRINTTAB(12);"";:PRINTUSING P$;TF#(X);:PRINT
TAB(26);"";:PRINTUSING C$;DL(X);:PRINTTAB(38);"";:PRINTUSING C$;
OH(X);:PRINTTAB(50);"";:PRINTUSING C$;BL(X)
6110 IF X/10=INT(X/10) THEN GOSUB 11000:CLS:PRINT BC$
6120 NEXT X
6130 IF OS=1 THEN LPRINT"":LPRINTTAB(9);"ASSEMBLY COST";:LPRINTT
AB(26);"";:LPRINTUSING C$;TD;:LPRINTTAB(38);"";:LPRINTUSING C$;T
A;:LPRINTTAB(50);"";:LPRINTUSING C$;TB:LPRINT:LPRINT:GOTO6998
6140 PRINT:PRINTTAB(9);"ASSEMBLY COST";:PRINTTAB(26);"";:PRINTUS
ING C$;TD;:PRINTTAB(38);"";:PRINTUSING C$;TA;:PRINTTAB(50);"";:P
RINTUSING C$;TB
6998 GOSUB11000
6999 GOTO 100
7000 CLS:INPUT"EDIT WHICH STATION";X:IFX<1 OR X>NS THEN7000
7010 PRINT@64,"EDIT WHICH ITEM";:INPUT A:Y=1
7015 IF IN(X,Y)=A THEN7030 ELSEIF Y=20 THEN7020ELSE Y=Y+1:GOTO70
15
7020 PRINT"ITEM";A;"NOT FOUND. CONTINUE TO EDIT (Y/N)";:INPUTA$:
IFA$="Y"THEN7000ELSE7999
7030 CLS:PRINT@0,"STATION NUMBER:";X;:PRINT@128,"ITEM NUMBER:";I
N(X,Y);:PRINT@192,"PART NUMBER:";PN$(X,Y);:PRINT@256,"QUANTITY:"
;QT(X,Y);:PRINT@320,"RAW MINUTES FOR A SINGLE ITEM:";RM(X,Y)/QT(
X,Y);
7040 PRINT@448,"EDIT (1) ITEM NUMBER (2) PART NUMBER (3) QUANTIT
Y
     (4) RAW MINUTES (5) QUIT";:INPUT A:IFA<1 OR A>5 THEN 7040
7045 PRINT@576,"":ON A GOTO 7050,7060,7070,7080,7090
7050 PRINT@640,"NEW ITEM NUMBER";:INPUT IN(X,Y):GOTO7030
7060 PRINT@640,"NEW PART NUMBER";:INPUT PN$(X,Y):GOTO7030
7070 PRINT@640,"NEW QUANTITY";:INPUT A:RM(X,Y)=RM(X,Y)/QT(X,Y):Q
T(X,Y)=A:RM(X,Y)=RM(X,Y)*QT(X,Y):GOTO7030
7080 PRINT@640,"NEW RAW MINUTES";:INPUT RM(X,Y):RM(X,Y)=RM(X,Y)*
QT(X,Y):GOTO7030
7090 FM(X,Y)=RM(X,Y)*PF:FH#(X,Y)=FM(X,Y)/60
7999 GOTO 100
8000 ONERRORGOTO12000:CLS:INPUT"ENTER NAME FILE IS TO BE SAVED U
NDER ";F$
8005 GOSUB 13500
8010 INPUT"INSERT DATA DISK IN DRIVE AND PRESS <ENTER>";A$
8020 OPEN"O",1,F$:PRINT#1,NS,PF
8025 FORX=1TONS:PRINT#1,PS(X):NEXTX
8030 FORX=1TONS:FORY=1TOPS(X)
8040 PRINT#1,IN(X,Y),QT(X,Y),RM(X,Y),FM(X,Y),FH#(X,Y)
8050 NEXTY:NEXTX
8060 FORX=1TONS:FORY=1TOPS(X)
8070 PRINT#1,PN$(X,Y)
8080 NEXTY:NEXTX
8090 CLOSE1
8998 GOSUB 11000
8999 GOTO 100
9000 ONERRORGOTO12000:CLS:INPUT"ENTER NAME OF FILE TO BE RETRIEV
ED ";F$
9010 INPUT"INSERT DATA DISK IN DRIVE AND PRESS <ENTER>";A$
9020 OPEN"I",1,F$:INPUT#1,NS,PF:GOSUB14000
9022 PRINT"PERFORMANCE FACTOR IS ";PF
9025 FORX=1TONS:INPUT#1,PS(X):NEXTX
9030 FORX=1TONS:FORY=1TOPS(X)
9040 INPUT#1,IN(X,Y),QT(X,Y),RM(X,Y),FM(X,Y),FH#(X,Y)
9050 NEXTY:NEXTX
9060 FORX=1TONS:FORY=1TOPS(X)
9070 INPUT#1,PN$(X,Y)
9080 NEXTY:NEXTX
9090 CLOSE1
9998 GOSUB11000
9999 GOTO 100
10000 CLS:END
11000 PRINT"PRESS <ENTER> TO CONTINUE"
11010 A$=INKEY$:IFA$=""THEN11010ELSERETURN
12000 CLOSE:PRINT ERROR:GOSUB11000:RESUME100
13000 P=PEEK(14312):IF P>128 THEN CLS:INPUT"PRINTER NOT READY. A
BORT (Y/N)";A$ ELSE GOTO13020
13010 IF A$="Y"THEN OS=0:GOTO13020:ELSE INPUT"READY PRINTER AND
PRESS <ENTER>";A$:GOTO13000
13020 RETURN
13500 PRINT"+ + SCANNING ARRAYS + +":FORX=1TONS:S=1
13510 IFS>20THEN13520ELSEIF IN(X,S)=0THEN13520ELSES=S+1:GOTO1351
0
13520 PS(X)=S-1:NEXTX:RETURN
13600 CLS:PRINT"PERFORMANCE FACTOR IS ";PF:INPUT"NEW PERFORMANCE
 FACTOR ";PF
13610 GOSUB 13700
13620 FOR X=1TONS:FOR Y=1TOPS(X)
13630 FM(X,Y)=RM(X,Y)*PF:FH#(X,Y)=FM(X,Y)/60
13640 NEXTY:NEXTX:GOSUB11000:GOTO100
13700 PRINT"+ + ADJUSTING ARRAYS + +":RETURN
14000 CLS:PRINT CHR$(23):PRINT@464,"INITIALIZING"
14010 DIM IN(NS,20),PN$(NS,20),QT(NS,20),RM(NS,20),FM(NS,20),FH#
(NS,20),TF#(NS),DL(NS),OH(NS),BL(NS),PS(NS)
14020 FORX=1TONS:FORY=1TO20
14030 IN(X,Y)=0:PN$(X,Y)=" ":QT(X,Y)=0:RM(X,Y)=0:FM(X,Y)=0:FH#(X
,Y)=0
14040 NEXTY:NEXTX:RETURN
```



The MX-100. Not just better. Bigger.

Epson.

Our MX-80 was a pretty tough act to follow. I mean, how do you top the best-selling printer in the world?

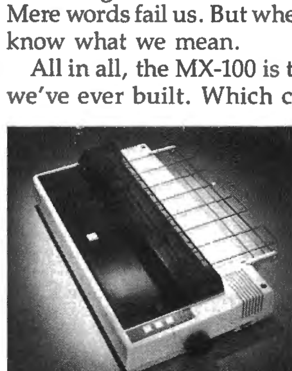
Frankly, it wasn't easy. But the results of all our sleepless nights will knock your socks off.

The MX-100 is a printer that must be seen to be believed. For starters, we built in unmatched correspondence quality printing, and an ultra-high resolution bit image graphics capability. Then we added the ability to print up to 233 columns of information on 15" wide paper to give you the most incredible spread sheets you're ever likely to see. Finally, we topped it all off with *both* a satin-smooth friction feed platen *and* fully adjustable, removable tractors. And the list of standard features goes on and on and on.

Needless to say, the specs on this machine — and especially at under $1000 — are practically unbelievable. But there's something about the MX-100 that goes far beyond just the specs; something about the way it all comes together, the attention to detail, the fit, the feel. Mere words fail us. But when you see an MX-100, you'll know what we mean.

All in all, the MX-100 is the most remarkable printer we've ever built. Which creates rather a large problem for those of us at Epson.

How are we going to top this?



Your next printer.

EPSON

EPSON AMERICA, INC.

3415 Kashiwa Street • Torrance, California 90505 • (213) 539-9140

See the whole incredible Epson MX Series of printers at your Authorized Epson Dealer. ✓404

Instant Software™ Inc.

Peterborough, NH 03458 – A Division of WAYNE GREEN INC.

$1 *This coupon may be redeemed for one dollar off any Instant Software package at all participating dealers in the United States. Limit—one coupon per package. Not valid for mail order purchases. All information must be filled out completely. Dealers—redeem coupon through Instant Software.* $1

Program you are purchasing ____________________

Your name ____________________

Address ____________________

City ____________________ State ________ Zip ________

Dealer name ____________________

____

Address ____________________

____________________ Zip ________

-A41

Instant Software™ Inc.

Peterborough, NH 03458 – A Division of WAYNE GREEN INC.

$1 *This coupon may be redeemed for one dollar off any Instant Software package at all participating dealers in the United States. Limit—one coupon per package. Not valid for mail order purchases. All information must be filled out completely. Dealers—redeem coupon through Instant Software.* $1

Program you are purchasing ____________________

Your name ____________________

Address ____________________

City ____________________ State ________ Zip ________

Dealer name ____________________

____

Address ____________________

____________________ Zip ________

-A41

Instant Software™ Inc.

Peterborough, NH 03458 – A Division of WAYNE GREEN INC.

$1 *This coupon may be redeemed for one dollar off any Instant Software package at all participating dealers in the United States. Limit—one coupon per package. Not valid for mail order purchases. All information must be filled out completely. Dealers—redeem coupon through Instant Software.* $1

Program you are purchasing ____________________

Your name ____________________

Address ____________________

City ____________________ State ________ Zip ________

Dealer name ____________________

____

Address ____________________

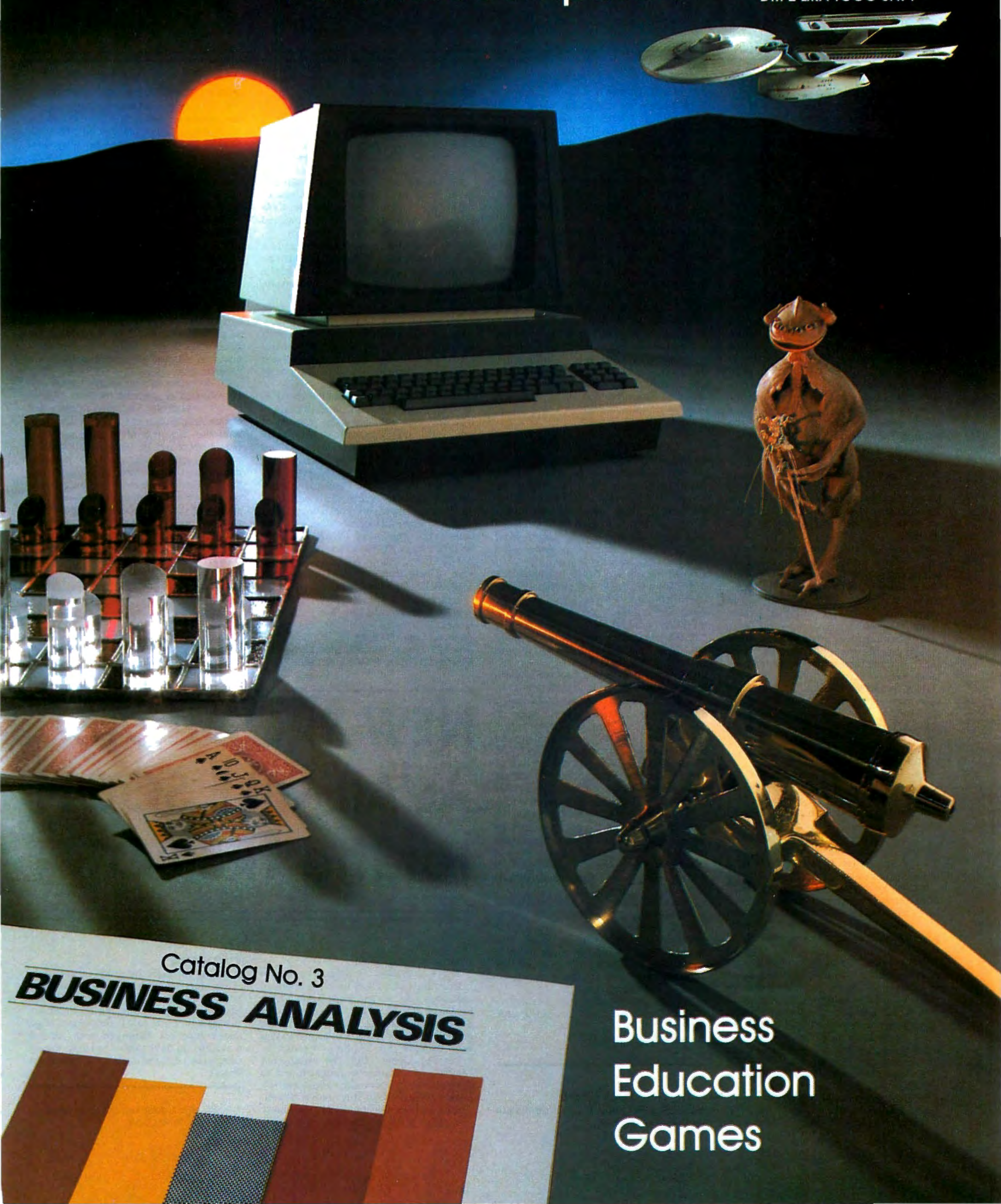
____________________ Zip ________

-A41

Instant Software™

For Microcomputers

U.S. $1.00 UK 50p FFR 5
DM 2 LIRA 1000 SFR 1



Business
Education
Games

Instant Software™

ALL NEW!!

NEW PRODUCTS

TEMPLE OF THE SUN

Your quest for the ancient Rites of the Shaman has finally led you to a remote South American ruin, long rumored to be the legendary TEMPLE OF THE SUN. Now, at last you are able to explore the site for the last clues to the most powerful order of Wizards left in the world. Octo-directional, the matrix of this adventure is nearly endless, as are the ways to die. Model I, Level II, 16K.

5012R (Tape) $19.95
5011RD (Disk) $29.95

THE RUSSIAN DISK

Well, here it is: THE RUSSIAN DISK. Everything your APPLE computer needs to put you into the international picture. Instant Software has packed two multi-part packages onto this single disk, so that you can master the Russian (or Cyrillic) Alphabet in no time flat, learn to type in Russian, and to read and recognize a variety of Russian words for foods, drinks, stores, and even names.

The first set of programs eases you through the Cyrillic alphabet in short, rapidly digested and thoroughly exercised steps, starting with those letters which most resemble the English counterparts and introducing the more exotic ones bit by bit. Enjoy that lift which comes from the discovery that you have learned the entire Russian alphabet, almost without trying.

The second set of programs develops your reading ability in Russian, and expands your Russian vocabulary as well. You'll learn the words for various foods, drinks and places to eat, for signs and names of stores—everything the traveller needs to know.

Furthermore, the program will allow you to practice typing in Russian, using the Cyrillic characters. Recommended age level, 10 to adult.

RUSSIAN DISK requires 32K RAM, one disk drive, an Apple II or Apple II Plus and Applesoft.

0283AD (Disk) $24.95

·MASTER REVERSI

Are you looking for a game you can spend hours with; a game that's simple to play, yet infinitely challenging; a game that you can play with a friend, or against your computer in the wee hours of the morning; a game that can be played at parties, and at serious tournaments?

Look no farther! Master Reversi will meet all your needs!

Master Reversi is a tournament-winning game program that has more features than any other reversi program on the market. It will challenge and teach you no matter what your degree of expertise.

What makes Master Reversi really special is its ability to allow in-depth analysis of moves and games. You may examine the computer's evaluation and choice of moves. You can save and replay interesting moves and games. You will be able to study and manipulate dozens of tournament-level games which are provided in the program's vast library.

Master Reversi will enable you to overcome any barriers standing between you and a world championship.

Model I, Level II, 16K.

0378RD (Disk) $29.95

CAPITALIZATION

As the first module in the Instant Software Aided Instruction (CAI) English Language Series, CAPITALIZATION is an educationally solid program for the teaching and review of the rules governing capital letters. With this program and an Apple computer, the student, the writer, the reporter—anyone can learn or review the 12 fundamental rules of capitalization quickly and effectively.

The program is designed for students of junior high school level and beyond. Each rule is concisely explained on the screen, and examples are given. Next, the student is allowed to exercise the usage of the rule. The computer keeps score and reports the level of mastery at the end of each set of exercises. The student may then choose to study the next usage, or may return for further practice to the previous exercise. A Model Mastery Worksheet is included in the documentation as an aid to teacher and student. CAPITALIZATION requires 32K, one disk drive and an Apple II or II Plus with Applesoft Basic.

0339AD (Disk) $24.95

SPACE SHUTTLE

Inside the cockpit of the space shuttle, Columbia, the pale green light from three computer driven TV screens illuminates the craft's 1,400 switches and circuit breakers. The Command Pilot gives the necessary instructions to begin what NASA calls the "deorbit" procedure. It's time to bring this baby home.

Just 55 hours ago, 5.3 million pounds of thrust from two solid fuel boosters, essentially 15-story firecrackers, sent the somewhat chubby Space Shuttle into orbit. Now, on-board computers performing 325,000 separate operations every second will help guide the craft safely and gracefully back to its roost.

Over the Indian Ocean, the OMS engines burn for two minutes to slow the craft down from its orbital speed—a speed equivalent to 25 times the speed of sound! The shuttle craft begins to drop and is turned nose up so that its tile coated belly will absorb the enormous heat of reentry. Those 34,000 silicon glass tiles are all that prevent the craft from being burned up. Thirty-five minutes later, the orbiter enters the Earth's atmosphere over Midway Island. By now, the shuttle is banking and flying wide traverses to control its speed. The strain on its superstructure is tremendous. Finally, the runway at Cape Canaveral is dead ahead. At 190 miles per hour, the shuttle touches down and ends its perilous journey.

Instant Software's SPACE SHUTTLE puts you in the Command Pilot's chair of one of America's first reusable space vehicles. You will bring her back from space. Simply stated, this will be the most fascinating and breathtaking experience you will have until the space shuttle becomes a commuter run.

SPACE SHUTTLE is the ultimate air flight simulation. Model I, Level II, 16K; Model III, 16K.

0332R (Tape) $14.95

USER'S TIPS

Never remove a floppy disk from its protective plastic casing.

DRAGONQUEST
THE PRINCESS
HAS BEEN KIDNAPPED

And you, Sir Knight, have the task of returning her to her father, the King, before nightfall, when SMAEGOR, the most foul, will certainly do her in.

Explore the unfamiliar lands beside the River Delta as you search for clues to SMAEGOR'S hiding place. Get the tools you will need: weapons, food, and magical items if you can find them. Prepare for encounters with the creatures of the land for some of them may be in league with The Dragon. Seek answers in the Land of the Dead, but enter cautiously for Ill awaits you there.

When, at last you are ready, confront the Mighty SMAEGOR in his lair, for M'lady awaits and is in much danger. CAN YOU DO IT? Can you vanquish SMAEGOR and beat the setting sun? Many surprises await you in this DRAGONQUEST—and one of them is named DEATH.

In a desperate race against the sun, you search for SMAEGOR, Monarch of Dragonfolk, who has kidnapped the princess of the Realm and holds her in a distant and unknown place. In a quest for Honor and glory, you must search the land, seeking out the tools needed for the ultimate confrontation. On the River Delta, in the abandoned Temple of Baathteski, Goddess of the Blade, everywhere clues abound. But WHERE is the Princess? Now, as never before, the genius of CHARLES FORSYTHE shines in this machine language ADVENTURE DRAGONQUEST. Can YOU save M'lady from the iron clutches of SMAEGOR? Tape version: Model I, Level II, 16K; Model III, 16K. Two individual disk versions available for Model I and Model III.

5006R (Tape) $15.95
5007RD (Disk, MODEL I) $21.95
5010R3D (Disk, MODEL III) $21.95

THE ELECTRONIC BREADBOARD

Computer Aided Design (CAD) is one of the newest applications of microcomputer technology. THE ELECTRONIC BREADBOARD permits the design and analysis of analog circuits. It can be used to evaluate voltages, currents, impedance and the frequency response of any circuit. This package is ideal for audio component repairmen, ham radio technicians, hobbyists, electrical engineers, telecommunications engineers, audiophiles and students of electronics. Tests can be made on any analog circuit such as transmitting and receiving equipment, amplifiers and the like. The program will allow: adding or removing components; calculating impedance at a particular point in the circuit; obtaining the voltage at a particular point (or all points); setting of the desired frequency or a sweep range of frequencies for analysis; and saving of circuit designs on cassette tape.

Components in a simulated circuit can be removed or inserted for fast, efficient design or analysis. The possible causes of circuit failure can easily be determined, reducing the amount of time spent on troubleshooting. In conjunction with a course in analog electronics, the program can be used as a training aid. Students can verify electrical theory by entering example circuits and determining frequency response. If you're an electronic doughboy, then you need THE ELECTRONIC BREADBOARD! Model I, Level II, 16K; Model III, 16K.

0287R (Tape) $49.95

NEW PRODUCTS AVAILABLE SOON for the TRS-80*

DOMES OF KILGARI

The Ion Station at Kilgari is rich in history and as far away from the civilized galaxy as you can get. Your Survey craft has been disabled, her Ion engines are dead for lack of fuel. Only by retrieving an Ion rod for the mass converter can you hope to leave the Outrim and return to Federation space. But the automated Digitron Ion Station is programmed to destroy all intruders. Can you solve the puzzle presented by this classic Adventure? Or will you be doomed to the lonely wasteland of Kilgari forever?

TIMEQUEST

Sifting through the sands of time, you attempt to solve the mystery as it unfolds before you. From the distant past to the obscure future your Time machine tracks Adventure in its truest form. High speed machine language with all the features you have grown to enjoy. Easily one of the best and destined to become a classic.

*TRS-80 is a trademark of Radio Shack, a division of Tandy Corp.

Table of Contents

Our Guarantee

Defective software may be returned for exact replacement at no cost to you, or for full credit, within thirty days of the invoice date. You MUST enclose dated proof of purchase for any replacement to be made, so please keep your invoice.

Should a disk or cassette become defective after the warranty period, Instant Software will still protect you. You may return the original defective cassette along with $4.00, or any disk with $5.00 for a replacement. Again, you must provide us with a copy of your invoice for any return to be made.

On the Cover: sculpture entitled "The Undersecretary in Charge of Protocol" by Don Simpson, courtesy of the Earthlight Gallery, Boston.

Photo by Martin Paul

Instant Software™ Inc.

Peterborough, New Hampshire 03458

Get a glimpse of the future today—at your nearest Instant Software dealer.

ALABAMA
THE COMPUTER SHOP, Gadsden
ALASKA
COMPUTER TALK, Anchorage
JUNEAU ELECTRONICS, Juneau
WAPATCO, Anchorage
ARIZONA
COMPUTER STORE, Phoenix
MESA ELECTRONICS, Mesa
MILLET'S ELECTRONICS, Mesa
PERSONAL COMPUTER PLACE, Mesa
SIMUTEK, Tucson
SOFTWARE STATION, Tempe
TOY BOX, Sierra Vista
ARKANSAS
MICRO COMPUTER SYSTEMS, Hot Springs
CALIFORNIA
ADVANCE RADIO (R/S DEALER), Grass Valley
ADVANCED COMPUTER PRODUCTS, Santa Ana
ALLTRONICS, San Jose
AMCO ELECTRONIC SUPPLY, Azusa
ASAP COMPUTERS, Signal Hill
BYTE INDUSTRIES, Hayward
BYTE SHOP, Mountain View
CAPITOL COMPUTER SYSTEMS, Sacramento
COAST ELECTRONICS, Morro Bay
COMPUSOUND, Orange
COMPUTER MART OF CALIFORNIA, INC., Diamond Bar
COMPUTER PLUS, Sunnyvale
COMPUTER STORE, San Leandro
COMPUTER WORLD, Westminster
DEROSEAR APPLIANCE & REPAIR, Weaverville
DIMENSIONAL SOFTWARE, San Diego
EXATRON, Santa Clara
GAME-O-RAMA, Santa Barbara
GRAY MATTER, San Francisco
HOBBY WORLD ELECTRONICS, Northridge
HUNTINGTON COMPUTING, Corcoran
MALIBU MICROCOMPUTING, Malibu
MICRO SPECIALTIES, Daily City
MN&T INDUSTRIES, Lompoc
OPAMP/TECHNICAL BOOKS, Los Angeles
OPPORTUNITIES FOR LEARNING, Chatsworth
PC COMPUTERS, El Cerrito
Q I COMPUTERS, INC., Lawndale
RAC PRODUCTS, San Jose
RADIO SHACK, El Cajon
RADIO SHACK, Palm Springs
RADIO SHACK, San Diego
R&V SOUND (R/S DEALER), Fortuna
SALINAS HOBBY CENTER, Salinas
SHAVER RADIO, San Jose
SILVER SPUR ELECTRONICS, Chino
SOFTWARE PLUS, El Toro
STACEY'S BOOKSTORE, San Francisco
STRAWFLOWER ELECTRONICS (R/S DEALER), Half Moon Bay
THE COMPUTER STORE, Santa Monica
THE FEDERATED GROUP, Commerce
THE SOFTWARE STORE, Huntington Beach
WABASH APPLE, El Toro
WARRIORS, WIZARDS & ROBOTS, Garden Grove
WENNER BUSINESS SYSTEMS, Los Altos
COLORADO
APPARAT, Denver
FISTEL'S MICRO ELECTRONICS, Denver
POOR RICHARD'S CALCULATORS, Fort Collins
SOFTWARE GOURMET, Denver
CONNECTICUT
AMERICAN BUSINESS COMPUTERS, Groton
BYTE ME COMPUTER SHOP, New London
COMPUTER LAB, New London
INSTRUCTIONAL SYSTEMS COMPUTERS, Manchester
TECHNOLOGY SYSTEMS, Bethel
THE COMPUTER STORE, Stamford
DELAWARE
MICRO PRODUCTS, Wilmington
OMNIFAX, Wilmington
DISTRICT OF COLUMBIA
THE PROGRAM STORE, Washington, D.C.
FLORIDA
ADVENTURE INTERNATIONAL, Casselberry
ALL SYSTEMS GO, Winter Garden
AMF MICROCOMPUTER CENTER, Tampa
COMPUTER CENTER, West Palm
COMPUTER JUNCTION, Fort Lauderdale
COMPUTER WORLDS, Clearwater
COMPUTERLAND, Jacksonville
COMPUTERLAND, Sarasota
COMPUTERLAND, Tampa
COMPUTERLAND, West Palm Beach
HEATHKIT ELECTRONIC CENTER, Hialeah
H&H HOBBY SALES, Sarasota
MICROCOMP LTD., Miami
MICROCOMPUTER SYSTEMS INC., Tampa
MINI CONCEPTS, Holly Hill
SAV-A-BYTE, Jacksonville
SOUND IDEAS, Gainesville
SOUTH EAST MICRO DATA, Orlando

GEORGIA
ATLANTA COMPUTER MART, Atlanta
BAILEY'S COMPUTER SHOP, Augusta
DELTA DATA DYNAMICS, Tucker
ENERGY LOGIC, Columbus
FLEMING DRUG CO., Wrens
HAWAII
HONOLULU ELECTRONICS, Honolulu
MILLS ELECTRONICS, Lahaina
RADIO SHACK ASSOC. STORE, Honolulu
IDAHO
DENNIS STONE ENTERPRISES, Fruitland
ELECTRONIC SPECIALTIES, Boise
R&L DATA SYSTEMS, Idaho Falls
ILLINOIS
ALPINE COMPUTER CENTER, Rockford
BELL ELECTRONICS, Edinboro
CHICAGO MAIN NEWSTAND, Evanston
COMPREHENSIVE MICRO SYSTEMS, Chicago
COMPUTER JUNCTION, Elmhurst
COMPUTER STORE, Rockford
COMPUTERLAND, Niles
CREATIVE PROGRAMMING, Charleston
GARCIA & ASSOCIATES, Chicago
ILLINOIS CUSTOM COMPUTERS, Harrisburg
MAIN STREET COMPUTER CO., Decatur
MIDWEST MICRO COMPUTERS, Lombard
THE PILOT HOUSE, Gladstone
WALLACE COMPUTERS, Peoria
INDIANA
ABC HOBBY, Evansville
DIGITAL TECHNOLOGY, Lafayette
FALL CREEEK ELECTRONICS, Pendleton
GAME HUT, Bloomington
GAME PRESERVE, Indianapolis
PROFESSIONAL MICROCOMPUTER SOFTWARE, Muncie
SIMONTON LAKE DRUGS, Elkhart
THE BOARDROOM, Indianapolis
THE HAM SHACK, Evansville
IOWA
LENWOOD SYSTEMS, Center Point
MEMORY BANK INC., Bettendorf
SERNETT LEISURE CENTER, Carroll
KANSAS
AMATEUR RADIO EQUIPMENT, Wichita
CENTRAL KANSAS COMPUTERS, Herington
COMMUNICATIONS CENTER, Lincoln
GOSUB INTL, Wichita
HATCH COMPUTER CENTER, Alliance
HIGH TECHNOLOGY, Wichita
KENTUCKY
COMPUTER MAGIC, Louisville
PERRY'S COMPUTER, Bremen
LOUISIANA
ACME BOOK CO., Baton Rouge
COMPUTER SHOPPE, Metairie
MAINE
FRYEBURG COMPUTER CENTER, Fryeburg
MAINE MICRO SYSTEMS INC., Auburn
MID-MAINE COMPUTER COMPANY, Auburn
PORTSMOUTH COMPUTER CENTER, Portsmouth
MARYLAND
COMM CENTER, Laurel
PROGRAM STORE, Baltimore
WILLS COMPUTER STORE, Marlow Heights
MASSACHUSETTS
COMPUTER CITY, Charlestown
HI FI EXCHANGE, Northampton
LAND OF ELECTRONICS, Lynn
MARK GORDON COMPUTERS, Cambridge
OMNITEK SYSTEMS, Tewksbury
SOUND COMPANY, Springfield
STAR COMPUTING, Framingham
THE GAME SHOP, East Acton
TUFTS RADIO ELECTRONICS, Medford
MICHIGAN
ALL FOR LEARNING, W. Bloomfield
ALTERNATE SOURCE, Lansing
A.M. ELECTRONICS, Ann Arbor
COMIC KINGDOM, Detroit
COMPUTER CENTER, Garden City
COMPUTER CONNECTION, Farmington Hills
COMPUTER MART, Clawson
COMPUTER MART, Flint
COMPUTERLAND, Kentwood
COMPUTERLAND, Southfield
COMPUTRONIX, Midland
EIGHT BIT CORNER, Muskegon
FERRIS RADIO, Hazel Park
HOBBY HOUSE, Battle Creek
LEARNING CENTER LTD., Ann Arbor
MAIN SYSTEMS, INC., Flint
MID-MICHIGAN MEMORY, Dimondale
NEWMAN COMPUTER EXCHANGE, Ann Arbor
TRI-COUNTY ELECTRONICS & SOUND CENTER, Fenton
WEATHERWAX DRUGS, Brooklyn
WIZARD'S ARSENAL, East Lansing
YE OLDE TEACHERS SHOPPE, Ypsilanti

MINNESOTA
CODE ROOM, Eden Prairie
MINNESOTA SOFTWARE, White Bear Lake
RURAL AMERICA ENTERPRISES, Marshall
ZIM COMPUTERS, Brooklyn Center
MISSISSIPPI
C-COM, Jackson
DYER'S INC., West Point
SOFTWAREHOUSE, Jackson
MISSOURI
CENTURY NEXT COMPUTERS, Columbia
COMPUTER CENTER, Joplin
COMP-U-TRS, Florissant
CRC COMPUTERS, Joplin
D'S, Cameron
HOUSE OF COMPUTERS, Joplin
SOFTWARE SHACK, Belton
MONTANA
COMPUTER STORE, Billings
THE COMPUTER PLACE, Kalispell
NEBRASKA
APPLETREE SOFTWARE, Battle Creek
COMPUTERS WEST, Omaha
GREAT RACE HOBBY PLACE, Lincoln
LEZOTTE STUDIO, Ogallala
NEVADA
CENTURY 23, Las Vegas
HURLEY ELECTRONICS, Las Vegas
NEW HAMPSHIRE
BITSNBYTES COMPUTER CENTER, Concord
COMPUCRAFT, Keene
COVER CRAFT, Amherst
PAUL'S TV, Fremont
PORTSMOUTH COMPUTER CENTER, Portsmouth
RADIO SHACK ASSOC. STORE, Keene
STURDIVANT AND DUNN, Conway
NEW JERSEY
ABC'S TV SALES & SERVICE, Glassboro
ADELMAN'S STATIONERY CO., Union City
BARGAIN BROTHERS, West Trenton
CHANNEL 1 RADIO SHACK, Medford
COMPUTER FORUM, Redbank
COMPUTER MADNESS, Englishtown
COMPUTERWORLD OF WESTFIELD, Westfield
CROWLEY'S, Whitehouse Station
DAVE'S ELECTRONICS, INC., Pennsville
ELECTRONIC WORLD, Mantua
HOBBYMASTERS, Red Bank
J&J ELECTRONIC WORLD, Mantua
LASHEN ELECTRONICS, INC., Denville
MIDAS DATA SYSTEMS INC., Marlton
OMNIFAX, Cherry Hill
RADIO SHACK ASSOC. STORE, Moorestown
RADIOS UNLIMITED, Somerset
SOFTWARE CITY, River Edge
NEW MEXICO
AUTEL ELECTRONICS CO., Albuquerque
J&W ENTERPRISES, Clovis
MITCHELL MUSIC, Carlsbad
THOMAS E. CARR JEWELER, Alamogordo
WARGAMES WEST, Albuquerque
NEW YORK
ARISTO CRAFT DISTINCTIVE MINIATURES, New York
ASD HOME COMPUTER CENTER, Poughkeepsie
BERLINER COMPUTER CENTER, New Hyde Park
C. HABILD OF NEW DORP, Staten Island
COMPUTER CORNER, White Plains
COMPUTER RESOURCES, Williamsville
COMPUTER SHOP, Kingston
COMPUTERLAND, Carle Place
COMPUTERLAND OF NYC, New York
DIGIBYTE SYSTEMS, New York
80-MICROCOMPUTER SERVICES, Cohoes
FUTURE VISIONS COMPUTER STORE, Melville
PHANTOS RESEARCH, Camillus
PROGRAMS UNLIMITED, Jericho
MR. COMPUTER, Wappingers Falls
OMNIFAX, DeWitt
SOFTRON SYSTEMS, Rensselaer
WATERLOO HOBBIES, Mineola
WORLD OF COMPUTERS, Port Chester
NORTH CAROLINA
RAYBURN MICRO-ELECTRONICS, Sylva
SOUND MILL, Havelock
OHIO
ABACUS II, Toledo
ALTAIR SYSTEMS, INC., Dayton
ASTRO VIDEO ELECTRONICS, INC., Lancaster
CINCINNATI COMPUTER STORE, Cincinnati
COMPUTER STORE, Toledo
COMPUTERLAND, Columbus
COMPUTERLAND, Mayfield Heights
COMPUTERLAND, North Olmsted
COMPUTERLAND, Warren
JOBAR ENTERPRISES, Middlefield

MICROAGE, Columbus
MICRO COMPUTER CENTER, Centerville
MICRO ELECTRONICS INC., Columbus
MICRO-MINI COMPUTER WORLD, Columbus
TWENTY-FIRST CENTURY SHOP, Cincinnati
WANNA PLAY, Cincinnati
OKLAHOMA
COMPUTER STORE, INC., Tulsa
COMPUTER WORLD, Tulsa
PROFILE INC., Moore
RADIO SHACK ASSOC. STORE, Guymon
SOUNDS, ETC., Watonga
OREGON
COMPUTER PATHWAYS, Salem
L&R ELECTRONICS, Grant Pass
PIONEER ELECTRONICS, Sandy
TRS-80 PRODUCTS LTD., Portland
PENNSYLVANIA
ALLIED HOBBIES, Philadelphia
ARTCO ELECTRONICS, Kingston
BELL ELECTRONICS, Girard
COMPUTERLAND, Gibsonia
COMPUTERLAND OF HARRISBURG, Mechanicsburg
ERIE COMPUTER, Erie
JNE COMMUNICATIONS, Altoona
MAFEX ASSOCIATES, Johnstown
OMNIFAX, Feasterville
OMNIFAX, Philadelphia
PERSONAL COMPUTER CORP., Paoli
PITTSBURG COMPUTER STORE, Pittsburg
STEVENS RADIO SHACK DEALER, Phoenixville
ROUTE 30 ELECTRONICS, Latrobe
RUMPELSTILSKIN TOY SHOP, New Hope
TELEVISION PARTS COMPANY INC., New Brighton
PUERTO RICO
MICRO COMPUTER STORE, Caparra Terrace
RHODE ISLAND
COLONIAL ENTERPRISES, Foster
SOUTH CAROLINA
OMNI ELECTRONICS, Charleston
TENNESSEE
ACE MINI SYSTEMS, Clarksville
CHATTANOOGA COMPUTER CENTER, Chattanooga
COMPUTER WORLD, Nashville
COMPUTERLAB, Memphis
TEXAS
CODEDATA INC., Arlington
COMPUSHOP, Bellaire
COMPUSHOP/FM 1960W, Houston
COMPUSHOP/N FWY, Houston
COMPUTER 'N THINGS, Austin
COMPUTER PROGRAMMING ASSOC., Lubbock
COMPUTER PORT, Arlington
COMPUTER SALES & SERVICE, Fort Worth
COMPUTER SOLUTIONS, San Antonio
COMPUTERLAND OF SW HOUSTON, Houston
COMPUTERS BY O'NEILL, Lake Jackson
COMPUTERS TO GO, Austin
COMPUTEX, Webster
CORSAIR, Ft. Worth
GATEWAY ELECTRONICS, Houston
KA ELECTRONICS, Dallas
MAC'S TV, Fairfield
MARYMAC INDUSTRIES (R/S DEALER), Houston
PAN AMERICAN ELECTRONICS (R/S DEALER), Mission
R.L. COLE'S ELECTRONICS, San Antonio
ROY'S CB & ELECTRONICS, Aransas Pass
80 SOFTWARE, San Antonio
WAGHALTER BOOKS, INC., Houston
UTAH
COMPUTERLAND, Salt Lake City
FOOTHILL MODELERS, Salt Lake City
VIRGINIA
COMPUTER SOLUTIONS, Leesburg
COMPUTER WORKS, INC., Harrisonburg
HOME COMPUTER CENTER INC., Virginia Beach
LITTLE SOLDIER, Alexandria
SYSTEMS MARKETING, Arlington
WASHINGTON
AMERICAN MERCANTILE COMPANY, Seattle
COMPUTERLAND, Bellevue
COMPUTERLAND, Federal Way
COMPUTERLAND, Spokane
EMPIRE ELECTRONICS, Sunnyside
J.B. SALES, Snohomish
LORDS, Port Angeles
UNIVERSITY VILLAGE MUSIC, Seattle
U.S.S. ENTERPRISE, Kirkland
WESTERN MICROCOMPUTER CENTER, Bellingham
WEST VIRGINIA
COMPUTER CORNER, Morgantown
COMPUTER STORE, Huntington

OHIO VALLEY ELECTRONIC SYSTEMS, Wheeling
SOUND & ELECTRONIC SPECIALTIES, Morgantown
WISCONSIN
BYTE SHOP, Milwaukee
COMPUTERLAND, Madison
COMPUTERLAND OF FOX RIVER VALLEY, Oshkosh
COMPUTER WORLD, Green Bay
MAGIC LANTERN COMPUTER, Madison
OMEGA MICROS, Milwaukee
PETTED MICROSYSTEMS, Milwaukee
S&O TV SALES, Monroe
WYOMING
COMPUTER CONCEPTS, Cheyenne

AUSTRALIA
DE FOREST SOFTWARE, Nunawading, Vic.
CANADA

Distributor:
MICRON DISTRIBUTING,
Toronto, Ont.

ALLIED COMPUTER CENTRE, Thunder Bay, Ontario
ARKON ELECTRONICS, Toronto, Ontario
AULL COMPUTER SYSTEMS, Victoria, B.C.
BITS & BYTES, Dartmouth, N.S.
CENTRAL DISTRIBUTORS LTD., Lachine, Quebec
COMPUCORNER, Smithers, B.C.
COMPUMART, Ottawa, Ontario
COMPUTER BARN, Sarnia, Ontario
COMPUTER CIRCUIT, London, Ontario
COMPUTER INNOVATIONS, Ottawa, Ontario
COMPUTER SHOP LTD., Calgary, Alberta
COMPUTER WORLD, Vancouver, B.C.
COMPUTERLAND OF BURLINGTON, Burlington, Ontario
CREATIVE COMPUTERS, Victoria, B.C.
DATATEC COMPUTER SYSTEMS LTD., Saskatoon, Sask.
ELECTRONICS 2001, Willowdale, Ontario
GALACTICA COMPUTERS LTD., Edmonton, Alberta
IRISCO DU QUEBEC, Ste. Foy, Quebec
LYONS LOGIC LTD., London, Ontario
MICROMATION, Toronto, Ontario
MICROSHACK, Saskatoon, Sask.
MICRO SHACK, Regina, Sask.
M&W COMPUTERS, Mississauga, Ontario
NIP & TUCK VARIETY, London, Ontario
OFFICE CENTRE, Kingston, Ontario
SAULT OFFICE MACHINES, Sault St. Marie, Ontario
STATUS COMPUTER SYSTEMS, St. Catherines, Ontario
TOTAL COMPUTER SYSTEMS, Ajax, Ontario
WEST WORLD COMPUTERS, Edmonton, Alberta
ITALY
BITS & BYTES, Milan
NEW ZEALAND
VISCOUNT ELECTRONICS, Palmerston North
NORWAY
A/S SORLUND, Vedavagen
SWEDEN
SENTEC AB, Jarfalla
UNITED KINGDOM
CALISTO COMPUTERS, Birmingham
THE SOFTWARE HOUSE, London
WEST GERMANY
MICROSTUFF, Frankfurt
NETHERLANDS & BELGIUM

Distributor:
SOFTWARE IMPORT BRABANT,
Eindhoven, Neth.

COMPUTER COLLECTIEF, Amsterdam
MICRO COMPUTING, Krommenie
MICRO DYNAMICS, Eindhoven
MUSICPRINT-CHIP, Arkel
OVEL GONNE, Drachten
R.A.L. MICROCOMPUTERS, The Hague

YOUR NAME COULD BE HERE. CALL: 1-800-258-5473

Find out what you're missing.

You heard the lady. Your microcomputing life *is* going to be very exciting. But don't leave it to chance—leave it to us! We have a brand new catalog full of packages and valuable coupons to help make it happen. It's free . . . if you fill out the coupon and mail it today. (Or make a photocopy and spare your magazine).

NAME: ______________________________

ADDRESS: ______________________________

CITY: ____________________ **STATE:** ________ **ZIP:** ________

Instant Software™ Inc. Peterborough, N.H. 03458 603-924-7296 ·A21

Introduction

Wayne Green's anticipation of the growing demand for microcomputer software brought about his establishment of Instant Software back in early 1978. At that time, Instant Software consisted of a handful of employees, and produced six programs.

Today, Instant Software employs fifty five people and produces over one hundred software packages for the TRS-80, Apple, Pet, TI, and Northstar. We have over 200 more in the development stages, including programs for the TRS-80 Color, Atari and Casio computers. The rapid rate of development of microcomputer software by today's top-notch programmers pushes the industry's standards higher. For this reason, several months ago, we began an extensive review of each Instant Software product. The results . . . temporarily, a smaller product line, but a top-notch one.

Those packages which did not meet our standards are being rerun through our development pipeline. We are working closely with the authors to upgrade these packages, and make them available to you once again.

Speaking of upgrading . . . you'll want to keep your eyes on our new products, packaging and our advertising campaigns in the upcoming months. These all create an image in which we see . . .

Instant Software in your future.

Instant Software, Inc.

PMC-80 COMPATIBILITY

At this time, the following TRS-80 programs are known to be compatible with the PMC-80 computer:

0017R	Air Flight Simulation
0023R	Oil Tycoon
0058R	Programmer's Converter
0106R	Air Mail Pilot
0117R	Night Flight
0127R	Surveyor's Apprentice
0133R	Renum/Compress
0136R	Beginner's Russian
0137R	Everyday Russian
0140R	Business Analysis
0141R	Battleground
0159R	Jet Fighter Pilot
0168R	Label
0171R	Flight Path
0218R	Dr. Chips
0219R	House of 30 Gables
0223R	Cosmic Patrol
0228R	Dynamic Device Drivers
0230R	TLDIS
0232R	Disassembler
0240R	Alien Attack Force
0246R	Compression Utility Pack
0250R	IRV
5003R	Ultra-Mon

Instant Software™

PETERBOROUGH, N.H. 03458

Order Form

TOLL FREE ORDER NUMBER: 1-800-258-5473

Name: ______

Address: ______

City: ______ State____ Zip____

Method of payment:

☐ check ☐ money order ☐ VISA ☐ AMEX ☐ MC

Card No. ______ Expiration

Signature ______ Date______

Yes, I would like to see my local software dealer carry more INSTANT SOFTWARE:

Store Name ______

Address ______ City______ State______ Zip______

QUANTITY	ORDER NO.	PROGRAM NAME	UNIT COST	TOTAL COST
		+ shipping		2.50
			Total	

You may photocopy this order form if you wish to keep your catalog intact. M

ATTENTION PROGRAMMERS

How to Submit Programs to Instant Software

The guiding idea behind Instant Software is a simple one. It takes time, talent, training and experience to write good programs; it also takes time, money, training, experience and talent to establish an efficient marketing system. You have the first set of qualifications; we have the second. We want to team up with you.

Instant Software is set up to evaluate submitted programs, to publish the best ones in quantity, and to market these programs through a network of computer stores and other retail outlets.

You have our catalog in your hands. Look at the breadth of the programs available: games, simulations, utilities, educational and business programs. While games have always been good sellers and are likely to remain so, bright futures for microcomputer software lie in the fields of business, educational and utility programs. Every business person wants to put his computer to work saving time and money. Good bookkeeping, accounting, inventory and billing programs are sure sellers. As microcomputers start to appear in schools, the need for good CAI programs will grow. Finally, programmers will continue to need the tools to write, edit and debug programs.

Thinking of programming for publication? If, in light of what is now being sold, your programs perform a unique function, work faster, are more convenient or easier to use, then we want to see them.

The kind of software we're soliciting consists of programs that not only solve old problems but also do jobs that couldn't be done before. We want software that incorporates the latest developments in the art of programming.

One piece of advice—write what you know. This advice applies especially to writers of computer software. If you're an accountant, you understand the problems faced by accountants. An accounting program authored by you has most likely addressed just those problems, and is probably a good one. The same goes for an educational program. A good teacher knows from training and experience precisely what students need in order to learn, and how to give it to them.

Please note: Instant Software is supporting all popular microcomputer systems. We want software for the TRS-80 Models I, II, III and the Color Computer, Apple II and III, Atari 800, PET, TI 99/4, Casio 2, and any other successful microcomputer system. At present the Instant Software lab, one of the best equipped computer laboratories in the world, has over 15 different computer systems running. What's more, we have the staff and the expertise to run and evaluate programs for all of these systems. We also seek conversions of successful programs from one system to another.

For the full details on program submission, write for a Submission Package to:

Submission Department
Instant Software, Inc.
Rt. 101 and Elm St.
Peterborough, NH 03458

In brief, the package will tell you that we need a complete description of what your program does, how it does it, how to use it and what benefits the user will get from the program. When you are ready, this information, plus a disk or cassette copy of your program, should be sent to the above address. We will send you a card of acknowledgement upon receipt of your material.

If we feel that your program has marketing potential, we'll send you a contract. Upon receipt of the signed contract, we'll send your program to up to five associate editors. We try to match the editor's background to the subject matter of the program. The associate editors will evaluate your program, tell us what they think of it, make suggestions for changes and suggest an estimated retail price. Our programmers will contact you about any changes.

Once the program has been completed, we'll write the user documentation, using materials you provide plus the experience of our trained documentation staff. Next, we design the package and write the package and catalog advertising. Then we print the documentation, make copies of the program on cassette or disk and release the package to our marketing department.

Our marketing department will set the price for the package, arrange the advertising campaign and coordinate the distribution of the package through our marketing representatives and Instant Software dealers.

Business/Applications for your TRS-80

FIND IT QUICK

Put an end to the misplaced information syndrome! Here is a reliable, fast, subject-oriented information locator—FIND IT QUICK! This information indexing and retrieval system is versatile and flexible—the ideal tool for doctors, lawyers, engineers, businessmen, educators, anyone who needs to locate information.

Designed with a journal/magazine format, FIQ is versatile enough to be used to store and retrieve any type of reference information including client lists, vendor advertisements, report bibliographies and more—even poems and famous quotations. Subjects can be indexed or called according to author, journal and date or by up to nine keywords that you determine. Keywords can be used singly or in combination during the search routines.

FIQ can be used by anyone. It requires no special skill, just the ability to type. Article references to be stored can be preselected and then entered into the system by a secretary or assistant. Even the contents of entire file cabinets can be indexed and stored!

FIND IT QUICK requires a minimum system consisting of TRS-80 Model I, Level II with 16K of RAM, an Expansion Interface and at least one disk drive. Up to 1080 items can be stored on one data disk. Instant Software's Tiny DOS operating system is included on the program disk so that you can use the program without fuss or bother.

Make your computer work like your own personal librarian with FIND IT QUICK. Can you afford to wait another day?

0258RD (Disk) $49.95

THE WORDSLINGER

At last. . . simplified letter writing! The Wordslinger is an economical program designed for the individual user or small business. The program has automatic formatting for letterhead stationery and envelope addressing. You can write and edit letters, reports, forms, even schoolwork! Text is stored on cassette tapes for fast, inexpensive retrieval. Once you've used the Wordslinger, you won't want to go back to your typewriter.

One of the Wordslinger's major advantages is that you don't need disk drives or an Expansion Interface. the minimum system required is the TRS-80 Model I, Level II with 16K of RAM and an appropriate printer (you may need an Expansion Interface, if your printer can't operate with just a printer interface cable, such as Radio Shack's part #26-1411 or 26-1416).

0129R (Tape) $29.95

CLIMATE-COMP

Come rain or come shine, you'll love being "in the know" with Climate-Comp, our two-program weather package.

WEATHER FORECASTER—Gives you a short-range forecast of upcoming weather in your local area. Just enter past and present barometric pressure to obtain a "preview" of your weather for the next 24 hours.

WEATHER PLOT—Provides you with accurate, average weather data for most of the major cities of the United States. This program includes data files for the Continental U.S.

This is a chance for all amateur meteorologists to generate useful forecasts and weather data. Model I, Level II, 16K.

0316RD (Disk) $24.95

MAIL/FILE

from Galactic Software Ltd.
A Mailing List for the TRS-80
Model I or Model II

Instant Software always tries to provide you with the best software on the market. Although the Mail/File mailing list program is not published by us, it is so good that we want you to try it.

We have two versions of this mailing list. Pkg. 5000RD is for the Model I with the 5-inch disk drive and Pkg. 5001RD is for the Model II with the 8-inch disk drive. The programs are essentially identical except for the storage media and their respective capacities. With the 5-inch drive, you can store up to 600 names per disk without DOS, or 300 names per disk with DOS. With the 8-inch drive, you can store up to 2500 names per disk, with or without DOS. (If your list is larger than the single disk maximum, it can be distributed over several disks.)

The program maintains separate alphabetical and ZIP code files under constant sort. When you add a name to your list it will be inserted into its correct position in the files. You will never have to sort your list; it will always be ready to print labels.

The program will record your data in nine fields: two for NAME, and one each for ADDRESS, CITY, STATE, ZIP CODE, PHONE NUMBER, PHONE EXTENSION, and a five-character CODE field. When you print labels, you have a choice of three different label formats: a three-line label, a four-line label or a user-defined label. In the three-line and user-defined label formats, you may include a message line on your label.

The best feature of this program is the sort process that lets you determine which labels will be printed. You may specify either alphabetical or ZIP code order for all or any part of your list. For example, you can print labels for everyone on your list whose name begins with the letter A, or for all of those people who have the same ZIP code. You can even print labels for only those people named Jones, who are living in a given city or state. (Note: The Model II version can search for both first and last names, e.g., John Jones.) Furthermore, you can choose to print labels by using any single field (i.e., specific cities, states, phone numbers, etc.). You may assign specific codes to any name in the CODE field. For example, ACT could stand for active accounts, and INACT for inactive accounts. If you wanted to send a letter to all of your inactive accounts, you would specify the CODE INACT, and labels would be printed only for your inactive accounts. When you print labels, you may specify up to nine different CODES at one time. If your data matches any one of the CODES a label will be printed.

Files created with the Model I version of this program can be transferred to the Model II version, when you upgrade your hardware.

Package 5000RD requires the following minimum system:

1. A TRS-80 Model I, Level II with 16K RAM.
2. An Expansion Interface with 16K RAM (or more).
3. One (or more) mini-disk drives.
4. A compatible printer (80 or 132 columns).
5. TRSDOS Version 2.3.

5000RD Model I (Disk) $99.00
5001RD Model II (Disk) $199.00

BUSINESS ANALYSIS

(Formerly Oracle-80)

Business Analysis will provide you with analytical and forecasting capabilities previously available only in large computer and time-sharing systems. It is a flexible, professional time series package that can be used in sales analysis and forecasting, product planning, business planning, etc. The professional forecaster will recognize BUSINESS ANALYSIS as a tool which incorporates all the power of the X-11 model and extends it. Investors can analyze stocks, market trends and growth rates. Financial managers and economists can analyze the general economic climate and investigate business cycles. Even families can find this program useful in analyzing spending or energy consumption trends, for it is ideal for anyone who needs to analyze and forecast monthly, quarterly or annual data.

Even though this package uses advanced statistical methods, you don't have to understand higher math to use it. Designed to be used and understood by the typical business person, its powerful analytical capabilities will satisfy even the professional forecaster. All inputs and outputs are written in plain English and the documentation carefully explains all functions.

Adding, deleting and modifying data is accomplished with a very flexible editing routine. Automatic scaling of numbers, ability to choose from several output formats, a calendar format that identifies all your data by month/quarter/year, and the ability to add and subtract values while inputting from the keyboard provide added user flexibility. You can use several methods—moving average, rate of change, seasonal indices or cyclical indices—to analyze your data. The unique graphing capability lets you visualize your historic data or any of the modified data series you calculate, and direct any chart or graph to your printer.

Business Analysis will forecast future data values using trend, moving averages or seasonal methods. You may choose either constant unit-trend or a constant percentage-growth-trend forecast for even more flexibility.

This is one of the most powerful and useful business tools you will ever use. It puts the future in your hands.

The package requires the following minimum system:

1. A TRS-80 Level II with 16K RAM.
2. An Expansion Interface with 16K RAM.
3. One or more disk drives.
4. Any compatible Disk Operating System.
5. An optional line printer.

Note: Tape version *only* is Model III compatible.

0140R (Tape) $75.00
0152RD (Disk) $99.95

LABEL

Are you tired of trying to remember which line numbers of your program correspond to which routines in it?

Well, Instant Software has solved your problem! With the LABEL program in place, you can forget about such trivial things and rely on the names you have given the routines. You can use LABEL to keep track of these things for you and get down to serious programming.

LABEL allows you to use labels in place of line numbers in GOTOs, GOSUBs, IF. . . THENs, THEN. . . ELSEs, RESUMEs, ON ERRORs, ON X GOTOs, and ON X GOSUBs.

After you finish with your program, you can have it converted into a normal program (with line numbers in place of the labels in GOTOs, etc.) and execute it in computers without the LABEL program. With this conversion you have three options:

1. The labels are left at the beginning of the routines, but changed to line numbers in the program statements.
2. The labels at the beginning of the routines are changed to REM statements, and changed to line numbers in program statements.
3. The labels at the beginning of routines are deleted, and changed to line numbers when in program statements.

While you are using LABEL to write your program, you can have it list all the lines which have label identifiers (to either the video of the line-printer), you can search the program for the line number of a particular label, or you can use it to delete all the lines that begin with a REM.

LABEL is compatible with TRSDOS 2.1, 2.2, 2.3, and with NEWDOS and Percom MicroDOS. Model I, Level II, 16, 32 or 48K.

0168R (Tape) $24.95

ONE-D MAILING LIST

Here is a mailing list system that can be run on only one disk drive! You can have up to 17 fields of selection for name/address retrieval.

Disk versatility allows you to add, delete, or change the numerous details stored in the system.

Features of the One-D Mailing List include:

- Automatic name sort (with zip code option).
- Rapid access to any name on file.
- Easy error correction and recovery.
- Prints selective name listings.
- Revise or update listings at any time.
- Up to 2500 names on-line (with 4 drives).
- Prints a list of all names on file.
- Prints mailing labels.

This package requires the following minimum system:

1. A TRS-80 Model I, Level II with 16K RAM.
2. An Expansion Interface with 0 to 32K memory.
3. A single disk drive (with automatic upgrade for up to three additional disk drives).
4. Any compatible Disk Operating System.

0123RD (Disk) $24.95

USER'S TIPS

When writing BASIC programs, place the most often used routines at the front of the program. This will enable the program to run faster.

Utilities for your TRS-80

DYNAMIC DEVICE DRIVERS

Are you tired of working around all of the little "obstacles" that are built into your TRS-80? Ever wish that there was some way to "repair" those imperfections?

Well, here it is! The DYNAMIC DEVICE DRIVERS package has all of these features:

PROGRAMMABLE KEY DEBOUNCE—Your keyboard can be "tuned" to your typing style.

PROGRAMMABLE REPEATING KEY FUNCTION—Every key has a repeat function.

LOWERCASE MODIFICATION SUPPORT—You have choice of standard or shift-for-lowercase letters. (A lowercase hardware modification must be installed.)

BETTER THAN NOTHING GRAPHICS—Graphics characters will be converted to the closest ASCII character.

PRINTER/SCREEN AUTO SWITCHING—If your printer is accidentally turned off, your program won't bomb.

PROGRAMMABLE PRINTER FORMS CONTROL—You control the format for printer output.

PROGRAMMABLE KEYBOARD LOCK—Only you will know the secret code to unlock your keyboard.

With the Dynamic Device Drivers package, you can look forward to working WITH your TRS-80, instead of against it! Model I, Level II, 16K.

0228R (Tape) $19.95
0199RD (Disk) $24.95

DISK EDITOR

Disk Editor is a powerful machine-language utility program that will allow you total access to ANY byte of information in ANY sector of ANY track of your disks. It is a fast, simple, and efficient method of modifying files—whether BASIC programs, SYSTEM programs, or just data. All commands are readily accessible, with no need to refer to a command table.

With Disk Editor you can examine, alter, add, and delete information with ease. Information can be retrieved from the disk by supplying track and sector information, or by giving the filespec. You can even search the disk for a specific string of characters (up to 8 characters long).

If you need hardcopy, use the LINEPRINT command to send a copy of the video display to your lineprinter.

You can transfer command from Disk Editor to Radio Shack's DEBUG and back, allowing dynamic debugging of disk I/O procedures.

Disk Editor is compatible with TRS-DOS 2.1, 2.2, and 2.3, as well as with Apparat's NEWDOS. It is even capable of reading disks made by Percom's Micro-DOS.

There are two versions of Disk Editor: one is for a 35-track DOS, and the other is for a 40-track DOS. Both are included in this package.

This package requires the following minimum system:

1. A TRS-80 Model I, Level II, 16K RAM.
2. An Expansion Interface.
3. A single Disk Drive.
4. Any compatible Disk Operating System.

(Disk Editor is not compatible with VTOS 3.0).

0180RD (Disk) $39.95

MASTER DIRECTORY

Wasn't it just yesterday you threw the cat into the washing machine because you were frustrated at not being able to find that Adventure that you saved on disk last week? Or, was it just yesterday that you force-fed that leftover hot tamale sauce to your mother-in-law because you had searched in vain for your backup copy of the Electric Pencil? And you say your three-year-old built a house of cards out of your carefully-filed business data disks and now you don't know which is which? Well, chin up, Bucky, 'cause MASTER DIRECTORY is here!!

The MASTER DIRECTORY is a disk file storage program that reads the files on all your disks, stores the file names and extensions, and even records the free space on each disk. All you have to do is number the disks in your library and the MASTER DIRECTORY will keep track of their contents. You can read the names, displayed alphabetically, search the DIRECTORY for file names and extensions, delete disks and search for free space. You can store 5000 files or 320 disks, whichever comes first.

Your disk storage problems are over now that the MASTER DIRECTORY is here. Model I, Level II, 16K RAM, Expansion Interface plus 1 disk drive.

5005RD (Disk) $29.95

DISK SCOPE

Need to check out a disk? Perhaps you want to see how the files are stored, or you forgot your password. No problem! Now you've got Disk Scope!

If you know the name of the file, the Fileloc program will show you what tracks and sectors on the disk contain that file, as well as how much memory the file takes when loaded into RAM. This works for both program and data files. Fileloc then allows you to print the information, restart the program or exit to BASIC. The information obtained allows you to use the CDISK program effectively.

CDISK is a powerful little BASIC utility and test program. It will allow you to view any track and sector on your disks in ASCII, Hex and screen POKEs. It totally disregards protection codes. It can also be used to randomly check all 350 sectors of your disk for read errors.

You don't know the whole file name if you haven't got the password, so the PASSWORD program has been included in the Disk Scope package. This machine-language utility not only gives you a password for files, but for the whole disks as well. Whether you're a novice or a pro, if you use a disk system, you need Disk Scope.

This package requires the following minimum system:

1. A TRS-80 Model I, Level II, 16K RAM.
2. An Expansion Interface.
3. A single disk drive.
4. Any compatible Disk Operating System.

0139RD (Disk) $19.95

TLDIS & DLDIS

You've bought a super machine-code program, but now wonder how it works. Maybe you even used a quick PEEK routine to glance through it when it was in memory. If so, you definitely noticed the complete lack of comments in the code, making it almost impossible for you to decipher and understand it.

Well, Instant Software's Labeling Disassemblers are the answer to your problem.

TLDIS (Tape-based Labeling DISassembler) and DLDIS (Disk-based Labeling DISassembler) make three passes to assign labels (where appropriate) to the routines in a machine-language program. Their output is almost identical to that of a hand-assembled source code.

You can send the disassembly to a lineprinter (Radio Shack parallel port) for either TLDIS or DLDIS. (The difference between these utilities is the storage mode of the disassembly.)

TLDIS can send the disassembly to cassette tape, DLDIS can send it to disk; both send it to the video monitor. The stored disassembly from TLDIS may be reassembled with Radio Shack's EDTASM™—the disassembly from DLDIS, with Apparat's extension of EDTASM™. Because of the use of labels, it is a simple matter to change any object code program by disassembling it and then making changes to the resultant source code, without losing track of jump/load addresses. Labels start with "AA00" and increment up, in even numbered steps (AA02, AA04, etc.). The odd numbers (AA01, AA03, etc.) are left for you to use for the source code during reassembly.

The printing of the disassembly may be temporarily halted by using ((SHIFT)) @ (just as in BASIC) or it may be ended by pressing the ((BREAK)) key. It also has a comments column to display ASCII characters used in a LD or CP opcode.

TLDIS and DLDIS may be relocated in memory to avoid conflict with the program you disassemble.

The next time you need to "climb inside" a machine-code program, take DLDIS or TLDIS with you. We promise that it will be an easier journey. Model I, Level II, 16K; Model III, 16K.

0230R (TLDIS-Tape) $14.95
0231RD (DLDIS-Disk) $19.95

RENUM/COMPRESS

Add a new command to your TRS-80! RENUM/COMPRESS fits in the top 767 bytes of memory and you call it like any BASIC command.

You forgot an INKEY$ routine and there's no space for a new line number? The command allows you to renumber your BASIC program. You can set the new starting line number and the increment or (like any good command) it will default automatically to standard values.

You think memory is going to be tight? This command allows you to compress the program instantly by removing REM statements and any spaces outside of quotes!

If you're a BASIC programmer, this is the one utility you should have loaded and ready to go! Add the command: RENUM/COMPRESS! Note: Not disk compatible. Model I, Level II, 16K.

0133R (Tape) $14.95

IRV

IRV, one of the most powerful utility programs available, turns your keyboard into a SUPERKEYBOARD.

Now you can have single key programming. IRV comes complete with its own keyboard definitions, or up to 255 characters can be assigned to every key, including (ENTER) and (BREAK). You can enter often-used BASIC words, variable names or even entire lines. Even functions, such as RUN, LIST, or EDIT can be entered with a single keystroke.

The relocate feature of IRV is unique, in that it allows single line relocation and renumbering. You can merge lines using the EDIT function and a single keystroke.

As a video editor, IRV is so powerful, you'll wonder how you got along without it. Full cursor control, blinking cursor, block movement and special erase functions are just the beginning. Frequently used video graphics blocks can be saved and used again and again. Even IRV's minor virtues are impressive. You can have auto repeat with any key, including programmed functions. You won't have to pull plugs or fiddle with a control box to rewind or fast-forward a tape. The cassette recorder can be controlled from the keyboard.

If you are a creative programmer (or wish to be), you need the power and convenience of IRV! Tape version, Model I, Level II, 16K. Disk version, Model I, Level II, 16K, Expansion Interface plus two disk drives.

0250R (Tape) $24.95
0350RD (Disk) $29.95

COMPRESSION UTILITY PACK

Do you want to add sound routines to a Space Trek program that already uses 16K? Or maybe you need an extra column in that financial report program, but when you run it, you get OM errors?

With a wave of your hand, and a little help from either of the COMPRESSION programs in this package, your problems are over.

COMPRESS-80—Fits in 265 bytes, deletes spaces, and offers the choice of leaving REMark line numbers in the program, or deleting them altogether.

SUPERCOMPRESS—Uses 767 bytes and can do everything COMPRESS-80 can do PLUS, it packs the program into the smallest possible number of multiple statement lines.

With the Compression Utility Pack and your own programming skill, you can add all those little extras to your BASIC programs. Model I, Level II, 16K.

0246R (Tape) $19.95

THE DISASSEMBLER

This is a single-pass, hex-notation disassembler that will send its output either to tape or to a lineprinter (Radio Shack parallel port). The tape output is directly compatible with Tandy's EDTASM™. Thus, you can take an object code tape, disassemble and output it to tape, then use EDTASM™ to add, delete, change and even re-assemble your new version.

In addition, it shows the displacement and absolute address of any relative jumps made by the disassembled program. It also displays any ASCII characters used in an LD or CP opcode.

Since the Disassembler works only on in-memory programs, it has been made relocatable so that you may move it around in memory to avoid conflict with the program you wish to disassemble. As an added option, you may also jump to memory locations and transfer control between Disassembler and other utility programs in your computer.

The Disassembler—use it to examine and analyze any resident machine-code program! Model I, Level II, 16K; Model III, 16K.

0232R (Tape) $9.95

Education for your TRS-80

PROGRAMMER'S PRIMER

Sometimes thousands of words can't give a clear picture of complicated theories and concepts. There are times when a chart or picture is worth those kilo-words, and more! The novice computer programmer is offered this breath of fresh air in an otherwise stuffy situation—the PROGRAMMER'S PRIMER.

This program functions as a Computer Aided Instruction package to introduce the novice programmer to several important computer concepts. It graphically explains the relationship of decimals to hexadecimals in the Decimal:Hexadecimal Conversion routines. Data storage is explained with the Subscription of a Variable and Three Dimensional Array routines. The logic of program flow is shown in the For-Next Loops flow chart demonstration. Finally, the Bubble Sort sorting technique is shown, in a fascinating display of what a computer does best.

Don't be a "scratch your head in wonder" boy any longer. Let the PROGRAMMER'S PRIMER package from Instant Software help you on your way down the primrose path of programming! Model I, Level II, 16K; Model III, 16K.
0245R (Tape) $9.95

PROGRAMMER'S CONVERTER

This package contains three programs that can aid you in converting to other number-base systems.
BASE CALCULATOR—Turn your TRS-80 into a calculator. Convert numbers to any base from 2 through 16 and perform calculations in that base. Memory, sign change, one's and two's complement are all available. It will even handle fractions.
HEXADECIMAL/DECIMAL CONVERSION TRAINING—A dual-purpose program. First, it's a handy converter that changes decimal numbers to hexadecimal notation (and vice versa). Second, it's a teaching/testing program that gives you practice in making those conversions yourself.
NUMBER BASE CONVERSIONS—Converts any decimal, binary, octal, or hexadecimal number (up to $FFFF), to its equivalent value in the other three bases and displays all four values simultaneously.

You'll have a command of octal and hex notation—instantly! Model I, Level II, 16K; Model III, 16K.
0058R (Tape) $9.95

ULTRA-MON

ULTRA-MON is a unique and powerful machine-language monitor. It is ROM independent and will function in Level II or DOS BASIC. With ULTRA-MON, you will be able to write, modify, study and debug machine-language programs. Plus, you'll be able to avoid the frustration and "bomb-outs" usually associated with machine-language programming. ULTRA-MON displays, disassembles, traces (hardcopy trace disassembly, too!), modifies, relocates memory, prints and even relocates itself with simple commands. Using interpretive execution, ULTRA-MON allows you to put breakpoints in ROM. This powerful monitor can even fetch, decode, disassemble and analyze each instruction individually so that your program cannot bomb out.

ULTRA-MON is designed for the beginning machine-language programmer as well as the professional. The documentation contains a Simple Demonstration section geared to the novice. Consequently, the program is a learning device as well as an extremely useful programming tool.

If you are serious about programming, you need to add this powerful utility to your library today. Model I, Level II, 16K.
5003R (Tape) $24.95

TEACHER'S AIDE

Now you can have the benefits of Computer Assisted Instruction (CAI) in your own home. The Teacher's Aide program will let you create a teaching system for any conceivable subject. The program allows you to create a question and answer lesson (you can input up to 8000 characters per lesson). You can then save this lesson on the disk and create an entire sequence of lessons.

Your lessons can be tailor-made for you or your students. The options available are: (1) Review the material prior to taking the lesson, (2) provide hints to help answer questions, and (3) offer a graphics display as a reward for correctly answering all the questions. The Teacher's Aide program will even allow for spelling errors!

Teacher's Aide is perfect for parents, teachers, and students who need the unlimited patience and undivided attention that only a computer can provide. Readin', Ritin', and 'Rithmetic will never be the same—now that you have the Teacher's Aide package from Instant Software. Model I, Level II, 16K, Expansion Interface + one disk drive. Recommended for teachers and parents.
0214RD (Disk) $39.95

**RECENTLY REVISED—
NOW BETTER
THAN EVER!!!**

TYPING TEACHER

A complete seven-part package that guides you from familiarization of the keyboard through typing words and phrases to mastery of touch-typing. Your video monitor becomes a bottomless page for typing practice, and your own private TYPING TEACHER, ready to teach when you're ready to learn. Model I, Level II, 16K; Model III, 16K.
0099R (Tape) $12.95

GEOGRAPHY EXPLORER: USA

The Geography Explorer: USA package is the most fascinating (and least painful) way of learning social studies that we've seen yet.

The program displays computer-generated maps of the United States, its seven regions, and its individual states. The student then answers questions about the states of a given region. These questions may be in Multiple Choice, Recognition or Fill-in format.

You can learn each state's name, capital, largest city, nickname, population, population rank, population density and percentage in urban areas, the state's flower, bird, tree, song and motto! Furthermore, Geography Explorer has a unique TEACHER mode which allows the teacher or parent to choose multiple options of how the material is to be presented. This permits directed learning for the student.

The package is also just plain fun! When the student answers the questions correctly, there are a variety of graphic rewards that flash on the screen to provide immediate, positive reinforcement.

As a bonus, this educational package can use a compatible Light Pen! The Light Pen can be used to respond to the MENU or to answer multiple-choice questions; simply point to the flashing square of your choice. (If you do not have a Light Pen, or you prefer not to use it, all responses can be entered from the keyboard.)

The union of sophisticated software with the speed and novelty of the Light Pen represents a milestone in Computer Assisted Instruction. As advanced as this package is, it can be readily used by grade-school students. Truly, it is educational software for any age.

We think you'll be equally impressed once you've explored these United States with the Geography Explorer. Model I, Level II, 16K, Expansion Interface + one disk drive. Light pen optional. Recommended age level six to adult
0071RD (Disk) $49.95

ULTRA-MON AND MODEL III

Ultra-Mon is compatible with the Model III computer with this short lineprint patch. Enter the listed code at the designated address. The program must be located at its original loading address, 6E00.

7BBE	E5	PUSH HL	;SAVE HL
7BBF	F5	PUSH AF	;SAVE AF
7BC0	DB FB	IN A, (0F8H)	;GET PRINTER STATUS
7BC2	CB 6F	BIT 5,A	;PRINTER SELECTED?
7BC4	28 04	JR Z, 7BCAH	;SKIP IF NOT
7BC6	CB 7F	BIT 7,A	;PRINTER BUSY?
7BC8	20 F6	JR NZ, 7BC0H	;LOOP IF IT IS
7BCA	F1	POP AF	;GET AF
7BCB	D3 F8	OUT (0F8H),A	;SEND A TO PRINTER
7BCD	E1	POP HL	;RESTORE HL
7BCE	C9	RET	;RETURN
7BCF	00	NOP	
7BD0	00	NOP	

To enter the program from Ultra-Mon's register display mode, type the following sequence (s denotes space bar): MM7BBE s E5 s F5 s DB s F8 s CB s 6F s 28 s 04 s CB s 7F s 20 s F6 s F1 s D3 s F8 s E1 s C9 s 00 s 00 (enter).

Your printer will now operate normally. Please note that this patch does not check to see if there is paper in the printer.

In addition, the documentation describes 6CC as the way to go to basic. Model III basic is more effectively entered from location 1A19. Where the documentation suggests the command E6CC,418E (enter), you should type E1A19,418E.

SURVEYOR'S APPRENTICE

This program makes applied math fun! Actually, we use the concept of area every day. You buy land by the acre, rent office space by the foot, and buy advertising space by the column-inch. The Surveyor's Apprentice can teach you the formula used to find the area of any plane figure.

Programs cover rectangles, circles, triangles, parallelograms (including both rhomboids and rhombuses) trapezoids, and polygons. This three-part package displays on-screen diagrams of all the figures, gives the formula for the area of each figure, and even gives you examples of computations. You can even request a quiz to see how well you're doing.

Here's an excellent refresher course for people who use calculations in their work. Why not broaden your area of expertise with the Surveyor's Apprentice! Model I, Level II, 16K; Model III, 16K. Recommended age level Jr. High to adult.
0127R (Tape) $9.95

BEGINNER'S RUSSIAN

In order to understand a foreign culture, you must know its language. In today's international politics and commerce, one of the most valuable languages to know is Russian. The Beginner's Russian package can put you well on the road to learning this vital language.

The three programs in this package will give you on-screen displays of the Cyrillic letters, detailed instructions on their proper pronunciation, and exercises that will have you recognizing and speaking simple Russian words.

This package is ideal for students, businessmen, scientists—perfect for anyone who is interested in learning the Russian language. Model I, Level II, 16K; Model III, 16K. Recommended age level 10 to adult.
0136R (Tape) $9.95

EVERYDAY RUSSIAN

This package is the second in Instant Software's Russian language series. Everyday Russian will acquaint you with the words for various foods, places to eat, signs, and the names of stores—exactly what a traveller needs to know. You'll also learn the order of the Cyrillic alphabet.

Each of the three parts in this package will not only teach you the words but will also present you with a quiz as well. Just pick the words you want to work on. The computer will score how well you've learned your lesson.

You can even practice typing in Russian. The program will allow you to type in the letters, or words, using the complete Cyrillic alphabet. Practice writing words such as hotel names, tourist attractions, and street addresses.

Why be deaf and mute when faced with the richness of the Russian language? Discover it for yourself with the Everyday Russian package. Model I, Level II, 16K; Model III, 16K. Recommended age level 10 to adult.
0137R (Tape) $9.95

RUSSIAN DISK

Beginner's Russian and Everyday Russian are available together on disk. Model I, Level II, 16K, Expansion Interface + one disk drive; Model III, 16K.
0212RD (Disk) $24.95

Games, Simulations and Entertainment for your TRS-80

AIR FLIGHT SIMULATION

Air Fright might be more like it! Instrument takeoffs and landings are no picnic—ask any pilot—and this computer simulation is certain to keep you on the edge of your seat.

You'll feel the adrenalin flowing, which accounts for the great popularity of flight simulation. Here's a program that provides a real sense of accomplishment, as you progress from takeoffs to tailspins and from landings to loops.

You begin with a full tank of gas and a flight plan that calls for a simple takeoff and landing—at least until you get the hang of it. Pay close attention to your instrument panel, especially the angle of ascent/bank indicator and air speed indicator—too steep a bank and your air speed will drop like a stone and so will your plane.

It's about as close to the real thing as you can get this side of a runway; and once you get some flight time under your belt, the sky's the limit. You can use the program to fly a course against a map—even try your hand at acrobatic maneuvers! Fun for the whole family. Model I, Level I; Model I, Level II, 16K; Model III, 16K.

0017R (Tape) $9.95

ALIEN ATTACK FORCE

(Formerly Invaders)

The invaders are coming! Earth's defenses are dead except for your Laser base. Your assignment is to destroy the approaching ALIEN ATTACK FORCE before it destroys Earth. Before Earth's sensors failed, they detected 550 armed invaders in space, speeding toward us in 10 attack formations of 55 in each group. The sensors detected four different types of attack craft: Large, Medium, Small, and a short profile craft which is the most difficult to destroy. If you cannot stop these space attackers they will stop Earth . . . for good. Model I, Level II, 16K; Model III, 16K.

0240R (Tape) $9.95

BALL TURRET GUNNER WITH SOUND

For years the Petro Resource Conglomerate has attacked our photon collection stations and strangled our deep-space trade routes. The PRC Exxonerator Class light fighters (code name: Gnat) have been their main weapon. Now you can strike back, by joining the Ball Turret Gunner Service.

Imagine yourself at the control console of an LW-1417 Stratoblazer (Type B Strategic Laser Weapon). Your Hindsight Director informs you that a Gnat fighter is coming in for an attack. You pivot your gigawatt laser turret until you can see the target on your monitor. The Range Indicator shows him coming in fast. The Targeting Computer studies his course and speed as your finger tenses over the firing key. You know you'll have only a fraction of a second in which to react. The Gnat fighter's evasive maneuvers cause him to dance in your sights. Suddenly, you see the FIRE command and you react instinctively. Your laser beam lashes out and reduces the Gnat to an expanding ball of ionized gas. Mission accomplished!

Ball Turret Gunner, with your choice of multiple levels of difficulty, optional sound effects and superb graphics, is more than just a game. It's an adventure. Experience it! Model I, Level II, 16K.

0051R (Tape) $9.95

JET FIGHTER PILOT

The Jet Fighter Pilot package takes you as close to real combat flying as possible . . . without pulling G's.

In this brilliantly realistic simulation, you become the pilot of a high performance, twin turbo-jet fighter. Total control of the aircraft is yours.

At the start of your mission, you'll go through an entire engine start procedure before your flight (provided your ground maintenance is up to par). Your takeoff will be from either the deck of an aircraft carrier (via a steam catapult) or from an airfield.

All controls respond the same as they would on a real jet fighter. You'll have to constantly monitor your display and make adjustments to your throttle, flaps, rudder and air spoilers. You decide when to retract flaps, landing gear and release the auxiliary fuel drop-tanks.

Your on-board navigational computer will direct you to your selected airport. The Glideslope/Localizer information will aid you in approaching and landing on an aircraft carrier deck or airfield.

The Weapons Control Computer will arm your missiles, provide you with the range and bearing to a target, and tell you when to attack. And, if things should get a little too hot, you have an ejection seat command for egress.

For a carrier-based landing, you'll have to deploy your tail hook. For a land-based landing, you'll need reverse thrust and to deploy your drag 'chute.

After you've flown a few missions with the Jet Fighter Pilot package, you'll know you've earned your wings. Model I, Level II, 16K; Model III, 16K.

0159R (Tape) $14.95

SWAMP WAR

Your (formerly) trusty ship, The Stellar Spaniel, has lumbered its last lightyear. An unfortunate encounter with a neutron star has damaged your hyperspace drive. you are stranded on a water-covered planet. Fortunately, you have managed to make it to one of the nine adjacent islands that compose this world's only dry land.

On each island, you discover the remains of an earlier landing party. Apparently, they were wiped out before they could finish building their supply transporters. A brief inspection reveals that each may safely transport you as far as one of the other islands.

There is evidence of all sorts of reptilian life—some small, some not so small, and some INTELLIGENT! You have an uneasy feeling that your sojourn here may not be entirely without incident . . . This feeling is transformed to terror when you find yourself suddenly surrounded by sapient swamp creatures shooting at you.

Your only hope is to use your maintenance 'droids to collect all the transporters and to clear the islands of the swamp creatures. Duck, dodge, shoot back . . . do what you must—but survive the Swamp War!!

The Swamp War program includes variable levels of difficulty, automatic scorekeeping and fast-paced graphics with optional sound effects. Model I, Level II, 16K; Model III, 16K.

0312R (Tape) $14.95

DANGER IN ORBIT

(Formerly Asteroid)

DATE: 28.02.2047

LOCATION: 270 million miles from Terra

MISSION: Maintaining Terra's Space Lanes

Briefing will follow:

1.1 Your mission is to destroy any asteroids in your sector and to prevent alien spacecraft from infiltrating the Terran Defense Network.

1.2 Your ship is armed with an anti-matter cannon. You can shoot large asteroids, but this turns them into many smaller asteroids, each capable of destroying your ship.

1.3 In addition, alien ships can make instantaneous hyperspace jumps into your area and start firing on your ship.

1.4 You'll need lightning reflexes and nerves of steel to survive DANGER IN ORBIT. We have no use for non-survivors!

DANGER IN ORBIT, a real-time, machine-language game, features variable levels of difficulty, superb high-speed graphics, sound effects and automatic scorekeeping. Tape version: Model I, Level II, 16K; Model III, 16K. Disk version: Model I, Level II, 16K, Expansion Interface + one disk drive.

0237R (Tape) $14.95
0247RD (Disk) $19.95

AIRMAIL PILOT

Picture the Pony Express with wings—that's Airmail Pilot! In place of hostile tribes and highwaymen, the pioneers of the airmail routes faced other dangers. They fought treacherous winds and thermals, sudden storms, and a constant battle with time. All of this to prove early airmail service was dependable and worthwhile.

Here's a program that lets you go back in time to the early days of aviation. You must fly the mail from Columbus to Chicago. Your Jenny, a cloth-covered biplane, must take you through unpredictable winds and electrical storms. The onboard clock will time your flight. You must get the mail through in the shortest time possible.

It's more than a game. You become totally involved in your mission. Not only is your reputation as pilot at stake, the future of Airmail—the bold new arm of the postal service—rides with you. Model I, Level II, 16K; Model III, 16K.

0106R (Tape) $9.95

DR. CHIPS

When the problems of your world begin to pile up, you can now turn to your own livingroom "psychiatrist"—DR. CHIPS.

The good Doctor is there whenever you need him for as long as you need him. He may not solve any of your problems but we're sure you'll be amused at his responses.

Why shoulder the burdens of modern life in silence? Put your TRS-80 and DR. CHIPS to work today and get it all off your chest! Model I, Level II, 16K; Model III, 16K.

0218R (Tape) $9.95

FLIGHT PATH

Experience all aspects of modern aviation with FLIGHT PATH.

MOUNTAIN PILOT—Transforms you into a daring bush pilot as you fly badly needed supplies to a remote gold mining camp. You must cross a hazardous mountain range, while struggling with headwinds, tricky navigation and diminishing fuel.

Watch your airspeed, altitude and rate-of-climb or you could stall and crash. If you deliver your supplies, you must then return over those mountains with a heavy cargo of gold bullion.

O'HARE—A control tower simulation in which you become an Air Traffic Controller. You are responsible for hundreds of human lives, as you guide the aircraft through your control sector to a safe landing.

You'll have to deal with different aircraft requirements, wind change warnings and potential midair collisions. But no matter what happens, you must bring in each of the 20 aircraft in your tour of duty.

PRECISION APPROACH RADAR—Combines the skills of pilot and Air Traffic Controller. You become the pilots' "eyes" as they try to land in limited visibility conditions. Your commands guide the aircraft in its approach to the field and a safe landing.

The flight path package involves you in both sides of flight procedure, from the thrill of flying to the tense drama of air traffic control. Model I, Level II, 16K; Model III, 16K.

0171R (Tape) $9.95

THE FLYING CIRCUS

The Flying Circus package covers the full range of flying from old time biplanes to modern day air traffic control. These seven programs offer you the daring realism of flight:

AIR FLIGHT SIMULATION—Allows you to learn and practice the essentials of piloting.

NIGHT FLIGHT—A nighttime photo reconnaissance mission.

AIRMAIL PILOT—Return to the early days of aviation, where it's just you and your aircraft against the elements.

MOUNTAIN PILOT—Fly to a remote mining camp with desperately needed supplies.

O'HARE—Transforms you into an air traffic controller.

PRECISION APPROACH RADAR—Requires the skills of both an air traffic controller and a pilot.

JET FIGHTER PILOT—Takes you as close to real combat flying as possible . . . without pulling G's. You become the pilot of a twin turbo-jet, supersonic fighter. Total control of the aircraft is yours from takeoffs to landings to launching your missiles and destroying the target.

Your dream of flying becomes a reality with The Flying Circus package. Model I, Level II, 16K, Expansion Interface 16K & one disk drive.

0205RD (Disk) $39.95

HOUSE OF THIRTY GABLES

Gold and treasures tempt us all, but only the stout of heart and swift of mind should attempt a trek through the legendary House of Thirty Gables. To be alert is to be successful against the strange array of creatures (and other terrors) waiting to end your quest for riches. Serpents and trolls are merely minor obstacles in this dungeon of mystery.

You make your way through these dank halls by typing two-word commands, such as GO WEST, READ SIGN, THROW AX, etc. You'll find various objects along your way—some of which you may need in order to solve the problems endemic to this enigmatic residence.

Lest you forget, there are heaps of treasure to be had for the adventurous and crafty explorer. However, the real reward is the pleasure you will have by outwitting the sinister inhabitants of the HOUSE OF THIRTY GABLES! Model I, Level II, 16K.

0219R (Tape) $9.95

COSMIC PATROL

Skilled players soon master many difficult computer games, but COSMIC PATROL is in a world all its own. The challenge intensifies! Supporting graphics and sound (optional) make each encounter an exciting new experience. It all adds up to a Super 3-S package . . . Skill, Sight and Sound.

Scenario: The COSMIC PATROL program puts you in the command chair of a small interstellar patrol craft. Your mission is to defend Terran space and prey on the Quelon supply ships which carry essential parts and lubricants for that implacably hostile robotic force. The drone freighters are fairly easy pickings for the accomplished starship pilot, but beware of the I-Fighter escorts. They're armed, fast and piloted by intelligent robots linked to battle computers. They never miss.

The Cosmic Patrol program is not just another search and destroy game. With its fast, real-time action, impressive sound option and superb graphics, this machine-language program is the best of its genre.

Don't keep putting quarter after quarter into arcade games or spending big bucks for video game cartridges. Get COSMIC PATROL from Instant Software—and get the best for less! Tape version Model I, Level II, 16K; Model III, 16K. Disk version Model I, Level II, 16K.

0223R (Tape) $14.95
0224RD (Disk) $19.95

BATTLEGROUND

Here is a program that should interest you war-gamers:

It is late 1944, and the Allied Forces are sweeping toward Berlin. As General of your sector, you have at your command tanks, planes, artillery, infantry, engineers and vehicles—an awesome array of fighting men and the machines of war. From intelligence reports you know that the enemy General is a shrewd tactician, not to be underestimated. It will take planning and strategy to outwit this wily old campaigner.

The battle map of your sector will fill with markers, each showing the deployment of your forces. You and another player will slip into the roles of opposing German and American commanders as yet another battle unfolds. Battleground allows you to experience the total responsibility of a battle-area command. It will be up to you to deploy your forces. On your shoulders rests the decision whether to call for direct artillery gunfire or to order your planes into the air. You will constantly be watching for an enemy airdrop, always carefully maneuvering your Forces.

The stark reality of World War II comes alive in BATTLEGROUND. Model I, Level II, 16K; Model III, 16K.

0141R (Tape) $9.95

NIGHT FLIGHT

Participate in one of the pivotal events of World War II—an awesome responsibility for you! Ready to tackle it?

It's May, 1941, the dreaded Axis battleship, the Bismarck, has broken out of the North Sea and is now somewhere in the North Atlantic. Your mission: make a nighttime photo reconnaissance flight over the Bismarck. You can help the Admiralty determine the extent of damage suffered by the Bismarck in a previous battle and whether the British fleet has a chance of sinking the German pocket battleship.

The Night Flight program lets you take off, fly and land a propeller-driven aircraft. Practice approaches and landings with a full on-screen display of the landing field information. The program will practically teach you to fly.

Somewhere out in the cold, gray North Atlantic, the Bismarck tries to elude her pursuers. Your photos are vital. Launch yourself into the night sky with the Night Flight package. Model I, Level II, 16K; Model III, 16K.

0117R (Tape) $9.95

THE ALL STARS

Here's a collection of our best selling games . . . the ones you'll want to play again and again. It includes: SANTA PARAVIA AND FIUMACCIO—Become the ruler of a medieval city state as you struggle to create a kingdom. Up to six players can compete to see who will become the King or Queen.

OIL TYCOON—Avoid oil spills, blowouts and dry wells as you battle to become the world's richest oil tycoon. Two players become the owners of competing oil companies . . . and there's room for only ONE at the top!

PARADISE TRADER—You're the captain of a trading schooner, sailing the Caribbean in search of profits. As you sail from island to island, beware the dangers of the deep: pirates, ghost ships and hurricanes. Good sailing, skipper!

MILLIONAIRE—Here's $1000. Can you turn it into a million dollars in fifteen (simulated) years? It depends on your strategy as you buy and sell properties, negotiate bank loans, collect rentals and accept sealed bids.

TIMBER BARON—An in-depth experience of the timber business, from dropping trees, until the milled lumber reaches the market. Your transactions are affected by those unexpected eventualities that can upset even the most careful plans.

BATTLEGROUND—European Theater, 1944: Your forces are sweeping toward Berlin. At your command are tanks, planes, artillery, infantry, engineers and vehicles. The battle map will fill with markers showing the development of your forces and the location of the enemy in one day of fighting. This two-player game will provide hours of entertainment. Model I, Level II, 16K, Expansion Interface + one disk drive.

0213RD (Disk) $34.95

SANTA PARAVIA AND FIUMACCIO

Buon giorno, Signore!

Welcome to the province of Santa Paravia. As your steward, I hope you will enjoy your reign here. I feel sure that you will find it, shall we say, profitable.

Perhaps I should acquaint you with our little domain. It is not a wealthy area, Signore, but riches and glory are possible for one who is aware of political realities. These realities include your serfs. They constantly request more food from your grain reserves, grain that could instead be sold for gold florins. And should your justice become a trifle harsh, they will flee to other lands.

Yet another concern is the weather. If it is good, so is the harvest. But the rats may eat much of our surplus and we have had years of drought when famine threatened our population.

Certainly, the administration of a growing city-state will require tax revenues. And where better to gather such funds than the local marketplaces and mills? You may find it necessary to increase customs duties or tax the incomes of the merchants and nobles. Whatever you do, there will be far-reaching consequences . . . and, perhaps, an elevation of your noble title.

Your standing will surely be enhanced by building a new palace or a magnificent *cattedrale*. You will do well to increase your landholdings, if you also equip a few units of soldiers. There is, alas, no small need for soldiery here, for the unscrupulous Baron Peppone may invade you at any time.

To measure your progress, the official cartographer will draw you a *mappa*. From it, you can see how much land you hold, how much of it is under the plow and how adequate your defenses are. We are unique, in that here, the map IS the territory.

I trust that I have been of help, Signore. I look forward to the day when I may address you as His Royal Highness, King of Santa Paravia. Buona fortuna—or, as you say, "Good luck".

For Model I, Level II, 16K.

0043R (tape) $9.95

PERFECT PONG

The last bastion of propriety has been breached. You, the staid, serious TRS-80 user are about to enter a world where no person can resist the lure, the excitement, the frenzy of the Arcade. Yes, you are about to meet—PERFECT PONG!

Instant Software presents Perfect Pong to add some zip and zest to your computer life. Enough of those math, bill paying and utility programs. It's time to add some fun to your life!

Perfect Pong is two programs offering nine different game boards and sound. Written in machine language, they are both fast and tricky. You control the paddles—but can you control your emotions in the middle of a rousing contest?

Why perpetuate the "stuffed shirt" image? Unleash the force of Perfect Pong and your computer will never by the same. Model I, Level II, 16K; Model III 16K.

0120R (Tape) $14.95

OIL TYCOON

What would it be like to be one of the world's biggest oil producers? You and your friends can find out with this action-packed simulation as you compete to become one of the oil industry's wealthiest tycoons.

Beginning with $2 million, and armed with information from geologists' reports, you'll explore for new wells, exploit existing wells, and invest heavily in research and development in an effort to make your oil the most competitive available.

Once you have oil to sell, you can name your price—but don't get too greedy, or you'll find the demand for your product dwindling till the prices become more attractive.

The game involves elements of both strategy and chance. Whether you wind up as one of the world's wealthiest men, or the bankrupt victim of too many oil spills, blowouts, and dry wells, you're sure to find Oil Tycoon both challenging and exciting. Model I, Level II, 16K; Model III, 16K.

0023R (Tape) $9.95

Home and Personal Programs for your TRS-80

QSL MANAGER

Did you remember to send a QSL card to the op you worked last week? Maybe you sent a QSL, but can't recall getting one in return. The QSL Manager program will help you set up a computerized log book for instant access to your records.

Make complete log entries which include: date, time, call sign, name, band, both the Sent and Received signal reports, the mode, QSL sent/received, and any remarks you may want to add.

No more fumbling with index cards during a QSO, because the QSL Manager has a built-in search function to locate and display information on any call sign in your records. You can even list all the QSO's for a particular date, time, band worked, mode or a specific signal report.

The program will automatically check to see how much memory you have in your system. If you have a two disk, 32K system, you may have a maximum of 1000 entries in your log (500 entries if you have a single disk system). A 48K system limits you to 1400 entries.

These limits are caused by the fact that you have only 350 sectors on a disk. Each entry used 1/4 of a sector, thus the 1400 entry limit.

The program has built-in editing features that help you keep your log book up-to-date.

There's also a command that lets you output your log entries to a printer for hard copy.

In the next QSO, knock their socks off with your infallible memory. Model I, Level II, 16K, Expansion Interface + one disk drive.

0151RD (Disk) $19.95

USER'S TIPS

Keep disk and tape cassettes away from magnetic fields (transformers, speakers, AC motors, magnets, etc.). Strong magnetic fields will destroy information on disks and tapes.

USER'S TIPS

When using a disk drive system, always power up peripherals (expansion interface, disk drives, printer, etc.) before you turn on the TRS-80 CPU/keyboard.

MUSIC MASTER

This quartet of programs lets you make beautiful music—or shall we say, YOUR kind of music.

MICRO ORGAN—A machine-language program that enables you to turn your TRS-80 into an electronic organ or harpsichord. You can choose from a range of four octaves and three voices.

KALEIDOPY—A combination kaleidoscope and player piano. Your computer creates a repeating pattern on the screen and then plays it as music. A special "messing around" section encourages you to modify and improvise the program.

COMPOSER—You generate "loose" or quasi-random music. You learn how to create music and how to control certain aspects of it. Also contains a "messing around" section.

KEYMANIA—It's Gametime! Up to four people can compete, as they try to remember and reproduce a random computer composition. It's a multi-level game, in which you set the tempo.

Bring harmony into your life with Music Master! Model I, Level II, 16K; Model III, 16K.

0084R (Tape) $9.95

Games, and Simulations Entertainment for your APPLE

PADDLE FUN

This new Apple disk package requires a steady eye and a quick hand at the game paddles! We've included four different games to challenge and amuse you. They include:

INVADERS—You must destroy an invading fleet of 55 flying saucers while dodging the carpet of bombs they drop. Keep a wary eye for the mother ship directing the incursion. Your bomb shelters will help you—for a while! Our version of a well-known arcade game. Requires Applesoft in ROM.

HOWITZER—This is a one- or two-person game in which you must fire upon another howitzer position. This program is written in HIGH-RESOLUTION graphics, using different terrain and wind conditions each round, to make this a demanding game. The difficulty level can be altered to suit the ability of the players. Requires Applesoft in ROM.

SPACE WARS—This program has three parts: (1) two flying saucers meet in laser combat—for two players; (2) two saucers compete to see which can shoot out the most stars—for two players; and (3) one saucer shoots the stars in order to get a higher rank—for one player only. Requires Applesoft.

GOLF—Whether you win or lose, you're bound to have fun on our 18-hole Apple golf course. Choose your club and your direction and hope to avoid the sandtraps. Losing too many strokes in the water hazards? You can always increase your handicap. Get off the tee and onto the green with Apple Golf. One of its nicest features is you'll never need to cancel a golf date due to rain. Requires Applesoft.

The minimum system requirement for this package is an Apple II or Apple II Plus computer with 32K of memory and one minidisk drive.

Disk-based version.

0163AD (Disk) $19.95

DOCTOR CHIPS

DR. CHIPS is a great icebreaker at cocktail parties. And if the need arose, he could entertain your early arriving guests while you finished dressing! Your friends will be asking you how the guy got into the computer.

Why shoulder the burdens of modern life in silence? Put your APPLE and DR. CHIPS to work today and get it all off your chest!

You'll need an Apple II (or II Plus), one disk drive, and 20K of RAM.

0254AD (Disk) $14.95

MIMIC

How good is your memory? Here's a chance to find out! In Mimic, your Apple will flash on the screen a sequence of figures on a 3×3 grid. You'll have to respond with exactly the same sequence. It sounds easy, but don't be fooled. There are five different, increasingly difficult versions of the game, including one that will keep going until you're ready to quit. Mimic is exciting, fast-paced, and challenging—a fun game for family and friends. You'll need Integer BASIC and at least 24K of RAM.

0025A (Tape) $9.95

APPLE FUN

We've taken five of our most popular programs and combined them into one tremendous package full of fun and excitement. This disk-based package now offers you these great games:

MIMIC—How good is your memory? Here's a chance to find out! Your Apple will display a sequence of figures on a 3×3 grid. You must respond with exactly the same sequence, within the time limit.

There are five different and increasingly difficult versions of the game, including one that will keep going indefinitely. Mimic is exciting, fast-paced and challenging—fun for all!

AIR FLIGHT SIMULATION—Your mission is to take off and land your aircraft without crashing. You're flying blind, on instruments only.

You start with a full tank of fuel, which gives you a maximum range of approximately 50 miles. The computer will constantly display updates of your air speed, compass heading and altitude. Your most important instrument is the Angle of Ascent/Bank Indicator. It will tell if the plane is climbing or descending, whether banking into a right or left turn.

After you've acquired a few hours flying time, you can try flying a circuit against a map or doing acrobatic maneuvers. With a little more flight time under your belt, the sky's the limit!

COLORMASTER—Test your powers of deduction as you try to guess the secret color code in this Mastermind-type game. There are two levels of difficulty, and three options of play to vary your games. Not only can you guess the computer's color code, but it will guess yours! It will also serve as referee in a game between two human opponents. Can you make and break the color code?

STARSHIP ATTACK—Your mission is to protect our orbiting food station satellites from destruction by an enemy starship. You must capture, destroy or drive off the attacking ship. If you fail, our planet is doomed.

TRILOGY—This fascinating contest of logic has its origins in the simple game of tic-tac-toe. The object of the game is to place three of your colors in a row into the delta-like, multi-level display. The rows may be horizontal, vertical, diagonal and wrapped around, through the "third dimension". Your Apple (or human opponent) will be trying to do the same, and there are many paths to victory. You can even have your Apple play against itself!

Minimum system requirements are an Apple II or Apple II Plus computer with 32K of memory and one minidisk drive. Mimic requires Applesoft in ROM, all others run in RAM or ROM Applesoft.

0161AD (Disk) $19.95

GOLF

Without leaving the comfort of your chair, you can enjoy a computerized 18 holes of golf with a complete choice of clubs and shooting angles. You need never cancel because of rain. One or two players can enjoy this game on the Apple, with Applesoft II and 20K.

0018A (Tape) $9.95

SKYBOMBERS II

Two countries, separated by The Big Green Mountain, are at war. Both nations are equipped with only one means of attack—SKYBOMBERS.

You and your opponent, each representing the nations at war, command opposing fleets of fighter-bombers armed with bombs and missiles. As enemy commanders, each of you has specific orders: Cross that mountain and bomb the enemy blockhouse into oblivion!

Flying over that innocent-looking mountain is not easy for either air force. The aircraft can fire missiles at each other; if that fails, they can ram each other. Sometimes, aircraft encounter falling bombs and are blown to pieces in flight. Desperate pilots can even crash into the enemy blockhouse.

Flight personnel are sometimes forced to parachute from badly damaged aircraft. As they float slowly to earth, they become helpless targets for the enemy to destroy in mid-air.

The sounds of battle are there to remind each commander of his grim responsibility.

Explosions are graphically displayed for both commanders. The scores for both countries are constantly updated at the bottom of the display screen.

Flying these missions develops into a gripping fascination. Air warfare becomes a vivid reality, as you both play the deadly game of Skybombers II.

Tape version requires 16K RAM & Applesoft or Integer Basic. Disk version requires 32K RAM & one disk drive. Both require an Apple II or Apple II plus and game paddles.

0183A (Tape) $9.95
0271AD (Disk) $19.95

SANTA PARAVIA AND FIUMACCIO

Buon giorno, Signore!

Welcome to the province of Santa Paravia. As your steward, I hope you will enjoy your reign here. I feel sure that you will find it, shall we say, profitable.

Perhaps I should acquaint you with our little domain. It is not a wealthy area, Signore, but riches and glory are possible for one who is aware of political realities. These realities include your serfs. They constantly request more food from your grain reserves, grain that could instead be sold for gold florins. And should your justice become a trifle harsh, they will flee to other lands.

Yet another concern is the weather. If it is good, so is the harvest. But the rats may eat much of our surplus and we have had years of drought when famine threatened our population.

Certainly, the administration of a growing city-state will require tax revenues. And where better to gather such funds than the local marketplaces and mills? You may find it necessary to increase customs duties or tax the incomes of the merchants and nobles. Whatever you do, there will be far-reaching consequences. . . and, perhaps, an elevation of your noble title.

Your standing will surely be enhanced by building a new palace or a magnificent *cattedrale*. You will do well to increase your landholdings, if you also equip a few units of soldiers. There is, alas, no small need for soldiery here, for the unscrupulous Baron Peppone may invade you at any time.

For the 48K Apple with Applesoft in ROM.

0174A (Tape) $9.95
0229AD (Disk) $19.95

AIR FLIGHT SIMULATION

Your aircraft is on the runway loaded with fuel, instruments feeding the computer a constant stream of information.

A glance at your flight screen gives you airspeed, altitude, and compass heading. After you take off, the all-important Ascent/Descent-Turn/Bank Indicator will tell you the attitude of your aircraft at a glance, whether you are climbing or diving, whether you are banking into a left or right turn.

Your mission is a short one. You have a maximum possible range of about 50 miles, on one precious tank of fuel. Your objective is to take off, fly the aircraft, and land without crashing.

Since you may not have been at the controls of an aircraft before, the basic flight instructions enclosed will be invaluable. Included are explanations on basic aerodynamics and principles of flight, plus illustrations telling you how to recover from dangerous maneuvers.

Your aircraft will respond rapidly to the controls, and your movements must be delicate. Too much airspeed and your aircraft could explode from overstress. If the airspeed is too slow, you might stall and crash. A clumsy turn, and you might find yourself flying upside down, fighting to regain control.

With Air Flight Simulation and enough flight time, the sky's the limit!

Requires 16K of RAM & Applesoft.

0148A (Tape) $9.95

SAHARA WARRIORS

Now you can enjoy all the gritty realism of desert warfare with the Sahara Warriors package. COMMANDO—You must send your commandos to trap a German general and cut him off from his troops. You'll have the choice of two levels of difficulty.

FRENCH FOREIGN LEGION—The battalions of the French Foreign Legion are in a race with the Arabs. Which side will get its battalions into the oasis in the shortest time? You and a friend can find out. This game has four different versions, which even include an occasional sandstorm.

NOTE: Both programs in this package require an Apple with at least 8K and Integer BASIC. The French Foreign Legion program requires the use of game paddles.

0080A (Tape) $9.95

OIL TYCOON

Could you be the world's wealthiest oil magnate? Find out with this action-packed simulation of real-world, cutthroat competition.

Beginning with two million dollars, and armed with the geologist's reports, you'll explore for new wells, exploit existing wells and invest heavily in research and development.

Once you have oil to sell, you can name your own price. Don't get too greedy, or the demand for your oil may dwindle until prices become more attractive.

When the game is over, you may find yourself the wealthiest tycoon around—or you may be the bankrupt victim of too many oil spills, blowouts and dry wells. No matter how you fare, you're sure to find Oil Tycoon both challenging and exciting on your Apple. Requires Applesoft in ROM, 16K.

0079A (Tape) $9.95

for your HEATH

MENTAL GYMNASTICS

Pit your mind against the challenge of these ancient games.

REVERSI—As you and a friend (or the computer) place your pieces on the board, you must each try to capture the opponent's pieces. The score can fluctuate wildly, and nobody can tell who'll win until the last move.

WARI—You can play a friend or the computer in this simple yet intriguing game. The two players take turns removing pieces from one cup and placing them in the other cups. As play continues, the number of pieces decreases. The last player who has a piece to move wins the game. To enjoy these ageless games, you'll need the Heath H-8 with 8K above Benton Harbor Extended BASIC.

0087H (Tape) $9.95

Home/Personal for APPLE

SOLAR ENERGY FOR THE HOME

With the price of fuels rising to astronomical heights, solar space heating systems are starting to become very attractive. But, how do you know if a solar heating installation will be economically feasible? This program can answer that question.

All you do is supply the location, size, interior details and the amount of window space in your home. The program will calculate the current heat loss of your home and the amount of heat gained from any south-facing windows. Then, enter the data for the contemplated solar heating installation. The program will compute the net heating gain and give you the cost of using conventional fuels versus using solar heat. It will even calculate the payback period and show you if your contemplated investment in solar heating will save you money.

For anyone seriously considering the advantages and economics of a solar conversion, this package can provide practical guidance. For professionals designing and installing solar systems, it's a must.

Requires Applesoft BASIC, a single disk drive and a minimum of 28K of RAM.

0235AD (Disk) $34.95

ASTROLOGY

Astrology is at once a science and an art. The heavenly bodies are believed to exert an influence upon the destinies of each individual—a destiny predetermined by a universal order resulting from (or coinciding with) the position of the planets in our solar system.

The astrological chart, or horoscope, is drawn as a map of the heavens at the exact time of a person's birth. A horoscope is an illustration of (1) the "planets" and their location within (2) the twelve signs of the Zodiacal Circle and (3) the twelve houses through which the heavenly bodies must pass. The sign which is in the ascendant (Eastern Horizon) at the time of birth is also of great significance.

This program draws your astrological chart using either the tropical or sidereal zodiac. After you enter your birth data in response to the on-screen prompts, your Apple will calculate and then display your individual horoscope. Once the horoscope has been cast, the program will list the angular relationships (aspects) of all of the "planets".

Note: This program makes no attempt to interpret the multitude of facts and relationships of a specific horoscope. For this information, we recommend that you consult a standard text on astrology or an experienced astrologer.

Knowledge of Astrology a necessity to use this program.

Minimum System: Apple II (or II Plus), one disk drive, 28K of RAM and Applesoft.

0242AD (Disk) $19.95

for your NORTHSTAR

TRAFFIC ACCIDENT ANALYSIS AND RECONSTRUCTION
by Haim Reizes

Highway accidents, resulting in fatalities, injuries, and property damage, generally lead to prolonged litigation. The Traffic Accident Analysis and Reconstruction (TAAR) System will provide you with programs and analytical techniques to solve the equations involved in motor vehicle accidents. The System will present a reconstruction methodology utilizing physical evidence at the scene and a conclusive determination of the probable cause.

The purpose of this system is to provide accurate conclusions regarding the operational factors and dynamics contributing to highway accidents. The TAAR System deals with the basic calculations of the common types of accidents and also explains and applies the various formulas of automobile kinematics.

The TAAR System is the result of Mr. Reizes' 25 years in the accident investigation field. These programs are invaluable to insurance casualty companies, accident investigators, plaintiff and defense attorneys, fleet safety directors, and police officials handling automobile accident cases.

This package requires the North Star Horizon II microcomputer with 48K of memory, two disk drives and the North Star Disk Operating System (DOS) version 5.0.

0173ND (Disk) $499.95

Texas Instruments

SANTA PARAVIA AND FIUMACCIO

The year is AD 1400, and you are the ruler of a tiny Italian city-state. You are ambitious by nature and intend to build your little city-state into a powerful kingdom.

So begins Santa Paravia and Fiumaccio, where you and your fellow players compete as rulers of neighboring cities. You control the grain harvest, feed your people, set tax rates, exercise justice, and invest in public works.

Life was short back then, and you'll have only a limited amount of time in which to build your kingdom. The lives of your serfs will depend on your decision. If they are wise, then your city-state will grow and you will acquire loftier titles. If your rule is incompetent, your people will starve and your city-state may be invaded by your neighbors.

You can play the game yourself or up to six players may compete at one time. Either way, you're sure to find your route to the throne a challenging road.

How will you rule your kingdom? Will you become unscrupulous and follow the example set by Niccolo Machiavelli in his book on government, The Prince—or will you be a benevolent ruler—an iron fist in a velvet glove? Only you can answer that question—with the Santa Paravia and Fiumaccio program. For TI99/4 Microcomputers.

0273TI (Tape) $9.95

for your PET

CODE NAME: CIPHER

Now you can enjoy the feeling of intrigue with the CODE NAME: CIPHER software package. This enigmatic quartet of programs is a puzzle lover's delight. They include:

MEMORY GAME—The computer will shuffle a deck of cards and lay them out face down. You take turns choosing two cards and trying to match a pair. You can adjust the computer's "IQ" so that its memory is as fallible as yours.

CODEMASTER—One player types in a word, phrase or sentence and the PET translates it into a cryptogram. The other player must break the code and solve the cryptogram in the shortest time possible. The computer keeps score and will offer hints—but you lose points if you accept them.

DECEITFUL MASTERMIND—This isn't your ordinary mastermind type of game. You must guess the five letters in the hidden code word. The computer will give you hints as to how close your guesses are. But it can (optionally) give you false hints. . . up to nine per game, that can lead you astray. For one player.

CODE BREAKER—Cracking this code won't be easy. The computer (or a human opponent) will enter a sequence of three, four or five secret symbols. You'll have to guess which symbols are used and their correct sequence. The fewer guesses, the higher your score. If you want to pit your wits against the relentless logic of the computer, then Code Name: Cipher is for you. Requires an 8K PET.

0112P (Tape) $9.95

PET UTILITY I

This two-program package includes:

MONITR—You can edit, save and verify any machine-language program while still allowing you access to BASIC. It allows you to: (1) display or modify memory in hexadecimal notation, (2) disassemble memory, (3) execute programs, (4) save, verify or load machine language tapes, and (5) move back and forth between BASIC and MONITR. It automatically displays register contents and processor status bits whenever it encounters a BRK instruction. It resides in high memory, so it is compatible with BASIC. (Old ROM only.)

PROGRAMMER'S CALCULATOR—This program will do more than the Texas Instruments' Programmer Calculator. Not only will the program convert numbers into binary, octal, decimal and hexadecimal systems and function as a floating point calculator, but it will also display all four numbers, simultaneously. It will handle numbers so large, they're limited only by the size of your screen! (Either ROM.)

The PET Utility I package, a tool for all your tomorrows. For 8K PETs.

0105P (Tape) $9.95

DUNGEON OF DEATH

"Those who seek the Holy Grail
Must fight and walk a tortuous trail.
Eleven monsters thou must fight,
Whilst searching through eternal night."

Count Stan of Santa Paravia
A.D. 1380-1402

Thus begins your quest in search of the Holy Grail, wherein you must descend through the twelve levels of the Dungeon of Death, find the Holy Grail, and return to the surface with it.

The Holy Grail is guarded by Smaug, the most fearsome monster of all, and ten lesser breeds of monsters. These gruesome beings will challenge your every step. You can only survive by using all the powers at your command.

You'll be able to cast magic spells, drink potions that may or may not help you, and open chests in search of gold or items that will help you fight the monsters.

Step softly in the darkness. Treasure or sudden death is only a footfall away in the Dungeon of Death! Requires an 8K PET.

0064P (Tape) $9.95

QUBIC 4/GO MOKU

Play two ancient games on your modern PET. The two programs included are:

QUBIC 4—Play a multi-dimensional game of tic-tac-toe.

GO-MOKU—Line up five of your men while blocking the PET's moves.

These one player games require 8K of memory.

0038P (Tape) $9.95

MIMIC

Test your memory and reflexes with the five different versions of this game. You must match the sequence and location of signals displayed by your PET. This one-player program includes optional sound effects with the PET. 8K.

0039P (Tape) $9.95

SANTA PARAVIA AND FIUMACCIO

The year is AD 1400, and you are the ruler of a tiny Italian city-state. You are ambitious by nature and intend to build your little city-state into a powerful kingdom.

So begins Santa Paravia and Fiumaccio, where you and your fellow players compete as rulers of neighboring cities. You control the grain harvest, feed your people, set tax rates, exercise justice, and invest in public works.

Life was short back then, and you'll have only a limited amount of time in which to build your kingdom. The lives of your serfs will depend on your decision. If they are wise, then your city-state will grow and you will acquire loftier titles. If your rule is incompetent, your people will starve and your city-state may be invaded by your neighbors.

You can play the game yourself or up to six players may compete at one time. Either way, you're sure to find your route to the throne a challenging road.

How will you rule your kingdom? Will you become unscrupulous and follow the example set by Niccolo Machiavelli in his book on government, The Prince—or will you be a benevolent ruler—an iron fist in a velvet glove? Only you can answer that question—with the Santa Paravia and Fiumaccio program.

This classic program will give you hours of enjoyment. Your software library isn't complete without Santa Paravia and Fiumaccio. Requires a 16K PET.

0175P (Tape) $9.95

AIRMAIL PILOT
The Chicago Sun
July 18, 1922
—WANTED—
Airmail pilot for the Columbus to Chicago run. Must be willing to fly in every kind of weather. Only the foolhardy need apply.

Let the Airmail Pilot package take you to the early days of aviation history. Your plane is the Curtis JN4-D, affectionately known as the Jenny. You must fly the mail from Columbus to Chicago.

The Jenny carries only 26 gallons of fuel. You'll have to stop along the way. Bad weather may force you down. Electrical storms may turn your aircraft into a mass of flaming wreckage, or ice may form on your wings and plunge you to certain death below. But, the mail must get through.

Experience the thrills of flying, when aircraft were mere fragile machines of wood and fabric, with the Airmail Pilot package. (Scarf and flying helmet optional.) All you need is your TI99/4 microcomputer.

0274TI (Tape) $9.95

INDEX BY PROGRAM NAME

PRICES SUBJECT TO CHANGE WITHOUT NOTICE

A compilation of wisdom on when and when not to compile.

A Macroprocessor For Basic—Part V

J. Alan Olmstead
J. Olmstead Financial
Engineering Systems
3843 West Street, Moritz Lane
Phoenix, AZ 85023

The MetaBasic compiler consists of three program modules. The first pools all literals; the second generates Assembly language source code from the MetaBasic commands; and the third posts public names (if any) to the external reference file. Prior to assembly, but just after all program modules in the group have been precompiled, the external reference generator uses this file to compute linkage addresses among all the various concurrent program modules.

Literal pooling is a feature common to nearly all compilers. Numbers, such as zero and one, are used continuously throughout the program as are certain character constants. Since their repetition consumes considerable space in main memory, the compiler sets them up once as a data constant and thereafter substitutes the compiler-generated name whenever the same literal appears again in the user's source code. The maximum length of a character literal is 48 bytes, and the maximum magnitude of a numeric literal is 99,999,999.99.

The generation of source Assembly language from the MetaBasic commands is actually the easiest part of the entire compilation task. The generated code consists of a call followed by address definitions. The calls are to service routines contained within ZMonitor and are automatically defined as external to the user's program. The external reference generator will resolve them just prior to assembly.

If the Assembly language generator encounters DEFPUB annotation to any equate statement, the third compiler program is called automatically. It posts the name of the program module and the name which was declared to be public. The address is not computed at this time, however, on the presumption that changes to the program will be made. Similarly, if any reference is found to an external name, the program posts the module name as containing references needing to be resolved prior to assembly.

The Last Language: MetaBasic

MetaBasic is not merely an attempt to smooth out the simplification of general into detailed code, or to add new commands to the repertoire. It is an attempt to end the development of new languages and new extensions of old languages by making language itself a nonproprietary component of the computer. In simplest terms, the goal of MetaBasic is to make the construction of command words which do not exist possible, without modification of the compiler itself. In this sense, the command words resulting from the precompilation stages from interpreter Basic are only the JOFES suggested commands; they can be changed at the user's will. No compiler can be all things to all users, and it is vaulting ambition to make such an attempt. To realize this goal, the very nature of compilation must be changed.

Changing Its Nature

Traditionally, a compiler searches an input source program file for recognizable command words. Finding them, the compiler creates an output file in a cycle of activities described as "This—Equals—That." As long as the compiler must search for recognizable command words in the source file, however, the compiler fails to meet the needs of some users.

Alternatively, consider a compilation process in which the compiler did not recognize command words but command syntax, or a sequence of symbols which collectively comprised a command. It could then consult a library of commands prepared by the user and, from the library, complete the compilation process. In this way, the user could revise and extend the compiler to adapt to the continuously changing hardware and applications environment of modern microcomputing. The capability might actually extend the life of an older model computer by several years, where it is serviceable except for its ability to be programmed in a new way.

Although the concept sounds exciting, there is a serious hitch in the rope that ties it all together. Making up new command sequences is easy; programming their long and detailed Assembly language service

"The solution is to recognize reality and open the hush-hush inner sanctum of the operating system."

subroutines is the work of a professional systems programmer. How does one Basic programmer, with little more than a beginner's introduction to Assembly language, make use of such a device? The solution is to recognize reality and open the hush-hush inner sanctum of the operating system.

A MetaBasic compiler works precisely because all the elemental Assembly language subroutines needed are already contained in the operating system and are available to the user through the Call command. There is a specific repertoire of such subroutines, documented in programming manuals and well known by systems programmers; they are the first subroutines programmed during the development of a new operating system. Those TRS-80 Model I users who have written disassemblers know where these TRSDOS subroutines are located and how to use them. The only difference introduced by JOFES' ZMonitor operating system is that this detective work has been obviated by publishing the information up front.

Given that essentially no Assembly language programming will be needed beyond the trivial to use the MetaBasic compiler, the modestly capable Basic/Assembly programmer may set up virtually any kind of high-level command he needs. The only restrictions include data formats and syntax.

Formatting both numeric and character data in memory is a picky and nasty business. After every step of program operation, the status of referenced data elements must be reviewed and updated where changed. Because of this, data formats are recognized and managed by the MetaBasic compiler. While limiting the general user in the same manner as traditional compilers, the trade-off given is freedom from a truly exasperating maintenance function.

The syntax problem is much simpler. The next few pages illustrate MetaBasic syntax by making up a new command and showing the library module which would be prepared for the compiler if the command were really used.

MetaBasic Syntax

The MetaBasic Compiler is actually not a compiler, but a "parsing macro processor." With very few exceptions, MetaBasic does not recognize command words. Instead, it analyzes or "parses" a command format before using the parsed elements to specialize a standard JOFES macro library module. The MetaBasic command line format is in Table 1.

The MetaBasic Compiler will process virtually anything which conforms to this format, in the following steps:

1. The MetaBasic line is read from the input file.
2. The line is determined to be a command line, or:
3. Comment or Assembly language line, the latter two being written directly to the output file.
4. Command line is parsed into its functional components and arranged in a macro call argument table in memory.
5. A final check is made for certain non-macro call elements.
6. The called macro is found in one of the macro libraries.
7. The macro lines are specialized from the command line arguments and written to the output file.
8. Return to step 1 until end of input file.

Before reviewing these steps in detail, it may be useful to examine a typical MetaBasic command line which is not available but for which a particular user has great need. His application consists, in this example, of retrieving a large number of customer accounts at random times in a large number of programs. He decides to invent the command: 1000 :FETCH ACCOUNT$ FROM FILE$ ON DRIVE$. The command word is Fetch. ACCOUNT$ is functional element one, From is element two, FILE$ is element three, On is element four, and DRIVE$ is element five. He might also have designed the command syntax: 1000 :FETCH ACCOUNT$ (FILE$,DRIVE$) in which case there would be only the command word and a single functional element ACCOUNT$, associated with parenthetical modifiers. The analysis of these lines proceeds as follows:

● Step 1—Read Input: A MetaBasic input file is an ASCII-format sequential disk file. Each record is a line of the program. All characters between the first position and the first blank space on the line must be a number from 1-99999. If any character is not 0–9, and if the number is not terminated by a blank space, the line and the job will be aborted.

● Step 2—Determine Line Type: The first nonblank character after the first blank on the line must be one of three kinds: a colon (:) to signify a MetaBasic command line; a semi-colon (;) to signify an Assembly language comment line; or a capital letter (A–Z) to signify an Assembly language command. Macro call commands ('MACRO = NAME) are not permitted in the input file; such calls must be processed prior to compilation through MetaBasic. (The Precompiler, however, will permit macro calls without disturbing them.) Assembly language and comment lines are written directly to the output file. Unknown line types are flagged for a Bad Form message and converted into comments. Only the MetaBasic commands remain for subsequent processing.

● Step 4—Parse MetaBasic Command Line: the first nonblank character after the colon begins the parsing process. Note that the position of the command elements is not as important as their sequence. For example, wherever a blank character appears any number of blanks may appear. The interpreter searches for the next nonblank character until the end of the line. The first nonblank character after the colon must be a capital letter; any other character will abort the line. This first character begins the command word, which continues until the next blank space. The command word may be any length, comprised of A–Z, or 0–9, and the special characters !,#,@,$,%,& and period (.) and is terminated only by a blank space (any other character will cause an abort). All but the first six characters will be truncated from the word, and the remaining 1–6 characters will be examined for the exclusive values JP, CALL and RET. If a perfect match is found, the line will be converted into an Assembly language line and written to the output file. If those values appear in a larger sequence, however, they will not be recognized. For example, JPHERE, CALLME and RETURN will fail this examination.

Returning to the blank space which terminated the command word, the compiler will search for the next nonblank character, which must also be a capital letter. This character signifies the beginning of the first functional element, the end of which is terminated by a blank space, a comma, or any of the symbols +, –, *,/,<,>,<>,><,< =, =<,> =, =>. In addition, a semicolon will terminate because it is the beginning of an in-line comment; a colon will terminate

Format Element	Description	Status
1	line number	required
2	blank space	required
3	colon	required
4	command word	required
5	blank space	required if not end
6	functional element 1	optional
7	terminator symbol	required if not end
8	functional element 2	optional
9	terminator symbol	required if not end
10	functional element 3	optional
11	terminator symbol	required if not end
12	functional element 4	optional
13	terminator symbol	required if not end
14	functional element 5	optional
15	terminator symbol	required if not end
16	functional element 6	optional
17	terminator symbol	required if not end
18	functional element 7	optional
19	terminator symbol	required if not end
20	functional element 8	optional
21	terminator symbol	required if not end
22	functional element 9	optional
23	terminator symbol	required if not end

Table 1. MetaBasic command line format.

Poor Man's Floppy

HIGH SPEED CASSETTE SYSTEM

FOR TRS-80*

[Reprint of June 1980 Review, *80 Microcomputing*]

Now the widely acclaimed JPC Cassette System is available for your TRS-80* computer. The price is only $90.00

TC-8 Cassette System
JPC Products
Albuquerque, NM
Kit: $90
Assembled: $120

by Carl A. Kollar

I guess I don't have to tell any TRS-80 owners how frustrating the cassette system that comes with the computer can be. Even with the factory mod that's available, the annoyance of loading and checking programs becomes just barely tolerable.

If you're like me, after you've just plunked down a chunk of money for a Level II 16K machine, "you ain't got nuttin left" for even one disk drive at 500 bucks apiece. So you suffer.

A reasonable alternative is the Exatron Stringy Floppy (ESF). This will cost you about 250 bucks and totally eliminates your loading and saving problems, automatically and fast. I've had one of these for about six months and love it!

But, if the price is still too steep, have I got a device for you!

The Device

The February 1980 issue of *Microcomputing* had an ad that intrigued the hell out of me. It was a high-speed cassette system by JPC Products acclaimed as a "poor man's floppy." It made all sorts of seemingly ridiculous claims such as "loads five times faster," "stores 50,000 bytes on a 10-minute cassette," "less than one bad load in a million bytes with the volume control anywhere between one and eight."

All this for a measly [90] bucks? How could this be? A call to Albuquerque answered a few questions: Yes, it had its own power supply, and, it stored programs five times faster because it utilized higher density data. The computer outputs the information at a higher rate out of the rear keyboard connector.

The ad had even claimed anyone could build it even if you have never soldered before. JPC would make it work, if you couldn't—for free. I was sold. I placed my order, and it arrived about two months later (parts shortage).

I work in electronics, so I found the unit exceptionally easy to build. It took about an hour. The manual is superb. (That's better than great.) It was clear, concise and exact with no ambiguities. Important parts placements are stressed (polarity markings on electrolytics, bands on diodes, etc.).

JPC was right! With these instructions, you couldn't go wrong. The board quality is excellent. It is double-sided and parts locations are clearly marked on the component side of the board. There are no jumper wires to install. JPC utilizes PC traces and plated-through holes for connections to traces on the other side of the board.

Also, there are absolutely no adjustments or settings to bother with.

The documentation is a sheaf of 8½ × 11 papers stapled together. It is written in the nicest format I've seen in a while. Each command and/or subjects is covered on its own sheet in large type. All explanations are in easy to read English—not computerese.

Commands and Features

SAVE"filename": Saves your BASIC program on cassette.
LOAD: Reads the next BASIC program from the cassette.
LOAD"filename": Searches for and loads the specified file from cassette.
LOAD? and LOAD?"filename": Reads file from cassette, and compares contents to memory.
LOADN: Prints a list of all the programs on a cassette, until interrupted by the "break" key.
LOADN"filename": Same as above except the tape will stop at the end of the program named.
KILL: Removes the file manager program from memory so that the extra memory can be used by large programs.
RSET: Allows the operator to rewind and position the tape on tape recorders that have these functions tied to the motor control jack.
RUN"filename": TC-8 searches for a specified program and runs it immediately.
PUT"filename": Same as SAVE "filename", except it is for use with system tapes.
GET: Same as LOAD, except it is for use with system tapes.
GET"filename": Same as LOAD "filename", except it is for use with system tapes.
GET? and GET?"filename": Same as LOAD? and LOAD?"filename", except it is for use with system tapes.
GETN and GETN"filename": Same as LOADN and LOADN"filename", except it is for use with system tapes.
OPEN: Required before cassette input or output of a data file can be attempted.
CLOSE: Required to end a cassette data file.
PRINT#: Allows numerical or string data to be output to a cassette file.
INPUT#: Allows numerical or string data to be input from a cassette file.

I haven't counted them, so I don't know about the "one load in a million bytes" claim, but my son, Anthony (age 11), loaded about 30 of his programs from his Radio Shack format tape to a new TC-8 format tape. He's run them all and found no bad loads.

Unlike the standard tape system, you can position your tape anywhere before the program you want and not have to look for a blank spot between programs. The TC-8 patiently waits for the program you want and then starts loading without getting confused by the portion of the previous program you just fed it.

Try that on your regular cassette system; you'll wear out the reset button. ■

ORDER NOW

To order your TC-8 kit, send your check or money order for $90.00 plus $3.50 postage and handling to JPC PRODUCTS CO., 12021 Paisano Ct., Albuquerque, NM 87112 (New Mexico residents add 4% sales tax). Credit card orders accepted by phone or mail. Personal checks will delay shipment. We will otherwise immediately ship you the TC-8 kit, the cabinet, the ribbon cable, the power adapter, an instruction manual, and a cassette containing the software.

✓190

JPC PRODUCTS CO.
Phone (505) 294-4623
12021 Paisano Ct.
Albuquerque, N.M. 87112

✓See List of Advertisers on page 402

"...JPHERE, CALLME and RETURN will fail this examination."

Argument Number	Contents
1	command word
2	line number of source input file
3	nth time this command has been used in this program
4	number of functional elements in this call
5	
6	
7	
8	
9	
10	Functional element number 1
11	number of parenthetical subelements in this element
12	subelement number 1
13	subelement number 2
14	subelement number 3
15	subelement number 4
16	subelement number 5
17	subelement number 6
18	terminator symbol which ended this functional element
19	
20–29	Same as 10–19, for functional element number 2
30–39	Same as 10–19, for functional element number 3
40–49	Same as 10–19, for functional element number 4
50–59	Same as 10–19, for functional element number 5
60–69	Same as 10–19, for functional element number 6
70–79	Same as 10–19, for functional element number 7
80–89	Same as 10–19, for functional element number 8
90–99	Same as 10–19, for functional element number 9

Table 2. Macro call structure.

because MetaBasic allows only one command per line. The colon will be converted into a semicolon without comment or error. If a left parenthesis "(" is encountered while searching for the terminator symbol, the compiler will search only for a right parenthesis ")" before resuming search for the terminator symbol. The first functional element will be separated from its parenthetical unit (e.g., "ABS(A)", where "ABS" will be separated from "(A)"), and the functional element will be truncated to six characters if it is longer. It will then be examined for an exclusive match against "EQU." If a match is found, the line will be converted into an Assembly language command and written directly to the output file.

If a parenthetical unit was found attached to the first functional element, it will be examined internally for one to six subelements. A parenthetical subelement may contain the same character range as the command word; multiple subelements must be separated only by a comma.

Returning to the symbol which terminated the first functional element, the compiler will search for the next nonblank character

". . . an external name needs to be defined only once for the entire program."

as the beginning of the second functional element. Its analysis will continue in the manner described above for the first functional element. There may be as many as nine functional elements.

The Macro Library

The internal macro call table is essential to implement compiler command lines. The macro library module providing the Assembly language used by the compiler must reference these syntactical elements; it does so according to the standard JOFES 99-argument macro call structure (Table 2).

Standard macro writing procedures, referencing these argument numbers, will include the correct Assembly language in the compiled program. For example, the MetaBasic command line, 1000: FETCH ACCOUNT$ (FILE$,DRIVE$) where ACCOUNT$ is the functional element associated with two subelements FILE$ and DRIVE$, will be serviced by the macro library module:

```
1000 'MACRO=FETCH
1010 FT&&02 EQU $ ;LINE &&02
1020 'BOOL ((&&00)=(1))
1030 FETCH EQU 0000H;DEFEXT
1040 'ENDBOOL
1050           CALL FETCH
1060 'BOOL ((&&04)=(1))
1070           DEFW &&10
1080 'ENDBOOL
1090 'BOOL ((&&04)< >(1))
1100 ;*** ERROR IN ACCOUNT NAME ***
1110 'ENDBOOL
1120 'BOOL ((&&11)=(2))
1130           DEFW &&12
1140           DEFW &&13
1150 'ENDBOOL
1160 'BOOL ((&&11)< >(2))
1170 ;***ERROR IN FILE NAME OR DRIVE
     NUMBER ***
1180 'ENDBOOL
1190 'ENDMAC
```

The result of this command will generate the Assembly language.

The entry tag FT1000 was generated from argument two so the source-code programmer could refer to this routine by a constant name. The same information appears on the end of the line in a comment. Because this is the first time this macro has been called in this program, the external reference is generated at line 1010. In a second or successive call this line would be dropped out; an external name needs to be defined only once for the entire program. The address for the data variable ACCOUNT$ has been truncated to the most significant six characters, "ACCOUN." Since the names of the file and drive were not longer than six characters, they were defined in their complete form. If the account name had been missing, or if either of the file name or drive number parameters had been missing, the appropriate error message would have been included in the program instead.

The program Fetch called in this sample has been assumed to be external to this program and, therefore, flagged with the DEFEXT indicator. If it were another macro called into this program in source form the macro programmer would not have included the DEFEXT comment on line 1030 of the macro module. Instead, a macro call to Fetch would appear elsewhere in the program, and the program would have passed through the Assembly language macro pro-

See List of Advertisers on page 402

"...the manufacturer who insists upon continuing to do every little thing for its customers will defeat its own purposes."

cessor before compilation. This illustrates a major difference between macro processing in Basic and in Assembly language. The Basic macro processor recognizes common macros and calls them into the program only once during each specialization run. It also arranges the various macros at different locations of the program. By contrast, the MetaBasic Compiler includes every macro exactly where it is called, every time it is called. The macros intended to function as common within the Assembly program must be brought in expressly by the programmer; the Compiler could not otherwise know when a macro is intended to be internal or external.

A good way to become accustomed to writing Assembly language macros is to compile a small program, list it, and then list the macros in the macro libraries used during specialization. The names of the macros are the first six letters of each MetaBasic command word. Thus, Load is both a MetaBasic command word and the name of a macro library module.

Much of what is called "common sense" is information acquired through noninstitutionalized education—just being around the subject and around other persons involved in that subject. Thirty years ago, there was no common sense surrounding the subject of computers. Today thousands of people have their own computers, get together in a variety of commercial and social contexts, and exchange their knowledge. The appearance of a body of "common sense" about computers is the hallmark of a maturing, socially integrated industry which will take a new shape and direction of its own, essentially beyond the control or influence of any manufacturer.

While it was formerly required that the computer hardware manufacturer provide a broad range of assistance devices in the software sphere because of the general lack of common sense in the user community, that is no longer true. Like a parent who cannot adjust to the maturity of his or her child, the manufacturer who insists upon continuing to do every little thing for its customers will defeat its own purposes. It is not unfair to state that the single greatest limitation on the usefulness of microcomputers today, and those microcomputers in the TRS-80 Model I class in particular, is the limitation imposed by the "helpfulness" of the manufacturer.

Optimal economies cannot be achieved by so many different kinds of users until the manufacturer—or some other source—provides tools which continue to help in a different way. The twenty-five-year-old "child" occasionally needs help, but not in the same way he needed help at age ten. The user community stands in this posture today as we seek optimized economies in the application of our microcomputers. We do not need to be told what to do and how to do it at every turn.

MetaBasic is one, perhaps the first, attempt to move into this new sphere of mature support for the microcomputer user. It will not be the final solution to this need. With another five years' experience, the community as a whole will determine new and vastly improved devices for acquiring that support appropriate to contemporary economic need. ■

For those of you who must edit and run.

Chain Command Implementation

Roger L. Pape
7545 Marble Drive
Liverpool, NY 13088

Have you ever wanted to change a program after spending much time entering variables? This article describes a program chaining that will allow you to edit programs and continue execution without losing previously entered or calculated values.

The routines are designed primarily for cassette-based systems. The chain command is essentially equivalent to the CLOAD and run commands, including an optional file label character or starting statement number.

The procedure used to save variables while modifying source code in the current program is split into two function calls. The first call moves the variables to upper memory and protects them. Then control returns to the command level. Use existing system commands to delete, insert or edit lines. After all modifications are made, a second function call restores the variables to their normal position and resumes execution at any point in the program. The second function should only be permitted if the variables had been moved earlier. Unique keywords should be defined for the three functions. The choice of Chain is obvious. For the command to protect variables before modifying the program, I selected Change. Restart was selected to denote the rerunning of the program.

Because the preserve-while-editing feature might be used at any time, these routines will remain in memory. Therefore, the package is as compact as possible and installed in protected high memory. Finally, this version is self-installing, relocatable below any other patches already loaded in upper memory.

Calling Convention

The standard system commands and functions are converted to single byte tokens when statements are entered. This process not only saves space, but also eliminates the time required to match strings of characters to a table of keywords when the program is executed. When additional commands are included, you should also use tokens, particularly if the commands are inserted in program statements. Otherwise, the character strings in each statement must be compared to the additional keywords as the statements are interpreted, slowing down execution time significantly. Unfortunately, though, almost all the possible 128 token values are used in the TRS-80 Level II software and there is no provision to extend the set with a second token byte.

Rather than add functions as separate commands, treat these functions as exten-

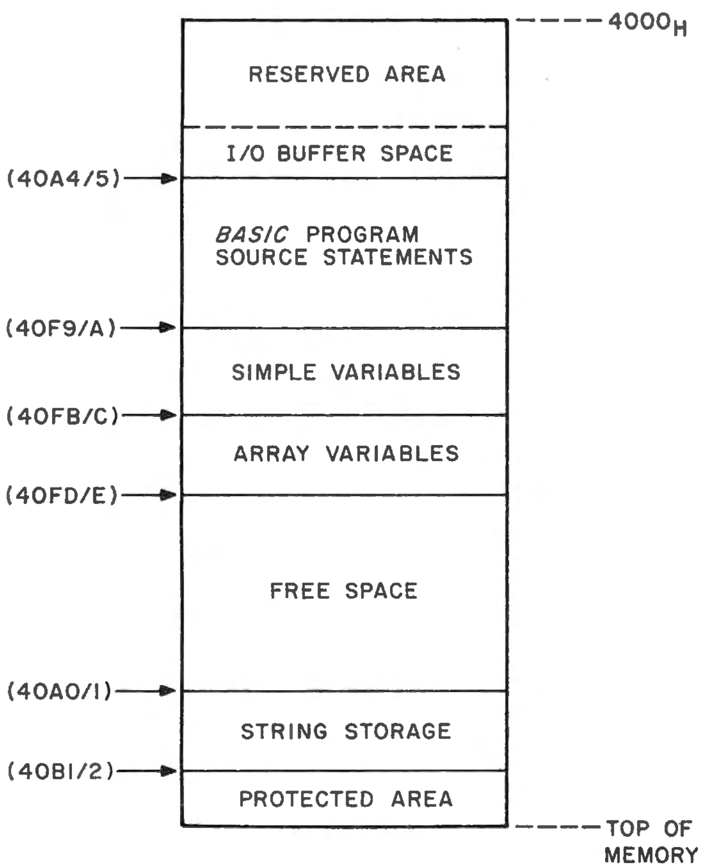


Fig. 1. RAM Memory Organization and Pointers in TRS-80 Level II.

Maxi Manager

The finest Data Base Manager Available

DATA MANAGEMENT PROGRAM COMPARISON CHART

FILE CAPACITY & FORMAT	CCA DATA MANAGER	AIDS III with CALS	MAXI MANAGER	ADEX 10	PROFILE
Maximum # of disks per file	1	1	4	31	4
Maximum # of records per file	2450	Note 1	32,767	10,199	65,535
Maximum record length	249	254	800	255	255
Maximum # of characters per field	249	254	40	254	255
Maximum # of fields	24	20	20	127	153
Maximum # of characters per field label	15	10	19	12	765
Variable length records (pack sectors)	No	Note 2	Yes	No	No
FIELD TYPES					
Alphanumeric	Yes	Yes	Yes	Yes	Yes
Numeric	Yes	Yes	Yes	Yes	No
Fixed decimal numeric	Note 4	Yes	Yes	No	No
Date (MM/DD/YY)	Yes	No	Yes	No	No
Extended date (MM DD YYYY)	No	No	Yes	No	No
Calculated equation	Note 5	Note 6	Yes	No	No
Permanent fields	Yes	No	No	No	No
SORTING					
Machine language assisted	No	Yes	Yes	Note 7	Yes
Sort by any field	Yes	Yes	Yes		Yes
Number of Sort Key files	1	1	5		1
Numeric sort	Yes	Yes	Yes		No
Ascending sort	Yes	Yes	Yes		Yes
Descending sort	Yes	Yes	Note 11		Yes
Sort within a selected range	No	No	Yes		No
Sort multiple fields simultaneously	Yes	Yes	No		No
FILE MAINTENANCE					
Fixed length input fields	Yes	Yes	Yes	Yes	Yes
Single key entry of common data	No	No	Yes	No	No
Single field EDIT selection	Yes	Yes	Yes	Yes	Yes
Skip record (next or previous)	Yes	Yes	Yes	No	Yes
Search & EDIT record	No	Yes	Yes	No	Yes
Search & DELETE record	No	Yes	Yes	No	No
Auto rejection of alphanumeric data in numeric field	Yes	No	Yes	No	No
RECORD SELECTION TECHNIQUES					
Record number	[illegible]	[illegible]	Yes	[illegible]	[illegible]
Binary search (high speed)	[illegible]	[illegible]	Yes	[illegible]	[illegible]
Maximum # of simultaneous keys	[illegible]	[illegible]	10	[illegible]	[illegible]
RELATIONAL COMPARISONS					
Equal	No	Yes	Yes	Yes	Yes
Not equal	No	Yes	Yes	No	Yes
Greater than	No	Yes	Yes	Yes	Yes
Less than	No	Yes	Yes	Yes	Yes
Instring	Yes	No	Yes	Yes	No
AND / OR	No	No	Yes	Yes	No
Wild card masking	No	No	Yes	No	No
PRINTING					
User specified page title	Note 8	Yes	Yes	No	Note 10
User specified column headings	No	No	Yes	No	Yes
Automatic page numbering	Yes	Yes	Yes	Yes	Yes
Right justification	No	Yes	Yes	No	No
User defined column widths	Yes	No	Yes	Yes	Yes
User defined column separators	No	No	Yes	No	No
Keyboard entered columnar values	No	No	Yes	No	No
Merge data into form letters	No	No	Yes	No	No
Form filling applications	No	No	Yes	No	No
Columnar totals	Yes	Yes	Yes	No	No
Columnar subtotals generated upon change in a specific field	Yes	Yes	Yes	No	No
Built in screen print	No	No	Yes	No	No
MISCELLANEOUS					
Cost	$75.00	$94.90	$99.95	$99.00	$79.95
Punctuation allowed within data fields	Yes	?	Yes	Yes	Yes
Upper / Lower case	Note 3	Note 3	Yes	Note 3	Note 3
Built in RS-232-C driver	Note 3	Note 3	Yes	Note 3	Note 3
Built-in TRS-232 driver	Note 3	Note 3	Yes	Note 3	Note 3
Programmer's interface	Note 9	Note 9	Yes	No	Note 9
Sample DATA disk	No	No	Yes	No	No
Documentation (# of pages)	?	?	120	38	29

NOTE 1 File size is dependant on memory size.
NOTE 2 Sequential files only.
NOTE 3 User must apply own driver routine.
NOTE 4 Hard copy print out only
NOTE 5 Four functions (+ - * /) only
NOTE 6 Same as note #5 with a maximum of two calculated fields.
NOTE 7 Available as a separate program for $99 95.
NOTE 8 120 character maximum.
NOTE 9 Data structures defined in manual.
NOTE 10 132 characters maximum.
NOTE 11 User option (files can be read from ascending or descending order).

The jury is in and the verdict is . . . "outstanding!" Reviews from all of you who purchased MAXI MANAGER (not to mention raves by many top microcomputing magazines) have heralded it as the definitive data base managing system. We knew that business owners and hobbyists demanded the finest data base managing system available. To all of you who praised us for MAXI MANAGER, we extend our thanks. And to those of you who have yet to try MAXI MANAGER, we invite you to experience this incredible system today. But don't take our word for it (or our jury's). Judge for yourself.

JUST CHECK SOME OF THESE FEATURES

- Supports six different relational search techniques.
- Comes with programmer's interface.
- Over 120 pages of documentation.
- Supports up to 20 user defined fields of 40 characters each.
- Record length up to 800 characters.
- Files can be up to four disks in length.
- Compatable 35, 40, 77 & 80 track drives with proper operating system.
- Has calculated equation fields.
- Complete report generator.
- Works hand in hand with any word processor.

MODEL 1 version requires TRSDOS 2.3 and is compatable with NEWDOS 2.1 & NEWDOS 80

MODEL 3 version comes on TDOS, a special version of the DOSPLUS operating system.

Requires 48K of RAM and one disk drive minimum.

For the TRS-80 Model 1 & 3
Order No. 012-0096

$99.95

A Division of Scott Adams, Inc.
Adventure International • Box 3435 • Longwood, FL 32750
TOLL-FREE ORDER NUMBER: (800) 327-7172
IN FLORIDA (305) 862-6917
SHIPPING AND HANDLING CHARGES ARE EXTRA
PRICES ARE SUBJECT TO CHANGE WITHOUT NOTICE

"The routines are designed primarily for cassette-based systems."

sions of an existing command. Certain commands include "hooks" in the reserved RAM area to provide for changes. At some point in the processing of these commands, program execution jumps to the RAM area. The default code stored there at power-up is simply a Return instruction. However, it can be modified to a jump that transfers execution to an area where the extensions have been loaded. One of the modifiable commands is Run. Therefore, it was decided to invoke the functions with Run Chain, Run Change, and Run Restart as the form of the commands. The patch will be entered only when a Run command is being processed, not as each line of code is interpreted.

Description of Routine

To obtain the compactness of Program Listing 1, I used existing Level II system routines and pointers stored in the reserved area. All the code to be installed is "position independent" so that it can be automatically relocated below existing routines. Returns to the system assume Level II (cassette-based) operation. Modifications would be required to operate with a DOS.

Code to Install Patch

The first portion of Program Listing 1 simply assigns labels to the various entry points in ROM and the storage locations in the reserved area where values and pointers needed in the routines can be found. Lines 440 to 610 are used only when first loaded to move the routines up into the highest available memory space, install the vectors to the patches and set the memory size pointer to protect the relocated routines. The installation code is not protected, so it will be overwritten later by the string storage. The value in the ORG statement is arbitrary. It may be any convenient starting point in memory where the machine code can be stored temporarily as it is loaded from tape.

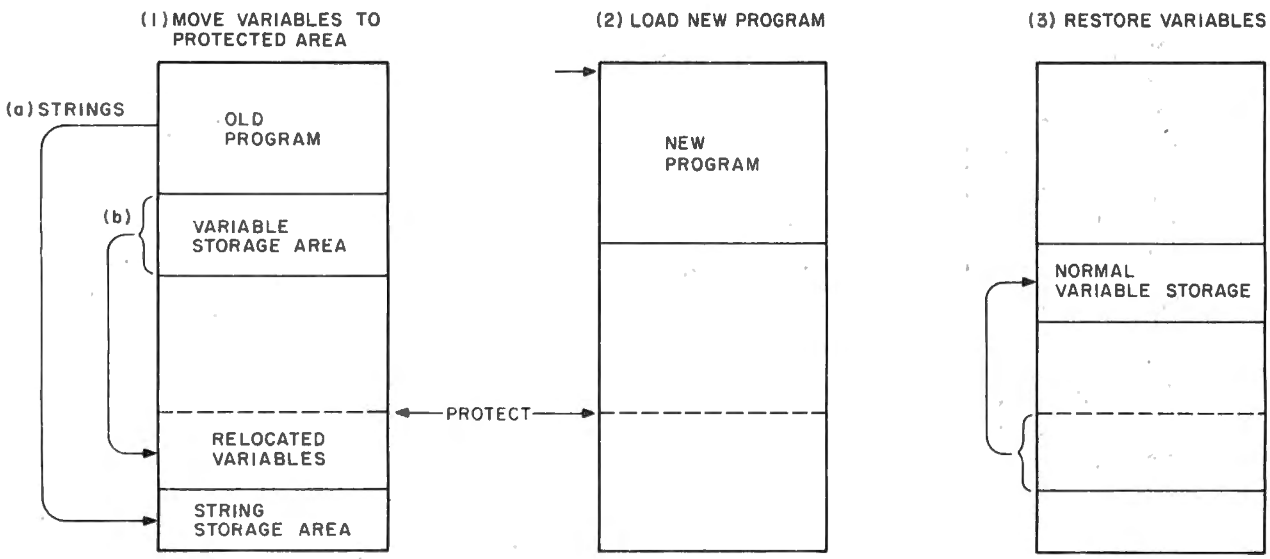


Fig. 2. Chain Sequence

✓See List of Advertisers on page 402

New

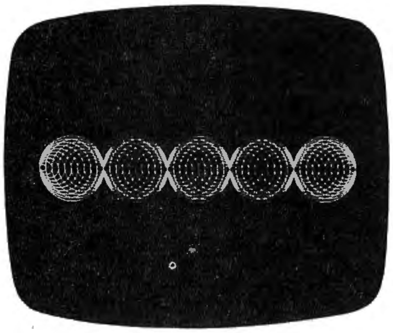
XCEL™ GRAPHICS

512 x 240 RESOLUTION

FOR TRS-80® MODEL II

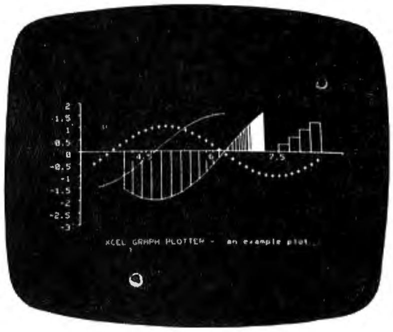
Also for Zenith Model 89 & Superbrain™

XCEL™ GRAPHICS UNIT — A completely assembled hardware retrofit package providing high resolution graphics display, normal video display or both displays simultaneously. **$895.00**



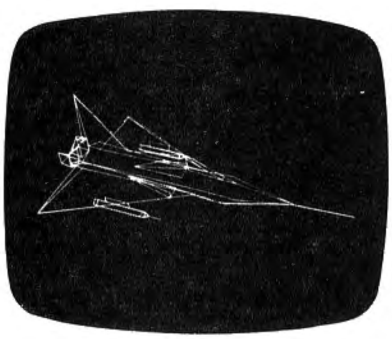
$175.00

XCEL™ SYMBOL GENERATOR—Subroutines to generate characters and strings with bold face, and 90° rotation options, circles, arcs, vectors, rectangles, area fills with reverse video option. All callable from your programs.



$175.00

XCEL™ GRAPH PLOTTER-Choice of four representations (line, graph, histogram, bar graph and scatter plot) with automatic annotation of axes and automatic scaling that allows any number of functions to be displayed on one set of axes. All callable from your programs.



$345.00

XCEL™ 3-D GENERATOR—User accessible routines for creation, editing and viewing "wire-frame" objects from any angle and in any orientation with scaling "zoom" and a graphics editor.

$395.00

XCEL™ SURFACE PLOTTER-True perspective view, with full hidden line removal, of curved surfaces from a user defined viewpoint callable from your programs.

$395.00

XCEL™ GRAPHICS TERMINAL—Emulates a subset of the Tektronix® 4010 Series Graphics Terminal functions. Simple menu-driven keyboard commands for configuration and dot-dash and erase mode vectors.

XCEL™ software packages operate under CPM® and utilize the Microsoft parameter passing mechanism for compatibility with high level languages, such as BASIC, FORTRAN, COBOL, PL1, and PASCAL. XCEL software can be loaded into high memory and called from your programs or integrated directly into your programs through use of the Microsoft linking loader.

FOR ADDITONAL PRODUCT INFORMATION
IN CONTINENTAL U.S. CALL TOLL FREE: **(800) 421-1423**

IN CALIFORNIA CALL
(213) 320-6604

MAXTEK, INC. 2908 Oregon Court, Torrance, CA 90503 ↙585

TRS-80-registered trademark Tandy Corp.® Superbrain-trademark Intertec Data Systems. Tektronix-registered trademark Tektronix, Inc. CPM-registered trademark Digital Research.

"...any absolute memory reference to the table must be avoided if the code is to be relocatable..."

After moving the stack pointer temporarily to the I/O buffer area, the routine points the HL register pair to the end of the code as loaded from tape and points DE to the last available memory location as determined from the value stored in the reserved area. It then performs a block transfer from the end to the start. Jump instructions are then loaded into the appropriate reserved area locations to provide for the Run command extension and to intercept the return from the CLOAD routine. Finally, the value of the first location of the code that was just installed is again loaded into HL before returning to the system software in ROM.

This re-entry point corresponds to the code following the memory size dialog, so the value in HL is decremented and stored as the new memory protect value. The string storage area and stack are also relocated below this point. This is a convenient technique that can be used when any machine language routine is loaded in high memory. Simply load the address of the lowest location to be protected into HL and jump to 00E7H when returning to Basic. This will automatically protect the routine and relieve the user of having to remember what memory size value to enter.

Scanning a Keyword Table

The remaining part of the code in Listing 1 represents the routine that remains resident in protected memory. Lines 630 to 880 check the characters following Run for the new keywords used in this patch. Although the table of new keywords is stored within this portion of code, any absolute memory reference to the table must be avoided if the code is to be relocatable in any area of memory *without modification.*

This feat is accomplished by the Call GetPC statement. A short routine in ROM is called which simply POPs the return address from the stack into HL and jumps back, so that HL contains the address following the call statement on return.

The keyword table is set up in the same way as the Basic keyword table in ROM. The first character of each keyword has its most significant bit set. After the last word, a value of 80H is used to terminate the table. By defining the keyword in this manner, we can use the same routine in ROM that is used to scan incoming statements for Basic keywords. This system routine attempts to match the character string pointed to by HL to the keywords in a table pointed to by DE. If a match is found, a value is returned in both the B and C registers which corresponds to the index of the keyword in the table plus 80H. A value of 80H is returned if Chain is found, 81H for Change, or 82H for Restart. If no match is found, we can exit from the patch.

When a keyword match occurs, the pointer to the remaining characters of the line must be saved for later, in case a file name and/or line number is included in the command. When Restart is encountered, check to make sure that a Change command had been issued earlier. If not, the command is invalid and is ignored.

Moving the Variables

Lines 890 through 1360 are common to

✓See List of Advertisers on page 402

"... some string variables may be located within the Basic statement storage area."

the processing of both the Chain and Change commands. After saving the current token, this section moves all variables to the upper portion of memory and protects them. Remember that some string variables may be located within the Basic statement storage area. These must be moved to the string storage area; otherwise they will be lost when new statements are loaded or the memory references will be incorrect if statements are moved in the editing process. Fortunately, there is a system routine in ROM (part of the FRE command processing) which packs the string storage area before reporting the remaining space available. Normally, this routine would only operate on strings in the string storage area. However, we can also pull strings out of the Basic statement area simply by changing the beginning location for the scan to the start of the Basic statements before the packing routine is called.

The code given in Listing 1 does not check to ensure that the other variable area is not overwritten as the strings are moved. This could be a problem if a lot of strings are

Program Listing 1. Assembly Language Routine for Chain.

```
                ;          COPYRIGHT (C) 1979
                ;          R. L. PAPE   LIVERPOOL, NY   SEPT. 1979
                ;
000B            GETPC      EQU 000BH         ;ENTRY FOR ROUTINE TO GET PROGRAM CO
UNTER
00E7            MEMSET     EQU 00E7H         ;REENTRY POINT IN LVL II TO SET MEMO
RY LIMIT
1A19            RESTRT     EQU 1A19H         ;RESTART POINT FOR LEVEL II BASIC
1B8F            INITM      EQU 1B8FH         ;ENTRY FOR INITIALIZATION ROUTINE
1BFE            KWSRCH     EQU 1BFEH         ;ENTRY FOR KEYWORD SEARCH ROUTINE
1D1E            XEQ        EQU 1D1EH         ;RETURN POINT TO START PROGRAM
1D91            RESTOR     EQU 1D91H         ;ENTRY FOR DATA RESTORE ROUTINE
1EAC            RUNLN      EQU 1EACH         ;ENTRY FOR RUN WITH LINE NUMBER
28E6            STRPK      EQU 28E6H         ;ENTRY FOR STRING PACKING ROUTINE
2C1F            CLOAD      EQU 2C1FH         ;ENTRY FOR CLOAD ROUTINE
                ;
40A0            SSTR       EQU 40A0H         ;RESERVED LOC FOR START OF STRING SP
ACE
40A2            LNNUM      EQU 40A2H         ;RESERVED LOC FOR CURRENT LINE NUMBE
R
40A4            SSRC       EQU 40A4H         ;RESERVED LOC FOR START OF BASIC SOU
RCE
40B1            MEMLIM     EQU 40B1H         ;RESERVED LOC FOR MEMORY PROTECT POI
NTER
40D1            LSTKN      EQU 40D1H         ;UNUSED LOC FOR SAVING TOKEN
40D6            CSPTR      EQU 40D6H         ;RESERVED LOC FOR CURRENT STRING STO
RAGE POINTER
40D8            INPTR      EQU 40D8H         ;RESERVED LOC FOR INPUT POINTER
40DF            SADDR      EQU 40DFH         ;RESERVED LOC FOR START ADDRESS
40F9            SVAR       EQU 40F9H         ;RESERVED LOC FOR START OF VARIABLE
STORAGE
40FB            SARR       EQU 40FBH         ;RESERVED LOC FOR START OF ARRAY STO
RAGE
40FD            SFRE       EQU 40FDH         ;RESERVED LOC FOR START OF FREE SPAC
E
                ;
41B5            CLDHK      EQU 41B5H         ;HOOK FOR CLOAD RETURN
41C7            RUNHK      EQU 41C7H         ;HOOK FOR RUN XXXX COMMAND
42E8            TMPSTK     EQU 42E8H         ;TEMP STACK START (IN I/O BUFFER ARE
A)
                ;
                ;     FOLLOWING SECTION INSTALLS MOD IN HIGHEST AVAILABLE MEM
ORY SPACE
                ;          AND AUTOMATICALLY PROTECTS ITSELF
                ;
6000                       ORG 6000H         ;SET ORIGIN TO ANY CONVENIENT FREE M
EMORY
6000 31E842     SETUP      LD SP,TMPSTK               ;SET STACK IN I/O BUFFER SPA
CE TEMP
6003 210661                LD HL,XIT+2                ;LOAD HL WITH CURRENT ADDR O
F LAST BYTE
6006 ED5BB140              LD DE,(MEMLIM)             ;LOAD DE WITH ADDR OF LAST A
VAILABLE LOC
600A 01DF00                LD BC,XIT-RUNX+3           ;PUT LENGTH IN BC
600D EDB8                  LDDR                       ;MOVE CODE
600F EB                    EX DE,HL
6010 23                    INC HL                     ;HL NOW POINTS TO START OF R
ELOC CODE
6011 E5                    PUSH HL                    ;SAVE FOR LATER
6012 3EC3                  LD A,0C3H                  ;JP INSTRUCTION CODE
6014 32C741                LD (RUNHK),A               ;HOOK FOR RUN COMMAND EXTENS
ION
6017 22C841                LD (RUNHK+1),HL
601A 019200                LD BC,CLRTN-RUNX           ;OFFSET
601D 09                    ADD HL,BC                  ; TO POINT HL TO CLRTN
601E 32B541                LD (CLDHK),A               ;HOOK TO INTERCEPT CLOAD RET
URN
6021 22B641                LD (CLDHK+1),HL
6024 E1                    POP HL                     ;RESET HL TO START OF CODE T
O PROTECT
6025 C3E700                JP MEMSET                  ;REENTER LVL II, SETTING MEM
ORY LIMIT
                ;
6028 EB         RUNX       EX DE,HL                   ;SAVE INPUT STRING POINTER I
N DE
6029 CD0B00                CALL GETPC                 ;HL WILL POINT 2 LOC BEFORE
TABLE BELOW
602C 1813                  JR SCAN                    ;SKIP OVER TABLE
602E C3         KWTBL      DEFB 0C3H                  ;1ST CHAR OF KEYWORD HAS MSB
 SET
602F 48                    DEFM 'HAIN'                ;REST OF KEYWORD
6033 C3                    DEFB 0C3H                  ;2ND KEYWORD
6034 48                    DEFM 'HANGE'
6039 D2                    DEFB 0D2H                  ;3RD KEYWORD
603A 45                    DEFM 'ESTART'
6040 80                    DEFB 80H                   ;TABLE TERMINATOR
6041 23         SCAN       INC HL
6042 EB                    EX DE,HL                   ;SWAP POINTERS FOR NEXT ROUT
INE
6043 CDFE1B                CALL KWSRCH                ;CHECK FOR KEYWORD MATCH
6046 EB                    EX DE,HL                   ;PUT STRING POINTER BACK IN
HL
6047 78                    LD A,B                     ;PUT TOKEN FROM SCAN IN A
```

Program continues

"...one can then POP off the memory protect value and the block lengths."

moved from a program which nearly fills memory. After the strings have been packed, the routine checks to see if the lower limit of the string space must be moved down.

The variable storage area can now be moved up immediately below the string area. Although the entire area can be moved as a block, the offset to the start of the array variables must also be computed in addition to the total length. The simplest way to save these lengths is to locate the stack immediately below the temporary variable storage and push the two values on the stack, along with the old memory protect pointer. The value of the stack pointer is then stored as the new memory size value.

If the command was Run Change, we have completed the necessary steps at this point. If the current token indicated Change, exit from the patch back to the command mode in Basic. The only other command that would have brought us to this point is Chain. Before jumping to the tape loading routine (CLOAD), the string pointer in the line being interpreted is restored in case a filename character is included in the Chain command.

Program continued

```
6048 B9        00780        CP C                 ;B AND C SAME IF MATCH
6049 C0        00790        RET NZ               ;RETURN IF NO MATCH
604A 23        00800        INC HL               ;SKIP PAST LAST CHAR MATCHED

604B 22DF40    00810        LD (SADDR),HL        ;SAVE STRING POINTER
604E 21D140    00820        LD HL,LSTKN          ;POINT HL TO TOKEN STORAGE
6051 FE82      00830        CP 82H               ;CHECK IF CURRENT TOKEN IS R
EPEAT
6053 2005      00840        JR NZ,SVTKN          ;IF NOT, SKIP OVER FOLLOWING

6055 96        00850        SUB (HL)             ;CHECK IF CHANGE COMMAND EAR
LIER
6056 3D        00860        DEC A                ;IF SO, DIFFERENCE IS 1
6057 2870      00870        JR Z,UNPK            ;CONTINUE BELOW
6059 C9        00880        RET                  ;ELSE RETURN
605A 77        00890 SVTKN  LD (HL),A            ;SAVE TOKEN
605B 21E842    00900        LD HL,TMPSTK
605E F9        00910        LD SP,HL             ;RELOCATE STACK
605F 2AA040    00920        LD HL,(SSTR)         ;GET START OF STRING STORE A
REA
6062 E5        00930        PUSH HL              ; AND SAVE
6063 2AA440    00940        LD HL,(SSRC)         ;REPLACE WITH START OF BASIC
 SOURCE
6066 22A040    00950        LD (SSTR),HL
6069 CDE628    00960        CALL STRPK           ;PACK ALL STRINGS AT TOP OF
MEMORY
606C 2AD640    00970        LD HL,(CSPTR)        ;GET BOTTOM OF PACKED STRING
S
606F EB        00980        EX DE,HL
6070 E1        00990        POP HL               ;GET ORIG STRING STORAGE STA
RT
6071 DF        01000        RST 18H              ;COMPARE VALUES
```

Program continues

Returning from CLOAD

The return from the CLOAD routine is intercepted at lines 1380 to 1430 because of the hook installed when the patch was initialized. The section will also be entered when a normal CLOAD command is issued (as well as the continuation of the Chain command). The first thing that must be done at this point is to check the current token. If it does not correspond to Chain, the patch is exited immediately. Otherwise, the string pointer is again saved in case a line number is included in the Chain command.

Restoring the Variables

Lines 1440 to 1640 move the variable block from upper memory back to its normal position immediately following the end of the new program statements. This is done automatically as part of the Chain command processing or is initiated by the Restart command after completing the statement editing (following a Change command). The steps performed in this section are essentially the reverse of those when the variable block is moved to upper memory and protected. By setting the stack pointer to the bottom of the protected area, one can then POP off the memory protect value and the block lengths.

A block transfer instruction moves the variables back. At the end of the transfer, the DE register pair will point to the start of free space, so its value is stored in the cor-

✓See List of Advertisers on page 402

"Once installed, the routine remains resident in protected high memory."

Program continued

```
6Ø72 38Ø1       Ø1Ø1Ø          JR C,RSSTR              ;IF CARRY, HL IS LOWER
6Ø74 EB         Ø1Ø2Ø          EX DE,HL                ;ELSE SWAP WITH DE
6Ø75 22AØ4Ø     Ø1Ø3Ø RSSTR    LD (SSTR),HL            ;RESET STRING STORAGE START
TO LOWER VAL
6Ø78 ED5BF94Ø   Ø1Ø4Ø          LD DE,(SVAR)            ;GET START OF VARIABLE STORA
GE
6Ø7C 2AFD4Ø     Ø1Ø5Ø          LD HL,(SFRE)            ;GET START OF FREE SPACE
6Ø7F B7         Ø1Ø6Ø          OR A                    ;CLEAR CARRY FLAG
6Ø8Ø ED52       Ø1Ø7Ø          SBC HL,DE               ;CALC SIZE OF TOTAL VARIABLE
 SPACE
6Ø82 44         Ø1Ø8Ø          LD B,H
6Ø83 4D         Ø1Ø9Ø          LD C,L                  ;TRANSFER SIZE TO BC
6Ø84 2AAØ4Ø     Ø11ØØ          LD HL,(SSTR)            ;GET START OF STRING SPACE
6Ø87 B7         Ø111Ø          OR A
6Ø88 ED42       Ø112Ø          SBC HL,BC               ;CALC START FOR RELOCATED VA
RIABLES
6Ø8A F9         Ø113Ø          LD SP,HL                ;RELOCATE STACK BELOW BLOCK
6Ø8B E5         Ø114Ø          PUSH HL                 ;SAVE POINTER DURING NEXT CA
LC
6Ø8C 2AFB4Ø     Ø115Ø          LD HL,(SARR)            ;GET START OF ARRAY STORAGE
6Ø8F B7         Ø116Ø          OR A
6Ø9Ø ED52       Ø117Ø          SBC HL,DE               ;CALCULATE OFFSET FROM START
 OF VARIABLES
6Ø92 E3         Ø118Ø          EX (SP),HL              ;PLACE ON STACK, RESTORING P
OINTER IN HL
6Ø93 C5         Ø119Ø          PUSH BC                 ;ALSO SAVE TOTAL LENGTH OF V
ARIABLES
6Ø94 78         Ø12ØØ          LD A,B                  ;TEST CONTENTS OF BC (BYTE C
OUNT)
6Ø95 B1         Ø121Ø          OR C
6Ø96 28Ø3       Ø122Ø          JR Z,SVML               ;SKIP BLOCK TRANSFER IF ZERO

6Ø98 EB         Ø123Ø          EX DE,HL                ;SWAP POINTERS FOR NEXT INST
R
6Ø99 EDBØ       Ø124Ø          LDIR                    ;RELOCATE BLOCK OF VARIABLES

6Ø9B 2AB14Ø     Ø125Ø SVML     LD HL,(MEMLIM)          ;GET OLD MEMORY PROTECT POIN
TER
6Ø9E E5         Ø126Ø          PUSH HL                 ;SAVE ON STACK
6Ø9F 21ØØØØ     Ø127Ø          LD HL,ØØØØH
6ØA2 39         Ø128Ø          ADD HL,SP               ;GET CURRENT STACK POINTER V
ALUE
6ØA3 22B14Ø     Ø129Ø          LD (MEMLIM),HL          ;PROTECT ALL ABOVE
6ØA6 22AØ4Ø     Ø13ØØ          LD (SSTR),HL            ;ALSO RESET STRING SPACE STA
RT HERE TEMP
6ØA9 3AD14Ø     Ø131Ø          LD A,(LSTKN)
6ØAC FE81       Ø132Ø          CP 81H                  ;CHECK TOKEN AGAIN
6ØAE CA191A     Ø133Ø          JP Z,RESTRT             ;IF CHANGE COMMAND, DONE; RE
ENTER BASIC
6ØB1 2ADF4Ø     Ø134Ø          LD HL,(SADDR)           ;RESTORE STRING POINTER IN H
```

Program continues

responding reserved area locations. The HL register pair will be pointing to the start of the string storage space and the stack is relocated there. Finally, the start of the array storage is computed and saved.

Starting the Program

Lines 1650 to 1730 perform the necessary initialization before program execution and reenter the ROM after the point where the variable storage pointers are reset. In this way the current variables in the storage area are preserved. The line number stored in the reserved area is reset to −1 ($FFFF_H$). If no line number is specified in the command string, execution will start at the beginning of the program. But if there are more characters in the command string (as determined by the RST 10H instruction), they are assumed to be the line number at which to enter the program. The alternate return in this case is the point at which the GOTO statement processing interprets the line number.

Therefore, we have provided the option of specifying a starting line number or starting at the beginning of the program by default if no line number is given.

Using the Routine

After the routine is assembled and saved, it can be loaded after any other patches, such as a keyboard debounce or special device driver, have been installed. The setup portion must then be run (by typing a / and Enter after loading) in order to install the routine. Once installed, the routine remains resident in protected high memory. The

LOW PRICES ON TRS-80*
From Radio Ranch, Inc.

In business 27 years at the same location providing electronic equipment and service.

We Are Better Equipped To Provide

TRS-80* COMPUTERS & Accessories Software

TRS-80* Mod II - 64K	3440.00
TRS-80* Mod III - 32K, 2 Dr	2195.00
TRS-80* Line Printer 6	1020.00
CORVUS** 5-10-20 M Hard Disks	Call For Prices

"CORDOS" and "OASIS" operating systems

Full Factory Warranty - Complete Service Department

Radio Ranch, Inc. ✓133

Cert. Check
Cashiers Ck.

Route 3, Radio Ranch Airport
Polo, Illinois 61064

Call Toll Free 800-435-8310 **In Ill. Call 815-946-2371**

*Trademark Tandy Corp. **Trademark Corvus Systems, Inc.

MODELS I AND III

ALL PROGRAMS USE TESTED

MICRO-TYPIT A text editing program that uses the keyboard "as is" like a typewriter. All edit and prompt functions are "built in" so instruction or learning is minimum. Text can be generated three pages at a time and printed either numbered, or unnumbered single or double spaced. Also, right justification is optional. Does not require substitution of @ for commas or any other character revision. Slowest function is transferring text to and from tape. Neat error free text with large print titles. Excellent for specifications, agreements, instructions, form letters or announcements. Complete with sound for TRS-80 Model I or III and Microteck printer easily adaptable to most other printers. PRICE.....$25.00

CLIENT FILES Allows keeping several contracts for up to 20 clients. Enter contracts including either payment amounts or percentages. As bills are sent out and payments received, the program is a record by contract of amounts outstanding and not billed including a summary of receivables and work in progress. Works well with one to 3 projects for a dozen or so clients. PRICE.....$25.00

ENGINEERING PROGRAMS

LIGHTING CALCULATIONS Calculates the required number of fixtures for a room given the dimensions and reflectances by the zone cavity method. Three standard fixtures are "built in" so that 8 room calculations are made to compare results such as watts, footcandles and number of fixtures. PRICE.....$30.00

HVAC PROGRAM This program calculates sensible and latent cooling and heating load by zone each two hours of the typical hottest and coldest days. Equations per 1977 ASHRAE Fundamental Guide page 25.2. Three programs are: Calculations, Geographic and Variable Data and Data Generator. PRICE.....$35.00

All tapes furnished with full documentation and instructions. Documentation, Instructions and Source List alone.....$15.00

All programs on tape require Level II and 16K. Printer is required for MICRO-TYPIT and optional on all others. Do not forget to include the Model Number (I or III) with your order.

Send Check or Money Order (no COD's please) to: ✓52

Coolidge & Associates

1317 S. CONGRESS #10 AUSTIN, TEXAS 78704 (512) 443-5329

EVEN SIGMON CRAVES A LITTLE SECS.

SIGMON

SIGMON is a complete machine Language development tool for your TRS-80 Color Computer and it's super powered 6809E microprocessor.

Features:

MONITOR: Display memory in HEX and ASCII, Direct HEX and Decimal entry into memory/registers, Move, Find, Tape Read and Write, Output to printer and more.

DISASSEMBLER: Display Addresses, Opcode operand, Mnemonics, and Address expressions for specific range of instructions.

MINI ASSEMBLER: Allows you to enter instructions in Symbolic form and have them converted to Machine Language.

DEBUGGER: Allows Stepping and Break point execution of Machine Language programs.

With SIGMON you can awaken the awesome powers that lurk within the dark recesses of your color computer.

Cassette and Source Code for the TRS-80 Color Computer $29.95

S.E.C.S.

(Screen Edit Control System)

SECS adds new power to your TRS-80 Color Computer. If you have been waiting for Enhanced BASIC and it's capabilities, then you are ready to take the next logical step.

Features:

FULL SCREEN EDITOR: Full Cursor control with auto repeat, Insert, Delete, Join lines, Relocate lines and Audible error warning.

HI RES GRAPHICS: Two modes of Hi Res graphics, Set point, Set line, Set color, Set programmable character, Set screen, and Toggle between Low Res and Hi Res/Normal and Inverted Screens.

HI RES CHARACTER GENERATOR: Sixty-four definable characters, Save and Load Characters sets to and from tape, Define Characters, and Place on screen in any color and in any position.

The entire system is linked into BASIC and is completely transparent to the user. As a BASIC programming tool, SECS is unsurpassed.

Cassette for the TRS-80 Color Computer $29.95

Software Utilities for the TRS-80* Color Computer

Order through your local software dealer or send check or money order plus $2.00 Shipping & Handling to:

✓407

Datasoft Inc.
COMPUTER SOFTWARE

19519 Business Center Drive, Northridge, CA 91324
(213)701-5161

*TRS-80 is a registered trademark of Radio Shack, a division of Tandy Corporation.

✓See List of Advertisers on page 402

"If Basic encounters a syntax error during program execution, it will automatically put you in the Edit mode."

added commands can then be called at any time as needed.

To chain from one program to another, any of the following forms may be used:

```
RUN CHAIN
RUN CHAIN "f"
RUN CHAIN "f" 1n
```

where f is the filename character used to save the program on tape
1n is a valid line number

In the first case, the next program on tape is loaded and execution will start at the beginning of the program. In the second case, the specified program will be loaded (if found) and will start from the beginning. In the third case, the named program will be loaded and entered at the specified line number. (If a starting line number is given, the filename character *must* be included.) Any of these Chain commands can be included as a statement in the first program. As the Chain routine is executed, it will be typed out Ready when the second program has been loaded and automatically start this new program.

In order to preserve the values of variables while a program is being modified, two separate commands are used. Before any editing, one must first type Run Change to protect the current values. This routine moves all variables to upper memory, protects them, and returns control to the command level of Basic. Then use any of the standard system commands to change the program statements, or use Edit to modify

Program continued

```
L
60B4 22D840    01350          LD (INPTR),HL         ;RESAVE IN LOC USED BY CLOAD

60B7 C31F2C    01360          JP CLOAD              ;JUMP TO CLOAD ROUTINE
60BA 21D140    01370 CLRTN    LD HL,LSTKN           ;NORMAL COMMANDS ENTER HERE
ALSO
60BD 7E        01390          LD A,(HL)             ;CHECK TOKEN STORAGE
60BE FE80      01400          CP 80H                ;IF CHAIN, CONTINUING FROM A
BOVE
60C0 C0        01410          RET NZ                ;ELSE RETURN
60C1 ED5BD840  01420          LD DE,(INPTR)         ;TRANSFER STRING POINTER
60C5 ED53DF40  01430          LD (SADDR),DE         ;TO OTHER STORAGE LOCATION
60C9 AF        01440 UNPK     XOR A
60CA 77        01450          LD (HL),A             ;CLEAR FLAG
60CB 2AB140    01460          LD HL,(MEMLIM)
60CE F9        01470          LD SP,HL              ;PUT STACK AT BOTTOM OF PROT
ECTED AREA
60CF E1        01480          POP HL                ;RESTORE
60D0 22B140    01490          LD (MEMLIM),HL        ;  OLD MEMORY PROTECT POINTE
R
60D3 210400    01500          LD HL,0004H
60D6 39        01510          ADD HL,SP             ;HL POINTS TO START OF RELOC
 VARIABLES
60D7 ED5BF940  01520          LD DE,(SVAR)          ;GET END OF NEW PROGRAM
60DB C1        01530          POP BC                ;GET LENGTH OF VARIABLE BLOC
K
60DC 78        01540          LD A,B                ;CHECK BYTE COUNT
60DD B1        01550          OR C
60DE 2802      01560          JR Z,RSTPT            ;SKIP TRANSFER IF ZERO
60E0 EDB0      01570          LDIR
60E2 C1        01580 RSTPT    POP BC                ;GET OFFSET TO ARRAYS
60E3 ED53FD40  01590          LD (SFRE),DE          ;DE POINTS TO START OF FREE
SPACE, STORE
60E7 22A040    01600          LD (SSTR),HL          ;HL POINTS TO STRING SPACE,
RESET POINTER
60EA F9        01610          LD SP,HL              ;ALSO RESET STACK THERE
60EB 2AF940    01620          LD HL,(SVAR)          ;CALCULATING
60EE 09        01630          ADD HL,BC             ;  START OF ARRAYS
60EF 22FB40    01640          LD (SARR),HL          ;STORE
60F2 21FFFF    01650          LD HL,0FFFFH
60F5 22A240    01660          LD (LNNUM),HL         ;RESET LINE NUMBER
60F8 CD8F1B    01670          CALL INITM            ;MORE INITIALIZATION
60FB CD911D    01680          CALL RESTOR           ;RESTORE DATA READ POINTER
60FE 2B        01690          DEC HL
60FF D7        01700          RST 10H               ;GET NEXT CHAR FROM INPUT ST
RING AND TEST
6100 C2AC1E    01710          JP NZ,RUNLN           ;IF MORE, MAY BE LINE NUMBER

6103 EB        01720          EX DE,HL              ;ELSE POINT HL TO START OF P
ROGRAM
6104 C31E1D    01730 XIT      JP XEQ                ;AND START EXECUTION
6000           01740          END SETUP
```

"... one can often segment a very large program into a sequence of modules which will fit in a modest sized machine."

```
>LIST
10 X=1:Y=10:Z=999
20 A$="THIS IS A TEST OF CHAIN"
READY
>RUN
READY
>RUN CHAIN
READY
STARTING NEW PROGRAM
THIS IS A TEST OF CHAIN
 1             10             999
READY
>

>LIST
10 X=1:Y=10:Z=999
20 A$="THIS IS A TEST OF CHAIN"
READY
>RUN
READY
>RUN CHAIN"B" 30
READY
 1             10             999
READY
>LIST
10 PRINT"STARTING NEW PROGRAM"
20 PRINT A$
30 PRINT X,Y,Z
READY
>
```

Program Listing 2. Examples of Chain Command Usage.

existing lines. Then restore the variables and start the program with the following command:

RUN RESTART 1n

1n is the line number at which to resume program execution. If no line number is entered, it will restart at the beginning by default. The Restart command will be ignored if a Change command was not issued earlier.

If a line number is included in either the Chain or Restart commands, it must be a number that exists in the current program or else the program will terminate with an error message. The line number should also be chosen to bypass any Clear statement or avoid redimensioning any variable array that had been specified earlier.

Sample Runs

In Listing 2, the first program simply sets the values of the variables. This program loads the values into the storage area. The A$ variable is a test of strings imbedded in the source statements.) The Run Chain command is typed to load the next program on tape. A Ready message indicates that the second program has loaded. This program automatically starts running and prints out the variables from the first program. A listing of the program shows that the new statements were loaded and that the variables printed out were indeed transferred from the first program. The second part of the figure shows a similar exercise, except that the command Run Chain "B" 30 causes the chained program to start execution at line 30, bypassing the first two print statements.

Program Listing 3 illustrates the case of a program with an error that needs to be corrected. When the program is run, the printout indicates that the value of Y had not been specified. To fix the problem, type the Run Change command to save the values that had been entered via the Input statement. After line 35 has been inserted, restart the program at line 35 by typing:

RUN RESTART 35

The final printout shows that the first three statements are skipped over and that the values of X and Z, which were input in the earlier program, have been retained.

If Basic encounters a syntax error during program execution, it will automatically put you in the Edit mode. Type Run Change and then return to the Edit mode. The variables must be protected before any changes are made.

Summary

An optional filename character can also be entered to specify the file to be retrieved from tape. The second program can be started at any specified line number (or default to the beginning if none is entered). With this added command, one can often segment a very large program into a sequence of modules which will fit in a modest sized machine.

The Chain routine is installed before any Basic programs are loaded since the code is position independent. A total of 239 bytes of memory are required for these three commands.

A Parting Note to DOS Users

The implementation presented is for a cassette-based Level II TRS-80. To modify this routine for a disk-based system, several system entry points would have to be changed. In particular, the hook to the CLOAD routine would be replaced by the corresponding routine which retrieves a program file from disk. Since the patch is entered as the Run command is processed, be familiar with the DOS extension for this command. Also, because DOS extensions are overlaid in lower RAM memory, it is more appropriate to install this patch in the DOS area, rather than waste space in upper memory. ■

```
>LIST
10 INPUT"ENTER X AND Z";X,Z
20 A$="TESTING CHANGE AND RESTART"
30 PRINT A$
40 PRINT X,Y,Z
READY
>RUN
ENTER X AND Z? 3,1
TESTING CHANGE AND RESTART
 3             0              1
READY
>RUN CHANGE
READY
>35 INPUT"ENTER Y";Y
>RUN RESTART 35
ENTER Y? 2
 3             2              1
READY
>
```

Program Listing 3. Example of Chain and Restart Command Usage.

✓See List of Advertisers on page 402

H & E COMPUTRONICS INC.

•••EVERYTHING FOR YOUR TRS-80™•••

* TRS-80™ is a trademark of the Radio Shack Division of Tandy Corporation

100 SUPER PROGRAMS

MASTER PAC 100

2nd EDITION (COMPLETELY REVISED)

FOR YOUR TRS-80™ LEVEL II MICROCOMPUTER

ALL ON CASSETTE OR DISKETTE

BUSINESS AND PERSONAL FINANCE
1. CHECKBOOK MAINTENANCE
2. TIME FOR MONEY TO DOUBLE
3. FEDERAL FICA & WITHHOLDING TAX
3. COMPUTATIONS
4. HOME BUDGET ANALYSIS
5. ANNUITY COMPUTATION
6. UNIT PRICING
7. CHANGE FROM PURCHASE
8. NEBS CHECK PRINTER
9. DAYS BETWEEN DATES
10. MORTGAGE AMORTIZATION TABLE
11. INVENTORY CONTROL
12. PORTFOLIO VALUE COMPUTATIONS
13. VALUE OF A SHARE OF STOCK
14. SALES RECORD KEEPING SYSTEM
15. FUTURE VALUE OF AN INVESTMENT
16. EFFECTIVE INTEREST RATE (LOAN)
17. PRESENT VALUE OF A FUTURE AMOUNT
18. RATE OF RETURN-VARIABLE INFLOW
19. RATE OF RETURN-CONSTANT INFLOW
20. REGULAR WITHDRAWAL FROM INVESTMENT
21. STRAIGHT LINE DEPRECIATION
22. SUM OF DIGITS DEPRECIATION
23. DECLINING BALANCE DEPRECIATION
24. BREAK EVEN ANALYSIS
25. SALVAGE VALUE OF INVESTMENT
26. PAYMENT ON A LOAN
27. FUTURE SALES PROJECTIONS
28. CREDIT CARD FILE
29. ECONOMIC ORDER QUANTITY (EOQ) INVENTORY MODEL
30. VALUE OF HOUSE CONTENTS
31. TEXT EDITOR
32. MONTHLY CALENDAR
33. DAY OF WEEK
34. CASH FLOW VS. DEPRECIATION
35. COMPLETE MAIL SYSTEM
36. INTEREST RATE ON A LEASE

BUSINESS

PERSONAL FINANCE

STATISTICS AND MATHEMATICS
37. RANDOM SAMPLE SELECTION
38. ANGLO-METIC CONVERSION
39. MEAN, STANDARD DEVIATION, MAXIMUM AND MINIMUM
40. SIMPLE LINEAR REGRESSION
41. MULTIPLE REGRESSION ANALYSIS
42. GEOMETRIC REGRESSION
43. EXPONENTIAL REGRESSION
44. SIMPLE MOVING AVERAGE
45. SIMPLE T-TEST
46. CHI-SQUARE TEST
47. NORMAL PROBABILITIES
48. BINOMIAL PROBABILITY
49. POISSON PROBABILITY
50. MATRIX ADDITION AND SUBTRACTION
51. MATRIX TRANSPOSE
52. MATRIX INVERSE
53. MATRIX MULTIPLICATION
54. SOLUTION OF SIMULTANEOUS EQUATIONS
55. QUADRATIC FORMULA
56. LINEAR EQUATION SOLUTIONS
57. ROOT HALF INTERVAL SEARCH
58. ROOTS OF POLYNOMIALS
59. ROOTS-NEWTON'S METHODS
60. PRIME FACTORS OF INTEGER
61. LEAST COMMON DENOMINATOR
62. RADIAN-DEGREE CONVERSION
63. NUMERICAL INTEGRATION

STATISTICS

MATH

UTILITIES
64. QUICK SORT ROUTINE
65. PROGRAM STORAGE INDEX
66. MULTIPLE CHOICE QUIZ BUILDER
67. FORM LETTER WRITER
68. SHELL SORT
69. CASSETTE LABEL MAKER
70. CODES MESSAGES
71. MERGE TWO FILES
72. SORT WITH REPLACEMENT

GRAPHICS
73. DRAWS BAR GRAPH
74. DRAWS HISTOGRAM
75. MOVING BANNER DISPLAY

GAMBLING AND GAMES
76. RANDOM SPORTS QUIZ
77. GOVERNMENT QUIZ
78. HORSE RACE
79. MAGIC SQUARE
80. ARITHMETIC TEACHER
81. HIGH LOW GAMBLE
82. UNSCRAMBLE LETTERS
83. HANGMAN
84. GAME OF NIM
85. RUSSIAN ROULETTE
86. ROULETTE GAME
87. ONE-ARMED BANDIT
88. HIT THE TARGET
89. WALKING DRUNK
90. STATE CAPITAL QUIZ
91. TIC-TAC-TOE
92. DICE GAME
93. LUNAR LANDAR GAME
94. BIORHYTHM
95. HORSE SELECTOR (CLASS CALCULATOR)
96. RANDOM DICE ROLL
97. RANDOM ROULETTE ROLL
98. RANDOM CARD DEALER
99. GUESS THE NUMBER
100. WHITE OUT SCREEN

GAMBLING

INCLUDES 110 PAGE USER MANUAL

GUARANTEED SATISFACTION
30-DAY MONEY BACK GUARANTEE ON ALL SOFTWARE

COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™ ✓9

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

PLEASE SEND ME:

☐ MASTER PAC 100 CASSETTE VERSION $59.95
☐ MASTER PAC 100 DISKETTE VERSION $59.95
☐ MASTER PAC 100 (TRS-80 MODEL II VERSION) $99.95

24 HOUR ORDER LINE
(914) 425-1535

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

★ *All orders processed within 24-Hours*
★ *30-Day money back guarantee on all Software*

CREDIT CARD NUMBER .... EXP. DATE ....

SIGNATURE ....

NAME ....

ADDRESS .... CITY .... STATE .... ZIP ....

*** ADD $3.00 FOR SHIPPING IN UPS AREAS • ADD $4.00 FOR C.O.D. OR NON-UPS AREAS • ADD $5.00 OUTSIDE U.S.A, CANADA & MEXICO ***

See List of Advertisers on page 402

✓See List of Advertisers on page 402

✓See List of Advertisers on page 402

Meet your fellow TRS-80 owner.

Users Groups

Many of you have been waiting for it, and here it is: an international list of TRS-80 users groups. This list is undoubtedly incomplete and some of the addresses are over a year old, but we believe this list to be an excellent representation of what is available to the TRS-80 owner.

80 Microcomputing intends to update this list periodically as a service to our readers. So, if anyone is forming a new group or the group you are in did not make this list, please notify us. Also, if your group has changed its name or is no longer in existence let us know.

Users groups are an excellent way to get to know other people in the microcomputer hobby/business as well as a good resource for help in solving your computer problems. We hope our readers find this list useful.■

STATE	GROUP NAME	ADDRESS	CITY	ZIP	CONTACT
	REGINA OPERATORS OF MICRO SYSTEMS	BOX 1001	REGINA	S4P3B2	WENDY MOFFATT
	BRASILIAN CLUB	RUA SAMBAIBA 516. LEBLON	RIO DE JANEIRO, 2245	BRASIL	DOUGLAS GILSON
AK	ANCHORAGE USERS GROUP	BOX 10, 385 SOUTH STATION	ANCHORAGE	99511	
AL	CENTRAL ALABAMA COMPUTER SOCIETY	2073 REXFORD RD	MONTGOMERY	36116	L GARRISON
AL	G2C3	4307 OLD SHELL RD	MOBILE	36608	G. REAGAN
AL	CENTRAL AL MICRO SOCIETY JUNIOR CHAPTER	RT. #3 BOX 570	MONTGOMER	36110	FREDERICK FERGUSON
AZ	USERS GROUP OF ARIZONA	6218 W. MARLETTE	GLENDALE	85301	ROD PURCELL, PRES.
AZ	PHEONIX USERS GROUP	1850 EAST MARYLAND #27	PHEONIX		
AZ	ARIZONA USERS GROUP	4322 EAST FAIRMONT	PHOENIX	85018	R WARREN
CA	USERS AND ABUSERS	1350 GRANT RD.	SUNNYVALE	94040	RADIO SHACK
CA	CAUSE (SOFTWARE EXCHANGE)	18651 VON KARMAN	IRVINE	92713	SEL HANDLER
CA	USERS GROUP	712-C COUNTRY WOOD	WALNUT CREEK	94598	
CA	PACIFICA USERS GROUP	338 ALIDA WAY #306	SOUTH SAN FRANCISCO	94080	J. STRAZZARINO
CA	REDWOOD EMPIRE USERS GROUP	7136 BELITA AVE.	ROHNERT PARK	94928.	JOHN REVELLE
CA	SOUTH BAY USERS GROUP	3605 PINE AVE	MANHATTEN BEACH	90266	S. NIMS
CA	EAST BAY USERS GROUP	17 ECHO AVE	OAKLAND	94611	
CA	SAN FRANCISCO TRS-80 USERS GROUP	338 ALIDA WAY #306	SOUTH SAN FRANCISCO	94080	JOHN F. STRAZZARINO
CA	HOMEBREW COMPUTER CLUB	BOX 626	MOUNTAIN VIEW	94042	
CA	INLAND COMPUTER SOCIETY	3359 SECOND ST	RIVERSIDE	92501	
CA	SOUTH BAY USERS GROUP	BOX 6302	STANFORD	94305	B. DEVENDORF
CA	TRS-80 FREE PROGRAM EXCHANGE	4418 MORROW RD.	MODESTO	95350	
CA	COUNTRYWIDE USER GROUP	10409 STATE ST.	SOUTH GATE	90280	J. MCGORTEY/S. NIMS
CA	LITTON CALULATOR/COMPUTER CLUB	5500 CANOGA AVE	WOODLAND HILLS	91364	W. BENNETT
CA	MARIN COUNTY USERS GROUP (MCTUG)	45 SELFRIDGE WAY	HAMILTON AFB	94934	PAT MCMAHON
CA	SONOMA COUNTY COMPUTER CLUB	BOX 945	COTATI	94928	
CA	IEEE COMPUTER SOCIETY	5855 NAPLES PLAZA, SUITE 301	LONG BEACH	90803	
CA	MONTEREY.BAY USERS GROUP	1002 HALSEY DR	MONTEREY	93940	CURT HESS
CA	TRS-80 NIBBLERS	2555 HESPERIAN BLVD	HAYWARD	94545	
CA	VALLEY COMPUTER CLUB	3311 WEST THIRD, APT. 1-319	LOS ANGELES	90020	KENNETH YOUNG
CA	PAMONA COMPUTER SOCIETY	4155 OAK HOLLOW RD.	CLAREMONT	91711	
CA	THE FORTH INTEREST GROUP	PO BOX 1105	SAN CARLOS	94070	
CA	ORANGE COUNTY USERS GROUP	3521 E. COMMONWEALTH	FULLERTON	92631	ED FAULK, PRES.
CA	SAN FRANCISCO USERS GROUP	HQ USARRIX	PRESIDIO OF SAN FRAN	94129	P. MCMAHON
CA	VENTURA COUNTY TRS-80	2534 NORTH TEMPLE AVE.	CAMARILLO	93010	CLUB NEWSLETTER
CA	USERS GROUP	7465 HOLLISTON AVE., SUITE 23	GOLETA	93017	
CA	VENTURA COUNTY TRS-80 COMPUTER CLUB	567 WEST LOOP DR.	CAMARILLO	93030	L. STEINMETZ
CA	VALLEY TRS-80 USERS GROUP (VTUG)	19116 NASVILLE ST	NORTHRIDGE	91326	WILLIAM WHITE
CA	SACREMENTO USERS GROUP	1237 BEARD WAY	CARMICHAEL	95608	MIKE MOORE
CA	SAN JOSE USERS GROUP	3490 BON AIR COURT	SAN JOSE	95117	C S SHANKS
CA	ET-3400 USERS GROUP	11231 OAK ST.	EL MONTE	91731	CHARLES VAN DYKE
CA	TRS-80 USERS GROUP OF NAPA	4432 SPRINGWOOD	NAPA	94558	
CA	OASIS USERS GROUP	PHASE ONE SYSTEMS	OAKLAND		FRED BELLAMY
CA	SAN GABRIEL VALLEY TRS-80 USERS GROUP	750 E. 5TH ST. #75	AZUSA	91702	DAN DRESSELHAUS
CAN	AURORA COMPUTER SOCIETY	BOX 4342	EDMONTON, ALBERTA	T5H1R5	BOB HUNTINGFORD
CAN	USERS GROUP OTTAWA	2 LARSEN CT.	KANATO, ONTARIO	K2L1Y8	
CN	KITCHENER-WATERLOO MICRO CLUB	UNIVERSITY OF WATERLOO	WATERLOO, ONT	N2L3G1	
CT	CONNECTICUT COMPUTER CLUB	18 RIDGE COURT WEST	WEST HAVEN	06516	TAYLOR
CT	SOUTHERN NEW ENGLAND COMPUTER SOCIETY	267 WILLOW ST	NEW HAVEN	06511	
CT	NORTH EASTERN BASIC FOUR USER GROUP	22 TOBEY RD.	BLOOMFIELD	06002	JOHN PEARSE
CT	FAIRFIELD COUNTY TRS-80 USERS GROUP	10 RICHLEE RD.	NORWALK	06851	ALAN ABRAHAMSON
CT	CONN MICROISTS	8802 WENDY LANE	WESTPORT	06881	
DC	TCUG	2617 42ND ST NW #2	WASHINGTON	20007	R. DALY
DE	DELAWARE USERS GROUP	1116 PIPER RD	WILMINGTON	19803	TIM J. IHDE
DE	DELAWARE USERS OF MICROS	318B CHAPEL AVE	CLAYMONT	19703	JODIE S. HOBSON
ENG	NORTH LONDON HOBBY COMPUTER CLUB	HOLLOWAY RD.	N. LONDON	N7 8DB	R. BRADBEER/N LONDON P
FL	TBUG-80	PO BOX 247	TAMPA	33602	RAY ECHENBACH
FL	CLUB OF CENTRAL FLORIDA	152 MILL RUN DRIVE	LAKE MARY	32746	SILOM HORWITZ
FL	ORLANDO, FL, COMPUTER CLUB	ADVENTURE INTERNATIONAL	ORLANDO	32802	SCOTT ADAMS
GA	USERS GROUP OF ATLANTA, LTD.	1315 RUSTIC RIDGE DR, N.E.	ATLANTA	30319	BOB GREEN 4/81
GA	CSRA COMPUTER CLUB	BOX 284	AUGUSTA	30903	
GA	USERS GROUP OF ATLANTA, LTD.	3408 CLAIRMONT RD, N.E.	ATLANTA	30319	
GRC	USERS GROUP	CHRISOSTOMOU SMYRNIS 30	NEO PSYHIKO		G. ANTIMISIARIS
IA	DES MOINES USERS GROUP (DM-TUG)	#303 4215 GRAND AVE	DES MOINES	50312	DENNIS R. SOLOMON
IA	MARSHALLTOWN COMPUTER CLUB	1101 SOUTH 2ND AVE	MARSHALLTOWN	50158	D. GROVES

STATE	GROUP NAME	ADDRESS	CITY	ZIP	CONTACT
IA	DES MOINES USERS GROUP (DM-TUG)	4215 GRAND AVE, #303	DES MOINES	50312	BOB HEATHCOTE
IA	TRS-80 USER GROUP	1400 EAST POST B	MARION		STEVEN CREE
IL	COMPUTER RENTAL SERVICE	RR BOX 1138	CRETE	60417	R BEPTIES
IL	PEORIA AREA COMPUTER CLUB	2019 NORTH IDAHO	PEORIA	61604	
IL	CHICAGO TRS-80 USERS GROUP	3950 N.LAKE SHORE DRIVE	CHICAGO	60613	MANNY GARCIA
IL	QUAD CITY COMPUTER CLUB	4211 1/2 7TH AVE.	ROCK ISLAND	61201	
IL	CHICAGO AREA COMPUTER HOBBYIST EXCH	BOX 52	SOUTH HOLLAND	60473	LEAH R O'CONNER
IN	USERS GROUP	2203 CORD ST.	INDIANAPOLIS	46224	J. HILL
IN	GROUP OFF SOUTHWEST INDIANA	BOX 3284	EVANSVILLE	47732	M ANDERSON
IN	INDIANAPOLIS TRS-80 USERS GROUP	2203 CORD ST.	SPEEDWAY	46224	JAMES R. HILL
KS	COMPUTER NETWORK OF KANSAS CITY	7631 BROADMOOR LANE	OVERLAND	66204	
KS	GREATER KANSAS CITY USERS GROUP	8909 WENONGA	LEAWOOD	62206	
KS	KBUG	BOX 1398	WICHITA	67201	
LA	CRESCENT CITY COMPUTER CLUB	PO BOX 1097, U. OF NEW ORLEAN	NEW ORLEANS	70122	JOAN LAMARE
MA	NEW ENGLAND COMPUTER SOCIETY	PO BOX 198	BEDFORD	01730	
MA	ALCOVE COMPUTER CLUB	230 MAIN ST	NORTH READING	01824	
MA	WESTERN MASS COMPUTER CLUB	134 BRECKWOOD CIR.	SPRINGFIELD	01119	D. CHANDLER
MA	CULPRIT/EDP-AUDITOR USER GROUP	20 WILLIAM ST.	WELLESLEY	02181	JAMES BLAKE CULLINANE
MA	NEW ENGLAND SYSTEM/3, INC.	PO BOX 1275	LOWELL	01853	J.R. QUARTARONE
MA	PRIME COMPUTER, INC.	40 WALNUT ST.	WELLESLEY	02181	ROLAND KITTREDGE
MA	USERS MEETING	61 LAKESHORE RD.	NATICK	01760	MILLER
MA	BOSTON COMPUTER SOCIETY	17 CHESTNUT ST	BOSTON	02108	
MA	TRUGEM	61 LAKE SHORE RD	NATICK	01760	R. MILLER
MA	NATIONAL COMPUTER SHOWS	BOX 678	BROOKLINE VILLAGE	02147	
MA	GOSUB TRS80 USERS GROUP	PO BOX 712	WORCESTER	01613	JAMES MERCANTI
ME	AUGUSTA USERS GROUP	BOX 2143	AUGUSTA	04330	
ME	TRS-80 NEWSLETTER	96 DOTHAN ST.	ARLINGTON	02174	
MI	SOUTHEASTERN MICHIGAN COMPUTER	BOX 9578	DETROIT	48202	
MI	CENTRAL MICHIGAN USERS GROUP	938 WILDEWOOD	EAST LANSING	48823	JANE HILBERT
MI	MID-MICHIGAN COMPUTER CLUB	15151 RIPPLE DR.	LINDEN	48451	
MI	FARGO-MOORHEAD COMPUTER CLUB	111 SO MAIN ST.	DILWORTH	56529	T. AUGER
MI	CENTRAL MICHIGAN USERS GROUP	5582 CORAL WAY	HASLETT	48840	G WILLIAMS
MI	EDUCATIONAL RECREATIONAL CLUB (ERCC)	POBOX 325	OWOSSO	48867	JOHN HORVATH
MI	MICROCOMPUTER USERS INTERNATIONAL	1804 WEST 18TH ST	SAULTE STE. MARIE	49783	JACK DECKER 4/81
MN	MINNESTOTA COMPUTER SOCIETY	BOX 35317	MINNEAPOLIS	55435	
MO	ST. LOUIS AREA COMPUTER CLUB	PO BOX 28924	ST. LOUIS	63132	
MS	OXFORD Z-80 USERS GROUP	PO BOX 847	UNIVERSITY	38677	
NC	NATIONAL COMPUTER SOCIETY	PO BOX 41205	FAYETTEVILLE	28304	
NC	USERS NOTES	7554 SOUTHGATE RD.	FAYETTEVILLE	28304	R. GORDON LLOYD
NC	TRS-80 USERS GROUP	7554 SOUTHGATE RD.	FAYETTEVILLE	28304	R. GORDON LLOYD
NH	GOSUB USERS GROUP	346 S. TAYLOR ST.	MANCHESTER	03103	SCOTT.MITCHELL
NH	NORTHERN NEW ENGLAND SOCIETY	PO BOX 69	BERLIN	03570	ALBERT BRUNELLI
NJ	BASIC FOUR USER GROUP, MID-ATLANTIC	PO BOX 2214	CLIFTON	07015	ROBERT MENTUS
NJ	NORTHERN NJ AMATUER COMPUTER CLUB	6 BRYSON ROAD	FAIR LAWN	07410	HERMAN S. SACHS
NJ	AMATEUR COMPUTER CLUB OF NJ	UCTI 1776 RARITAN RD	SCOTCH PLAINS	07076	
NV	NORTHERN NEVADA AMATUER COMPUTER CLUB	PO BOX 9068	RENO	89507	UNSCC
NY	LONG ISLAND COMPUTER ASSN	36 IRENE LANE E	PLAINVIEW	11803	
NY	TRS-80 SPECIAL INTEREST GROUP	RD 2	BLOSSVALE	13308	MIKE TOLLERTON
NY	SAINT PETER'S COMPUTER CLUB	JAMES ST.	ROSENDALE	12472	
NY	INTERNATIONAL SOFTWARE A.G. USERS GROUP	1 NEW YORK PLAZA	NEW YORK	10004	MICHAEL BERMAN, INT NI
NY	ZWEIBRUECKEN MICROCOMPUTER USERS GROUP	TAMMC ARMT DIV.	APO NEW YORK	09052	D. GOBER
NY	CENTRAL NEW YORK TRS-80 USERS GROUP	26 JAMESVILLE VE. J-4	SYRACUSE	13210	PETER MOLLER
NY	CP/M USERS GROUP	164 WEST 83RD ST	NEW YORK	10024	
NY	TRS-80 USERS GROUP	245 MAPLEVIEW RD	CHEEKTOWAGA	14225	
NY	USERS GROUP	RT #1, BOX 8	MILTON	12547	KLAUS RITTENBACK
NY	CP/M USERS GROUP	1651 THIRD AVE.	NEW YORK	10028	
NY	TRS-80 USERS GROUP, CHURCH APPLICATIONS	PO BOX 41	MASONVILLE	13804	MERRILL COOK
NY	ONONDAGA COUNTY	BAKER HALL,ELECTRONIC'S PKWY	SYRACUSE		
NY	USERS GROUP (DIVISION OF C.H.I.P.S.)	407 N. 6TH ST.	FULTON		RICHARD DE FOREST
NY	KINGS BYTE TRS-80 USERS GROUP	1063 EAST 84 ST.	BROOKLYN	11236	MORTY LIBOWITZ
OH	BASIC FOUR USER GROUP, N. OHIO	3203 W. 71ST ST.	CLEVELAND	44102	KENNETH SMITH, ROHCO,
OH	TRS-80 CLUB	8106 QUAILWOOD CT	WEST CHESTER	45069	J LINDSLY
OH	PRIDE USERS ASSOCIATION, INC.	1248 SPRINGFIELD PIKE	CINCINNATI	45215	TIM BRYCE
OH	ACSCO USER GROUP	3521 GRACE ST.	COLUMBUS	43228	JOHN CRAMER
OH	DAYTON AREA USERS GROUP	231 GREEN ST	DAYTON	45402	
OK	TULSA USERS GROUP	PO BOX 1133	TULSA	74101	
OR	PORTLAND USERS GROUP	13485 S.W. DRIFTWOOD	BEAVERTON	97213	J. WARNER
OR	TEKTRONIX, INC.	PO BOX 500	BEAVERTON	97077	PATRICIA KELLY
OR	CENTER FOR THE STUDY OF THE FUTURE	4110 NE ALMEDA	PORTLAND	97212	
PA	ATLANTIC SOFTWARE, INC.	901 LAFAYETTE BLDG., 5TH & CH	PHILADELPHIA	19106	USERS GROUP RELATIONS
PA	AUUA, INC.	FLEET MATERIAL SUPPORT OFF.,	MECHANICSBURG	17055	JAMES STINE, U.S. NAVY
PA	PITTSBURGH AREA COMPUTER CLUB	400 SMITHFIELD ST.	PITTSBURGH	15222	
PA	PHILADELPHIA AREA COMPUTER SOCIETY	PO BOX 432	AUDUBON	19403	THE COMPUTER BUS (PACS
PA	DELAWARE VALLEY COMPUTER CLUB	BOX 551	LEVITTOWN	19058	VICTORIA MILLER
PA	CRANBERRY COMPUTER CLUB	RADIO SHACK STORE, CRANBERRY	CRANBERRY	16319	LOUIS DARVASSY
TN	CHATTANOOGA MICROCOMPUTER CLUB	4429 PAULA LANE	RED BANK	37415	DAVE MCGLUMPHY
TX	ALAMO COMPUTER ENTHUSIASTS	4847 CASTLE SHIELD	SAN ANTONIO	78218	
TX	USERS OF HOUSTON	3723 PURDUE	HOUSTON	77005	BEN TAYLOR
TX	NASA/BAY AREA USERS GROUP	PO BOX 57116	WEBSTER	77598	RAY CONE
TX	USERS OF HOUSTON	6900 S. RICE	BELLAIRE		
TX	KILLEEN USERS GROUP	BOX 510	KILLEEN	76541	
TX	PERMIAN BASIN COMPUTER GROUP	BOX 3912	ODESSA	79760	J. RABENALDT, ECTOR SC
TX	PAN HANDLE COMPUTER SOCIETY	3440 BELL SUITE 226	AMARILLO	79109	
TX	COMPUTER HOBBYIST GROUP OF NORTH TEXAS	BOX 1344	GRAND PRAIRIE	75051	
TX	TRS-80 USERS GROUP	7822 SHADY HOLLOW LANE	SAN ANTONIO	78255	GENE A. KENT
TX	TEXHOMA MICROCOMPUTER ENTHUSIASTS	PO BOX 4391	WICHITA FALLS	76308	COL.J.WESLEY B. TAYLOR
TX	WICHITA VALLEY USERS GROUP	PO BOX 4391	WICHITA FALLS	76308	
TX	TEXAS USERS GROUP	5224 WINIFRED DR	FT. WORTH	76133	
TX	HIGH PLAINS USERS GROUP	POBOX 30545	AMARILLO	79120	TOM WHITTENBURG.
TX	USERS GROUP OF S.W. HOUSTON	3723 PURDUE	HOUSTON	77005	B. TAYLOR
TX	USERS GROUP	1516 DALLAS ST.	KILLEEN	76541	C. MANIS
TX	TRS-80 USERS GROUP	PARK ROW AT COLLINS	ARLINGTON	76010	
VA	TIDEWATER TRS-80 USERS GROUP	BOX 10281	NORFOLK	23513	LES LOGAN
VA	TANDY COMPUTER USERS GROUP	8205 CHIVALRY RD	ANNANDALE	22003	ROD WRIGHT
VA	PRINCE WILLIAM COMPUTER CLUB		WOODBRIDGE		DON BENNETT
VA	AMATEUR RADIO REASEARCH AND DEV. CORP	1524 SPRINGVALE AVE	MCLEAN	22101	
VA	TCUG,INC	P.O.BOX 2826	FAIRFAX	22031	PARSONS PILCHER
VT	NORTHERN VERMONT COMPUTER CLUB		ROXBURY	05669	D. KITSZ
WA	NATIONAL CINCOM USERS GROUP	400 HOUSER WAY NORTH	RENTON	98055	BERT BERTRAM, PACCAR,
WA	USERS GROUP	3825 NORTH 26TH ST.	TACOMA	98400	
WA	USERS GROUP - WASHINGTON	21814 PACIFIC HWY SOUTH - LOT	DES MOINES	98188	WALT NASH
WA	USERS GROUP	SOUTH 3718 MANITO BLVD.	SPOKANE	99203	S. JOHNSON
WA	USERS CLUB--WASHINGTON	PO BOX 7112	TACOMA	98407	SCHMIDT
WA	USERS GROUP OF SPOKANE	S. 3718 MANITO BLVD	SPOKANE	99203	STEVE A. JOHNSTON
WGR	TUG	1M ORTFELD 12	4300 ESSEN 14		W. LUCE
WI	TRI-CO GSD USERS	PO BOX 768	MANITOWOC	54220	THOMAS SITKIEWITZ
WI	MADISON USERS GROUP	430 JEAN ST	MADISON	53703	P DALY
WI	USERS NOTES	61 13TH AVE. SOUTH	ONALASKA	54650	JOHN R. MARSH

Show your students beauty in number systems.

Alternate Course—Part IV

Michael A. Duffin
1507 East Avenue
Berwyn, IL 60402

One of the most difficult concepts to understand in data processing is that of different numbering systems. It is particularly difficult to grasp in an introductory course. Most courses introduce these concepts within the first few weeks of class. Thus, most students are inundated with binary, octal and hexadecimal manipulations within a time period of two to ten days. Then they never see the numbering systems again except on tests.

When I teach numbering systems in "An Introduction to Data Processing" at Morton College in Cicero, IL, I try to spread these ideas over a 12-week period. Since I believe that most people fear mathematics, I make my initial presentations in a nonmathematical fashion.

The first time I introduce binary numbers I use the chart shown in Table 1.

I ask my students if there is anyone who doesn't recognize these characters. (I have a real problem with those that don't.) I then ask them what numbers would be left if we could only use the numbers zero and one. From all of the numbers printed, the only ones that use just the digits 0 and 1 are 0, 1, 10, 11, 100, 101, 110 and 111. I write these numbers above the first line in the chart so that we have the following:

0	1	10	11	100	101	110	111		
0	1	2	3	4	5	6	7	8	9

The next thing to do is to look for a pattern in these numbers. (What is the next number if we follow the same pattern?) I receive some interesting answers the first time this question is asked. These answers range anywhere from 112 to 222. Finally, in a moment of inspiration (or frustration) someone will say 1000. From that point on it is easier to determine the numbers that follow. Once everyone understands the pattern, I usually have them construct a table with the first 40 decimal and binary numbers printed side by side.

Someone always asks, "Why do we use binary numbers?" I usually split this question into two parts. The first part of this question is "Why?" I explain that philosphers have been trying to figure out the answer to this question since the beginning of time. The second part of this question is "Do we use the binary numbering system?" The answer to this is "Yes." I realize that this doesn't tell the student a heck of a lot but it does keep their attention.

The reason we use the binary numbering system in data processing is that the computer uses these numbers. I tell them that a computer is similar to a machine made up of switches, each like a light switch. The only thing you can do with a light switch is turn it on or off. Thus, every time a computer sees a one it turns a switch on. Every time it sees a zero it turns the switch off.

I spread the above information over two or three 15-minute periods. Lets face it, numbering systems are boring. To expect a student to pay attention for an hour or so is a bit much to ask.

Once my students have constructed a decimal-binary chart, simple addition and subtraction problems become easy. For example, if we wish to add 1010 and 11000 we can look up the decimal equivalents of these numbers. We find that 1010 corresponds to decimal 10 and 11000 corresponds to decimal 24. If we look next to decimal 34, our answer is 100010.

When we have had our fill of binary (that doesn't take long) we then look at octal and hexadecimal. To introduce octal numbers we use the same chart that was used to introduce binary numbers. Only this time we choose only those numbers that have the digits 0, 1, 2, 3, 4, 5, 6 or 7 in them (see Table 2).

Then we add these numbers to our decimal-binary chart and we can again perform simple addition and subtraction problems using the above techniques.

Introducing hexadecimal numbers is more confusing to the students because they aren't used to seeing the characters A, B, C, D, E and F used as numbers. However, by adding these numbers to the chart, we can again perform simple addition and subtraction problems. Our chart now looks like Table 3.

The chart is useful for adding or subtracting small numbers in the numbering systems described. However, it does have its limitations. To be specific, it's no good for numbers above decimal 40 without some additional knowledge.

To add or subtract in any numbering system, you must first understand how to add in the decimal numbering system. Let's look at an example using decimal numbers:

$$\begin{array}{r} 869 \\ +732 \\ \hline \end{array}$$

To add these numbers we add each row individually: 9+2=11, so you bring down a 1 and carry a 1. 6+3=9 plus the 1 that was carried is 10, so we bring down the 0 and carry the 1. 8+7=15 plus the 1 we carried is 16. Thus, our final answer is 1601.

The same basic rules are true in almost any numbering system. The only thing that changes is when you have to carry a number, and what follows a number in that numbering system.

To illustrate this, let's look at another example; but this time

let's look at one in the hexadecimal numbering system.

```
 6A4
+3CA
```

Once again we perform our operations column by column. The first thing we have to do is add A and 4. The way my students were told to do this is to look at their charts. We find that a hexadecimal A is the same as a decimal 10 and a hexadecimal 4 is the same as a decimal 4. Since 10 + 4 in decimal is 14, we now look at our chart and see what a decimal 14 is in hexadecimal. We discover that the answer is E. We have no number to carry this time since E is a single-digit answer.

Next, we must add A + C. We find that hex A equals decimal 10 and hex C equals decimal 12. Adding 10 and 12 gives us 22 which is equivalent to hexadecimal 16. Thus, we bring down the 6 and carry the 1.

Finally, we add 6 and 3 plus the 1 we carried. This is the point where many students get screwed up. They forget the numbering system that they're working in. Many students incorrectly give the answer as 10. What they have forgotten is that they are working in hexadecimal and 6 + 3 + 1 is not 10 in hexadecimal. The number which follows 9 in hexadecimal is A.

Thus, the answer to our problem is A6E.

Now I'm sure that these techniques are causing many mathematicians to turn over in their graves because I haven't explained the numbering systems using the powers of the various bases or the Remainder Theorem. However, I am a strong believer in logical deduction and I would much rather have my students discover these formulas for themselves.

One thing that I haven't discussed up to this point is the answer to the question: "Why do we have to learn this junk?" and "What does the computer do with all this stuff?" As I mentioned earlier, most computers are binary machines, and that's why we must learn something about binary.

To explain why we must learn about hex and octal I usually bring in a dump. A dump is the result of a sick program or computer system. When a large system can't figure out what the heck your program is trying to do, it spits it out as a dump. An IBM system prints out its dumps in hexadecimal. A Honeywell system prints out its dumps in octal. The reason they don't print out their dumps in binary (which is what both computers are using) is that hex and octal use much less paper. When the computer gives up trying to execute a sick program, the first thing it does is convert all those little binary numbers into hex or octal.

To illustrate how the computer converts binary to octal and binary to hexadecimal, I use an example. Pick a binary number such as 100001 and split it up into groups of four starting from the right so that we have groups of binary numbers. In this example we get the numbers 10 and 0001. Now we take these individual numbers and look for them in the binary section of our chart. Then we look up the hexadecimal equivalents of these numbers. Thus, binary 10 is equivalent to hex 2 and binary 0001 is equivalent to hex 1. (Leading zeros are always dropped). Sure enough, if we look up hexadecimal 21 on our chart we find that it equals binary 100001.

A similar method converts a binary number to an octal number. Look at example 100001 again. This time split the original binary number into groups of three. So, the numbers we get are 100 and 001. If we look these numbers up in our table we find that a binary 100 is equal

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99
100	101	102	103	104	105	106	107	108	109
110	111	112	113	114	115	116	117	118	119

Table 1. Decimal Numbers

0	1	2	3	4	5	6	7
10	11	12	13	14	15	16	17
20	21	22	23	24	25	26	27
30	31	32	33	34	35	36	37
40	41	42	43	44	45	46	47
50	51	52	53	54	55	56	57
60	61	62	63	64	65	66	67
70	71	72	73	74	75	76	77
100	101	102	103	104	105	106	107

Table 2. Octal Numbers

Program Listing 1

```
10 REM ****************************************************
   **
20 REM *
24 REM *         MORTGAGE PROGRAM PLUS
28 REM * BY  MIKE DUFFIN
32 REM *     USED AS AN EXAMPLE PROGRAM
36 REM *     IN DPR 101
40 REM *     AT MORTON COLLEGE IN BERWYN,ILL.
44 REM *
48 REM ****************************************************
   ****
52 REM *
56 REM *         LIST  OF  VARIABLES
60 REM *
64 REM * X ----- AN ARRAY INDEX WITH 4 SPECIFIC VALUES
68 REM *         1  REQUIRED PAYMENT
72 REM *         2  REQUIRED PAYMENT WITH INFLATION CON
   SIDERED
76 REM *         3  REQUIRED PAYMENT + EXTRA PAYMENT
80 REM *         4  REQUIRED PAYMENT + EXTRA PAYMENT
84 REM *            WITH INFLATION CONSIDERED
88 REM *
92 REM *    VARIABLES THAT USE THE X INDEX
96 REM *    PL----- PRINCIPAL  REMAINING
100 REM *   AP----- ACTUAL PAYMENT FOR A PERIOD
104 REM *   TP----- TOTAL PAYMENTS FOR ALL  PERIODS
108 REM *   AI-----  INTEREST FOR A GIVEN  PERIOD
112 REM *   TI-----  TOTAL INTEREST FOR ALL PERIODS
116 REM *   LP----- LAST PAYMENT FOR A SPECIFIC INDEX
120 REM *   B ----- ACTUAL NUMBER OF PAYMENTS
124 REM *   TP----- ACTUAL PRINCIPAL PAID FOR A PERIOD
128 REM *
132 REM *
136 REM * A..... ORIGINAL AMOUNT BORROWED
140 REM * AY.... PERIOD OF TIME THAT LOAN EXTENDS WHEN
144 REM *        AN ADDITIONAL PAYMENT IS MADE
148 REM * A$.... VARIABLE WITHIN INKEY$ INSTRUCTION
152 REM *        THAT ALLOWS THE USER TO TAKE TIME
156 REM *        TO READ AT  HIS/HER  OWN PACE.
160 REM * CO.... COUNTER IN FOR-NEXT LOOP.
164 REM * DO.... VALUE OF ONE DOLLAR AFTER INFLATION.
168 REM * E..... REQUIRED  MONTHLY PAYMENT.
172 REM * EX.... MONTHLY PAYMENT PLUS EXTRA.
176 REM * IR.... USERS ESTIMATE OF ANNUAL INFLATION RAT
   E.
180 REM * IY.... INTEGER PART OF ACTUAL NUMBER OF YEARS
184 REM *        IT TAKES TO PAY OFF LOAN WHEN AN
188 REM *        ADDITIONAL PAYMENT IS MADE.
189 REM * K..... MONTHLY INFLATION RATE.
190 REM * M..... NUMBER OF MONTHS (REFER TO IY,AY)
191 REM * MI.... MONTHLY INFLATION RATE (REFER TO IR)
192 REM * MP.... USERS EXTRA MONTHLY PAYMENT.
193 REM * P..... INTERVAL BETWEEN PAYMENTS.
194 REM * Q..... FOR-NEXT COUNTER (BASED ON EP)
195 REM * SW.... SIGNIFIES LAST PAYMENT HAS  BEEN MADE.
196 REM * T..... INTEREST CHARGED BY LOAN COMPANY.
197 REM * TØ.... SIGNIFIES ONLY THE TOTALS ARE TO BE
198 REM *        PRINTED.
200 REM * Y.....  ORIGINAL NUMBER  OF YEARS FOR LOAN.
```

Program continues

Program continued

```
202 REM * Z..... ORIGINAL TOTAL NUMBER OF PAYMENTS.
204 REM *
206 REM ***********************************************
    **
220  CLS
240 PRINT "THIS PROGRAM WILL :"
242 PRINT "     1. DETERMINE PAYMENTS FOR A LONG  TERM
    LOAN"
244 PRINT "     2. DETERMINE HOW MUCH YOU WILL SAVE IF"
245 PRINT "        YOU PAY EXTRA"
246 PRINT "     3. CONSIDER THE IMPACT OF INFLATION ON"
247 PRINT "        BOTH TYPES OF PAYMENTS."
248 GOSUB 4000
249 CLS
250 PRINT
260 INPUT "WHAT IS THE AMOUNT BORROWED ($)";A
262 IF A=0 THEN GOTO 260
280 INPUT "            INTEREST CHARGED.":I
282 IF I=0 THEN GOTO 280
300 INPUT "            INTERVAL BETWEEN PAYMENTS (MONTH
    S)";P
302 IF P=0 THEN GOTO 300
320 INPUT "            TERM OF LOAN (YEARS)";Y
322 IF Y=0 THEN GOTO 320
330 INPUT "             EXTRA  PAID   PER MONTH (OPTION
    AL)";MP
334 INPUT "             ESTIMATED  YEARLY INFLATION RAT
    E ";IR
336 EP=MP
340 PRINT
350 INPUT "DO YOU WISH TO SEE THE TOTALS ONLY - INSTEAD
      OF THE ENTIRE TABLE - (1-YES, 0-NO)";T0
380 PRINT
390  IF T0 > 0 THEN GOTO 450
400 PRINT "           OUTSTANDING"
410 PRINT "           PRINCIPAL AT
      PRINCIPAL"
420 PRINT "             END             INTEREST
      REPAID AT"
430 PRINT "PERIOD      OF PERIOD        AT END OF PER
      END OF PER"
440 PRINT
445 'INITIAL VALUES'
450 Z=(Y*12)/P
460 K=(I*(P/12))/100
470 E=A*K/(1-1/(1+K)[Z)
472 E=INT(E*100 +.5)/100
480 DO=1
490 MI=IR/12/100
492 DEFDBL P,L
500 PL(1)=A: PL(2)=A: PL(3)=A: PL(4)=A
510 AP(1)=E:AP(2)=E:AP(3)=E+MP:AP(4)=E+MP
520 TP(1)=0:TP(2)=0:TP(3)=0:TP(4)=0
530 TI(1)=0 : TI(2)=0 : TI(3)=0 : TI(4)=0
540 FOR CO = 1 TO Z
550     IF MP=0 THEN Q=2 ELSE Q=4
560     IF PL(3)<= 0 THEN Q=2
570     FOR X=1 TO Q
580         GOSUB 1000
590     NEXT X
600     IF MP=0 THEN X=1 ELSE X=3
601    IF SW=1 THEN GOTO 630
602     IF LP(X)=0 THEN SW=1
604     IF T0>0 THEN GOTO 630
610     GOSUB 2000
620     GOSUB 3000
630     DO=DO - DO*MI
640 NEXT CO
644 REM  ***********************************************
648 REM *
652 REM *   THE FOLLOWING STATEMENTS WRITE  THE
656 REM *       SUMMARY REPORTS.
660 REM
664 REM ***********************************************
    **
684 GOSUB 4000
686 GOSUB 5000
688 CLS
696 PRINT "IF YOU MADE ONLY THE REQUIRED MONTHLY PAYMEN
    T OF "
700 PRINT "$"E
704 PRINT "THEN YOUR TOTAL INTEREST PAYMENT WAS "
708 PRINT "$"TI(1)"  AND THE TOTAL PRINTCIPAL REPAID W
    AS "
712 PRINT "$"A"  THUS THE TOTAL AMOUNT PAID WAS $"TP(1)
716 PRINT
720 GOSUB 4000
722 IF IR=0 THEN GOTO 740
724 PRINT "DUE TO THE "IR"% YEARLY INFLATION RATE THE T
    OTAL "
728 PRINT "AMOUNT ("TP(1)") IS ACTUALLY WORTH $"TP(2)
732 PRINT "OF TODAYS DOLLARS."
736 GOSUB 4000
740 IF MP=0 THEN GOTO 816
744 PRINT "BY PAYING $"E+MP"INSTEAD OF $"E" PER MONTH"
748 PRINT "YOU PAID $"TP(3)" INSTEAD OF $"TP(1)
752 AY=B(3)/12
756 IY=INT( B(3)/12)
760 IF AY=IY THEN GOTO 772
764 M=INT((AY-IY)*12)
768 AY=IY
772 PRINT
776 GOSUB 4000
780 PRINT "THE LOAN WILL BE REPAID"
784 PRINT "IN "AY" YEARS AND "M"MONTHS"
788 PRINT "INSTEAD OF "Y" YEARS"
792 PRINT "BY PAYING $"E+MP" PER MONTH"
796 PRINT "INSTEAD OF "E" PER MONTH."
800 GOSUB 4000
802 IF IR=0 THEN GOTO 832
804 PRINT "HOWEVER WITH THE "IR"% YEARLY"
808 PRINT "INFLATION RATE $"TP(3)" IS WORTH"
812 PRINT "$"TP(4)" IN TODAYS DOLLARS."
816 PRINT
818 IF IR=0 AND MP=0 THEN GOTO 832
820 PRINT "OK? WELL, LET'S SUM THIS UP."
824 PRINT
828 GOSUB 4000
832 CLS
840 PRINT " ";TAB(20);"REQUIRED-WITH EXTRA"
842 IF MP=0 THEN EX=0 ELSE EX=E+MP
```

Program continues

Program continued

```
844 PRINT "MONTHLY PAYMENT       $"E;TAB(29);"/$"EX
846 IF MP=0 THEN TP(3)=0
848 PRINT "TOTAL PAYMENT         $"TP(1);TAB(29);"/$"TP(3
    )
850 IF IR=0 THEN GOTO 860
852 PRINT "VALUE OF TOTALS       $"TP(2);TAB(29);"/$"TP(4
    )
856 PRINT "(IN TODAYS DOLLARS)"
860 PRINT "LOAN REPAID IN ";TAB(21);Y;TAB(24);"YEARS/ "
    AY;TAB(34);"YRS"M;TAB(39);"MO"
864 PRINT
866 IF MP=0 THEN GOTO 990
868 PRINT "AMOUNT SAVED OR LOST     $"INT((TP(1)-TP(3))
    *100 + .5)/100
870 IF IR=0 THEN GOTO 880
872 PRINT "ACTUAL AMOUNT SAVED"
876 PRINT "OR LOST("IR"% INFLATION) $"INT((TP(2)-TP(4))
    *100+.5)/100
880 PRINT "NOTE: NEGATIVE VALUES MEAN YOU WILL BE"
884 PRINT "       PAYING ADDITIONAL MONEY."
990 END
992 REM ***********************************************
    ****
993 REM * THE FOLLOWING  ROUTINE DETERMINNES  THE
994 REM *     ACTUAL INTEREST (AI),TOTAL  INTEREST  (TI
    )
995 REM *     ACTUAL PAYMENT (AP), TOTAL PAYMENT (TP)
996 REM *     AND DETERMINES IF WE HAVE THE LAST
997 REM *     PAYMENT (LP) FOR ALL VALUES OF X.
998 REM ***********************************************
    *****
1000 AI(X)=K*PL(X)
1001 AP(X) = E
1002 IF X>2 THEN AP(X)=G+MP
1006 LP(X)=PL(X)
1008 IF LP(X)=0 THEN GOTO 1070
1009 IF PL(X)<AP(X) THEN AP(X)=AI(X) :GOSUB 6000:GO
    020
1010 PL(X)= PL(X) - (AP(X) - AI(X))
1012 IF PL(X)<=0 OR CO=Z THEN GOSUB 6000
1020 IF X/2 <> INT(X/2) THEN GOTO 1050
1030 AI(X) = AI(X) * DO
1040 AP(X) =  AP(X) * DO
1050 TP(X) = TP(X) + AP(X)
1060 TI(X) = TI(X) + AI(X)
1070 REM
1080 RETURN
1090 REM *********************************************
1092 REM *
1094 REM * THE FOLLOWING ROUTINE ROUNDS THE
1096 REM *     PRINCIPAL LEFT (PL),ACTUAL INTEREST THI
    S
1098 REM *     PERIOD (AI),AND THE PRINCIPAL PAID
2000 REM *     THIS PERIOD (TP) TO 2 DECIMAL PLACES.
2002 REM *
2004 REM **********************************************
    ****
2010 AI(X) = INT( AI(X) * 100 + .5)/100
2020 TP = INT( ( AP(X) - AI(X) ) * 100 + .5 )/100
2030 RETURN
2800 REM **********************************************
    *
2810 REM *
2820 REM *  THE FOLLOWING ROUTINE PRINTS THE VALUES
2830 REM *      FOR THE CURRENT PERIOD (CO), THE
2840 REM *      PRINCIPAL LEFT (PL) AND
2850 REM *     THE INTEREST THIS PERIOD (AI) AND
2860 REM *     THE  PRINCIPAL PAID THIS PERIOD (TP)
2870 REM *
2880 REM **********************************************
    **
3000 REMARK
3010 PRINT CO;TAB(11);INT( PL(X)*100 + .5)/100;TAB(29);
    AI(X);TAB(48);TP
3090 RETURN
3900 REM *********************************************
3910 REM *
3920 REM * THE FOLLOWING ROUTINE ALLOWS THE USER
3930 REM *      TO KEEP A SCREEN UNTIL  HE OR SHE
3949 REM *      IS READY TO CHANGE IT.
3950 REM *
3969 REM **********************************************
    *
4000 REM
4010 PRINT "PRESS ANY KEY TO CONTINUE."
4020 A$= INKEY$: IF A$="" THEN GOTO 4020
4022 PRINT
4030 RETURN
4900 REM ********************************************
4910 REM *
4920 REM *  THE FOLLOWING ROUTINE ROUNDS
4930 REM *    AI,TI,TP AND E FOR ALL VALUES OF X.
4940 REM *
4950 REM *********************************************
5000 REM
5010 FOR X= 1 TO 4
5020      AI(X)=INT( AI(X)*100 +.5)/100
5030      TI(X)=INT( TI(X)*100 + .5)/100
5040      TP(X)=INT( TP(X)*100 + .5)/100
5042    E=INT(E*100 +.5)/100
5050 NEXT X
5060 RETURN
5900 REM **********************************************
    **
5910 REM *
5920 REM *  THE FOLLOWING ROUTINE DETERMINES
5930 REM *      THE FINAL VALUES OF AP,LP
5940 REM *      AND LP. IT ALSO SETS THE
5950 REM *      ENDING PERIOD (B) TO THE CURRENT
5969 REM *      VALUE OF THE PERIOD (CO) FOR
5970 REM *      FOR ALL VALUES OF X.
5980 REM *
5990 REM **********************************************
    *****
6000 REM
6002 IF PL(X)>0 THEN GOTO 6050
6010 AP(X)=PL(X)-AI(X)
6040 GOTO 6060
6050 AP(X)=AP(X)+PL(X)
6060 PL(X)=0
6062 B(X)=CO
6070 LP(X)=0
6100 RETURN
```


See List of Advertisers on page 402

to an octal 4 and a binary 1 is equivalent to an octal 1. If we look up an octal 41 in our table we find miraculously that it is equivalent to a binary 100001.

This method will always work so that if we have a binary number such as 110001101 which is too large for our table we can still convert it to octal or hex. The numbers 1, 1000, 1101 are equivalent to 1, 8, D in hex, so binary 110001101 equals 18D in hex. The numbers 110, 001, 101 are equivalent to 5, 1, 3 in octal, so binary 110001101 is equal to 513 in octal.

For those of you who would like to know why this works, we have to explain a little bit about the three numbering systems. First of all binary is a two-digit numbering system (0,1) which means that every binary number is actually some power of 2. Octal is an eight-digit numbering system (0–7) so every octal number is a multiple of a power of eight, and hexadecimal is a 16-digit numbering system (0–F) which means that every hexadecimal number is related to a power of 16. Now, if we think about these statements, we remember that 8 is a power of 2 and so is 16. (2 raised to the third power is 8 and 2 raised to fourth power is 16.) Although this doesn't entirely explain why the above technique works, I think some of the mathematicians out there are turning back around.

The program that I have included at the end of this article is one I call Mortgage Program Plus. The main body of it is virtually the same as many other mortgage programs. It involves no graphics nor does it use any unusual Basic capabilities. I use it with those students who feel that learning about computers has to involve a lot of pain, agony and number crunching. It uses a whole slew of variables. The uses of these variables are all explained within the program.

The difference between this program and most mortgage programs is that it takes the inflation rate into account. I strongly urge those of you who have loans and are considering paying them off early or those of you who are paying a little bit more than the required payment towards your mortgage to run this program first. By paying a little bit more than the required payment you will pay off your loan quicker. However, you may be losing money because of the inflation rate.

As you might have already guessed, this program has nothing to do with numbering systems. However, numbering systems are an intricate part of all programming activities. For those involved with assembler programming and those of you who will be required to interpret dumps, I hope that this article will be useful.

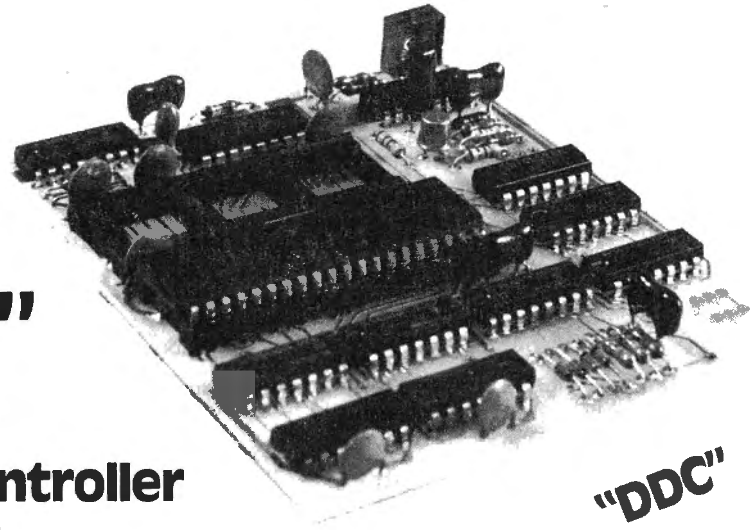
For those who are mathematicians I think that you will have a good time dissecting the mortgage program. For those who are teaching data processing, I hope I have provided some techniques that you will find useful in your classes. If you are data processing students. . .good luck! ■

Decimal	Binary	Hexadecimal	Octal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17
16	10000	10	20
17	10001	11	21
18	10010	12	22
19	10011	13	23
20	10100	14	24
21	10101	15	25
22	10110	16	26
23	10111	17	27
24	11000	18	30
25	11001	19	31
26	11010	1A	32
27	11011	1B	33
28	11100	1C	34
29	11101	1D	35
30	11110	1E	36
31	11111	1F	37
32	100000	20	40
33	100001	21	41
34	100010	22	42
35	100011	23	43
36	100100	24	44
37	100101	25	45
38	100110	26	46
39	100111	27	47
40	101000	28	50

Table 3. Chart of Decimal, Binary, Hexadecimal and Octal Numbers

✓See List of Advertisers on page 402

Aerocomp's Proven Best-By Test! The "DDC"



Double Density Controller

★ Technical Superiority

At last! A double density controller for Model I with HIGHER PROBABILITY OF DATA RECOVERY THAN WITH ANY OTHER DOUBLE DENSITY CONTROLLER ON THE MARKET TODAY! The "DDC" from Aerocomp. No need to worry about the problems that keep cropping up on existing products. AEROCOMP'S new analog design phase lock loop data separator has a wider capture window than the digital types currently on the market. This allows high resolution data centering. The finest resolution available with digital circuitry is 125 ns (nano seconds). The "DDC" analog circuit allows infinately variable tuning. Attack and settling times are optimum for 5-1/4 inch diskettes.
The units presently on the market use a write precompensation circuit that is very "sloppy". Board to board tolerance is extremely wide - in the order of $\pm$100 ns. The "DDC" is accurate to within $\pm$ 20 ns.
The bottom line is state of the art reliability!

★ Test Proven

Tests were conducted on AEROCOMP'S "DDC", Percom's "Doubler A"* and "Doubler II"* and LNW's "LNDoubler"** using a Radio Shack TRS80*** Model I, Level 2, 48 K with TRS80 Expansion Interface and a Percom TFD100* disk drive (Siemens Model 82). Diskette was Memorex 3401. The test diskette chosen was a well used piece of media to determine performance under adverse conditions. The various double density adapters were installed sequentially in the expansion interface.
The test consisted of formatting 40 tracks on the diskette and writing a 6DB6 data pattern on all tracks. The 6DB6 pattern was chosen because it is recommended as a "worst case" test by manufacturers of drives and diskettes. An attempt was then made to read each sector on the disk once - no retrys. Operating system was Newdos/80, Version 1.0, with Double Zap, Version 2.0. Unreadable sectors were totalled and recorded. The test was run ten times with each double density controller and the data averaged. Test results are shown in the table.

★ Features

TRS80 Model I owners who are ready for reliable double density operation will get (1) 80% more storage per diskette, (2) single and double density data separation with far fewer disk I/O errors, (3) single density compatibility and (4) simple plug-in installation. Compatible with all existing double density software.

★ Value

$149.95 for the BEST double density controller on the market.
$189.95 for "DDC" complete with DOSPLUS 3.3D
$239.95 for "DDC" complete with LDOS

★ TEST RESULTS ★

MFR & PRODUCT	SECTORS LOCKED OUT (AVG)
AEROCOMP "DDC"	0
PERCOM "DOUBLER II"	18
PERCOM "DOUBLER A"	250
LNW "LNDOUBLER"	202

Note: test results available upon written request. All tests conducted prior to 8-25-81

Aerocomp's 14 day money back guarantee applies to hardware only.
Specials will be prorated. Shipping $2.00 in Cont. US. See opposite page for details.

THE FIRST PRODUCTION RUN WILL GO FAST - ORDER TODAY!

Data Separators

The advances that make the "DDC" great are incorporated in the new AEROCOMP Single Density Data Separator ("SDS") and Double Density Data Separator ("DDS").

★ Has your original manufacturer left you holding the bag?

If you already own a Percom "Doubler A", "Doubler II" or LNW "LNDoubler", the AEROCOMP "DDS" will make it right. Look at the test results:

MFR. & PRODUCT	SECTORS LOCKED OUT	
	WITHOUT "DDS"	WITH "DDS"
PERCOM "DOUBLER II"	18	1
PERCOM "DOUBLER A"	250	0
LNW "LNDOUBLER"	202	0

Note: Same test procedures as "DDC".
* Trademark of Percom Data Co.
** Trademark of LNW
*** Trademark of Tandy Corporation

Plugs directly into your existing Double Density Controller.

★ "DDS"..... $49.95
(Use 1791 chip from your DD Controller)

★ "DDS" with disk controller chip included....... **$79.95**

★ Disk controller chip..... $34.95

(Shipping $2.00 Cont. US - see opposite page for details)

Do you need a Single Density Data Separator?

The internal data separator in the WD1771 chip (R/S Expansion Interface) is NOT recommended by WD for reliable data transfer. Do you have any of these problems: Lost data, tracks locked out, CRC errors, disk retry? YOU NEED ONE!

★ "SDS"...... $29.95
(For Mod. I; shipping $2.00)

387

See opposite page▸▸▸▸▸▸

See List of Advertisers on page 402

Tie a few tapes together.

Merge For Level II

William J. Dalesandry Jr.
355 Rockfield Rd.
Pittsburgh, PA 15243

Until now, Merge was only available to DOS users, but now you can have it on Level II. Merge offers an extremely useful additional command statement to an already powerful repertoire of commands available. The program is also compatible with Microsoft's Level III cassette Basic.

Normally when you CLOAD a program, it writes over any program already in memory and the resident program is lost. This new command, used either from the keyboard in command mode or as a program statement, loads two or more Basic programs into memory at the same time. The second program and all subsequent programs will merge with the root program or programs to form one contiguous program.

Because the program loads into memory at location 28000 (6D60H) then relocates itself to the bottom of Basic's user RAM area, it is compatible with Level III, Level II and other Basic extension programs. This relieves the inconvenience of reserving space in high memory. It requires no modification because of the size of your system. Basic starts at the same place in a 4K system as it does in a 48K system and because it requires no memory to be reserved, you can load it at any time without answering the question memory size?

Merge is such a useful command I cannot understand why it wasn't included in Level II or Level III.

Save Programming Time

Considerable programming time can be saved by keeping a cassette tape file of commonly used subroutines and merging them into your programs as you need them.

One subroutine that complements Merge is shown in Program Listing 3. It computes and displays the length of any program currently in memory. If several programs have been merged, the display will reflect the total length of the composite program. Merge the subroutine into memory and run or GOTO 60000.

The usual ?15562-MEM doesn't always yield the same number. The number displayed before running a program is not the same as the number displayed after running it. Any program, run either from the keyboard in command mode or through the interpreter, requires about 10 bytes of overhead. Therefore, I subtracted from 15562, not the usual 15572 (I have a 16K system).

The subroutine first gets the start-of-variables address, which is the last byte of the resident Basic program plus two. It then subtracts the start-of-Basic address, two bytes at the end of a program, and its own length. The result of this computation is then sent to the CRT screen and printer. If your system does not include a printer, delete the LPRINT statement.

The Game Sea

You may be drowning in a sea of game tapes. Why not merge several similar games into one program? Just write a short subroutine that asks the player which game he or she wants to play. Load the subroutine and merge all your similar games onto it.

Merge can be used the same as a CLOAD to load the root program, so you can store this subroutine with your other subroutines.

Another useful feature of Merge is that you can add and delete data lines quickly without entering them by hand.

Modular Programming

One use for the routine, possibly the most valuable, allows you to develop programs in short modules. The modular approach will save time loading and saving during the program development.

Modular programming techniques also allow easier maintenance and smoother program flow. Program sections will no longer seem to blend into each other.

How It Works

The assembly language Program Listing 1 was written in standard Zilog Z-80 mnemonics for a Level II system. It can be assembled as shown or poked in one byte at a time using T-Bug or a similar debug monitor. The program is initially located in an area of memory that doesn't interfere with T-Bug. By using the #J 6D60 command, you can relocate the program below T-Bug, but merging in any program over 20 bytes will write over a section of T-Bug.

If you do not have access to an assembler or debug monitor, you can use the Basic program shown in Program Listing 2 to load it. The merge and relocate routines are contained in decimal form in the data lines 100 to 190. Line 10 POKEs the program into the proper locations, and line 20 deletes the program, opening Basic user RAM for our use.

Merge is written in two modules. The first module relocates the merging module to the bottom of Basic's user RAM area. After relocation, this module is scrapped by letting it drift in the area of variables and strings. The second module is the actual Merge routine.

Normally, when you type and enter Merge an ?L3 error message is displayed on your CRT screen. However, Level II Basic was clearly written to allow for all kinds of expansions.

The various commands used by Level II are stored in memory in the form of a one byte code number. In this case, 168 (A8H) = MERGE. (For a list of these codes see "New Restored" and Hidden Codes & Missing Chips" in the January 1980 issue of *80 Microcomputing*.)

When Basic encounters one of these code bytes, it goes to a vector list, which in effect is a GOTO table to find out where the appropriate routine is stored. The jump vector (or GOTO) for Merge, stored at memory locations 16779 (418BH) to 16780 (418DH) normally points to the ?L3 error routine at location 012DH for Level II and 4A07H for Level III. We are going to change this jump vector to point to our Merge routine, bypassing the ?L3 error message.

Level II also maintains a list of addresses used in the normal operation of its business. The list is stored in the reserved RAM area just below Basic RAM. Table 2 is a partial list of these addresses. The address at 16548 (40A4H) and 16549 (40A5H) is where the first byte of a Basic program will start loading and one at 16633 (40F9H) and 16634 (40FAH) where variable storage starts.

The Relocation Module

Lines 00230 and 00240 get the start-of-Basic pointer from locations 40A4H and 40A5H and plug it into locations 418CH and 418DH as our new jump vector.

Basic user RAM starts at 17129 (42E9H) for Level II and at 22208 (56C0H) for Level III. Because of this difference in starting locations, we must calculate the distance from where the program is to where we want it to be. The displacement value is computed in lines 00260 to 00290 by subtracting (SBC) the Basic-start-address, already in register pair DE, from the unrelocated address of the Merge module, which we put into register pair HL in line 00260. The result of this subtraction is automatically put into register HL. Save the address of the Merge module for use later by pushing it into the stack in line 00280.

Lines 00300 and 00310 transfer the displacement value from register pair HL to register pair BC. Then in lines 00320 to 00370 we get the address of the LDBYTE subroutine and subtract the displacement value to get the new LDBYTE address. Then plug it into the subroutine calls in lines 00660, 00690, 00720 and 00840. A call in assembly language is the same as a GOSUB in Basic.

Relocation of the Merge module data block will be done for us by one of those elegant commands for which the Z-80 microprocessor is famous. The LDIR command (LDIR stands for load-increment repeat) takes care of all the details of a complete block move.

It works like this: The byte of the data located in the address pointed to by the HL register pair is moved to the address pointed to by the DE register pair. One is added to both the HL register and the DE register pair and one is subtracted from the BC register pair. The BC register pair is used as a down-counter. We set it to the length of the Merge module (80 bytes = 50H) in line 00390. BC is then checked for zero. If it has not downcounted to zero, the cycle is repeated until a zero is found. At that time, both the HL register pair and the DE register pair are incremented one more time and the program falls through to the next program line.

```
                 00100 ;  ************************************************
                 00110 ;  *                                              *
                 00120 ;  *               LEVEL 2 MERGE                  *
                 00130 ;  *          BY BILL DALESANDRY 1980             *
                 00140 ;- *                                              *
                 00150 ;  ************************************************
6D60             00160           ORG     6D60H      ; /28000  (131 BYTES)
40AB             00170 STORE     EQU     40ABH      ;THIS LOCATION IS RND SEED
                 00180 ;  **********************************************
                 00190 ;  *                                            *
                 00200 ;  *    THIS BLOCK IS THE RELOCATION MODULE     *
                 00210 ;  *                                            *
                 00220 ;  **********************************************
6D60 ED5DA440    00230 RELOC     LD      DE,(40A4H);PUT BASIC START IN DE
6D64 ED538C41    00240           LD      (418CH),DE;  AND PLUG IT IN AS NEW
                 00250·                             ; DOS VECTOR ADDRESS
6D68 21946D      00260           LD      HL,MERGE   ;PUT ADDRESS OF UNRELOCATED
                 00270                              ; MERGE PROGRAM IN HL AS SOURCE
6D6B E5          00280           PUSH    HL         ; ADDRESS AND SAVE FOR LATER
6D6C ED52        00290           SBC     HL,DE      ;HL NOW HAS DISPLACEMENT VALUE
6D6E E5          00300           PUSH    HL         ; TRANSFER IT
6D6F C1          00310           POP     BC         ; TO BC
6D70 21D56D      00320           LD      HL,LDBYTE ;PUT ADDRESS OF CALL IN HL
6D73 ED42        00330           SBC     HL,BC      ; SUBTRACT DISPLACEMENT
6D75 22AC6D      00340           LD      (NEXT+1),HL;PUT CORRECTED
6D78 22B16D      00350           LD      (LOAD+1),HL; CALL ADDRESS
6D7B 22B96D      00360           LD      (LOOK+1),HL;  AT FOUR
6D7E 22CC6D      00365           LD      (ZERO+1),HL;   CALL LOCATIONS
6D81 E1          00370           POP     HL         ;POINT HL TO MERGE LOCATION
                 00380                              ; (BASIC START IS IN DE)
6D82 015000      00390           LD      BC,0050H   ;SET COUNTER FOR 80 BYTES
6D85 EDB0        00400           LDIR               ; MOVE TO NEW LOCATION
6D87 ED53A440    00410           LD      (40A4H),DE;DE NOW POINTS TO LAST PROGRAM
                 00420                              ; LINE+1 -THIS IS NEW BASIC START
6D8B 13          00430           INC     DE         ;COUNT 2 MORE TO GET
6D8C 13          00440           INC     DE         ; NEW START OF VARIABLES
6D8D ED53F940    00450           LD      (40F9H),DE;  PLUG IT IN
6D91 C37A2C      00460           JP      2C7AH      ;A BIT OF HOUSEKEEPING AND
                 00470                              ; WE HAVE A LIFTOFF
                 00480 ;  **********************************************
                 00490 ;  *                                            *
                 00500 ;  *      THIS BLOCK IS THE MERGE PROGRAM       *
                 00510 ;  *                                            *
                 00520 ;  **********************************************
6D94 CD9302      00530 MERGE     CALL    0293H      ;TURN ON CASSETTE, READ LEADER,
                 00540                              ;FIND SYNC BYTE (A5), TURN ON **
6D97 0604        00550           LD      B,04H      ;SKIP NEXT FOUR BYTES
6D99 CD3502      00560 SKIP      CALL    0235H      ; (3-D3 HEADERS + FILE NAME)
6D9C 10FB        00570           DJNZ    SKIP       ; FILE NAME IS IN A
6D9E 323E3C      00580           LD      (3C3EH),A ;DISPLAY FILE NAME (LEFT *)
6DA1 2AF940      00590           LD      HL,(40F9H);GET VARIABLES  POINTER
6DA4 2B          00600           DEC     HL         ; SUBTRACT 2 TO GET THE STARTING
6DA5 2B          00610           DEC     HL         ; ADDRESS OF THE PROGRAM TO BE
6DA6 54          00620           LD      D,H        ; MERGED AND SAVE IT IN DE
6DA7 5D          00630           LD      E,L        ; AS LAST 'NEXT LINE POINTER'
6DA8 2B          00640           DEC     HL         ; SET HL TO START ADDRESS -1
6DA9 0E04        00650 FIRST4    LD      B,04H      ;GET AND LOAD FIRST
6DAB CDD96D      00660 NEXT      CALL    LDBYTE     ; 4 BYTES OF LINE (NEXT LINE
                 00670                              ; ADDRESS AND LINE NUMBER)
6DAE 10FB        00680           DJNZ    NEXT       ; - LOOK FOR TWO ZEROS -
6DB0 CDD96D      00690 LOAD      CALL    LDBYTE     ;GET AND LOAD A BYTE
6DB3 20FB        00700           JR      NZ,LOAD    ;IF NOT ZERO LOAD NEXT BYTE
6DB5 CD2C02      00710           CALL    022CH      ;IF ZERO -END OF LINE- BLINK *
6DB8 CDD96D      00720 LOOK      CALL    LDBYTE     ; THEN LOOK FOR A SECOND ZERO
                 00730                              ; (CP USED LATER)
6DBB E5          00740           PUSH    HL         ;SAVE CURRENT ADDRESS POINTER
6DBC 22AB40      00750           LD      (STORE),HL;STORE ADDRESS POINTER, (HL)
6DBF 21AB40      00760           LD      HL,STORE   ; AND POINT HL TO IT
6DC2 EDA0        00770           LDI                ;MOVE ADDRESS OF CURRENT LINE
6DC4 EDA0        00780           LDI                ; TO FIRST 2 BYTES OF LAST LINE
6DC6 E1          00790           POP     HL         ;GET CURRENT MEMORY ADDRESS
6DC7 54          00800           LD      D,H        ; AND PUT IT
6DC8 5D          00810           LD      E,L        ; IN DE
6DC9 20DE        00820           JR      NZ,FIRST4 ;(Z FLAG STILL RESET)
                 00830                              ; IF NZ LOAD NEXT LINE
6DCB CDD96D      00840 ZERO      CALL    LDBYTE     ;LOAD NEXT BYTE
6DCE 0603        00845           LD      B,03H      ; FIRST4 IS NOW FIRST3
6DD0 20D9        00850           JR      NZ,NEXT    ; NOT THIRD ZERO, LOAD NEXT LINE
                 00860                              ; END OF PROGRAM
6DD2 23          00870           INC     HL         ;THIRD ZERO +1 IS THE
6DD3 22F940      00880           LD      (40F9H),HL; NEW VARIABLES START ADDRESS
6DD6 C37A2C      00890           JP      2C7AH      ;ADJUST MEMORY SIZE POINTERS
                 00900                              ; TURN OFF CASSETTE AND GO HOME
6DD9 CD3502      00910 LDBYTE    CALL    0235H      ;GET A BYTE
6DDC 23          00920           INC     HL         ;INCREMENT ADDRESS POINTER
6DDD 77          00930           LD      (HL),A     ; AND LOAD THE BYTE
6DDE FE00        00940           CP      00H        ;IF ZERO, SET Z FLAG
6DE0 C9          00950           RET                ; AND BACK AGAIN
6DE1 00          00960           DEFB    00H        ;THESE ARE THE THREE
6DE2 00          00970           DEFB    00H        ; ZEROS WHICH INDICATE THE
6DE3 00          00980           DEFB    00H        ;  END OF THIS PROGRAM
6D60             00990           END     RELOC
00000 TOTAL ERRORS

FIRST4   6DA9
SKIP     6D99
ZERO     6DCB
LOOK     6DB8
LOAD     6DB0
NEXT     6DAB
LDBYTE   6DD9
MERGE    6D94
RELOC    6D60
STORE    40AB
```

Program Listing 1.

Our Merge module has now been moved to low memory, including the three zeros indicating the end of a program. The routine sometimes crashes without them.

Register pair HL now points to one address past our routine. This address is plugged into locations 40A4H and 40A5H as our new start-of-Basic address. We then add two by incrementing register pair HL twice and plug this address into locations 40F9H and 40FAH as our new start-of-variables address.

The only thing to do now is reset the various memory size pointers to allow for the 80 bytes just added. This chore is done by the Level II ROM routine located at address 2C7AH. That routine also sends us back to the Level II ready prompt, and Merge is operational.

		00	LEADER
THESE		00	CONSISTING OF
BYTES			
DO NOT		00	255 ZEROS
LOAD		A5	SYNC BYTE
INTO		D3	THREE
MEMORY		D3	D3
		D3	HEADERS
		41	FILE NAME "A"
START OF	42E9H	F5	42F5H IS ADDRESS
BASIC		42	OF NEXT LINE
STORAGE		0A	000AH = LINE 10
AREA		00	
		B2	PRINT
		22	"
	42EFH	54	T
	42F0H	45	E
		53	S
		54	T
		22	"
		00	END OF LINE
	42F5H	02	4302H IS ADDRESS
		43	OF NEXT LINE
		14	0014H = LINE 20
		00	
		B2	PRINT
		22	"
		50	P
		52	R
		47	G
		4D	M
	42FFH	2E	.
	4300H	22	"
		00	END OF LINE
	4302H	00	0000H = DUMMY
		00	LINE NUMBER
START OF	4304H	XX	VARIABLE TYPE
VARIABLE		XX	
STORAGE		XX	

Table 1.

16526	408EH	ENTRY FOR USR
16539	409BH	PRINTER CARRIAGE POSITION
16540	409CH	DEVICE - TAPE=FFH VIDEO=00H PRINTER=01H
16544	40A0H	START OF STRING SPACE
16548	40A4H	START OF BASIC USER RAM
16550	40A6H	LINE CURSOR POSITION
16554	40AAH	SEED FOR RND AND RANDOM
16559	40AFH	NUMBER TYPE - INT=02H SINGLE=04H DOUBLE=08H
16561	40B1H	TOP OF BASIC MEMORY
16563	40B3H	STRING WORK AREA POINTER
16565	40B5H	STRING WORK AREA
16598	40D6H	MEMORY SIZE
16629	40F5H	LAST LINE NUMBER EXECUTED
16633	40F9H	START OF VARIABLES
16635	40FBH	ARRAYS POINTER
16637	40FDH	FREE SPACE
16639	40FFH	NEXT DATA BYTE TO BE READ
16722	4152H	LEVEL 2 VECTOR TABLE

Table 2.

The Merge Module

To understand how the second module works, we must examine how CSAVE formats cassette tapes and how this information is loaded back into the computer as a coherent program. Save Program Listing 4 on a cassette using the file name "A".

Table 1 demonstrates the CSAVE format. The first column contains some explanatory comments. The second column is the memory location where the byte on tape will load. The third column is a representation (in hex) of the data that is contained on the cassette tape. The last column is a brief description of what the byte is or its function.

A saved tape starts with a leader of 255 zeros, then a sync byte (A5H) synchronizes the loading of all following data. Next comes a three-byte leader of three D3H bytes, followed by a one-byte file name for the following program, in this case 41H, an ASCII "A".

All the data to this point has been CSAVE housekeeping overhead and does not load into memory. The next two bytes are the first two of the program and will be the first two bytes to be loaded into memory. These bytes point to the address of the next program line. After Basic has finished with a program line, this is how it knows where to go for the next line. The two bytes are arranged in standard Z-80 format with the least significant byte (LSB) first, followed by the most significant byte (MSB). To read the address, read the last byte first (in hex).

The next two bytes are the program line number, again arranged LSB first. The following bytes in Table 1 are the various command codes and ASCII characters making up the program.

A zero indicates the end of a program line. If you have used multiple statement program lines, a zero will not be put between the statements. Instead, a 3AH byte (ASCII colon) will be there. The zero only appears at the end of a numbered program line. The end of a program is indicated by two more zero bytes following the end of line zero. These two zeros form a dummy line number. The next sequential address starts the area used for simple variable storage.

To see how Merge works, we will first call the very useful Level II ROM routine located at address 0293H. First, it turns on the cassette motor and selects default drive number one. Then it reads the 255 zero byte leader, locates the sync byte (A5H), and turns on the two asterisks in the upper right hand corner of the screen for us. Since Merge doesn't search for a particular file name, we can skip over the next four bytes in lines 00550 to 00570.

The DJNZ (decrement-jump (if) not zero) command in line 00570 creates the assembly language equivalent of a For... Next loop, using register B as a downcounter. When counting is to be done, it is always done through the B register or for counts longer than 255, through the BC register pair. In line 00550 we set register B to four (04H). When the B register has downcounted to zero, the command falls through to the next program line.

The byte in the accumulator, (register A) is the last byte loaded and the name of the program about to be merged. We can use that byte to replace the left asterisk (Listing 1, line 00580).

Next, we need to know where in memory to load the new program. Lines 00590 to 00640 give us this address by taking the start-of-variables address, which is in register pair HL and subtracting two. Register pair HL now points to the next address past the last zero byte of the last line of the resident program (see Fig. 1). This is where the program to be merged will start. We will use this address later to adjust the next-program-line pointer. For now, we will save it in register pair DE.

Later, we are going to look for the zero byte, which represents the end of a program line. If you

PROFESSIONAL

INCOME TAX SYSTEM

FOR TRS-80* MODEL I, II or III

For four successive years!

Our system, running on the magnificent line of TRS-80 computers, has prepared thousands of lightning-fast, error-free tax returns a day in accounting offices nationwide.

Now we introduce!

SYSTEM THREE!

SYSTEM THREE is an improved, still more interactive version — Totals W-2's and computes FICA overpayment — Includes "QWIKTAX," a fast tax calculator for tax planning — Has automated Client Billing — More!

And, as always—

1. Full interactive user control, in tax-form language only, line-by-line.
2. Screen display of full 1040 and all schedules, prior to printout.
3. All formats IRS and State approved.
4. Schedule amounts automatically entered on Form 1040.
5. Your Preparer's Information automatically printed at bottom of Page 2, Form 1040.
6. Built-in Validation Check tests entire system, hardware and software.
7. Special printer adjustment routines: Margin Offset, Text Position, etc.
8. Fills in pre-printed forms (we supply) or use overlays. Your choice.
9. AUTOMATICALLY COMPUTES: Tax — Earned Income Credit — Maximum/Minimum Tax — Least Tax Method — All Percentage of Income Limitations — All Fixed Limitations.
   DOES INCOME AVERAGING IN EIGHT SECONDS!
10. Full support through the tax season, no charge.
11. Inexpensive yearly updates in accordance with tax-law changes.

SYSTEM THREE comes in four modules. You select the ones you need. There is no system like it, for any computer, and the cost is literally nothing. In one season, our users have paid, out of savings, for the software ***and* the computer.**

35-page Descriptive Manual - $7.50
(Refundable on order)

MINIMUM SYSTEM REQUIRED: Model I, 48K, 1 Disk Drive

CONTRACT SERVICES ASSOCIATES ✓10
706 SOUTH EUCLID • ANAHEIM, CA 92802 • TELEPHONE: (714) 635-4055

*TRS-80 is a trademark of Tandy Corp.

refer to Table 1, you will see that the fourth byte to come in is a zero. Avoid confusing this zero with an end-of-line zero by loading the first four bytes without a zero search in lines 00650 to 00680, using the DJNZ command again.

Lines 00910 to 00950 contain the subroutine, labeled LDBYTE, which gets one byte of data from tape by calling the Level II ROM subroutine located at address 0235H and loads that byte into the memory location pointed to by the HL register pair.

Register pair HL is incremented by one to point to the next memory location and the last byte loaded (which is in the accumulator) is compared with zero in line 00940. If the result of this comparison is true (byte = 0), the Z flag (the seventh bit of an eight-bit binary number in the F register) will be set to one. If the result is false (byte ≠ 0) the Z flag will be reset to zero. We can now return to the main program.

Line 00700 checks the status of the Z flag. If the last byte compared was not a zero (NZ), the program jumps back to the label specified by the operand of the command. Another byte is loaded and another and another, until the last byte loaded is a zero, at which point the program falls through to line 00710, which calls a Level II ROM subroutine to blink the right asterisk.

We have found a zero that means the end of a program line. If the next two bytes are also zeros, we have arrived at the end of the last line of the program. Now we must adjust the next-line-pointer in the first two bytes of the last line loaded to point to the last byte loaded, which is the first byte of the next line.

The LDI commands in lines 00770 and 00780 work the same as the LDIR command except it does not use the BC register pair as a downcounter. It transfers data once and only once. This way not only saves a byte of program memory, but demonstrates both commands as well. We are moving two bytes, the LSB and MSB of the address. Therefore, we must use LDI twice.

We have already put the address of the first byte of the last line, in register pair DE, back in lines 00620 and 00630. Register pair HL holds the address of the current line. We must store the current line address somewhere and point register HL to it in order for LDI to do its job. The problem is where to store it. Locations 40AAH to 40ACH hold the seed used in the random number generator for Basic's RND and Random statements. We can store the address here without worrying about interfering with any of Basic's normal functions.

With all the housekeeping done, the two LDIs move the LSB and MSB of the address to the first two bytes of the last line. Now POP the current line address out of the stack and back into the HL register pair. Lines 00800 and 00810 transfer it to the DE register pair for later and the adjustment is complete.

The Z flag is still set or reset from the last compare. We will use the flag now to either load the next program line or look for another zero. If the zero flag test in line 00820 is true, we load another byte and set register B to 03H. This makes the First 4 routine into a First 3 routine, otherwise our downcount would be off by one if this bit isn't a zero.

If the last byte was a zero, the flag test in line 00850 will fall through to the next program line. Increment register pair HL one last time and put that address into locations 40F9H and 40FAH as the new start-of-variables pointer.

Remember the Level II ROM routine starting as address 2C7AH that was used in the relocation module? Use it again to adjust memory size to allow for the Basic program we just merged, then we are done.

The ROM routine at 2C7AH contains a call to another ROM routine at location 01F8H that turns off the cassette motor. The reason I mention this now is that many useful Level II routines contain calls to other subroutines. When using these routines, careful consideration must be given to what effect the subroutine will have on your program or the registers you are using. If the extra calls will not affect your program or registers, use them! If the subroutine affects your registers but not your program, push the registers in jeopardy into the stack and pop them back after the subroutine. Remember for every PUSH there must be a POP.

```
10 FOR P=28000 TO 28131:READ B:POKE P,B:NEXT
20 NEW
100 DATA 237,91,164,64,237,83,140,65,33.148,109,229,237,82
110 DATA 229,193,33,217,109.237,66,34,172,109,34,177,109,34
120 DATA 185,109,34,204,109,225,1,80,0,237,176,237,83,164
130 DATA 64,19,19,237,83,249,64,195,122,44,205,147,2,6
140 DATA 4,205,53,2,16,251,50,62,60,42,249,64,43,43
150 DATA 84,93,43,6,4,205,217,109,16,251,205,217,109,32
160 DATA 251,205,44,2,205,217,109,229,34,171.64,33,171,64
170 DATA 237,160,237,160,225,84,93,32,222,205,217,109,6,3
180 DATA 32,217,35,34,249,64,193,122,44,205,53,2,35,119
190 DATA 254,0,201,0,0,0
```

Program Listing 2.

```
60000 ZZ$="PROGRAM BYTES =":ZZ=(PEEK(16634)*256+PEEK(16633))-(PEEK
(16549)*256+PEEK(16548))-103:PRINTZZ$;ZZ:LPRINTTAB(10)ZZ$;ZZ
```

Program Listing 3.

Using The Program

In response to the memory size prompt, press Enter. If you need memory for some other purpose, reserve it. If you are using Level III, load and initialize it; if not, it would be a good idea to load a line renumbering program into high memory. If you have assembled the program or punched it using T-Bug, load it using the System command. Initialize Merge by typing "/" or /28000 and pressing Enter.

If you are using the Basic program shown in Program 2, CLOAD and run it. After it has run, the New statement in line 20 will clear it out of memory. Type system and press Enter, then type /28000 and press Enter. When the Ready prompt appears, you're in business.

```
10 PRINT"TEST"
20 PRINT"PRGM."
```

Program Listing 4.

Some Precautions

CLOAD the first program, but before merging in another program, some precautions should be observed. The next program should have line numbers higher than the line numbers of the root program. Actually the higher line numbers can be loaded first and the program renumbered but if line numbers are duplicated in the two programs, line statements such as GOTO or GOSUB will not renumber properly

If the earlier line numbers are higher, always renumber or unpredictable results can occur.

Merge does not have file selection ability. You must locate your cassette at a point just before the program to be merged. Be sure the volume on your cassette is properly set, then type Merge and press Enter. Two asterisks will appear in the upper right hand corner of your CRT screen. The left asterisk will be replaced by the file name of the program that is loading. Verify that it's the proper program. If the wrong program is loading, the routine can be aborted with the reset switch. The blinking asterisk will signal that the program is loading. When loading is complete, the ready prompt will appear. If you have loaded the wrong program, simply delete the unwanted lines and start over.

All other normal commands will operate as usual. Finally, list the composite program for a final verification that you have merged all the separate programs in the right order. Renumber the new program, if necessary, and save it on tape.■

Your best Model I/III peripheral buy is a modem.

Your best Model I/III modem buy is LYNX.

Considering expanding your TRS-80 Model I or III? Then you should know that **only one kind of peripheral** can give you all these **extra capabilities** . . . in a single unit:

- **Computer-to-computer data communications, for better personal or business management.**
- **Access to free "bulletin board" resources.**
- **Access to *Source, CompuServe* and other time-sharing systems.**
- **Services like electronic mail, instant news and financial reports.**
- **An almost limitless supply of new games and other enthusiast activities.**

Of course, we're talking about a modem. A far more **sensible** first step in expanding than, say, joysticks, or a voice box, or a printer. And, therefore, your **best peripheral buy.**

A modem opens up a whole new world to you. A world of **thousands of computers** and computer people just like you, in homes and businesses around the block and across the country. A world you tap through your telephone.

But don't simply settle for any modem. For the most modem satisfaction, you need to make the **best modem buy.**

And that's **LYNX.**

Lynx for Model I/III.

The new LYNX for TRS-80 Model I and Model III. It's the latest innovation from the people who are making data communications affordable. It's the best first step you can take in expanding. It makes your microcomputer a whole new animal.

The new LYNX is the ideal modem for your Model I or III. That's because it's the **key element** of a **total direct-connect telephone communications package.**

This package includes all serial and computer bus interfaces, cables and terminal software . . . and comes to you with these remarkable features:

- **SINGLE PRICE: $299.95!**
- **Compatible with both Model I and III.**
- **No separate purchases required.**
- **One-year warranty.**
- **Auto dial/auto answer.**
- **Works with any software, including ST-80 by Micklus.**
- **Active clear, break, and "control" keys.**
- **Half or full duplex, variable word length, parity and stop bits.**
- **Dial from phone, keyboard, or memory.**
- **Works with or without Model I expansion interface.**
- **Works with or without Model III RS-232 card.**
- **Can be placed on either side of Model III.**
- **No tools needed to install.**

No other modem gives you so much in a **single package.** And when you **compare** the LYNX price with the **total cost** of bringing any other modem on line, it could **save you OVER $100!**

ESI LYNX ✓278

123 LOCUST STREET LANCASTER, PENNSYLVANIA 17602
Phone 717/291-1116

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation.
FCC Registration Number: A909KE-68171-DM-N

VISA master charge

Consult your local microcomputer retailer, or call one of these LYNX handlers today—TOLL FREE:

ADVENTURE INTERNATIONAL
Longwood, FL
800/327-7172

ADVANCED COMPUTER PRODUCTS
Irvine, CA
800/854-8241

TSE HARDSIDE
Milford, NH
800/258-1790

COMPUTER PLUS
Littleton, MA
800/343-8124

THE PROGRAM STORE
Washington, DC
800/424-2738

STEVENS RADIO SHACK
Phoenixville, PA
800/345-6279

SIMUTEK
Tucson, AZ
800/528-1149

PROGRAMS UNLIMITED
Jericho, NY
800/645-6038

✓See List of Advertisers on page 402

Maximize the potential of your TRS-80*

Encyclopedia for the TRS-80*

What's the key to getting the most from your TRS-80*? No, it isn't disk drives or printers or joysticks. It's information. Without a continual supply of information and ideas, you cannot realize the full potential of the TRS-80*.

Our response to the clamor for additional information is the *Encyclopedia for the TRS-80*,* a ten-volume reference work of programs and articles carefully selected to help you make the most of your microcomputer. You can consider the volumes of the *Encyclopedia* to be an extension of the documentation that came with your TRS-80* Each book is full of material on programming techniques, business, language, hardware, games, tutorials, education, utilities and interfacing.

Unlike conventional encyclopedias, the *Encyclopedia for the TRS-80** will never become stale or out of date. That's because the volumes of the *Encyclopedia* are being issued one-at-a-time, approximately six weeks apart. This means that each new volume will reflect the latest developments and discoveries, making this a living encyclopedia for TRS-80* users.

The first four volumes are being issued during 1981. The remaining volumes will be issued during the first half of 1982. The deluxe COLLECTOR'S EDITION has a handsome green and black hard cover with a dust jacket.

A soft cover edition is also available.

DEALERS Please request discount information and catalog when ordering. Mail Dealer orders ATT: Wayne Green Books Dealer Sales.

WAYNE GREEN BOOKS
A division of Wayne Green Inc.
Pine Street
Peterborough, NH 03458

Encyclopedia Loader

The editors at Wayne Green Books want to help you use the programs in your *Encyclopedia for the TRS-80**. So to help you maximize the use of your microcomputing time, we created **Encyclopedia Loader™.**

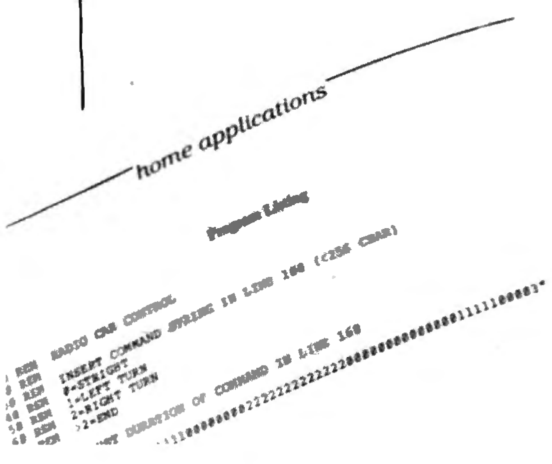
By a special arrangement with Instant Software™, Wayne Green Books can now provide you with selected programs contained in each volume of the *Encyclopedia for the TRS-80** on a special series of cassettes called **Encyclopedia Loader™**. Your encyclopedia provides the essential documentation but now you'll be able to load the programs instantly.

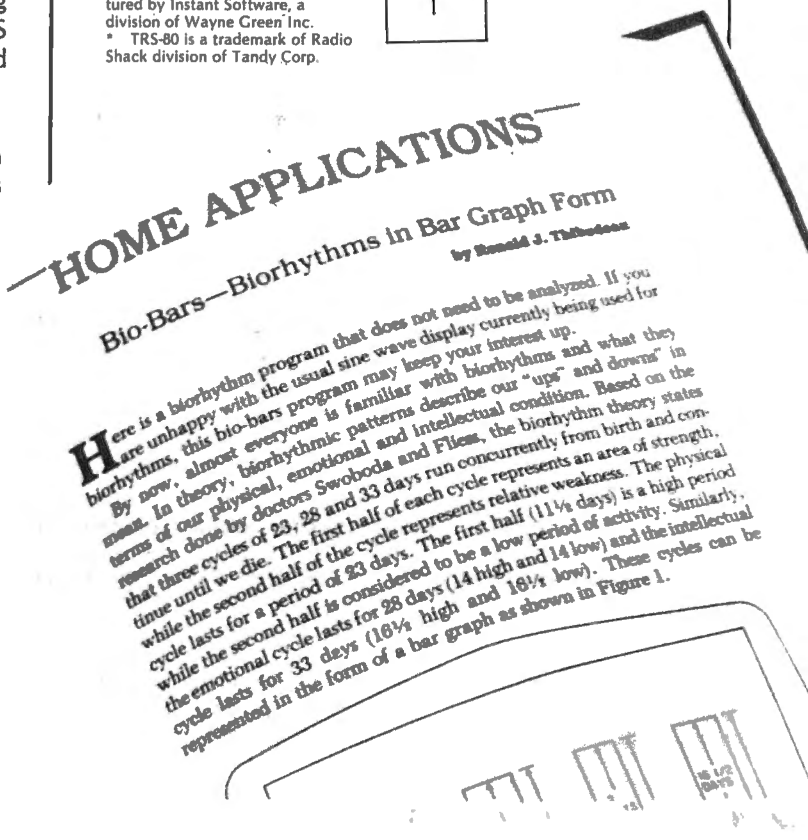
With **Encyclopedia Loader™** you'll save hours of keyboard time and eliminate the aggravating search for typos.

Save money with this introductory offer. **Encyclopedia Loader™** for Volume 1 of *Encyclopedia for the TRS-80** which will normally cost $14.95 is available **for a limited time only at the introductory price of $12.95.**

To order specify EL8001 $12.95.

The Encyclopedia for the TRS-80 is a Wayne Green publication. Encyclopedia Loader is manufactured by Instant Software, a division of Wayne Green Inc.
* TRS-80 is a trademark of Radio Shack division of Tandy Corp.





YES, I want your special offer!

____10 Volume deluxe hard cover COLLECTOR'S EDITION—a $199.50 value. EN8100 $164.00

____10 Volume soft cover edition—a $109.50 value. EN8080 $83.00

To order single hard cover volumes of the *Encyclopedia* please specify:		To order single soft cover volumes of the *Encyclopedia* please specify:		To order **Encyclopedia Loader™** cassettes please specify:	
____Volume 1	EN8101 $19.95*	____Volume 1	EN8081 $10.95*	____Volume 1	EL8001 $12.95*
____Volume 2	EN8102 $19.95*	____Volume 2	EN8082 $10.95*	____Volume 2	EL8002 $14.95*
____Volume 3	EN8103 $19.95*	____Volume 3	EN8083 $10.95*	____Volume 3	EL8003 $14.95*
____Volume 4	EN8104 $19.95*	____Volume 4	EN8084 $10.95*	____Volume 4	EL8004 $14.95*
____Volume 5	EN8105 $19.95*	____Volume 5	EN8085 $10.95*	____Volume 5	EL8005 $14.95*
____Volume 6	EN8106 $19.95*	____Volume 6	EN8086 $10.95*	____Volume 6	EL8006 $14.95*
____Volume 7	EN8107 $19.95*	____Volume 7	EN8087 $10.95*	____Volume 7	EL8007 $14.95*
____Volume 8	EN8108 $19.95*	____Volume 8	EN8088 $10.95*	____Volume 8	EL8008 $14.95*
____Volume 9	EN8109 $19.95*	____Volume 9	EN8089 $10.95*	____Volume 9	EL8009 $14.95*
____Volume 10	EN8110 $19.95*	____Volume 10	EN8090 $10.95*	____Volume 10	EL8010 $14.95*

PAYMENT ENCLOSED ______ PLEASE CHARGE TO MY ____VISA ____MASTERCARD ____AMEX

CARD # ______________________ INTERBANK # __________ EXPIRES __________

NAME ______________________________________

ADDRESS ______________________________________

CITY ______________________ STATE __________ ZIP __________

SIGNATURE ______________________________________

*These prices do not include shipping and handling. Please add $1.50 per item in the U.S. and Canada. Allow 4-6 weeks after publication date for delivery. (All volumes will be shipped UPS if the complete street address is provided; otherwise shipment will be 4th class book rate.) Foreign orders please add $10.00 for airmail per item or $5.00 for surface mail per item. Delivery outside the U.S. varies. Prices are subject to change without notice. No C.O.D.'s.

8011

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

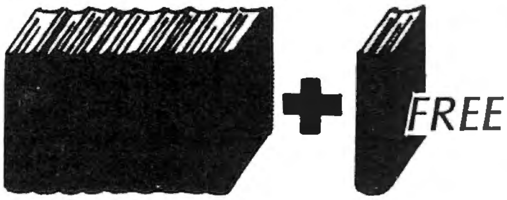
BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 80 PETERBOROUGH, NH 03458

POSTAGE WILL BE PAID BY ADDRESSEE

WAYNE GREEN BOOKS
SALES DEPARTMENT
PINE STREET
PETERBOROUGH, NH 03458

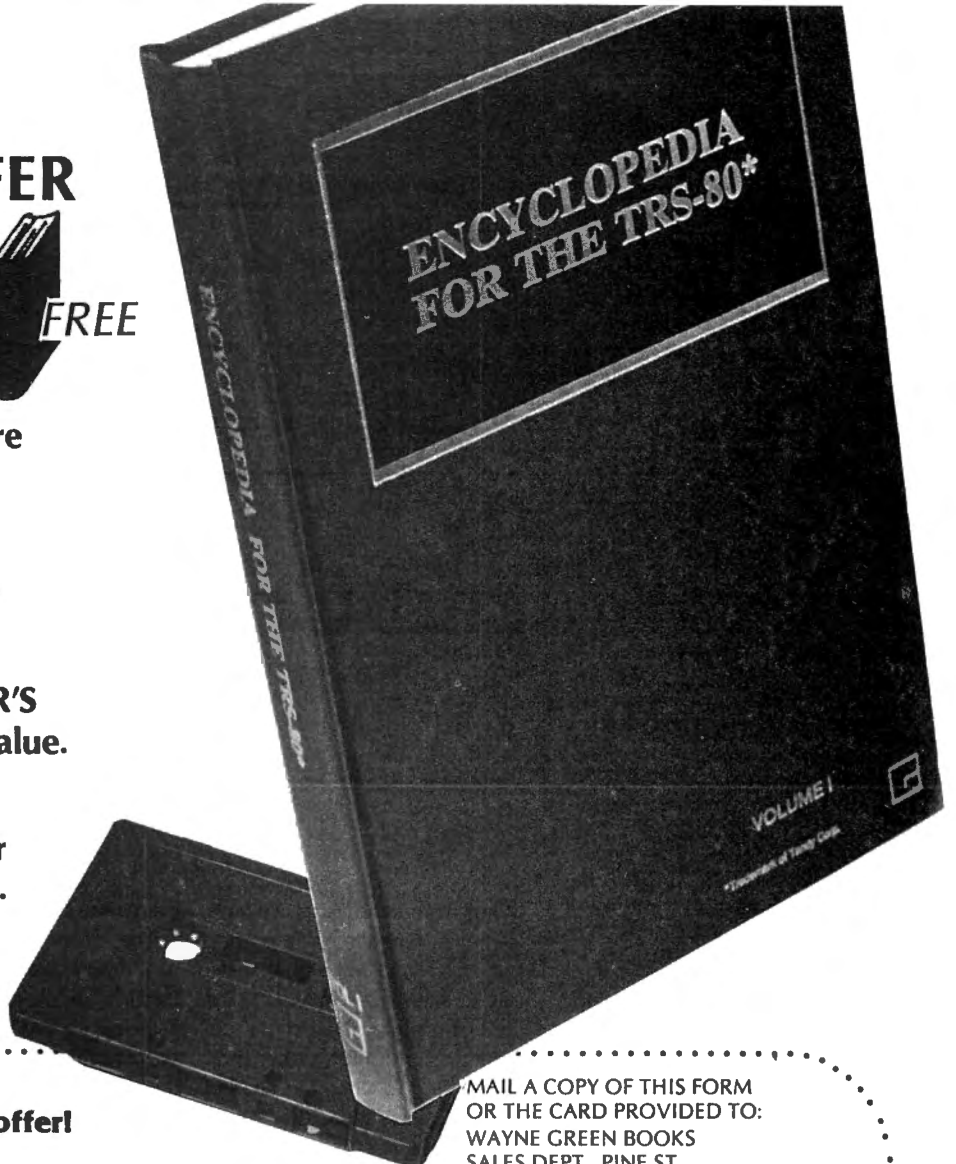
SPECIAL OFFER



If you order the entire ten volume set of the Encyclopedia for the TRS-80,* you'll receive the tenth volume FREE!

10 Volume deluxe hard cover COLLECTOR'S EDITION—a $199.50 value.
$164.00 EN8100 postpaid

10 Volume soft cover edition, a $109.50 value.
$83.00 EN8080 postpaid



YES, I want your special offer!

____10 Volume deluxe hard cover COLLECTOR'S EDITION—a $199.50 value. EN8100 $164.00

____10 Volume soft cover edition—a $109.50 value. EN8080 $83.00

MAIL A COPY OF THIS FORM OR THE CARD PROVIDED TO:
WAYNE GREEN BOOKS
SALES DEPT., PINE ST.
PETERBOROUGH, NH 03458
OR CALL **TOLL FREE**
1-800-258-5473

To order single hard cover volumes of the *Encyclopedia* please specify:		To order single soft cover volumes of the *Encyclopedia* please specify:		To order **Encyclopedia Loader™** cassettes please specify:	
____Volume 1	EN8101 $19.95*	____Volume 1	EN8081 $10.95*	____Volume 1	EL8001 $12.95*
____Volume 2	EN8102 $19.95*	____Volume 2	EN8082 $10.95*	____Volume 2	EL8002 $14.95*
____Volume 3	EN8103 $19.95*	____Volume 3	EN8083 $10.95*	____Volume 3	EL8003 $14.95*
____Volume 4	EN8104 $19.95*	____Volume 4	EN8084 $10.95*	____Volume 4	EL8004 $14.95*
____Volume 5	EN8105 $19.95*	____Volume 5	EN8085 $10.95*	____Volume 5	EL8005 $14.95*
____Volume 6	EN8106 $19.95*	____Volume 6	EN8086 $10.95*	____Volume 6	EL8006 $14.95*
____Volume 7	EN8107 $19.95*	____Volume 7	EN8087 $10.95*	____Volume 7	EL8007 $14.95*
____Volume 8	EN8108 $19.95*	____Volume 8	EN8088 $10.95*	____Volume 8	EL8008 $14.95*
____Volume 9	EN8109 $19.95*	____Volume 9	EN8089 $10.95*	____Volume 9	EL8009 $14.95*
____Volume 10	EN8110 $19.95*	____Volume 10	EN8090 $10.95*	____Volume 10	EL8010 $14.95*

PAYMENT ENCLOSED______ PLEASE CHARGE TO MY ____VISA ____MASTERCARD ____AMEX

CARD # ______________________ INTERBANK #__________ EXPIRES__________

NAME ____

ADDRESS ____

CITY ______________________ STATE__________ ZIP__________

SIGNATURE ____

*These prices do not include shipping and handling. Please add $1.50 per item in the U.S. and Canada. Allow 4-6 weeks after publication date for delivery. (All volumes will be shipped UPS if the complete street address is provided; otherwise shipment will be 4th class book rate.) Foreign orders please add $10.00 for airmail per item or $5.00 for surface mail per item. Delivery outside the U.S. varies. Prices are subject to change without notice. No C.O.D.'s.

8011

Chop and channel some machine-code subroutines for more powerful Basic.

Customized Commands

Dale W. Rupert
76 Virginia Ave.
Bridgeport, CT 06610

Every machine code utility, in a sense, customizes your computer. I have developed a method which increases the number of Basic commands and functions available to you in your Level II or Disk Basic machine. These functions are machine code subroutines. They are called by commands which are recognized but not utilized by the Level II Basic interpreter.

Some Background

A major advantage of machine language routines, aside from their speed, is their permanence. Impervious to New and CLOAD commands, which wipe out a Basic program in an instant, machine code programs remain protected beyond the boundary set up by your response to the memory size query. Only an erratic POKE or a System tape loaded into the same memory locations can alter your machine code routines.

With that important distinction understood, consider a command that would be useful in our customized computer—a Merge command. After we have loaded or typed in one program, this command will adjust some pointers. We may then CLOAD a second program which is appended to the first, rather than overwriting it. Once the second program is loaded, the Merge command must readjust a pointer before the combined programs are run.

The procedure for merging has been clarified in Roger Pape's article "Whip File Wipeouts" (*Kilobaud Microcomputing*, July 1979). My machine language version of this program is shown in Program Listing 1, lines 310–380. The comments in the listing explain the procedure.

Once this routine is stored in memory, we simply call it rather than using the cumbersome Basic sequence of PEEKing and POKEing to merge two programs.

A New Command

Before putting the Merge rou-

```
                 00100 ;*** CUSTOMIZED COMMANDS:  MERGE ROUTINE ******
                 00110 ;   DALE W. RUPERT     07/06/80
                 00120 ;
                 00130 ;THIS PROGRAM ALLOWS THE USE OF THE  MERGE  COMMAND
                 00140 ;IN LEVEL II BASIC.
                 00150 ;
                 00160 ;TO MERGE TWO BASIC PRGMS.:
                 00170 ;       1)HAVE FIRST PRGM. IN MEMORY
                 00180 ;       2) TYPE  MERGE  (ENTER)
                 00190 ;       3) CLOAD  SECOND PRGM.
                 00200 ;       4) TYPE  MERGE2  (ENTER)
                 00210 ;THE TWO PRGMS. ARE NOW ONE.
                 00220 ;
7F02             00230         ORG     7F02H           ;S.A.=32514
                 00235                          ;CHANGE LINE 240 TO RELOCATE
7F02 210B7F      00240 PATCH   LD      HL,START        ;PATCH TO START
7F05 228C41      00250         LD      (418CH),HL      ;MERGE VECTOR
7F08 C37200      00260         JP      0072H           ;TO BASIC
                 00270 ;
7F0P 7E          00280 START   LD      A,(HL)          ;GET ARGUMENT
7F0C B7          00290         OR      A               ;SET FLAGS
7F0D 200B        00300         JR      NZ,MERGE2       ;ARG. <> 0
                 00305 ;
7F0F 2AF940      00310 MERGE   LD      HL,(40F9H)      ;NEXT LINE PTR.
7F12 2B          00320         DEC     HL              ;SUBTRACT 2 ...
7F13 2B          00330         DEC     HL              ;... FROM IT
7F14 22A440      00340         LD      (40A4H),HL      ;INTO FIRST LINE PTR.
7F17 C37200      00350         JP      0072H           ;BACK TO BASIC
                 00355 ;
7F1A 21E942      00360 MERGE2  LD      HL,42E9H        ;NORMAL 1ST LINE ADDR.
7F1D 22A440      00370         LD      (40A4H),HL      ;INTO 1ST LINE PTR.
7F20 C37200      00380         JP      0072H           ;BACK TO BASIC
0000             00390         END
```

Program Listing 1.

See List of Advertisers on page 402

tine into memory, consider how we might call it. The traditional methods use the System command or the USR statement. Both are satisfactory, but a better way is to use the Merge command!

In Level II, type: MERGE (enter). You didn't get a syntax error—that means the computer recognized the word merge. The computer responded to merge with an L3 ERROR. Thus, the Merge command is built into the Basic interpreter, but it is interpreted to be a Disk Basic command, not used in Level II.

In Wes Thielke's article, "ROM Routines" (*80 Microcomputing*, February 1980), a table of Disk Basic command vectors shows that the Merge command sends the computer to address 418BH. The three bytes beginning at 418BH (C3 2D 01) cause the computer to jump to address 012DH. This jump leads right to the L3 Error dead end in Level II Basic. If we replace bytes 2DH and 01H with 0BH and 7FH, the computer jumps to location 7F0BH whenever it encounters the Merge command. Waiting at 7F0BH is our Merge routine.

```
                ;****** CUSTOMIZED COMMANDS *******
                ;      DALE W. RUPERT   07/06/80
                ;
                ;PROGRAM ALLOWS THE USE OF CMD0 THRU CMD7 IN LEVEL II BASIC
                ;TO CALL UP THE SOUNDS OF DENNIS KITSZ' BABYBEEP PRGM.
                ;IN '80 MICROCOMPUTING, 4/80, P. 68
                ;
                ;
                ;
7E30                     ORG     7E30H
                ;
7E30 21397E     PATCH    LD      HL,START          ;PATCH TO S.A.
7E33 227441              LD      (4174H),HL        ;CMD VECTOR
7E36 C37200              JP      0072H             ;TO BASIC
                ;
7E39 23         START    INC     HL                ;NEXT BASIC LOC.
7E3A E5                  PUSH    HL                ;SAVE IT
7E3B FE30                CP      30H               ;ARG. < "0" ?
7E3D FA557E              JP      M,NEXT            ;INVALID ARG.
7E40 FE38                CP      38H               ;IS ARG.>ASCII"7"?
7E42 F2557E              JP      P,NEXT            ;ARG. > 7
7E45 D630                SUB     30H               ;GET VALUE 0 - 7
7E47 21577E              LD      HL,TABLE          ;START OF TABLE
7E4A 85                  ADD     A,L               ;GET TABLE POSITION
7E4B 6F                  LD      L,A
7E4C 5E                  LD      E,(HL)            ;GET LSB
7E4D 167D                LD      D,7DH             ;BABYBEEP STARTS AT 7D00H...
                                                   ;...SO MSB IS 7DH
7E4F 21557E              LD      HL,NEXT           ;RETURN ADDRESS
7E52 E5                  PUSH    HL                ;SAVE IT
7E53 D5                  PUSH    DE                ;SET UP INDIRECT...
7E54 C9                  RET                       ;...SUBROUTINE CALL
7E55 E1         NEXT     POP     HL                ;RESTORE BASIC PTR.
7E56 C9                  RET                       ;BACK TO BASIC
                ;
7E57 00         TABLE    DEFB    00H      ;PHASER      (7D00H)
7E58 14                  DEFB    14H      ;BEEPS       (7D14H)
7E59 34                  DEFB    34H      ;DOODLES     (7D34H)
7E5A 57                  DEFB    57H      ;FANFARE     (7D57H)
7E5B B8                  DEFB    0B8H     ;SIREN       (7DB8H)
7E5C DF                  DEFB    0DFH     ;1 BLOOP     (7DDFH)
7E5D ED                  DEFB    0EDH     ;1 BLEEP     (7DEDH)
7E5E FB                  DEFB    0FBH     ;RASPBERRY   (7DFBH)
                ;
                ;
0000                     END
00000 TOTAL ERRORS
```

Program Listing 2

The Merge Routine

We must first patch our routine into the computer's sequence of steps. The Merge routine must put address 7F0BH into locations 418CH and 418DH. The location of this routine is arbitrary; I chose 7F0BH so it won't interfere with other routines in upper memory. If you have a 4K machine, replace 7FH with 4FH so your starting address is 4F0BH.

This patch causes the jump to our routine:

```
LD   HL,7F0BH    'get our starting address
LD   (418CH),HL  'store it as the Merge
                 vector
```

Since the Merge routine consists of two parts, one before the CLOAD and one after it, a single Merge command won't suffice. We must somehow indicate whether to jump to the first or the second part of the routine. We could do some POKEing before each Merge command, but that defeats the idea of customized, built-in routines.

Here is one solution. Consider the command mode statement: MERGE2. The computer first interprets the Merge command, and the two is then stored in the A register as 32, its ASCII code.

See for Yourself

To observe this process, using T-Bug:

- Put 0BH into 418CH and 7FH into 418DH (type M 418C 0B 7F); type X.
- Put a breakpoint at 7F0BH (type B 7F0B).
- Exit to Basic (type J 0072 or use the Reset button); clear.
- Type and enter: MERGE2. The computer goes to 418BH then to 7F0BH.
- The breakpoint puts you back into T-Bug. Type F to replace the breakpoint; then type R to display the registers. Notice the address in the HL register pair. (Also note the 32 in the A register.)
- Type M and the HL address to see 32H, the ASCII value for two.

Consequently, if we use Merge and Merge2, we can distinguish between our first and second entries to the routine. The following statements determine whether or not there is a non-blank character, such as 2, immediately after the word Merge:

```
LD  A,(HL)      'get chr. after Merge
OR  A           'set flags
JR  NZ,MERGE2   'if non-blank, jump to
                part two
...             'otherwise do part one
                here
```

Thus, if nothing follows the word Merge, the computer executes part one. If anything, such as two, follows the word Merge, there is a branch to part two. The complete Merge programs and patch are shown in Listing 1. You can use T-Bug to make a System tape of this program:

- Use the M command to store the program in memory (type M 7F02, then type the numbers in column two of the Listing: 21 0B 7F 22. . . etc.)
- Use the command P 7F02 7F22 7F02 MERGE to make a tape.

The starting addresses for the eight sound subroutines in Kitsz' program are shown in Listing 2. The CMD vector address is 4173H. Listing 2 first sets up the patch to our program. It then decodes the argument following the CMD, and finally calls the appropriate subroutine in the Babybeep program. The final return takes us back to the Basic program which continues from where it left off.

With this and the Babybeep programs in memory, you can generate a sound in the command mode by entering: CMD0.

Even more useful, you can use the CMD command within any of your Basic programs. This simple program demonstrates some of the possibilities:

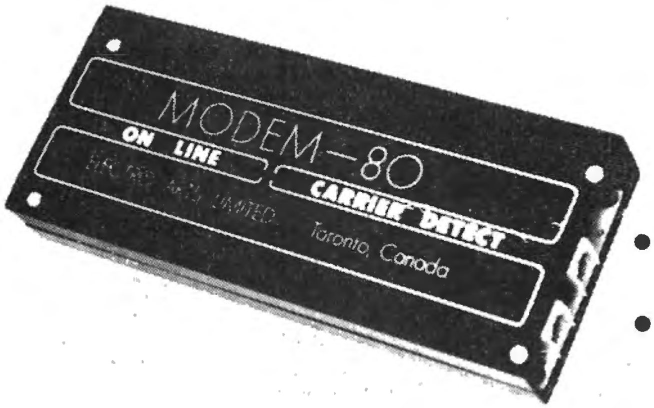
```
10 PRINT "PHASER"
20 CMD0
```

MODEM-80

TRS-80 MODEL III VERSION AVAILABLE SHORTLY

INQUIRIES WELCOME

THE TRS-80* DATA COMMUNICATIONS SYSTEM FOR THE EIGHTIES



- Direct connects to your home telephone jack
- Installs in a minute
- Program swapping mode plus terminal mode
- All software included
- No RS232C interface is required
- Lowest price ever

The MODEM-80 and your TRS-80 Model I Level 2 disk or tape system make the perfect combination to maximize your data communications capabilities.

This direct connect, auto dialing/answering modem will transmit programs and data between MODEM-80 equipped TRS-80 systems. In addition, it will transform your computer into a full-function intelligent terminal. And, no RS-232C interface is required, so tape-based systems do not need an expansion interface. The result is the lowest priced data communications system of its type available anywhere.

Program-swapping with your friends is a snap with the MODEM-80. Both machine-language and BASIC programs can be easily exchanged over the telephone line. And, accessing your favorite computer club bulletin board or any other Bell 103-type compatible time-sharing facility is no problem either.

The TRS-80 to TRS-80 software, included with the MODEM-80, extends your system's BASIC language with 8 new communications-related functions. These can be easily incorporated into your own BASIC programs allowing, for example, "computer vs. computer" competitions. BASIC programs are also provided which support disk, tape and memory transfer of programs and data, including automatic line error detection and re-transmission. This approach combines function with flexibility, reliability and ease-of-use.

When operating in the intelligent terminal mode, you have over 35 functions from which to choose. These include 8 user-definable function keys, baud rate selection from 25 to 300, forward and backward buffer scrolling, parity control, full and half-duplex operation, disk and tape file load and save, automatic dialing, parallel printer support. . .the list goes on and on.

You can install a MODEM-80 on your system in one minute. The ribbon connector plugs into the back of your keyboard or expansion interface. Power is provided by your computer's power module so no additional 110 V outlets are needed. An extension cable with standard miniature phone plugs is included for direct connection to the telephone line and there is even a spare jack to plug in a phone, if desired. Since the TRS-80 bus is extended through the MODEM-80, you can leave the unit permanently connected and still plug in other devices as you expand your system.

The MODEM-80 complete with all software and User Guides is **only** U.S. **$219.00** plus** U.S. **$5.00** for shipping, handling and insurance. We accept money orders, VISA and MasterCard. The MODEM-80 hardware has a 90 day manufacturer's warranty.

INTRODUCING TOMORROW'S PRODUCTS—TODAY!

**Subject to Canadian rate of exchange at time of purchase

*TRS-80 is a trademark of Tandy Corp.

TO ORDER CALL **416-923-7961** ANYTIME
AND ASK FOR ICROM'S ORDER DESK
OR WRITE
ICROM ENTERPRISES LTD. ✓40
1240 BAY ST.—SUITE 505
TORONTO, CANADA M5R 2A7

master charge

CHARGEX VISA

✓See List of Advertisers on page 402

Program Listing 3.

```
                00100 ;****** CUSTOMIZED COMMANDS *******
                00110 ;     DALE W. RUPERT   07/06/80
                00120 ;
                00130 ;PROGRAM ALLOWS THE USE OF CMD0 THRU CMD9 IN BASIC
                00140 ;TO CALL UP THE SOUNDS OF DENNIS KITSZ' BABYBEEP PRGM.
                00150 ;IN '80 MICROCOMPUTING, 4/80, P. 68
                00160 ;
                00170 ;SECOND PART OF PROGRAM ALLOWS THE USE OF CMDA THRU CMDZ
                00180 ;TO CALL UP SUBROUTINES WITH ADDRESSES STORED IN TAB2.
                00190 ;THE ADDRESSES ARE STORED WITH LSB FIRST AND MSB LAST.
                00200 ;     FOR EXAMPLE,  CMDA   WILL CALL THE SUBRTN.
                00210 ;     AT LOC. 7D00H  IF 00H IS STORED IN (TAB2+0)
                00220 ;                  AND  7DH IS STORED IN (TAB2+1).
                00230 ;       CMDB CALLS THE SUBRTN. WITH ADDRESS STORED
                00240 ;       IN (TAB2+2) AND (TAB2+3).
                00250 ;       .....
                00260 ;       CMDZ CALLS THE SUBRTN. WITH ADDRESS STORED
                00270 ;       IN (TAB2+50) AND (TAB2+51).
                00280 ;
                00290 ;
7E30            00300        ORG     7E30H
                00310 ;
7E30 3A7341     00320 PATCH  LD      A,(4173H)       ;CMD VECTOR
7E33 324A7E     00330        LD      (STORE),A       ;SAVE IT
7E36 2A7441     00340        LD      HL,(4174H)      ;   "
7E39 224B7E     00350        LD      (STORE+1),HL
7E3C 3EC3       00360        LD      A,0C3H          ;'JP '
7E3E 327341     00370        LD      (4173H),A
7E41 214D7E     00380        LD      HL,START
7E44 227441     00390        LD      (4174H),HL
7E47 C37200     00400        JP      0072H           ;BACK TO BASIC
                00410 ;
0003            00420 STORE  DEFS    3               ;OLD VECTOR STORAGE
                00430 ;
                00440 ;** DECODE  CMD 0  THRU  CMD 9  **
                00450 ;
7E4D E5         00460 START  PUSH    HL              ;SAVE BASIC POINTER
7E4E D5         00470        PUSH    DE
7E4F F5         00480        PUSH    AF
7E50 FE30       00490        CP      30H             ;IS ARG. < "0"?
7E52 F25B7E     00500        JP      P,OK            ;NO. CONTINUE
7E55 F1         00510 QUIT   POP     AF              ;RESTORE REG'S.
7E56 D1         00520        POP     DE
7E57 E1         00530        POP     HL              ;RESTORE HL
7E58 C34A7E     00540        JP      STORE           ;USE OLD VECTOR
7E5B FE3A       00550 OK     CP      3AH             ;IS ARG.>ASCII"9"?
7E5D F26C7E     00560        JP      P,GT9           ;ARG. > 9
7E60 D630       00570        SUB     30H             ;GET VALUE 0 - 9
7E62 218F7E     00580        LD      HL,TABLE        ;START OF TABLE
7E65 85         00590        ADD     A,L             ;GET TABLE POSITION
7E66 6F         00600        LD      L,A
7E67 5E         00610        LD      E,(HL)          ;GET LSB
                00620 ;   PUT MSB INTO D REG. IN THE NEXT LINE        ;
7E68 167D       00630        LD      D,7DH           ;(BABYBEEP STARTS AT 7D00H)
7E6A 1818       00640        JR      GO              ;SBTRN. ADDR. IN DE
                00641 ;
7E6C FE41       00642 GT9    CP      'A'             ;VALID ARG.?
7E6E FA557E     00643        JP      M,QUIT          ;QUIT IF NOT
                00644 ;
                00650 ;** DECODE CMD A  THRU  CMD Z **
                00651 ;
                00659 ;***CHANGE THE COMPARISON BYTE IN THE NEXT LINE TO EQUAL

2 ***
7E71 FE42       00660        CP      'B'             ;VALID ARG.?
                00661 ;
7E73 CA797E     00662        JP      Z,OK1           ;ARG. IS VALID
7E76 F2557E     00670        JP      P,QUIT          ;INVALID ARG.
7E79 D641       00680 OK1    SUB     41H             ;GET 0-25 FROM A-Z
7E7B 07         00690        RLCA                    ;DOUBLE IT
7E7C 21997E     00700        LD      HL,TAB2         ;START OF TABLE 2
7E7F 85         00710        ADD     A,L             ;TABLE POSITION
7E80 6F         00720        LD      L,A             ;HL HAS TABLE LOC.
7E81 5E         00730        LD      E,(HL)          ;GET LSB
7E82 23         00740        INC     HL
7E83 56         00750        LD      D,(HL)          ;GET MSB
                00760 ;
7E84 218A7E     00770 GO     LD      HL,NEXT         ;RETURN ADDR.
7E87 E5         00780        PUSH    HL              ;SAVE IT
7E88 D5         00790        PUSH    DE              ;SET UP INDIRECT...
7E89 C9         00800        RET                     ;...SUBROUTINE CALL
7E8A F1         00810 NEXT   POP     AF              ;RESTORE REGS.
7E8B D1         00820        POP     DE
7E8C E1         00830        POP     HL              ;RESTORE BASIC POINTER
7E8D 23         00840        INC     HL              ;NEXT LOCATION
7E8E C9         00850        RET                     ;BACK TO BASIC
                00860 ;
                00870 ;***  USE THIS TABLE FOR SUBRTNS. IN SAME PAGE OF MEMORY ***
                00880 ;       TABLE STORES LSB : D REG. CONTAINS MSB (7DH)
                00890 ;
7E8F 00         00900 TABLE  DEFB    00H     ;PHASER      CMD0
7E90 14         00910        DEFB    14H     ;BEEPS       CMD1
```

Program continues

```
30 PRINT "SOME OTHER SOUNDS":
   CMD1: CMD2 : CMD3
40 CMD7 : PRINT "HOW'S THAT!!!"
```

Total Flexibility

Two shortcomings of the previous program are: the eight decoded arguments are already used, and all the addresses must be in the same page of memory, that is, the most significant byte of each address is the same (7DH).

One final modification of the argument-decoding and table-lookup scheme will make this program even more useful. If all of our machine language subroutines are not in the same page of memory, we need a two-byte address table to locate them. Listing 3 allows us to create such a table. Furthermore, this program may be used with Disk Basic.

This program decodes arguments 0–9 (used with the Babybeep subroutine) and arguments A–Z. For arguments A–Z, the program calls the subroutine identified by the two-byte address in the second table (TAB2). For the example shown in Listing 3, CMDA calls the subroutine at 7D00H, and CMDB calls the subroutine at 7D14H. After executing the subroutine, the computer jumps back to Basic for the rest of the program.

The Inner Workings

Let's take a brief look at Listing 3. First read the comments and get an overall view of it.

Decoding arguments A–Z begins at line 650. A is stored as an ASCII 41, B is 42, . . . , and Z is 5A with all numbers in hexadecimal. Subtracting 41H from each value gives a range from 0–19H, or 0–25 in decimal. Doubling this result (line 690) allows two places in the table for each argument's address. Thus, positions zero and one correspond to A, two and three are for B, . . . , and 50 and 51 belong to argument Z.

After adding the table's starting address to the argument's value (line 710), the program puts the least significant byte (LSB), then the most significant byte (MSB), into the DE register pair.

ELECTRIC SPREADSHEET™

NEW **Considering VisiCalc™?**

ELECTRIC SPREADSHEET does everything you can do with a pencil, paper and calculator. Start with a blank screen. Move the fast cursor to desired location. Enter line or column labels or data. Select preprogrammed operators for line, column, or cell calculations. Run. Results appear on the screen, formatted for your printer. Revise. Run again.

PREPROGRAMMED OPERATORS for mathematics, finance, and statistics. Save data on tape or disk. The screen is your window to a larger spreadsheet. Output features scrolling and split-screen. Variable formats for labels and data.

UNLIMITED APPLICATIONS. P/L forecast. Personal budget. Real estate investment. Net worth forecast. Invoice. Cash flow estimate. Sales analysis. Check record. Business forms.

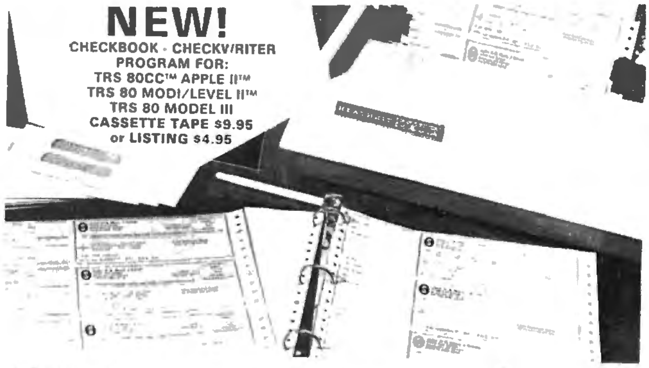
16K version has 50 operators. 32/48K version has 70 operators plus histogram plot, remote storage of data, global revise spreadsheet layout, alpha entries, and more.

16K Model I or III Tape $34.95
32/48K Model I or III Tape $64.95
48K Model III Disk $67.95

YOU CAN'T LOSE! Calif. orders add 6½% tax.

VisiCalc is a trade mark of Personal Software, Inc.

Dan G. Haney & Associates, Inc. ✓464
P.O. Box 687, San Mateo, CA 94401
(415) 493-4094



NOW. . .Continuous Checks

That Can be Used With or Without Your Computer!!
The Best in A Home Checking System

That's right. Continuous Checks fan-folded in a 3-to-a-page desk set design. And they can be computer printed, handwritten or typewritten — whichever suits the quantity or situation.

SPECIAL DESIGN

Our checks are not a high-volume business form adaptation. They're specially designed **Computer/Manual Checks** for the home user. And they're **easy to use**, too. No need to change printer form width when loading. Our checks are the same 9½" **width** as standard tractor feed printer paper. Check tear down size is the same as the classic **personal-sized check** issued by all banks.

COMPLETE HOME SYSTEM

With this system you can print the bulk of your monthly checks on your computer using our **program**. Your **checks** and stubs can then be stored in our attractive Data Ring Binder **Checkbook**. Later, if you have a few checks to write, there's no need to load them into a printer - just write a check at your desk as shown above. And you can mail your checks in our **dual windowed envelopes** to eliminate addressing chores.

UNIQUE

You won't find **continuous** checks like these anywhere. And, our special small quantity printing process will give you **excellent quality** and appearance. Standard **color-coordinated** imprinting and encoding is as shown above (logo can be omitted) on **blue, grey, green**, or buff checks.

PRICED RIGHT

Two Hundred checks are just **$29.95** (envelopes **$14.95**). Five Hundred checks are **$49.95** (envelopes **$27.95**). Data Ring Checkbooks are only **$4.95**.

Special "ORDER NOW" Offer

So that you can start using your computer immediately to pay your monthly bills, we'll make you a special package offer. If you order directly from this ad, we can send you:

200 Checks, 100 Envelopes, A Binder, and Program for $49.95
— OR —
500 Checks, 300 Envelopes, A Binder, and Program for $74.95

You'll save as much as **$9.85** over the separate purchase price. And with the special package you can begin your monthly checkbook balancing and bill paying as soon as you receive your checks. Just enclose a voided check (for encoding information) with your payment (VISA - MasterCard orders must show signature, expiration date, and account number). Or, send today for samples.

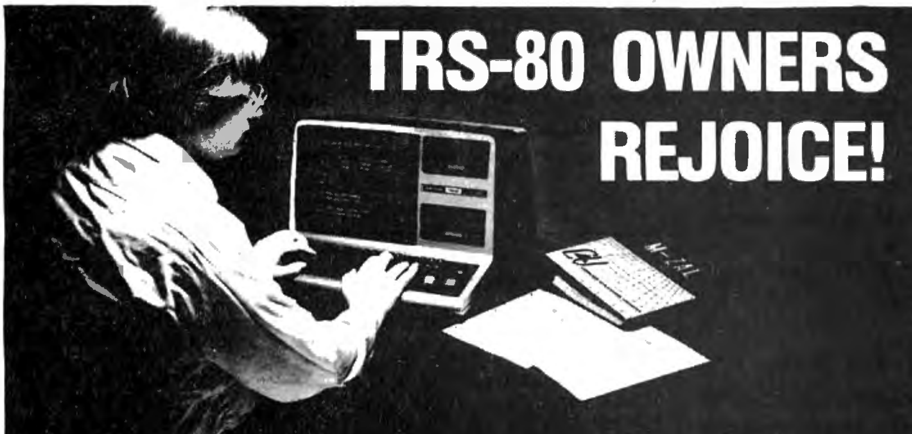
E-MAIL ADDRESS. SOURCE TCH-411 microNET CIS 70270,236

SYNERGETIC SOLUTIONS
PHONE: (813) 646-6557
4715 SHEPHERD RD.
DEPARTMENT M
MULBERRY, FL 33860

✓321

TRS 80™ Tandy, Inc Apple II™ Apple Computer, Inc

TRS-80 OWNERS REJOICE!



Our professional quality software development tools let you program your **TRS-80** with more ease and power than you ever imagined possible. Programming will become a pleasure when you use our full screen editors. These editors are not just patched up word processors. They have been specially designed to enhance the BASIC and ASSEMBLER programming languages. Our assemblers provide full screen option menus to relieve you of the burden of remembering command names. Combine user oriented features like this with the finest and most extensive documentation in the industry and you will see why programming a TRS-80* is now better than ever.

T-ZAL: TAPE BASED ASSEMBLER: Assemble to memory or tape. Create relocatable SYSTEM tapes! Includes relocating LOADER pgm. 8 character symbols with XREF. Upgrades to disk when you do!
Mod 3 Cat # 1250-20 $ 49.95

FULL SCREEN PROGRAM TEXT EDITORS:
All CAU full screen editors support four way cursor motion and scrolling with user selectable auto repeat and scroll speeds. Totally visual character insert, delete, and change. Block move, copy, delete. Global find and change. Editor for BASIC also provides automatic line renumbering and 26 user-definable macro keys.

For Tape and Disk BASIC:
Mod 1 Cat # 1010-20 $24.95
Mod 3 Cat # 1210-20 $29.95

For Mod 1 EDTASM source files:
Tape Cat # 1010-30 $24.95
Disk Cat # 1010-31 $34.95
(Disk version shipped on tape.)

M-ZAL:™ DISK EDITOR/ASSEMBLER:
Modular Assembler Language development system includes full screen text editor, multi-pass assembler, and object module linker. Allows 8 character symbols and source file chaining/nesting with *INCLUDE cmd. Full support for relocation and global symbols. Symbol table with cross reference. Req. 32K Disk system.
Mod 1 Cat # 1050-10 $149.00
Mod 3 Cat # 1250-10 $149.00

XBUG: DEBUGGING TOOL & MONITOR:
Display/Modify memory and regs. Set up to 10 invisible and transparent breakpoints. Multi-speed single step. Load/Copy/Create SYSTEM tapes. Uses only 2.5K and can be dynamically relocated to any address.
Mod 1 Cat # 1020-10 $19.95
Mod 3 Cat # 1220-10 $19.95

Note: these products are not available for level 1 machines.

When ordering specify Model, RAM size, # of disks.

Send check, money order, or MC/VISA numbers and expiration date to:
Computer Applications Unlimited
P.O. Box 214, Dept. EEW
Rye, New York 10580

N.Y. State residents add applicable sales tax.
Dealer Inquiries Invited.

COMPUTER APPLICATIONS UNLIMITED™ ✓107
A Div. of CAU, Inc.
Quality•Reliability•Service

*TRS-80 is a TM of Tandy Corp.

✓See List of Advertisers on page 402

Program continued

```
7E91 34      00920         DEFB     34H      ;DOODLES      CMD2
7E92 57      00930         DEFB     57H      ;FANFARE      CMD3
7E93 B8      00940         DEFB     0B8H     ;SIREN        CMD4
7E94 DF      00950         DEFB     0DFH     ;1 BLOOP      CMD5
7E95 ED      00960         DEFB     0EDH     ;1 BLEEP      CMD6
7E96 FB      00970         DEFB     0FBH     ;RASPBERRY    CMD7
7E97 00      00980         DEFB     00H      ;PHASER       CMD8
7E98 00      00990         DEFB     00H      ;PHASER       CMD9
             01000 ;
             01010 ;TAB2 STORES LSB THEN MSB  OF ADDRESS
             01020 ;
7E99 00      01030 TAB2    DEFB     00H      ;LSB    CMD A
7E9A 7D      01040         DEFB     7DH      ;MSB
7E9B 14      01050         DEFB     14H      ;LSB    CMD B
7E9C 7D      01060         DEFB     7DH      ;MSB
             01070 ;       .....
             01080 ; EXTEND THIS TABLE AS NEEDED FOR MORE SUBRTN. ADDRESSES
             01090 ;
7E30         01100         END      PATCH    ;STARTING ADDRESS IS 'PATCH'
```

An Underhanded Call

The section labeled Go is the least straightforward part of the program. We have the address of the subroutine to be called in DE. Unfortunately, there is no CALL (DE) command for the Z-80. But the solution is not difficult: Put the DE address onto the stack (PUSH DE), then execute a return statement.

Consider what the return statement does. It POPs the two-byte address from the top of the stack into the program counter, and the computer jumps to that address. But once we've gone to the subroutine, how do we get back?

A normal Call would PUSH the return address onto the stack before jumping to the subroutine. Now we must do that.

In line 780, we PUSHed the address labeled Next onto the stack. In the meantime, the DE address has been PUSHed and POPped, so the Next address is now on top of the stack. The return statement in the called subroutine POPs the Next address and jumps back to the calling program.

Finally, the Basic pointer, which we PUSHed in line 460, is restored to the HL registers and the other registers are restored. The final return statement jumps back to Basic.

I normally load this program when I first start up the computer. Answer the memory size question with 32513 (7F01H) or less. Type and enter: SYSTEM and MERGE. Once the program is loaded, type and enter "/". The first part of the program immediately loads the patch and returns to Basic.

Now whenever you wish to merge two Basic programs follow this simple procedure:

- Have the first program in memory.
- Type and enter: MERGE. The ready signal quickly appears.
- Type and enter: CLOAD to load the second program.
- Type and enter: MERGE2 when it is done loading.

Now your two programs are one. Note that the line numbers of the second program must be higher than those of the first program.

We have created a new command for our computer. Once its code is in memory, you may use it as any other command. This routine is so simple and so useful, you will wonder why it was not part of the original commands.

Adding Versatility

The simple decoding scheme in the Merge routine allowed only two subprograms to be distinguished. A more selective method for decoding the argument would greatly enhance our use of the built-in Disk Basic commands. For example, we could use the CMD command with a one-character argument (CMD3, CMDF, and so forth) to call 36 different subroutines (0–9 and A–Z).

As an example, we will use the CMD command to call up various sound-generating subroutines. Dennis Kitsz' article, "Babybeep" (*80 Microcomputing*, April 1980) tells how the TRS-80 can produce a variety of sounds. By using Kitsz' sound subroutines along with the customizing methods in this article, we will be able to produce a phaser sound as easily as typing CMD0.

Make It Work

I suggest you put Babybeep into memory from 7D00H to 7E2FH. Then enter Listing 3 into locations 7E30H to 7E9CH. To make a System tape with T-Bug, type: P 7D00 7E9C 7E30 CUSTOM. Use memory size 31999.

You later change the values in TAB2 to call your own custom commands. As written, CMDA and CMDB generate the phaser (7D00H) and beeps (7D14H) respectively.

For readers who are not assembly language enthusiasts, I have included Basic programs in Listing 4. These may be used in Level II only, not Disk Basic. They POKE the Merge routine (Listing 4a) and the CMD 0–7 routine (Listing 4b) into memory. Once the Basic programs are run, they delete themselves. The routines remain in protected memory, however, until the power is shut off.

A Challenge

This article has developed a procedure that creates additional Basic commands and functions. The format of Listing 3 may be used in either Level II or Disk Basic. There are also many other Disk Basic commands available.

You could further customize your TRS-80 with a screen-fill command, a decimal/hex converter, a pause command, a register display command, a search function.

You could also write routines which manipulate, rather than merely decode, the arguments. How about a memory dump called by CMD D 4200,4500, for example? Or CMD V 2,8 which clears lines 2–8 of the screen?

You are no longer limited to the Level II Basic commands! ■

```
1 '*** CUSTOMIZED COMMANDS ***
4 '
5 ' MERGE ROUTINE: MEM. SIZE? 32522 (OR LESS)
6 '
10 POKE 16780,11: POKE 16781,127   'PATCH
15 FOR M=32523 TO 32546: READ B: POKE M,B: NEXT
20 DATA 126,183,32,11,42,249,64,43,43,34,164,64
25 DATA 195,114,0,33,233,66,34,164,64,195,114,0
30 DELETE 1-30
```

Program Listing 4.

```
1 '*** CUSTOMIZED COMMANDS ***
4 '
5 'LEVEL II BASIC VERSION OF LISTING 2: CMD0-CMD7
6 '      YOU MUST HAVE BABYBEEP IN MEMORY
         BEFORE USING CMD0 THRU CMD7
7 '
10 POKE 16756,57: POKE 16757,126         'PATCH
15 FOR M=32304 TO 32350: READ B: POKE M,B: NEXT
20 DATA 33,57,126,34,116,65,195,114,0,35,229,254
25 DATA 48,250,85,126,254,56,242,85,126,214,48
30 DATA 33,87,126,133,111,94,22,125,33,85,126,229
35 DATA 213,201,225,201,0,20,52,87,184,223,237,251
40 DELETE 1-40
```

Program Listing 4a.

Some tips on remaining so when in the mode.

In Command

Jerry Rutledge
Box 123
Waseca, MN 56093

How powerful a tool can the command mode be? Recently I was writing a program that would be used only one time. It involved two different lists of 275 insurance agents (by four digit agent code numbers, like 1473). Some agents appeared on both lists and on one of the lists, some agents appeared as many as three times.

The idea was to sort through the two lists and delete all of the numbers that appeared more than once and find out who on the second list was not on the first list. Since the lists were not in numerical order, I figured it would be quicker to write a short program and input all 550 numbers into data statements than attempt to go through the lists manually.

It would have been, too, except that my novice programming ability got in the way. When you're writing your own sorting or comparing program, the TRS-80 is not the fastest computer in the world. Consider that this program was comparing one array of 275 elements with another of equal size. That means between 50,000 and 75,000 comparisons—a process that took about an hour.

The sort went without a hitch, and as I sat waiting for the printer to begin the final printout, there suddenly appeared on the screen one of those ghastly, dreaded error messages:

```
SUBSCRIPT OUT OF RANGE IN 2640
```

I made three attempts to detect where I had gone wrong in dimensioning those arrays. Each re-run meant a one hour wait for that dismal sort, and each time the same error message happened. When it appeared the third time, I did not immediately return to the edit mode to look at line 2640 again. Had the computer done the sort properly? Were the new arrays loaded properly after all of those doubled up numbers had been deleted? Why not ask and see? So I entered: PRINT A(275).5260. Voila! There it was just waiting to be printed out. If the computer will print on the screen in command mode, why won't it LPRINT the same way? Well, it will: LPRINT A(275). And sure enough, the printer clacked out 5260.

Thoroughly frustrated at this point, I figured nothing could be worse than sitting through another sort, so I began LPRINTA(1), then LPRINTA(2), etc. After typing that command about five times another thought struck me: Would the command mode respond to a For...Next statement and multiple commands? Only one way to find out: FOR X=1 TO 10:LPRINTA(X): NEXT X. Sure enough, out came the first 10 numbers. The rest was easy.

Since some of the array elements were equal to zero (their numbers having been deleted in the sorting process) I did not want to print those. I also wanted the list numbered so I didn't have to count the printed lists. Could all that be done in one set of command mode statements? You bet... here's how it looked:

```
C=1: FORX=1TO275: IFA(X)=OTHEN-
NEXT X ELSELPRINTC,".";A(X):C=C+1:
NEXTX
```

In less than two minutes I had my listing that I had been sweating for over four hours. It took about 30 seconds more to rewrite the command to print out array B, and another two minutes to print it.

I quickly typed New to wipe out any lingering memory of the experience. Later, mellowed by a couple of cans of beer, it occurred to me that it had been a valuable learning experience. Almost any string of commands can be used in the command mode provided they don't exceed the 256 byte limit.

Beware of the Edit Mode

Like all good things, there is a caveat and a limit to using a trick like this. There are at least two Level II situations where your '80 will set all of your stored variables back to zero. One is when you type Run and begin a program, and the other is when you execute into the edit mode.

Thus, if you are running a program which crashes on an error where your computer automatically puts you in the edit mode (such as a syntax error), your variables are automatically set back to zero and this trick will not work.

A Fix

If you like to hedge, there is a simple way to avoid the automatic execution into the edit mode when errors occur. For example, suppose you have written this simplistic program:

```
10 INPUT A,B,C
20 D=A*B*C
30 E=D/C
40 PRINTE
```

Take the program listed above and change line 40 so there will be a syntax error: 40 PLINTE. Now run the program. There's the syntax error, big as life, and you're left in the edit mode on line 40. Hit Enter so that you are back in the command mode and see what happens when you ask: PRINT E. 1. See, you've lost everything, right? Now add the following two lines to your program:

```
5 ON ERROR GOTO 1000
1000 PRINT"ERROR"
(Enter and Run)
RUN
? 2,3,4
ERROR
NO RESUME IN 1000
```

ATTENTION COMPARISON SHOPPERS

HOW DOES A $299 BYTEWRITER-1 STACK UP AGAINST A $650 EPSON MX-80? YOU DECIDE!

The Only 80 Column Dot Matrix Printer Under $300.

Why do we dare to compare the Bytewriter-1 to the Epson MX-80, the industry leader? Because we feel strongly that dollar for dollar, the Bytewriter-1 is tough to beat for performance and quality.

Our extensive testing has proved that the Bytewriter-1 interfaces problem-free to the TRS-80, the Apple II and the Atari 400 and 800.

We are not going to tell you that the Bytewriter-1 is better than the MX-80, but by comparison, and for half the cost, you get more than a reliable printer — you get a great value.

Call or write for more information today.

Comparable features. Uncomparable price.

MICROTEK inc. ✓ 362

9514 Chesapeake Drive
San Diego, CA 92123
(714) 278-0633
Outside CA call
TOLL FREE (800) 854-1081
TWX. 910-335-1269

FEATURES	BYTEWRITER-1	EPSON MX-80*
Print speed	60 lines per minute	46 lines per minute
Paper feed	Friction feed original plus 3 copies	Pin feed original plus 2 copies
Ribbon	Black, cartridge $9.95	Black, cartridge $14.00
Life expectancy	Printhead — 100 million char. Drive Mech. — 10 million char. Ribbon — 5 million char.	50 — 100 million char. 5 million char. 3 million char.
Dimensions	3.8″ × 15″ × 9″	5.2″ × 14.7″ × 12″
Character set	96 ASCII	96 ASCII
Interface	Parallel	Parallel
Warranty	90 days	90 days
Printhead replacement	$29.95	$30
Cost	$299	$650

*Data source: Epson MX-80 Operation Manual

TRS-80 is a trademark of Radio Shack, Div. of Tandy Corp.
Apple II is a trademark of Apple Computer, Inc.
Atari 400 & 800 are trademarks of Atari, Inc.

Now you know that there was an error in your program and you have also created a No Resume error in line 1000. But note that this error has not automatically put you back in the edit mode.

Now try the command, Print E (or D, A or any of the other variables), and there they are. Note that the Print"Error" line must be the last line of your program. Even adding the word End to line 1000 will put you back in the edit mode!

You're probably saying to yourself at this point, "That's fine for a small program like this one, but if my program has 250 lines in it and there is an error, how would I know if enough of the program has executed that I could retrieve all the variables in the command mode?" Good question. And there is an answer.

In your *Level II Reference Manual* there are two error-routine functions, ERR and ERL, and up to now they looked to me like interesting but relatively useless functions. ERL returns the line number in which the error occurred and ERR returns the code number of the error. So, let's change line 1000 to read:

```
1000 PRINT"ERROR LINE";ERL;" ER-
ROR CODE";ERR/2+1
(Run)
? 2,3,4
ERROR LINE 40 ERROR CODE 2
NO RESUME IN 1000
```

And there you have it, a syntax error in line 40 without being dropped into the edit mode (look up the error code number in the Appendix). In Disk Basic there may be a way of printing out the actual error message using the Error function described on page 7-6 of the *Disk Manual*, but so far it has eluded me.

At any rate, perhaps you have discovered some ideas for different ways to use the command mode. If you're lucky, you may never need to use them, but there is a lot of power there at your fingertips. ■

SOFTWARE FOR TRS-80

UNITERM
'The Universal Terminal Program'

The FIRST and ONLY terminal program for both the Model I and Model III TRS-80 computers. Written by Pete Roberts, this program will soon become the standard of terminal software. Unique features includes a Handsome binder with over 75 pages of instructions and examples. Free upgrade policy using local Bulletin Boards and MicroNET. Expanded functions such as 'Type to buffer', 'Display Buffer', Define Auto logon, polling, signon, and connect messages! Requires 32K disk system. Only **$79.95**

ST80III, OMNITERM, SMART80, SMARTIII owners!!! Upgrade to 'UNITERM'! Send your ORIGINAL program disk and receive 'UNITERM' for only $29.95!!

CONNECTION-80 BBS

Run your own Computer Bulletin Board. This software package when combined with your Model I or Model III TRS-80 and an Auto Answer Modem will convert your computer into a mini-times share system. Functions include message leaving, both public and private, message retrieving, Bulletins, Downloading, merchandise, product ordering, chat, etc. This is a full feature system, and well known for excellence among modem users. Only **$199.95**

UTILITY PACK

This package includes three of the most useful programs available for the serious TRS-80 user, by Mike Friedman.

1) **Spooler:** Very fast, allows page parameters and perferation skip over automatically. you can even switch DOS's without interupting printing!

2) **Erase:** Super Purge utility with multiple modes of operation. Allows you to display DIR and position cursor over file to be killed, and then a key stroke and it's gone! You will never use 'KILL' again!

3) **Map:** Places a sector map (LUMP map for NEWDOS/80) of your disk on the screen. Shows free, used and locked out grans.

ENTIRE PACKAGE ONLY. (Specify Mod I or III), **$49.95**

COPY III

This Model III Utility, written by Dick Balcom, allows you to load system tapes into your computer at either 500 or 1500 Baud and then copy them onto a new tape at either 500 or 1500 Baud. Includes 10 page instruction manual, **$9.95**

THE COPYRIGHT KIT

A self-instruction booklet on copyrighting the computer software you write. Includes step by step instructions, sample forms, as well as discussions of copyrights, patents and trade secrets, your rights secured by copyrights, legal remedies upon infringements, material not copyrightable and MUCH MORE! Written by Attorneys. Published by National Attorneys Publications and distributed exclusively through B.T. Enterprises. IF YOU EVER WROTE A COMPUTER PROGRAM, YOU NEED THIS BOOK! **$11.95**

DOS PLUS OPERATING SYSTEM

A fast, easy to use, and bug-free operating system for the TRS-80 Model I and III. This system is a must for business and hobby users. Easy to use, supports Single and Double density in the Mod III, (Mod I with Doubler). Includes a $100 reward for an error if you can locate one. (See Micro Systems Software ad for details.)

DOSPLUS 3.3S (Mod I Single Density), **$99.95**
DOSPLUS 3.3D (Mod I Double Density, Doubler required), **$99.95**
DOS PLUS 3.3III (Mod III Sing/Doub Density), **$99.95**

NEWDOS/80 OPERATING SYSTEM

The Hottest Disk Operating System for the TRS-80 Model I and III. Version 2.0 with full single and double density support, allows the use of and combination of disk drives types and densities. Full Double density support when used with a Doubler. On the Model I, you can read and write Model III compatable data disks. On the Model III, you can write Model I single or double density data disks for use on the Model I. Includes: Expanded directories, dynamic basic merge and delete, selective variable clearing, enhanced basic editing, Enhanced chaining functions, Superzap utility, Disk enhanced editor assembler and disassembler, and much more. Special Price, **$139.95**

M-ZAL

This is the most powerful Editor Assembler for the TRS-80 ever written. Features a full screen editor, a menu driven asembler, and an interactive linker which allows the linking of /CMD and /RLD files. Files can be loaded to Disk or Tape! Assembly Language programers like Dick Balcom, and Pete Robert claim that this is the best Editor Assembler on the market! Special Price, **$129.95**

OTHER MYSTERIES

We carry the full line of IJG 'Other Mysteries' books at discount prices. These books are a must for computer users.

TRS—80 Disk and Other Mysteries	~~$22.50~~	**$19.95**
Basic Decoded and Other Mysteries	~~$29.95~~	**$26.96**
Custom TRS-80 and Other Mysteries	~~$29.95~~	**$26.95**
Basic Faster & Better	~~$29.95~~	**$26.95**

MX-80 RIBBON RELOADS

Don't throw away your worn MX-80 ribbon cassettes. We carry endless loop ribbons to replace the worn ribbon in your MX-80 ribbon cassette and save money. Installation takes about 3 minutes each. Special offer **$3.50/ea., $35.00/doz.**

CASOP

The CASOP utility by Paul Spoltore allows you to copy system tapes, find load locations, check for hidden messages, modify program blocks, relocate program blocks, and merge two tape programs together. A sort of 'SUPERZAP' for tape users, **$24.95.**

AVAILABLE AT

Breeze Computing
P.O. Box 1013
Berkley, MI 48072
(313) 288 9422

Micro Images
146-03 25th Road
Flushing, NY 11354

All Systems Go
105 W. Plant Street
Winter Garden FL 32781
(305) 877-2830

CPU Computer Corp.
175 Main Street
Charleston, MA 02129

Programs Unlimited
20 Jericho Tpke.
Jericho, NY 11753
(800) 645-6038

Son John Enterprises
P.O. Box 1671
Freemont, CA 94538
(415) 651-4147

Contemporary Comm Corp.
5582 Coral Way
Haslett, MI 48840
(517) 339 1028

Stevens Radio Shack
562 Nutt Road
Phoenixville, PA 19460
(800) 345-6279

Soft Sector Marketing
6250 Middle Belt Road
Garden City, MI 48135
(313) 425-4020

Omni Systems Associates
P.O. Box 632
W. Caldwell, NJ 07006
(201) 226-9185

Bob's T.V.
93-15 86th Drive
Woodhaven, NY 11421
(212) 441-2807

Computer Serv. of Danbury
P.O. Box 993
Danbury, CT 06810
(203) 744-5516

BT ENTERPRISES

171 Hawkins Road
Centereach, New York 11720
✓ 122
(516) 981-8568 (Voice)
(516) 588-5836 (Data)
MNET-70331, 105

VISA MasterCard

Dealer Inquiries Welcome
Add $2.00 S & H
NYS res. add appr. tax

'THE SYSTEM'

New TRS-80 Model III

Model III, 48K 2 Siemans Disk Drive system, **$1799.00**, (plus shipping and handling) includes Dos Plus III operating system.

ADD DISK DRIVES

Internal Drive #1—Includes Micro Mainframes Controller board, Dual Drive Power Supply (for 2 internal disk drives), Mounting Hardware (for 2 internal disk drives), 40 Track Siemans Disk Drive, **$629.95**

Internal Drive #2—Includes 40 Track Siemans Disk Drive 12 ms Disk Drive, **$299.95**

Memory Upgrade, 150ns, Prime Chips, Add 16K, **$44.95**, Add 32K, **$79.95**

SPECIAL!!!

Internal Drive #1	$629.95
Internal Drive #2	$299.95
32K RAM Memory Chips	$79.00
All Together	~~$1009.85~~
Includes Complete Installation Instructions!	$849.95

External Drive #3—Includes case (as shown above, dual power supply, cable, mounting hardware for 2 external drives, **$429.95**
External Drive #4—(same as Internal Drive #2) 40 track disk drive, **$299.95**

External case (no drives)—Includes case, powers supply (for 2 drives) cable, mounting hardware, (shown above) **$179.95**.

SPECIAL!!! Dual External Drives & Case, **$629.95**

NOTE: Dual Sided Drives—add **$99.00** per drive.
Flippy Drives (single sided only)—add **$19.95** per drive.
Dos Plus III (3.3) Operating System, **$99.95** alone, **$49.95** with drive purchase.

ACCESSORIES

Put Your Printer on a Pedestal!

This printer stand allows you to place a stack of paper under the printer for neat paper stacking—Available for most printers.

(MX-80, MX-80F/T, LPIV, etc.)	**$24.95**
Larger stand (MX-100, Anadex, etc.)	**$29.95**
Extra Shelves (Shown Above)	**$9.95**

MODEL I DISK DRIVE CASES

Clear Cases for your model I disk drives, complete with power supply for 5-1/4 inch disk drives.

Single Drive Case	**$79.95**
Dual Drive Case	**$109.95**
Triple Drive Case	**$139.95**
Quad Drive Case	**$169.95**

Smoked Plexiglass Cases, Add **$5.00**/per drive.

LYNX MODEMS

The finest Auto Answer/Dial Modem on the market! This is a direct connect modem for your model I or III. Does not require an RS232-C interface in your computer! Season Special, ~~$299.95~~ **$249.95**

AMBER EASE

Amber filter for your computer. Amber has been shown to be the easiest on the eyes. Easy to install. (Shown on computer above.) Season Special, All TRS-80 Models, **$19.95**

TIC-TOC-80 MICROCLOCK

A microclock for your TRS-80 Model I. Gives your computer the correct time and date on powerup. Clock is maintained on a lithium battery for long life. Plugs onto the I/O Buss of the TRS-80 Model I and does not require any modifications for most computers. Complete with extensive documentation and a disk with utilities. SPECIAL NOW AVAILABLE IN KIT FORM!

Complete assembled and tested	**$99.95**
Kit with P/C board, parts and instructions	**$59.95**
Kit with P/C board, parts list and instructions	**$19.95**

BT ENTERPRISES

171 Hawkins Road
Centereach, New York 11720
✓122
(516) 981-8568 (Voice)
(516) 588-5836 (Data)
MNET-70331, 105

VISA MasterCard

Dealer Inquiries Welcome
Add $2.00 S & H
NYS res. add appr. tax

✓See List of Advertisers on page 402

The kids use the computer, so why not you , professor?.

Calling All Teachers

Roger Hedden
B22-8 151 Rte. 206
Flanders, NJ 07836

Computers are appearing in more schools each day. Their uses fall into one of three broad groups: Students interact with canned or teacher-written programs to reinforce previously learned skills; students use the computers to learn a programming language; or teachers use them to ease paperwork burdens. Many vendors have attempted to answer the first two needs—this article will help you begin resolving the third.

The programs included have been tested for two marking periods. I have received positive comments from colleagues, administrators, and parents for the increase in communication made possible by them. The exciting part of this is that I have been able to save time in the process!

The package consists of seven modules (each module is actually a separate program):

- The Data Base Builder
- The Grading Program
- The File Update
- The Section Roster Printer
- The Data Base Display
- The Grade Book Printer
- The Failure Notice Printer

Data Base Builder

The data base is actually a data file; it contains information on every student I have during a marking period. Usually that amounts to about 120 (there's room for 150). After the program has been loaded and run, the title page is displayed followed by instructions to list the requested information. You should enter the names of your students one at a time, with their sex (M/F), and section number. Once all the names are entered, you create a new student whose last name is End.

Once this name has been entered the program will direct you to place your data tape in the cassette recorder; press play and record and enter. Your entire list will be recorded on tape. This is the last time you will have to type this information. All other programs in this package are designed to read and/or rewrite this file. For your data tape use a high quality tape that won't lose data. It must also be long enough to handle the file. I presently have 117 students, which takes about seven minutes of tape.

There are some important flags built into this file. The End flag mentioned above is used to terminate reading the tape. Another is the section number. In my school, the student population is divided into five clusters which are labeled A,B,C,D,E. Each cluster has at least one grade and three to four homerooms in each grade. For example, a student's section might be 6A1, meaning he or she is a 6th grader in A cluster, homeroom number 1. Within a teaching section, all students will be of the same grade and cluster but may not be of the same homeroom (see Table 1). The programs that use the data file look for a change in either of the first two places of the section from one student to the next in order to know that it is now looking at the beginning of a new section. In the discussion of the other programs, you will see how this operates.

I have structured the programs to make modifications as simple as possible.

The data base builder is used once to establish a new list of students. Any changes thereafter can be made by the file update program.

File Update

One program I've found to be an absolute necessity is the file

```
10 REM *BUILD DATA BASE (INTERFACES WITH ALL OTHER MODULES)*
20 CLEAR 4800:CLS:PRINT@85,"**DATA BASE BUILDER**":PRINT:PRINTST
RING$(63,"*"):PRINTTAB(26)"WRITTEN FOR":PRINTTAB(21)"NORTHERN MI
DDLE SCHOOL":PRINTTAB(31)"BY":PRINTTAB(25)"ROGER J HEDDEN":PRINT
@982,"*ALL RIGHTS RESERVED";:GOSUB1010:CLS
30 DIM LN$(150),F$(150),MF$(150),SC$(150)
31  STU=0
40  PRINT"LIST THE INFORMATION REQUESTED, PLEASE":PRINT
50  FOR I=1TO150
60    INPUT"LAST NAME: ";LN$(I):STU=STU+1
65    IF LN$(I)="END" THEN GOTO 150
70    INPUT"FIRST NAME: ";F$(I)
80    INPUT"SEX(M/F): ";MF$(I)
90    INPUT"SECTION: ";SC$(I)
100   PRINT
120 NEXTI:CLS
150 PRINT"LOAD BLANK CASSETTE AND PRESS RECORD & PLAY.":PRINT
160 INPUT"ONCE LOADED, PRESS ENTER.";G$:CLS
161 PRINT"THE FOLLOWING DATA IS BEING PLACED ON THE DATA BASE TA
PE:":PRINT
170 PTS=0:PP=PTS
180 FORI=1TOSTU
181  PRINT"NAME: ";LN$(I);", ";F$(I)
182  PRINT"SEX: ";MF$(I)
183  PRINT"SECTION: ";SC$(I):PRINT
190  PRINT#-1,LN$(I),F$(I),MF$(I),SC$(I),PTS,PP:NEXT I
200 CLS:PRINT"LOAD COMPLETE.":PRINT:PRINT:PRINT
210 INPUT"IS THERE ANOTHER SECTION(Y/N)";G$
220 IF G$="Y" THEN GOTO 20
230 CLS:END
1000 END:'TIMER DELAY
1010 FOR ZZ=1TO1000:NEXT:RETURN
```

Program Listing 1. Data Base Builder.

update. This program allows you to add a new student, delete one who has left, and change any field in any record of the file. When all changes have been made, the updated file is saved on tape.

When you record this new data, use the other side of the tape. This is wise, because if something should happen during the recording phase of the program (@?#*&!) only the most recent data would be lost. Also, make a note on the cassette to remind you which side has the most recent data on it. If you go back later to use a program and, as the tape is being read, it encounters a bad record in the file, one of two error messages will be displayed—OD? or FD? Relax, type CONT and (enter). The program will resume. That record will be lost but you can put it back in with the update. If you find you are getting a lot of error messages, check the volume, clean the heads, or beat your head against the wall, but don't give up!

You only have to run the program to see how it works. All the instructions are presented on the video screen to lead you through the program. Be sure you read these carefully. Also, when you add a name, a graphics block is displayed on the screen to indicate that the changes are being made. You may notice a period when the indicator stops changing. This occurs because the computer is searching for more string space. Be patient.

The Grading Program

The grading program is based on a few simple concepts. Over a period of time a teacher gives tests, quizzes, homework and class assignments, all of which are graded and recorded in a grade book. Most teachers like to weight these grades—that is, a test counts more than a quiz, a quiz more than an assignment. This program allows for weighting. For example, John Doe has scored 15 out of 20 points on a quiz, four out of five on an assignment, and was absent for a second quiz. How would the program handle these grades? When the program starts up, it prompts you to place the data file in the cassette player. The file is then read. At this point, go get a cup of coffee and relax. For my 117 students, it takes just over seven minutes. If you have a faster I/O device, consider yourself lucky.

Once the tape file has been read, a grade book will be dis-

NORTHERN MIDDLE SCHOOL

Science—Hedden		SECTION ROSTER		Cluster A
Able	I. M.	EA2	50	
Crab	John	EA1	60	
Drab	Sam	EA3	100	
Drools	Mike	EA5	81	
Falls	Water	EA1	50	
Money	Owen	EA3	86	
Dolly	Piano	EA2	90	
Farce	Polly	EA1	94	
Toys	Attic	EA3	61	
		SECTION ROSTER		Cluster B
Frog	Kermit	6B5	89	
Goose	Father	6B3	98	
Hassle	Adam	6B7	51	
Kong	King	6B3	25	
Load	Cigar	6B8	58	
Anna	Annie	6B1	78	
Disco	Debbie	6B3	84	
Katz	Kitty	6B2	85	
Strain	Adel	6B2	69	

Table 1.

```
10 REM *FILE UPDATE FROM DATA BASE*
15 CLEAR 4800
20 CLS:PRINT@88,"**FILE UPDATE**":PRINT:PRINT STRING$(63,"*"):PR
INTTAB(26)"WRITTEN FOR":PRINTTAB(21)"NORTHERN MIDDLE SCHOOL":PRI
NTTAB(31)"BY":PRINTTAB(25)"ROGER J HEDDEN":PRINT@982,"*ALL RIGHT
S RESERVED";
30 DIM LN$(150),F$(150),MF$(150),SC$(150),PTS(150),PP(150):GOSUB
 1010
40 CLS:PRINT@512,"LOAD YOUR TAPE TO BE UPDATED.":INPUT"THEN PRES
S 'ENTER'";A$:GOSUB2010
60 CLS:PRINT"ENTER THE APPROPRIATE CODE":PRINT:PRINT"ADD NEW REC
ORD - ENTER  A":PRINT"DELETE A RECORD - ENTER  D":PRINT"CORRECT
A RECORD - ENTER  C":PRINT"SAVE CHANGES - ENTER S":PRINT:PRINT"Y
OUR CHOICE PLEASE";
70 G$=INKEY$:IF G$=""THEN 70
80 IF G$="A"THEN GOSUB3010:GOTO120
90 IF G$="D"THEN GOSUB4010:GOTO120
100 IF G$="C"THEN GOSUB5010:GOTO120
105 IF G$="S" THEN GOTO 130
110 PRINT@384,"HMMM - TRY THAT AGAIN":GOSUB 1010:PRINT@384,"YOUR
 CHOICE PLEASE    "
120 PRINT"ARE THERE MORE RECORDS TO BE CHANGED (Y/N)?"
121 A$=INKEY$:IFA$=""THEN121
122 IF A$="Y"THENGOTO60
123 IF A$<>"N"THEN CLS:GOTO110
130 CLS:PRINT"PREPARE DATA TAPE AND HIT ENTER":INPUT A$
135 CLS:PRINT"LOADING"
140 FOR S=1TOSU
150   PRINT#-1,LN$(S),F$(S),MF$(S),SC$(S),PTS(S),PP(S)
155   PRINT@916,"RECORDS WRITTEN =";S;
160 NEXT S
165 PRINT#-1,"END","OF","FILE","DATA",1,1
170 CLS:PRINT"JOB COMPLETE"
180 END
1000 END:'TIMER DELAY
1010 FOR ZZ=1TO1000:NEXT:RETURN
2000 END:'READ DATA TAPE
2010 FOR S=1TO150
2020   INPUT#-1,LN$(S),F$(S),MF$(S),SC$(S),PTS(S),PP(S):SU=S
2025   PRINT@90,"RECORDS READ = ";S;
2030   IF LN$(S)="END" THEN RETURN
2040 NEXT S: RETURN
3000 END:'ADDS NAME TO FILE
3010 CLS:PRINT"PLEASE LIST NAME OF PERSON IMMEDIATELY PRECEDING
NEW ENTRY":INPUT"LAST NAME:";PL$:INPUT"FIRST NAME:";PF$
3020 CLS:PRINTCHR$(23):PRINT"SEARCHING - PLEASE WAIT"
3030 FOR S=1TOSU:C=S
3040   IF LN$(S)=PL$ AND F$(S)=PF$ THEN 3060
3055 NEXT S
3060 CLS:PRINT LN$(C);", ";F$(C);" SECTION: ";SC$(C)
3070 PRINT:PRINT"ENTER NEW NAME AND INFORMATION:"
3080 INPUT"LAST NAME";NL$:INPUT"FIRST NAME";N$:INPUT"SEX(M/F)";S
X$:INPUT"SECTION";NS$:INPUT"POINTS EARNED TO DATE";PE:INPUT"POIN
TS POSSIBLE TO DATE";PO
3090 CLS:PRINTCHR$(23):PRINT"WAIT, PLEASE":FOR I=SU TO C+1 STEP-
1:PRINT@10,CHR$(166):LN$(I+1)=LN$(I):F$(I+1)=F$(I):MF$(I+1)=MF$(
I):SC$(I+1)=SC$(I):PTS(I+1)=PTS(I):PP(I+1)=PP(I):PRINT@10,CHR$(1
53):NEXTI
3095 SU=SU+1:LN$(C+1)=NL$:F$(C+1)=N$:MF$(C+1)=SX$:SC$(C+1)=NS$:P
TS(C+1)=PE:PP(C+1)=PO:CLS:RETURN
4000 END:'DELETES A NAME
4010 CLS:PRINTCHR$(23):F=0:PRINT@512,"WARNING, THIS ROUTINE IS
     POWERFUL!":GOSUB1010:PRINT"  ONCE A NAME IS DELETED, IT IS
 LOST.":PRINT"  DO YOU WANT TO DELETE A NAME  (Y/N)?"
4015 A$=INKEY$:IF A$=""THEN 4015
4016 IF A$<>"Y" THEN RETURN
4017 CLS:INPUT"LAST NAME OF STUDENT TO BE DELETED";NL$:INPUT"FIR
ST NAME";N$:INPUT"SECTION";NS$
4018 CLS:PRINT"THANK YOU - PLEASE WAIT."
4020 FOR S=1TOSU
4030   IF (LN$(S)=NL$)AND(F$(S)=N$)AND(SC$(S)=NS$) THEN F=1:C=S:
GOTO4060
4050 NEXT S
4060 IF F=0 THEN PRINT "STUDENT NOT FOUND IN FILE -- CHECK SPELL
INGS!":RETURN
4065 FOR S=CTOSU
4066   LN$(S)=LN$(S+1):F$(S)=F$(S+1):MF$(S)=MF$(S+1):SC$(S)=SC$(
S+1):PTS(S)=PTS(S+1):PP(S)=PP(S+1)
4067 NEXT S:SU=SU-1
4070 RETURN
5000 END:'CORRECTS ERRORS IN FILE
5010 CLS:F=0:PRINT"ENTER ALL INFORMATION REQUESTED FOR RECORD TO
 BE CHANGED":PRINT"REMEMBER - ENTER IT AS IT PRESENTLY APPEARS."

5020 INPUT"OLD LAST NAME";EN$:INPUT"    FIRST NAME";EF$:INPUT"
  SECTION";ES$
5025 CLS:PRINT"SEARCHING - PLEASE WAIT"
5030 FOR S=1TOSU
5040   IF (LN$(S)=EN$)AND(F$(S)=EF$)AND(SC$(S)=ES$)THEN F=1:C=S:
GOTO5080
5070 NEXTS
5080 IF F=0 THEN PRINT"NOTHING FOUND IN FILE MATCHING REQUEST":R
ETURN
5090 CLS:PRINT"WHAT FIELD IS TO BE CHANGED?"
5100 PRINT"LAST NAME  - ENTER L"
5101 PRINT"FIRST NAME - ENTER F"
5102 PRINT"SECTION  -   ENTER S"
5103 PRINT"SEX(!!)  -   ENTER X"
5104 PRINT"PTS EARNED - ENTER P"
5105 PRINT"PTS POSSIBLE-ENTER O"
5106 PRINT:PRINT"YOUR CHOICE?"
5107 A$=INKEY$:IF A$=""THEN5107ELSECLS
5110 IF A$="L" THEN INPUT "CORRECT LAST NAME";LN$(C):GOTO6000
5120 IF A$="F" THEN INPUT"CORRECT FIRST NAME";F$(C):GOTO6000
5130 IF A$="S" THEN INPUT"CORRECT SECTION";SC$(C):GOTO6000
5140 IF A$="X" THEN INPUT"CORRECT SEX";MF$(C):GOTO6000
5150 IF A$="P" THEN INPUT"CORRECT POINTS EARNED";PTS(C):GOTO6000

5160 IF A$="O" THEN INPUT"CORRECT POINTS POSSIBLE";PP(C):GOTO600
0
6000 PRINT"ARE THERE MORE CHANGES TO BE MADE TO THIS RECORD (Y/N
)?"
6010 A$=INKEY$:IF A$=""THEN 6010
6020 IF A$="Y"THEN 5090
6030 RETURN
6040 END
```

Program Listing 2. File Update.

played on the screen with a message requesting the number of grades for this section. Going back to our example, the response would be three (don't forget the absence). Once the response is given, the next question appears at the bottom, asking the maximum possible points for each grade. In our example, we would respond with 20,5,15—the quiz John missed was worth 15 points. After the third maximum is entered, the first student's name is displayed at the top of the page. For John we would enter 15,4 and −1 (I'll explain this later). At the right side of the screen, John's percent grade will appear. If you don't have a printer, you'll need to copy this grade or use the data base display program to record your grades. If your school uses letter grades, you'll need to convert them or add a few lines of code to display the letter grade.

As soon as John's grade is displayed, the next student's name is shown on the following line and the process continues. When the next section is encountered (remember the program detects this by the section number) you are given an opportunity to change the maximum number of points for each score and the number of scores.

Now, that −1 mentioned earlier is one of four codes available during the operation of the program. They are listed at the bottom of the grading page. They are: -1: The student was absent and the grade is to be ignored; -2: Correct the spelling of the individual's last name; -3: Correct the line because of a grade input error; -4: Correct the previous line because of an error.

After all grades have been entered for all sections, the screen is cleared and you are instructed to place your data tape in the cassette recorder; play, record, and enter. The new data will be recorded and the program will end. The percent grade displayed is not recorded on tape, but the number of points earned and the number of points possible for that student are recorded. This is one of the most important features of the program because it allows you to update your grade book at any time without worrying about throwing the grades out of balance. The next time the grading program is run, the previous grades are read and the new grades accumulated.

Display Data Base

This program was designed for schools that do not have a printer. It will display the contents of the data file. No changes can be made by this program. Execution can be halted by a shifted @ and resumed by depressing any key.

Printed Output

So far none of the programs provide printed output. If you have a printer, read on! The three following programs are for

```
10 REM *STUDENT GRADES FROM DATA BASE*
20 CLS:PRINT@86,"**STUDENT GRADES**":PRINT:PRINT STRING$(63,"*")
:PRINTTAB(26)"WRITTEN FOR":PRINTTAB(21)"NORTHERN MIDDLE SCHOOL":
PRINTTAB(31)"BY":PRINTTAB(25)"ROGER J HEDDEN":PRINT@982,"*ALL RI
GHTS RESERVED";:GOSUB1010:CLS
70 PRINT" PUT YOUR DATA TAPE IN THE CASSETTE PLAYER AND PRESS 'P
LAY'.":FOR X=1TO10:PRINT:NEXT:INPUT"THEN PRESS 'ENTER'";A$:CLS
80 PRINT CHR$(23):PRINT:PRINT"YOUR TAPE IS NOW LOADING. WHEN  IT
 HAS FINISHED LOADING, TURN   YOUR CASSETTE TO THE OPPOSITE   SI
DE AND REWIND IT TO THE BEGIN-NING.":FOR ZZ=1TO1000:NEXTZZ:CLS:P
RINT"yOUR DATA IS AS FOLLOWS:"
90 CLEAR 4800
100 DIM LN$(150),F$(150),MF$(150),SC$(150),PTS(150),PP(150),T(25
),I(25)
110 Q=129
120 GOSUB 3010:'*READS INPUT FILE*:CLS
130 SC$(0)=SC$(1):CLS:GOSUB 3030:GOSUB 3070
140 FOR S=1TO(SU-1)
150   IF LEFT$(SC$(S),2)<>LEFT$(SC$(S-1),2) THEN PRINT@Q,"END OF
 SECTION";:INPUT "//PRESS ENTER FOR NEXT SECTION";A$:GOSUB 3030:
GOSUB 3070
160 PRINT@Q,STRING$(63," "):PRINT@Q,LN$(S);", ";LEFT$(F$(S),1)
170 T=17
180 FOR I=1TON
190  PRINT@(Q+T),"";:INPUT ST(I)
200  PRINT@(Q+T),ST(I):T=T+3
210 IF ST(I)>=0 THEN PTS(S)=PTS(S)+ST(I):PP(S)=PP(S)+T(I):NEXTI:
ELSE IF ST(I)=-1THEN NEXTI:ELSE IF ST(I)=-2 THEN GOSUB 3210:PTS(
S)=0:PP(S)=0:NEXT S ELSE IF ST(I)=-3 THEN S=S-1:PTS(S)=0:PP(S)=0
:NEXTS ELSE IF ST(I)=-4 THEN S=S-2:PTS(S)=0:PP(S)=0:NEXT S
215 ON ERROR GOTO 240
220 F=INT((PTS(S)*100)/PP(S))
230 PRINT@(Q+58),F;:Q=Q+64:IF Q>832 THEN Q=129
240 NEXT S
250 PRINT@Q,STRING$(63," "):PRINT@Q,"LAST STUDENT OF LAST SECTIO
N--PRESS 'ENTER' TO PROCEED";:INPUT A$
260 CLS:PRINT CHR$(23)
270 INPUT"MAKE SURE YOUR TAPE IS WOUND    PAST IT'S LEADER. PRES
S 'PLAY'  AND 'RECORD'. wHEN YOU'RE READY PRESS 'ENTER'.";INPUT
A$:PRINT
280 PRINT"THE NEWEST DATA IS BEING RECORD-ED ON TAPE--THIS WILL
TAKE A FEW MINUTES."
290 GOSUB 3150
295 PRINT#-1,"END","OF","FILE","SECTION",1,1
300 CLS:PRINT CHR$(23):PRINT"ALL DONE --- BYE."
310 END
1000 END:'TIMER DELAY
1010 FOR ZZ=1TO1000:NEXT:RETURN
3000 END:'READS INPUT FILE FROM LINE 120
3010 FOR S=1 TO 150:INPUT#-1,LN$(S),F$(S),MF$(S),SC$(S),PTS(S),P
P(S):SU=S:IF LN$(S)="END" THEN RETURN
3015 PRINT LN$(S);", ";F$(S);" ";SC$(S);" ";:IF PTS(S)<>0 THEN P
RINT INT(PTS(S)/PP(S)*100);"%" ELSE PRINT "NO GRADE"
3016 NEXT S:RETURN
3020 END:'*PRINTS HEADERS*
3030 CLS:PRINT"NAME";TAB(20)"GRADES - - -";TAB(62)"%"
3040 FOR X=0TO127:SET(X,3):SET(X,44):NEXT
3050 PRINT@960,"CODES:STUDENT ABSENT(-1)**CORRECTIONS-NAME(-2)-A
LL(-3)-LAST(-4)";:RETURN
3060 END:'SET NUMBER OF INPUT GRADES
3070 PRINT@641,"";:INPUT"HOW MANY GRADES ARE THERE FOR THIS SECT
ION";N
3080 PRINT@641,STRING$(60," "):W=850
3090 PRINT@833,"MAX. POINTS";:FOR I=1 TO N:PRINT@W,"";:INPUT T(I
):PRINT@W,T(I):W=W+3:NEXT I
3100 RETURN
3110 END:'CORRECT NAME OR GRADE
3120 PRINT@Q,STRING$(63," "):PRINT@Q,"";:INPUT"CORRECT LAST NAME
";LN$(S):PRINT@Q,LN$(S);", ";LEFT$(F$,1)
3130 RETURN
3140 END:'OUTPUTS DATA TO TAPE.
3150 FOR S=1TOSU
3160  PRINT#-1,LN$(S),F$(S),MF$(S),SC$(S),PTS(S),PP(S)
3170 NEXT S
3180 RETURN
3200 END
3210 IF ST(I)=-2 THEN GOSUB 3120:S=S-1:RETURN
3215 END
```

Program Listing 3. Grading Program.

```
10 REM *LIST SECTION ROSTERS*
20 CLEAR500:CLS:PRINT@87,"**PRINT ROSTERS**":PRINT:PRINT STRING$
(63,"*"):PRINTTAB(26)"WRITTEN FOR":PRINTTAB(21)"NORTHERN MIDDLE
SCHOOL":PRINTTAB(31)"BY":PRINTTAB(25)"ROGER J HEDDEN":PRINT@982,
"*ALL RIGHTS RESERVED";
30 L=63:GOSUB 1010:CLS
40 INPUT"YOUR NAME PLEASE";T$
60 INPUT"WHAT SUBJECT";S$
70 CLS:PRINT"THANK YOU":GOSUB1010:PRINT@512,"LOAD YOUR DATA TAPE
 - THEN PRESS ENTER";:INPUT A$
80 LS$="AAA"
90 INPUT#-1,LN$,F$,MF$,SC$,PTS,PP
100 IF LEFT$(SC$,2)<>LEFT$(LS$,2)THEN GOSUB 2005
110 LPRINTTAB(5)LN$;TAB(20)F$;TAB(35)SC$;:IF PP<>0 THEN LPRINTTA
B(50) INT(PTS*100/PP)ELSE LPRINTTAB(50)"NO GRADE"
120 L=L+1:GOTO 90
1000 END:'TIMER DELAY
1010 FOR ZZ=1TO1000:NEXT:RETURN
2000 END
2005 IF LN$="END"THEN GOSUB 3010:END
2010 FOR Z=LTO66:LPRINT" ":NEXT Z
2020 LPRINTTAB(16)"N O R T H E R N   M I D D L E   S C H O O L
":LPRINT" "
2030 LPRINTTAB(5) S$;" - ";T$;TAB(60)"CLUSTER ";MID$(SC$,2,1)
2040 LPRINTTAB(5) STRING$(24,"*");" SECTION ROSTER ";STRING$(24,
"*"):LPRINT" "
2050 L=6:LS$=SC$:RETURN
3000 END
3010 CLS:PRINT"END OF FILE":PRINT:GOSUB1010:PRINT"TO ADVANCE PAP
ER, PRESS 'A'.":PRINT"TO END PROGRAM, PRESS 'E'."
3020 A$=INKEY$
3025 IFA$=""THEN 3020
3030 IF A$="A"THEN LPRINT" ":GOTO 3020
3040 IF A$<>"E" THEN GOTO 3020
3050 END
```

Program Listing 4. Section Roster Printer.

```
10 REM *CRT DISPLAY OF DATA BASE*
20 CLS:PRINT@85,"**DISPLAY DATA BASE**":PRINT:PRINTSTRING$(63,"*
"):PRINTTAB(26)"WRITTEN FOR":PRINTTAB(21)"NORTHERN MIDDLE SCHOOL
":PRINTTAB(31)"BY":PRINTTAB(25)"ROGER J HEDDEN":PRINT@982,"*ALL
RIGHTS RESERVED";:GOSUB1010:CLS:PRINTCHR$(23)
30 CLEAR 4800:DIM LN$(150),F$(150),MF$(150),SC$(150),PTS(150),PP
(150)
31 INPUT"PREPARE TAPE - THEN HIT ENTER";Q$
50 FOR I=1 TO 150
60  INPUT#-1, LN$(I),F$(I),MF$(I),SC$(I),PTS(I),PP(I)
65  IF LN$(I)="END" THEN GOTO 100
66  PRINT:PRINT"NUMBER ";I;" OF LIST."
70  PRINT LN$(I);", ";F$(I);" ";MF$(I);"--";SC$(I)
75  IF PP(I)>0 THEN PRINT"GRADE ";INT(PTS(I)*100/PP(I));"%"
80 PRINT
90 NEXT I
100 PRINT "JOB COMPLETE"
110 END
1000 END:'TIMER DELAY
1010 FOR ZZ=1TO1000:NEXT:RETURN
```

Program Listing 5. Data Base Display.

See List of Advertisers on page 402

use with a printer only. If you attempt to use them with no printer attached or the printer off, the computer will freeze when it attempts to execute a print.

Grade Book Printer

Remember that when you typed the names of your students into the data base builder, I said it was the last time you'd have to do it? Here's proof: The program is designed for 9.5 inch by 11 inch pin feed paper. After the print operation is completed, I tear off the perforations, punch three holes in it and put it in a looseleaf notebook. If the page gets full, I generate another. The program prints 10 characters per inch and six lines per inch vertically. It will print one section per page and automatically advance to the next page for the next section. I use Okidata's Microline 80. If you use this program, you should change line 1015 to reflect the name of your school.

There are two operator entries required at the beginning of the program, the teacher's name and subject. Once these have been entered and the tape is in place, the program takes it from there.

The Section Roster Printer program is similar to the grade book printer except that it prints the grade status of each student instead of a series of boxes.

The Failure Notice Printer

One of the objections often heard from parents is that the school doesn't communicate with them regarding their child's grade. Teachers have a tremendous amount of paperwork to do, and it is easy to ignore a failing grade. This program reads the data tape and generates a note to the parent of each child who is not passing the course. After all notices have been printed, the program prints a report for the teacher so he can keep track of the signed notes as they come back. I use roll paper to print these notes with a cut line for separating them. They are designed so they can be folded and stapled with the name and section showing.

If you want to ease into it, try the data base builder and grading program first. Once you've done this much, you'll either be hooked or hate it. If my experience is any indication, you'll spark a lot of other people's interest when you sit with your feet propped up reading your favorite professional journal while your colleagues sweat over their grade reports. What used to take me two hours now takes me 20 minutes! ■

```
10 REM *PRINTS A GRADE BOOK PAGE FOR EACH SECTION*
20 CLEAR 500:CLS:PRINT@85,"**GRADE BOOK PRINTER**":PRINT:PRINTST
RING$(63,"*"):PRINTTAB(26)"WRITTEN FOR":PRINTTAB(21)"NORTHERN MI
DDLE SCHOOL":PRINTTAB(31)"BY":PRINTTAB(25)"ROGER J HEDDEN":PRINT
@982,"*ALL RIGHTS RESERVED";:GOSUB5010:CLS
25 L=66:LS$="AAA":INPUT"TEACHER'S NAME ";T$:INPUT"SUBJECT ";S$:C
LS:INPUT"INSERT DATA TAPE AND PRESS 'PLAY'";G$:CLS:PRINT"WORKING
"
30 GOSUB 2010:'READ RECORD FROM INPUT FILE
40 IF LEFT$(SC$,2)<>LEFT$(LS$,2)THEN GOSUB1010
50 IF LN$<>"END"THENGOSUB3010:GOTO30
60 PRINT"JOB COMPLETE":END
1000 END
1010 IF L<>66THEN GOSUB4010:IF LN$="END"THEN RETURN
1015 LPRINTTAB(16)"N O R T H E R N   M I D D L E   S C H O O L
":LPRINT" "
1020 LPRINTTAB(5) S$;" - ";T$;TAB(60)"CLUSTER ";MID$(SC$,2,1):LP
RINTTAB(5) STRING$(70,"*"):LPRINT" ":L=6:LS$=CS$
1030 RETURN
2000 END
2010 INPUT#-1,LN$,F$,MF$,SC$,PTS,PP
2020 RETURN
3000 END
3010 LPRINTTAB(4) LN$;", ";LEFT$(F$,1);TAB(20)SC$;TAB(24)" ";
3020 FOR X=1TO17
3030   LPRINT"..:";
3040 NEXT X
3045 LPRINT" "
3050 L=L+1:LS$=SC$
3060 RETURN
4000 END
4010 FORS=LTO66:LPRINT" ":NEXT S
4020 RETURN
5000 END:'TIMER DELAY
5010 FORZZ=1TO1000:NEXT:RETURN
```

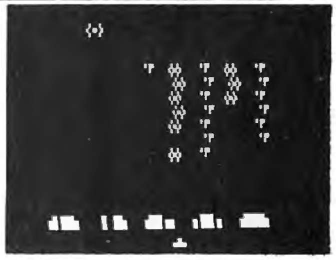
Program Listing 6. Grade Book Printer.

```
10 REM *PRINT FAILURE NOTICE REPORTS AND REPORT*
20 CLS:CLEAR 4800:E=0:I=1:PRINT@86,"**FAILURE NOTICES**":PRINT:P
RINTSTRING$(63,"*"):PRINTTAB(26)"WRITTEN FOR":PRINTTAB(21)"NORTH
ERN MIDDLE SCHOOL":PRINTTAB(31)"BY":PRINTTAB(25)"ROGER J HEDDEN"
:PRINT@982,"*ALL RIGHTS RESERVED";
25 DIM L1$(150),F1$(150),S1$(150),G1(150):GOSUB1010:CLS:INPUT"TE
ACHER'S NAME";T$:INPUT"TEACHER'S SUBJECT";S$:INPUT"REPORT DATE";
D$:CLS:INPUT"LOAD DATA TAPE, PRESS 'PLAY', THEN 'ENTER'";G$
30 INPUT#-1,LN$,F$,MF$,SC$,PTS,PP
40 IF LN$="END" THEN GOSUB 2010:END
50 IF INT(PTS*100/PP)<60 THEN GOSUB 3010
60 GOTO 30
70 '************ END OF MAIN PROGRAM **************************
1000 END:'TIMER DELAY
1010 FOR ZZ=1TO1000:NEXT:RETURN
2000 END:'PRINTS MASTER REPORT
2010 LPRINTSTRING$(80,"-"):FOR L=1TO3:LPRINT" ":NEXT
2020 LPRINTTAB(16)"N O R T H E R N   M I D D L E   S C H O O L
":LPRINT" "
2030 LPRINTTAB(5)S$;" - ";T$;TAB(60)D$
2040 LPRINTTAB(5)STRING$(24,"*");" FAILURE REPORT ";STRING$(24,"
*"):LPRINT" "
2050 FOR S=1 TO E
2060   LPRINTTAB(5) L1$(S);TAB(20) F1$(S);TAB(35) S1$(S);TAB(45)
 G1(S);TAB(55)"RETURNED-(  )"
2070 NEXT S
2080 RETURN
3000 END:'PRINTS A FAILURE NOTICE FOR EACH FAILING STUDENT
3010 IF MF$="M" THEN PN$="HIM":PO$="HIS" ELSE PN$="HER":PO$="HER
"
3030 LPRINTSTRING$(80,"-"):LPRINT" ":LPRINT" "
3040 LPRINTTAB(5)LN$;", ";F$;TAB(65)"SECTION ";SC$:LPRINT" ":LPR
INT" "
3050 LPRINTTAB(10)"I THOUGHT THAT YOU SHOULD KNOW THAT ";F$;"'S
";S$;" GRADE IS"
3060 LPRINT"PRESENTLY ";INT(PTS*100/PP);"% .  PLEASE TAKE A FEW
MINUTES TO DISCUSS THIS WITH ";PN$;"."
3070 LPRINT"IF YOU HAVE ANY QUESTIONS ABOUT ";PO$;" WORK, PLEASE
 GIVE ME A CALL.":LPRINT" "
3080 LPRINTTAB(10)"PLEASE SIGN AND RETURN THIS WITH ";F$;".  THA
NK YOU.":LPRINT" "
3090 LPRINTTAB(40)"SINCERELY,":LPRINT" "
3100 LPRINTTAB(40)T$;" - ";S$:LPRINT" "
3110 LPRINTTAB(5)STRING$(20,CHR$(95));TAB(60)D$
3120 LPRINTTAB(5)"Signature"
3130 FOR L=1TO15:LPRINT" ":NEXT
3140 L1$(I)=LN$:F1$(I)=F$:S1$(I)=SC$:G1(I)=INT(PTS*100/PP):I=I+1
:E=I
3150 RETURN
```

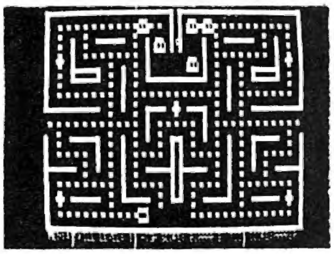
Program Listing 7. Failure Notice Printer.

Enter The Soft Sector Marketing
ARCADE CENTER

ALIEN DEFENSE
©1981 Soft Sector Mkt.
Mod I or III Tape, 16K
$19.95
Mod I or III Disk*, 32K
$24.95
Only Mod III version has Extended Graphics.

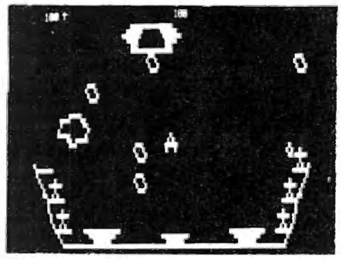


SUPER VADERS
©1981 Soft Sector Mkt.
Mod I or III Tape **$19.95**
Mod I Disk **$24.95**
Mod III Disk **$24.95**
– 2 Player - 10 Levels –
Ask for upgrade info for INVADERS PLUS & TRS-SUPER INVADERS

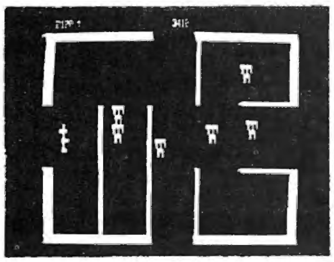


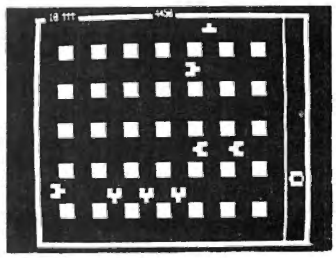
SCARFMAN
©1981 Cornsoft Group
Mod I & III Tape
$15.95
Mod I & III Disk
$19.95

METEOR MISSION 2
by Big 5
Mod I & III Tape
$15.95
Mod I & III Disk*
$19.95

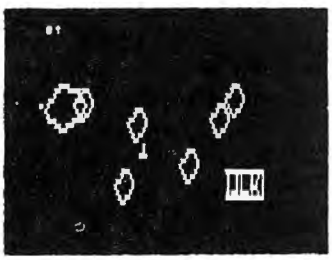


ROBOT ATTACK
by Big 5
Mod I & III Tape
$15.95
Mod I & III Disk
$19.95



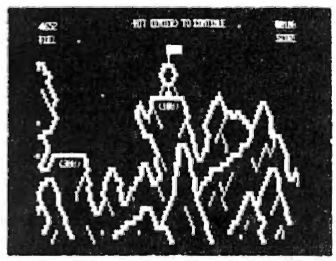


ATTACK FORCE
by Big 5
Mod I & III Tape
$15.95
Mod I & III Disk*
$19.95

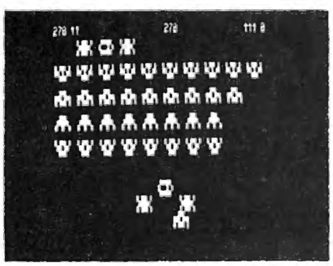


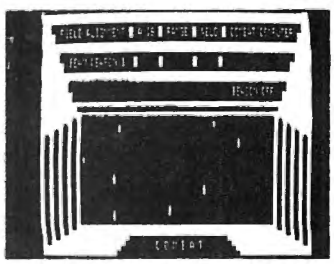
SUPER NOVA
by Big 5
Mod I & III Tape
$15.95
Mod I & III* Disk
$19.95

LUNAR LANDER
by Adventure
Mod I & III Tape
$15.95
Mod I & III Disk
$20.95



GALAXY INVASION
by Big 5
Mod I & III Tape
$15.95
Mod I & III* Disk
$19.95





STAR FIGHTER
by Adventure
Mod I & III Tape
$24.95
Mod I Disk
$29.95

*These Disk versions save high score to disk.

SPECIAL DEAL – This Month Only
Purchase 1 Game - List Price / Purchase 2 Games - Deduct 10%
Purchase 3 Games - Deduct 15% / Purchase 5 Games - Deduct 20%
– SPECIAL DEAL applies only to games purchased from this ad, at one time, and at list price –

SSM SOFT SECTOR MARKETING, INCORPORATED

MasterCard

✓515

Order Line Only
800-521-6504

6250 Middlebelt
Garden City, Michigan 48135

VISA

MI Orders & Questions
(313) 425-4020

Page 1 of 3

✓See List of Advertisers on page 402

Lazy Writer®

*A product of ABC Sales

GUARANTEE

Lazy Writer® comes with a
30 DAY, MONEY BACK† GUARANTEE
Guarantying There Is Nothing Even Close!
†Less 10% restocking charge.

If you are looking for The Finest In Word Processing Systems you can . . .

1. Purchase and examine all other word processing programs on the market, and then purchase **Lazy Writer®**

   or . . .

2. You can purchase **Lazy Writer®** first. **It's your money!**

Lazy Writer® Mod I still only **$125.00**
till Dec. 31, 1981
Lazy Writer® Mod III **$175.00**
*Requires 2 Disk Drives to convert only.

SPECIAL DEAL

Purchase Lazy Writer® and receive, this month only,
MICRO PROOF with the correction feature and patched to Lazy Writer®
Only **$149.50** extra
Reg. $184.50

Now available from

SSM SOFT SECTOR MARKETING, INCORPORATED 495

For those of you who requested
a spelling checker to use with Lazy Writer®

MICRO PROOF

Product of Cornucopia Software

- EASY TO USE
- 50,000 WORD VOCABULARY & EXPANDABLE
- RECOGNIZES PREFIXES & SUFFIXES
- REQUIRES ONLY 32K OF MEMORY & 2 DISK DRIVES
- AFFORDABLE & AVAILABLE IN THREE FORMS: fully independent program to identify errors; independent program to identify and correct errors; or with a conversion program that will permit MICRO PROOF to operate from within.

New Lower Price!	Was	Now Only
TRS-MOD I or III Version	$125.00	**$89.00**
Correcting Feature		Add **$60.00**
Word Processing Conversion for Lazy Writer*		Add **$35.00**

Let MICRO PROOF
Find Your Errors
Before Someone Else Does!

MasterCard

C.O.D.—certified check, M.O. or cash only. Most orders shipped next day. All orders must have shipping included. Please add 2% or $2.50, whichever is higher for shipping. Michigan residents, please add 4% tax. Add extra $1.50 for C.O.D. Personal checks take 3 weeks to clear.
Out of the country orders add $10.00 extra shipping.

*TRS-80 is a product of Radio Shack, division of the Tandy Corporation.

VISA

Order Line **800-521-6504**
MI Orders & Questions **313-425-4020**

✓See List of Advertisers on page 402

An old fashioned project for sentimental Model 15 owners.

Baudot LPRINT

Winford Rister
Route 10, Box 33
Lake City, FL 32055

It only took a few hours on my new TRS-80 for me to realize that some type of hard copy was essential for any serious programming. Not being willing to invest the amount of money necessary to purchase a good printer, and preferring not to buy an inferior one, I took a hard look at using my model 15 teletype, at least as a temporary measure.

Now, unless you either own a baudot machine or can get one for a short song, the concessions which go along with using one make it impractical. The major disadvantage is not the lack of speed, but the inability to print many of the standard ASCII characters. The snail's pace printing is annoying, but if you plan your evenings well you can overcome that part. Just do your LLISTs and long LPRINTs as you watch television or while having your evening meal.

If you are still reading I assume you are either as stubborn as I am, have your own baudot printer, or are curious. Assuming you have a current loop driver which is TTL compatible, the hardware interface is very simple, requires no modification to the TRS-80, no connection to the keyboard port, and does not interfere with any cassette operation. It does connect to the cassette port and therefore requires a method of connecting to the miniature phone plug that goes to the recorder. I installed a simple Y adapter in my recorder, plugged the lead from the keyboard into one side, and the wire going to the printer interface into the other side. This eliminates constantly plugging and unplugging the recorder, and running the risk of breaking the wires from fatigue. The two functions work well, though wired parallel.

Hardware

The circuit is shown in Fig. 1. A machine-language program takes input from the LPRINT function, converts the data to baudot, serializes it, and outputs it to the cassette plug.

This output is fed into one section of an LM3900 op amp to bring the 0.8 volt pulses from the recorder port up to TTL level. The two sections of the 7414 Schmidt trigger inverters serve as buffers. The output of the 7414 goes to your TTL drivable current loop. Several articles have been published in *73* and *Kilobaud* for 60 and 20 mA current loops.

In this configuration a mark condition is represented by a high TTL level. This causes the loop to be active when the power is on the interface except when the TRS-80 sends a space signal. This prevents the machine from running "open".

Since this is a temporary measure for me, I stopped with the breadboard version of the interface. I have been using it for several months without any problems.

Software

The program is straightforward, although it may not appear so. Program Listing 1 is the assembly listing (the program was hand assembled). The character to be LPRINTed is passed to the LPRINT routine in the C register by Basic. The normal LPRINT routine begins at location 058D H, which is pointed to by locations 4026H and 4027H of the printer device control block (DCB). This Basic program POKEs the starting address of the new routine into these locations. That is, it puts location 7F00H into locations 4026H and 4027H. All subsequent LPRINT or LLIST commands will now go to our new location.

The character to be printed is limited to 7FH (to limit the table size). The program checks the incoming character for specials such as space, carriage return, %, <, =, etc. After this checking, a determination is made as to whether or not the carriage should be shifted. Once this decision is made, a flag is checked to see if the printer is in the necessary condition, and if not, sets it to the proper condition. Since there is no feedback from the printer it is necessary to disable the "unshift on space" feature. (On the model 15 this is a lever just above the keyboard.)

Now that our routine has properly determined the printer condition, it gets a substitute character from the table 7E00H to 7E7FH, for the ASCII character to be printed. This character is passed to the OUTCH subrou-

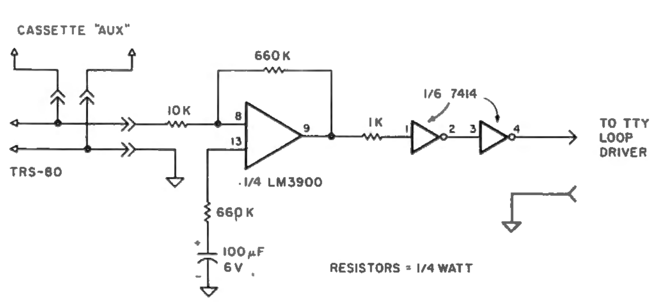


Fig. 1. Circuit Diagram.

6 ways to give your TRS-80* a +PLUS

Exhaustive research and professional standards bring you the best in business software.

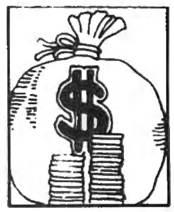
EASYTRAK™

KEEP YOUR SALES CURVING UPWARD WITH EASYTRAK™, a powerful, all-purpose activity tracking system. User-specified options enable you to monitor sales activity by territory, product line, salesman, or by combinations. Can monitor inquiries, sales calls and order booking. Production activity (by shift, by product, by day, by team, etc.), spending or income activity are simple to set up with EASYTRAK™. Extensive video displays make available individual, group or total summaries, relative averages, and percentage data all at the touch of a button. Use of an 80-column printer is optional.

MODEL I/III $125
MODEL II $150

+ACCOUNTS RECEIVABLE

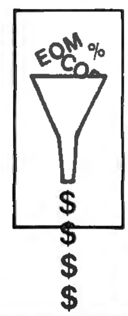
PUT MORE CASH IN YOUR BANK ACCOUNT with +Account Receivable's many collection/credit features, including average payment days, aging schedule and statements. Open-item system records sales activity this year and last, optional automatic discounts and salesman assignment. Cash receipts and A/R adjustments feature extensive on-line validation. And, yes, +Accounts Receivable handles the pesky new 9-digit zip code. Other optional features include service charges to selective customers; partial and "on-account" payments. Posts, at your option, to +General Ledger.



MODEL I/III $325
MODEL II $350

+ACCOUNTS PAYABLE

FUNNEL THAT STACK OF BILLS into +Accounts Payable system and create an orderly, manageable flow of vendor histories, cash requirements forecasts, check registers and checks, aging schedules, etc. Will accept manual checks, reversals, and adjustments. Pay by selected vendor, invoice, due date, discount dates, and optionally post directly to your +General Ledger. Supports 5-digit vendor numbers.

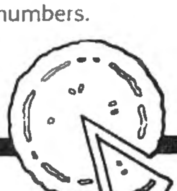


MODEL I/III $325
MODEL II $350

+GENERAL LEDGER

SLICE OUT MORE PROFIT by using +General Ledger to monitor your financial status. Features include fully-flexible financial statements, both budget and last year comparisons, special report options, monthly and quarterly trial balances, manual and/or automatic journal entries all with extensive audit trails. You structure the chart of accounts with up to 5 digit account numbers.

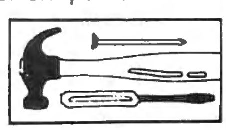
MODEL I/III $325
MODEL II $350



+FIXED ASSETS

A POWERFUL SET OF TOOLS TO CALCULATE and monitor fixed asset expenditures, book and tax depreciation, investment tax credit, etc. Will even forecast future depreciation for you. (Handy at budget time!) Code structure enables you to classify assets by departments, buildings, groups or other categories. Handles seven depreciation methods. Can post to +General Ledger for you.

MODEL I/III $325
MODEL II $350



+PAYROLL

KEEP YOUR EMPLOYEES HAPPY and smiling with prompt, accurate and professional payroll records. Will handle weekly, biweekly, semimonthly, monthly payrolls, produce checks, check registers, labor reports, earnings reports, Form 941's, W2's, etc. Can handle fixed/voluntary deductions, hourly/salary employees, bonuses, city and state taxes, etc. Easy to use and update tax tables. Automatic +General Ledger posting.

MODEL I/III $325
MODEL II $350

SPECIAL! All of the above: MODEL I/III $1600. MODEL II $1750.

+PLUS SOFTWARE comes to you feature-packed from on-line validation to machine-language file handling. Extensive documentation and thorough installation guide accompany each order. All systems are furnished on disc. Source code is included. Texas residents add 5% sales tax.

VISA master charge

To order, call TOLL-FREE 24 hours
1-800-231-7315

+PLUS

PLUS COMPUTER TECHNOLOGY, INC.
POST OFFICE BOX 1152
ANGLETON, TEXAS 77515 (713) 849-1108

✓131

Dealer inquiries invited

*TRS-80 is a registered trademark of Tandy Corp
*EASYTRAK is a trademark of Plus Computer Tech. Inc

✓See List of Advertisers on page 402

tine in the A register. By using a timing loop the character is shifted out serially through port FF, the cassette port. All bits except bit 0 are forced to 0 during the LPRINTing. This keeps the cassette motor relay from trying to open and close at 45 baud!

If you look at the baudot substitution table, Fig. 2, you will see that the five-bit character has an eight-bit code. That is because the substitute character contains the data bits, the start bit and two stop bits. The stop bit requirement is actually about one and one half times the period for a data bit. But I found that my printer works perfectly with only one data-bit-length stop bit. So I ignore the second stop bit by only shifting out seven bits (location 7FBBH). If your machine has problems you can change this to 08H and output eight bits. However, this will cost about 12 percent in speed.

The timing loop period is determined by the value stored in locations 7E84H, 7E85H. I found that there is a wide range of acceptable values, and this is the center of the range. If you have a faster baudot machine, this number will have to be reduced accordingly.

Location 7F31 contains the character number upon which an automatic carriage return and line feed will be issued. No characters are lost in this function. I made no provision for page format, yet a counter could be incremented after each CR,LF and checked for page length.

Using the Program

All addresses given are for a 16K system, but this program works well in a 4K system as well. Fig. 3 lists the changes for a 4K system. Program Listing 1 is the Basic listing.

To use the program enter a number less than 32255 to the memory size prompt. For simplicity, I always enter 32000 because it causes problems if I forget and put a five where a two should be. Next CLOAD the program and run it. As the program executes, it prints the hex code on the screen while it POKEs into memory. As soon as you get the Ready prompt you can remove this Basic program from memory and use Basic as usual. Now the LLIST and LPRINT functions will operate your printer.

For safety, I always LPRINT a number and letter to get the printer in position after I first load the routine. The motor should be turned on after the loop is activated *and turned off* before the loop is deactivated. Programs which do not heed the memory size restrictions such as the Editor/Assembler will have to be modified (*Kilobaud*, Jan. 80) if you want to use this hard copy routine with them.

Character	Hex Code	Character	Hex Code
A -	C6	Q 1	EE
B ?	F2	R 4	D4
C :	DC	S Bell	CA
D $	D2	T 5	E0
E 3	C2	U 7	CE
F !	DA	V ;	FC
G &	F4	W 2	E6
H #	E8	X /	FA
I 8	CC	Y 6	EA
J '	D6	Z "	E2
K (	DE	LF LF	C4
L)	E4	CR CR	D0
M .	F8	SP SP	C8
N ,	D8	LTRS	FE
O 9	F0	FIGS	F6
P 0	EC	IDLE	FF

Fig. 2. Substitution Table.

```
LINE 10 M = 19968 ' FIRST BYTE OF PROTECTED MEMORY
LINE 100 IF M = 20102 THEN GOTO 500
LINE 105 IF M > 20438 THEN END
LINE 500 M = 20224
LINES 1001 TO 1065 CHANGE ALL REMARKS OF 7EXX OR 7FXX TO 4EXX OR
4FXX RESPECTIVELY

1040  DATA  79,E6,7F,FE,0D,CA,AA,4F,FE,25,28,62,FE,20,28,67
1042  DATA  FE,3C,28,6C,FE,3D,28,73,FE,3E,28,7A,FE,5B,CA,A1
1044  DATA  4F,FD,21,00,4E,CD,2B,4F,C3,3A,4F,F5,3A,80,4E,3C
1046  DATA  FE,49,CC,AA,4F,32,80,4E,F1,C9,32,82,4E,FE,41,30
1048  DATA  11,FD,CB,81,46,28,1C,DD,2A,82,4E,DD,4E,00,CD,BA
1050  DATA  4F,C9,FD,CB,81,46,28,EF,FD,CB,81,86,3E,FE,CD,BA
1052  DATA  4F,18,E4,FD,CB,81,C6,3E,F6,CD,BA,7F,18,D9,3E,25
1054  DATA  CD,25,4F,CD,47,4F,C9,3E,C8,CD,2B,4F,CD,BA,4F,C9
1056  DATA  3E,4C,CD,25,4F,3E,54,CD,25,4F,C9,3E,45,CD,25,4F
1058  DATA  3E,51,CD,25,4F,C9,3E,47,CD,25,4F,3E,54,CD,25,4F
1060  DATA  C9,3E,1B,CD,25,4F,CD,47,4F,C9,3E,D0,CD,BA,4F,3E
1062  DATA  C4,CD,BA,4F,3E,00,32,80,4E,C9,06,07,F5,E6,01,EE
1064  DATA  01,D3,FF,F1,CB,2F,CD,CC,4F,10,F1,C9,11,01,00,2A
1066  DATA  84,4E,ED,52,20,FC,C9
```

Fig. 3. Substitutions for 4K System.

There are a few subtle traps which you should be aware of. An example is the substitution for less than, which is LT. If you list a program and later reenter it from the listing and don't catch the substitutions, you will run into problems. The break key does not seem to work while the printer is going, but it will if you will just hold it down until the TRS-80 issues a CR,LF.

In summary I would like to list what I think are the most important advantages and disadvantages of using a model 15 printer with your TRS-80.

Advantages

Baudot printers are easily obtained and cheap. This interface can be constructed for less than five dollars (excluding power supply). This routine is easily stored on cassette for easy loading and editing, and the POKEd version is about a half kilobyte. Regular paper can be used, which is much easier to read than 20 or 40 column print.

Disadvantages

All ASCII characters cannot be printed, so substitutes must be made. The print is very slow and very noisy, and some specialized forms are difficult to use. The processor is tied up doing the trivial task of counting while in the timing loop, and interrupts should be disabled while using this routine.

Now that you have some of the facts you can decide if it's worth the effort to get that cheap hard copy. Once I got the routine all debugged I wondered why I waited so long to do it. A good fast printer is certainly in my future, but for now I can print summaries, reports and even gummed labels for a mailing list which look as nice as many of those from expensive printers. With a little care in reading, I can list all my programs when they get too large to fit on the screen. My conclusion is: The effort was certainly worthwhile. ■

```
7FØØ  79                     LD    A,C          GET CHARACTER
7FØ1  E6 7F                  AND   A,7FH        LIMIT TO 7F
7FØ3  FE ØD                  CP    A,ØDH        CHK FOR CR
7FØ5  CA AA 7F               JP    Z,OUTCRLF
7FØ8  FE 25                  CP    A,25H        CHK FOR ''
7FØA  28 62                  JR    Z,OUT''
7FØC  FE 2Ø                  CP    A,2ØH        CHK FOR SPACE
7FØE  28 67                  JR    Z,OUTSP
7F1Ø  FE 3C                  CP    A,3CH        CHK FOR LT
7F12  28 5C                  JR    Z,OUTLT
7F14  FE 3D                  CP    A,3DH        CHK FOR EQ
7F16  28 73                  JR    Z,OUTEQ
7F18  FE 3E                  CP    A,3EH        CHK FOR GT
7F1A  28 7A                  JR    Z,OUTGT
7F1C  FE 5B                  CP    A,5BH        CHK FOR !!
7F1E  CA A1 7F               JP    Z,OUT!!
7F21  FD 21 ØØ 7E            LD    IY,7EØØH     INITIALIZE Y
7F25  CD 2B 7F      INIT1    CALL  INIT
7F28  C3 3A 7F               JP    CONVERT
7F2B  F5            INIT     PUSH  AF
7F2C  3A 8Ø 7E               LD    A,COUNTR
7F2F  3C                     INC   A
7F3Ø  FE 49                  CP    A,49H        CHK FOR 73RD CHCTR
7F32  CC AA 7F               CALL  A,OUTCRLF
7F35  32 8Ø 7E               LD    COUNTR,A
7F38  F1                     POP   AF
7F39  C9                     RET
7F3A  32 82 7E      CONVERT  LD    MBOX,A
7F3D  FE 41                  CP    A,41H        CHK FOR SHFT OR UNSHFT
7F3F  3Ø 11                  JR    NC,LTRSET
7F41  FD CB 81 46   FIGSET   BIT   Ø,(IY&FLAG) SHFT TST
7F45  28 1C                  JR    Z,OUTFIG
7F47  DD 2A 82 7E   CLOUT    LD    IX,MBOX      GET OFFSET
7F4B  DD 7E ØØ               LD    A,(IX&TSTRT)GET SUB CHRTR
7F4E  CD BA 7F               CALL  OUTCH
7F51  C9                     RET
7F52  FD CB 81 46   LTRSET   BIT   Ø,(IY&FLAG) SHFT TST
7F56  28 EF                  JR    Z,CLOUT
7F58  FD CB 81 86   OUTLTR   RES   Ø,(IY&FLAG) CLR
7F5C  3E FE                  LD    A,FEH        PRINT LTRS
7F5E  CD BA 7F               CALL  OUTCH
7F61  18 E4                  JR    CLOUT
7F63  FD CB 81 C6   OUTFIG   SET   Ø,(IY&FLAG) SET FLG
7F67  3E F6                  LD    A,F6H        PRINT FIGS
7F69  CD BA 7F               CALL  OUTCH
7F6C  18 D9                  JR    CLOUT
7F6E  3E 25         OUT''    LD    A,25H        PRINT ''
7F7Ø  CD 25 7F               CALL  INIT1
7F73  CD 47 7F               CALL  CLOUT
7F76  C9                     RET
7F77  3E C8         OUTSP    LD    A,C8H        PRINT SPACE
7F79  CD 2B 7F               CALL  INIT
7F7C  CD BA 7F               CALL  OUTCH
7F7F  C9                     RET
7F8Ø  3E 4C         OUTLT    LD    A,4CH        PRINT LT
7F82  CD 25 7F               CALL  INIT1
7F85  3E 54                  LD    A,54H
7F87  CD 25 7F               CALL  INIT1
7F8A  C9                     RET
7F8B  3E 45         OUTEQ    LD    A,45H        PRINT EQ
7F8D  CD 25 7F               CALL  INIT1
7F9Ø  3E 51                  LD    A,51H
7F92  CD 25 7F               CALL  INIT1
7F95  C9                     RET
7F96  3E 47         OUTGT    LD    A,47H        PRINT GT
7F98  CD 25 7F               CALL  INIT1
7F9B  3E 54                  LD    A,54H
7F9D  CD 25 7F               CALL  INIT1
7FAØ  C9                     RET
7FA1  3E 1B         OUT!!    LD    A,5BH        PRINT !!
7FA3  CD 25 7F               CALL  INIT1
7FA6  CD 47 7F               CALL  CLOUT        DO IT AGAIN
7FA9  C9                     RET
7FAA  3E DØ         OUTCRLF  LD    A,DØH        PRINT CRLF
7FAC  CD BA 7F               CALL  OUTCH
7FAF  3E C4                  LD    A,C4H
7FB1  CD BA 7F               CALL  OUTCH
7FB4  3E ØØ                  LD    A,ØØH        CLEAR CHRTR CNTR
7FB6  32 8Ø 7E               LD    COUNTR,A
7FB9  C9                     RET
7FBA  Ø6 Ø7         OUTCH    LD    B,Ø7H        SET FOR 1 STOP BIT
7FBC  F5            HOLD     PUSH  AF           SAVE CHARACTER
7FBD  E6 Ø1                  AND   A,Ø1H        PRESERVE OTHER BITS
7FBF  EE Ø1                  XOR   A,Ø1H
7FC1  D3 FF                  OUT   A,FFH        SEND DATA BIT TO PRTR
7FC3  F1                     POP   AF
7FC4  CB 2F                  SRA   A            SHIFT IN NEXT BIT
7FC6  CD CC 7F               CALL  DELAY
7FC9  1Ø F1                  DJNZ  HOLD
7FCB  C9                     RET
7FCC  11 Ø1 ØØ      DELAY    LD    DE,ØØØ1H
7FCF  2A 84 7E               LD    HL,CYCLES    GET DELAY PERIOD
7FD2  ED 52         LOOP     SBC   HL,DE        DECREMENT BY ONE
7FD4  2Ø FC                  JR    NZ,LOOP
7FD6  C9                     RET                END OF PROGRAM
```

Program Listing 1.

```
1 REM  LPRINT TO BAUDOT TTY ::: WR ::: JUNE 28,198Ø 16K
5 CLS:PRINT"WRITING PROGRAM---PLEASE WAIT"
7 POKE 16422,Ø:POKE 16423,127
1Ø M=32256 'FIRST BYTE OF PROTECTED MEMORY
2Ø READ H$
21 P$=H$
22 H=Ø
25 FOR I=1 TO LEN(H$)
3Ø A$=RIGHT$(H$,1)
4Ø IF A$>="A"ANDA$<="F"THENA=ASC(A$)-55 ELSE A=VAL(A$)
5Ø H=H+A*16[(I-1)
6Ø H$=LEFT$(H$,LEN(H$)-1)
7Ø NEXT
75 PRINTP$;"  ";
8Ø POKE M,H
9Ø M=M+1
1ØØ IF M=3239Ø THEN GOTO 5ØØ
1Ø5 IF M>32726 THEN END
11Ø GOTO 2Ø
5ØØ M=32512:GOTO2Ø
999 REM MEM LOCATION 7EØØ
1ØØØ DATA FF,FF,FF,FF,FF,FF,FF,CA,FF,C4,C4,C4,C4,DØ,FF,FF
1ØØ1 REM 7E1Ø
1ØØ2 DATA FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,DA,FF,FF,FF,FF
1ØØ3 REM 7E2Ø
1ØØ4 DATA C8,DA,E2,E8,D2,D6,F4,D6,DE,E4,DA,F4,D8,C6,F8,FA
1ØØ9 REM 7E3Ø
1Ø1Ø DATA EC,EE,E6,C2,D4,EØ,EA,CE,CC,FØ,DC,FC,FF,FF,FF,F2
1Ø11 REM 7E4Ø
1Ø12 DATA FA,C6,F2,DC,D2,C2,DA,F4,E8,CC,D6,DE,E4,F8,D8,FØ
1Ø13 REM 7E5Ø
1Ø14 DATA EC,EE,D4,CA,EØ,CE,FC,E6,FA,EA,E2,DA,FF,FF,C8,FE
1Ø19 REM 7E6Ø
1Ø2Ø DATA FE,C6,F2,DC,D2,C2,DA,F4,E8,CC,D6,DE,E4,F8,D8,FØ
1Ø21 REM 7E7Ø
1Ø22 DATA EC,EE,D4,CA,EØ,CE,FC,E6,FA,EA,E2,FF,FF,FF,FF,FE
1Ø29 REM 7E8Ø THRU 7E85
1Ø3Ø DATA ØØ,ØØ,ØØ,7E,AØ,Ø5
1Ø39 REM 7FØØ
1Ø4Ø DATA 79,E6,7F,FE,ØD,CA,AA,7F,FE,25,28,62,FE,2Ø,28,67
1Ø41 REM 7F1Ø
1Ø42 DATA FE,3C,28,6C,FE,3D,28,73,FE,3E,28,7A,FE,5B,CA,A1
1Ø43 REM 7F2Ø
1Ø44 DATA 7F,FD,21,ØØ,7E,CD,2B,7F,C3,3A,7F,F5,3A,8Ø,7E,3C
1Ø45 REM 7F3Ø
1Ø46 DATA FE,49,CC,AA,7F,32,8Ø,7E,F1,C9,32,82,7E,FE,41,3Ø
1Ø47 REM 7F4Ø
1Ø48 DATA 11,FD,CB,81,46,28,1C,DD,2A,82,7E,DD,7E,ØØ,CD,BA
1Ø49 REM 7F5Ø
1Ø5Ø DATA 7F,C9,FD,CB,81,46,28,EF,FD,CB,81,86,3E,FE,CD,BA
1Ø51 REM 7F6Ø
1Ø52 DATA 7F,18,E4,FD,CB,81,C6,3E,F6,CD,BA,7F,18,D9,3E,25
1Ø53 REM 7F7Ø
1Ø54 DATA CD,25,7F,CD,47,7F,C9,3E,C8,CD,2B,7F,CD,BA,7F,C9
1Ø55 REM 7F8Ø
1Ø56 DATA 3E,4C,CD,25,7F,3E,54,CD,25,7F,C9,3E,45,CD,25,7F
1Ø57 REM 7F9Ø
1Ø58 DATA 3E,51,CD,25,7F,C9,3E,47,CD,25,7F,3E,54,CD,25,7F
1Ø59 REM 7FAØ
1Ø6Ø DATA C9,3E,1B,CD,25,7F,CD,47,7F,C9,3E,DØ,CD,BA,7F,3E
1Ø61 REM 7FBØ
1Ø62 DATA C4,CD,BA,7F,3E,ØØ,32,8Ø,7E,C9,Ø6,Ø7,F5,E6,Ø1,EE
1Ø63 REM 7FCØ
1Ø64 DATA Ø1,D3,FF,F1,CB,2F,CD,CC,7F,1Ø,F1,C9,11,Ø1,ØØ,2A
1Ø65 REM 7FDØ
1Ø66 DATA 84,7E,ED,52,2Ø,FC,C9
1Ø7Ø END
```

Program Listing 2.

✓See List of Advertisers on page 402

A financial trend monitor for credit unions.

State of the Union

R. M. Reid, Jr.
6519 Velasco
Dallas, TX 75214

Credit union managers and directors spend a lot of time analyzing financial data. Many methods are used to determine the relative health of the organization, and one of the most popular is the calculation of ratios between items appearing on financial statements.

The consistent calculation of a significant number of relationships can be time consuming, especially since the results are most indicative when compared with previous relationships so trends can be detected.

An excellent discussion of some of the most significant indicators is presented in the August 1979 issue of *Credit Union Executive* magazine. The trend-monitoring sheet suggested in that article was adopted by the credit union of which I am a member and resulted in the preparation of this program.

The data for the program is obtained directly from the regular financial statements prepared by the credit union staff. Our credit union uses the account number structure recommended in the credit union accounting manual published by the National Credit Union Administration. I included these account numbers in the input requests so the information would be easier for me to locate and input. A sample financial and statistical report is shown in Fig. 1.

A sample of the output report used by the managers and directors is shown in Fig. 2. The tabular presentation makes it easy to scan the data for each indicator to see if significant trends are developing.

Ratios and Indicators

The ratios and indicators calculated by the program are as follows:

- The rate of return on assets is the total income earned during the period divided by average total assets, then annualized and multiplied by 100 to give a percentage answer.
- The rate of return on loans to members is the total loan interest received during the period divided by average total loans outstanding, annualized and converted to a percentage.
- The expense-to-income ratio represents the total operating expenses to date divided by total operating income to date.
- The percent loans-to-shares are 100 times total loans divided by total shares.
- The maximum allowable dividend rate is one calculation you might want to adjust so it fits the conditions of your credit union. As written, the program calculates 95 percent of the net operating income divided by 94 percent of total shares. This calculation assumes that five percent of net income must be put in reserves (thus the .95 multiplier) and that only 94 percent of total shares will be eligible for dividends. The 94 percent factor results from the way we calculate dividends and is a factor we learn from experience. You may edit program line 650 to include factors more applicable to your situation.
- The loans-to-assets ratio is the total loans outstanding divided by total assets.
- The reserves-to-delinquency ratio is the total reserves divided by the dollar amount of delinquencies.
- The percent reserves-to-loans are 100 times the sum of total reserves plus the allowance for loan losses divided by total loans.
- The percent liquidity reserve, depending on your investment portfolio, might also require some adjustment to fit your particular situation. The program is intended to calculate the product of 100 times cash plus all investments with maturities of less than one year, divided by total shares not pledged as loan collateral. As written, it will include all investments in jumbo CD's, because all our investments of this type have shorter-term maturities.
- The liquidation value of members' shares calculates whether members could anticipate receiving 100 percent of their shares if the credit union were liquidated. It is obtained by subtracting anticipated loan losses from the sum of shares and reserves, then dividing the result by total shares and multiplying the result by 100. Obviously, the indicator should never fall below 100, or the credit union is insolvent.
- The percent delinquencies-to-loans is calculated in dollars (delinquent amount divided by total loans) and number (number of delinquent loans divided by the number of loans outstanding). A comparison of these two measures can yield some information about the relative sizes of the delinquencies.
- The percent of members borrowing is the number of loans outstanding divided by the total number of members—a rough measure of member service.
- The loan turnover ratio is a measure of the rate at which loans turn over per year. It is the annualized quotient of total loans granted year-to-date divided by the average loans outstanding during the period.
- The average loan account balance is the total outstanding loan amount divided by the number of loans.
- The annual membership growth represents the annualized net increase in total membership year-to-date.
- The annual savings growth is the dollar growth in total shares on an annual basis. The

annualized growth indicators are possibly most useful in preparing budgets for future periods.

- The average share balance is the total shares divided by total membership. It is a measure of members' saving habits.
- The average annual savings per member is calculated by annualizing the year-to-date increase in total shares divided by the average membership during the period.
- Shares are determined by dividing the dollar amount in the money market certificates by the amount of total shares. As we began to offer money market share certificates in amounts over $10,000, we were concerned that the percent of total shares earning the higher interest rates might be worth watching.

The Program

The program proceeds with a minimum of jumps and no subroutines. Table 1 contains a list of all numerical variables used. In Program Listing 1, line 100 fulfills the necessary housekeeping duties. Lines 110–170 ask if you are about to make the first entries for a new year. If so, you are asked for the year and some beginning data. Since several of the indicators are calculated on an annualized basis, these previous year-end numbers are necessary.

If the answer to the line 110 question is that you are going to add data to a file, then line 180

FINANCIAL AND STATISTICAL REPORT

For Period Ended ______________ 19 ______ Charter No. ____________________

____

ADDRESS ____
STREET AND NUMBER CITY AND STATE ZIP CODE

STATEMENT OF FINANCIAL CONDITION

ACCT. NO.	ASSETS	END OF THIS PERIOD
701	Loans	
712	Loans Pur. from Liq. CUs (Less Disc.)	
	Total	
719	Less — Allowance for Loan Losses	
	Net	
730	Cash	
740	Investments	
752	Accrued Income on Investments	
760	Prepaid and Deferred Expenses	
774,5	Furniture & Equipment (net)	
790	Other Assets	
	TOTAL ASSETS	
	LIABILITIES AND EQUITY	
801	Accounts Payable	
820	Dividends Payable	
840	Taxes Payable	
860	Other Liabilities	
	TOTAL LIABILITIES	
901	Shares	
931	Regular Reserve	
940	Undivided Earnings	
960	Net Income (Loss)	
	TOTAL EQUITY	
	TOTAL LIAB. & EQUITY	

STATEMENT OF INCOME

ACCT. NO.	OPERATING INCOME	THIS MONTH	YEAR TO DATE
111	Interest on Loans		
113	Income from Loans of Liq. CUs		
	Total		
119	Less — Interest Refund		
	Net		
121	Income from Investments		
131	Fees and Charges		
151	Misc. Operating Income		
	Total Operating Income		
	OPERATING EXPENSES		
210	Compensation		
220	Employee Benefits		
230	Travel & Conference Expenses		
240	Association Dues		
250	Office Occupancy Expenses		
260	Office Operations Expenses		
270	Educ. & Promotional Exp.		
280	Loan Servicing Expenses		
290	Prof. & Outside Services		
300	Provision for Loan Losses		
310	Members' Insurance		
320	Fed. Supvr. & Exam Exp.		
330	Cash Over and Short		
340	Int. on Borrowed Money		
350	Annual Meeting Expense		
360	Misc. Operating Expenses		
	Total Operating Expenses		
	INCOME (Loss) FROM OPER.		
	NON-OPERATING GAINS (Losses)		
420	Gain (Loss) on Investments		
430	Gain (Loss) on Disp. of Assets		
	Total Non-Oper. Gains (Losses)		
	Income (Loss) Before Dividends		
380	Dividends		
	NET INCOME (Loss)		

STATISTICAL REPORT

		NUMBER	AMOUNT
1.	Delinquent Loans		
a.	2 to less than 6 months		
b.	6 to less than 12 months		
c.	12 months and over		
d.	Subtotal		
2.	Current & Less than 2 mos. delq.		
3.	Total Loans		
4.	Loans made:		
a.	Loans made year to date		
b.	Loans made since org.		
5.	No. Members' & Nonmembers' Accounts at End of Period		
6.	No. of potential members		
7.	Loans chgd. off since org.		
8.	Recov. on loans charged off Since Organization		

We certify that, to the best of our knowledge and belief, this report is true and correct and presents fairly the financial position and the results of operations for the periods covered.

Treas. ____________________

Official: ____________________

CU-10056 (May77)

Fig. 1. Typical Financial and Statistical Report.

E-SYSTEMS GARLAND FEDERAL CREDIT UNION
MONTHLY INDICATORS
1980

	JAN	FEB	MAR	APR	MAY	JUNE	JULY	AUG	SEPT	OCT	NOV	DEC
ANNUAL RATE OF RETURN ON TOTAL ASSETS	13.45	11.82	10.47	11.25	11.18	10.48	10.76	10.55	10.37	10.62	0.00	0.00
ANNUAL RATE OF RETURN ON LOANS TO MEMBERS	14.33	12.85	12.15	12.23	12.26	12.09	12.17	12.21	12.10	12.34	0.00	0.00
ACTUAL EXPENSE-INCOME RATIO TO DATE	0.30	0.26	0.27	0.25	0.24	0.27	0.26	0.25	0.27	0.26	0.00	0.00
PERCENT LOANS-TO-SHARES	79.22	78.80	78.44	81.58	78.74	75.97	73.24	69.89	67.36	64.80	0.00	0.00
MAXIMUM ALLOWABLE DIVIDEND RATE YEAR-TO-DATE	10.16	9.40	8.18	9.23	9.32	8.24	8.52	8.33	7.86	8.05	0.00	0.00
LOANS-TO-ASSETS RATIO	0.75	0.74	0.73	0.77	0.74	0.71	0.69	0.65	0.63	0.61	0.00	0.00
RESERVES-TO-DELINQUENCY RATIO	15.85	10.99	20.14	16.12	13.94	9.18	10.63	10.04	18.40	17.25	0.00	0.00
PERCENT RESERVES-TO-LOANS	4.66	4.75	4.66	4.82	4.91	5.07	5.13	5.23	5.15	5.53	0.00	0.00
PERCENT LIQUIDITY RESERVE (5% MIN)	9.17	10.35	11.71	7.12	10.92	14.63	18.22	22.06	25.54	27.25	0.00	0.00
LIQUIDATION VALUE OF MEMBERS' SHARES	103.50	103.51	103.49	103.75	103.68	103.52	103.43	103.34	103.35	103.44	0.00	0.00
PERCENT DELINQUENCIES-TO-LOANS ($)	0.29	0.42	0.23	0.29	0.34	0.53	0.46	0.50	0.28	0.31	0.00	0.00
PERCENT DELINQUENCIES-TO-LOANS (QTY)	0.48	0.62	0.42	0.52	0.61	0.70	0.93	1.03	0.66	0.80	0.00	0.00
PERCENT OF MEMBERS BORROWING	63.33	63.54	63.99	63.74	63.29	63.42	62.91	62.56	61.72	60.38	0.00	0.00
LOAN TURNOVER RATIO (YEARS)	0.72	0.74	0.81	0.85	0.82	0.82	0.84	0.83	0.84	0.85	0.00	0.00
AVERAGE LOAN ACCOUNT BALANCE	2465	2416	2401	2377	2351	2302	2263	2222	2206	2161	0	0
ANNUAL MEMBERSHIP GROWTH	420	276	304	291	225	234	282	268	284	303	0	0
ANNUAL SAVINGS GROWTH ($ X 1000)	−843	−722	−106	−982	−533	−146	163	417	608	615	0	0
AVERAGE SHARE BALANCE	1970	1948	1959	1857	1890	1921	1944	1989	2021	2014	0	0
AVERAGE ANNUAL SAVINGS PER MEMBER	−257.91	−220.68	−32.24	−297.75	−161.64	−44.07	48.99	125.23	181.52	182.40	0.00	0.00
CERTIFICATES/TOTAL SHARES	0.13	0.15	0.18	0.23	0.21	0.21	0.23	0.23	0.22	0.23	0.00	0.00

Fig. 2. Sample Output Report.

obtains all previous data from the disk.

Lines 200-360 ask for data from the balance sheet. You are reminded to make all entries in whole dollars. You also should omit all commas from the dollar entries.

Lines 370-410 ask for data from the income statement, and you are reminded to use year-to-date figures. The year-to-date data is necessary because some of the indicators are annual projections based on the average performance from the beginning of the year through the latest entry.

Lines 420-500 ask for data from the statistical report plus one item—shares pledged as loan collateral—that does not appear on any financial report. This last item is necessary for the computation of percent liquidity reserve. The minimum value for this reserve in federal credit unions is five percent, so the board of directors will want to watch the reserve very closely when it approaches the borderline.

Lines 600-720 contain most of the computations in the program. The only exceptions are some totals and multiple-use factors that were calculated at the time of input. (Refer to lines 220, 300 and 360.) The arithmetic is relatively straightforward.

Lines 730-770 permit you to enter data for as many months as you wish before printing the results. When you have finished entering data, the program proceeds immediately to print the results.

The print routine is contained in lines 900-1250. Lines 900, 910 and 915 are the three major print formats. Line 900 is the format for most indicators that are expressed as a number with two possible decimal places. Line 910 is used for those indicators that are always whole numbers. Line 915 is almost the same as line 900, except it allows four digits prior to the decimal point. This format is necessary for the larger results in the "average annual savings per member" calculation.

Lines 980, 990 and 1000 may

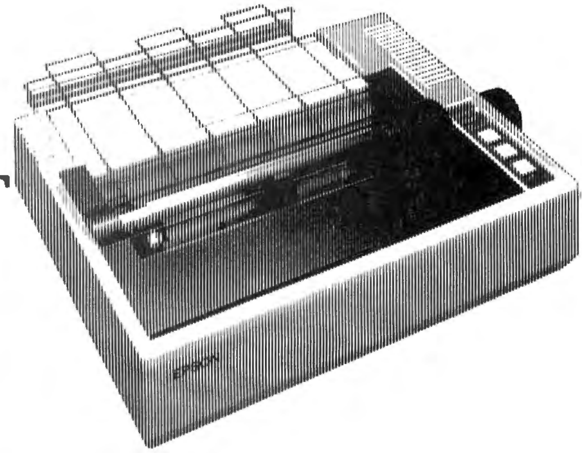
Variable	Definition
A	Total Assets
AA	Average Assets
AL	Average Loans
AS	Average Shares
BA	Beginning Assets
BL	Beginning Loans
BM	Beginning Number of Members
BS	Beginning Shares
C	Cash
CT	Common Trust Investments
CU	Deposits and Shares in Other Credit Unions
DA	Delinquent Loan Amount
E	Annualizer
F	Annualizer times 100
FA	Federal Agency Securities
I	Interest on Loans
L	Loans Outstanding
LA	Loan Amount, Year-to-Date
LL	Allowance for Loan Losses
MM	Money Market Certificate Investments
N	Month Number
ND	Number of Delinquent Loans
NM	Number of Members
OE	Total Operating Expense
OI	Total Operating Income
PS	Pledged Shares
RC	Reserve for Contingency
RR	Regular Reserve
S	Regular Shares
SC	Money Market Share Certificates
ST	Investments With Maturities Less Than One Year
TI	Total Investments
TL	Total Number of Loans
TR	Total Reserves
TS	Total Shares
X	Counter
Y	Year
Z	Counter

Table 1. Definition of Variables.

EPSON

MX-80

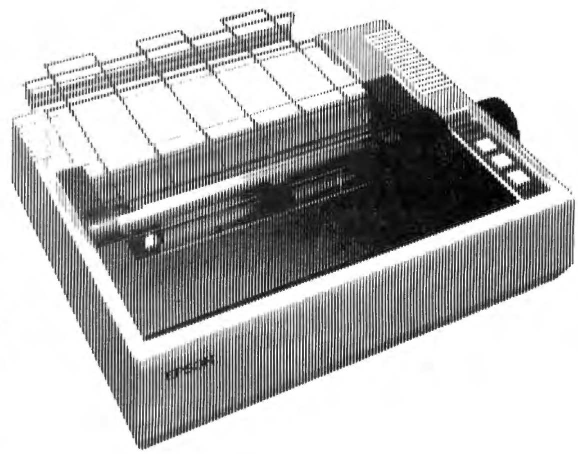
MX-80
MX-80 FT
MX-100



EPSON

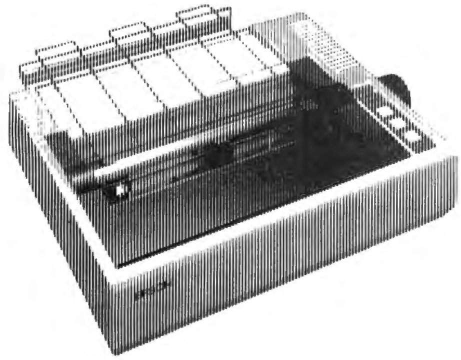
MX-70

EPSON
EPSON
EPSON



EPSON
EPSON

MX-100
MX-80 FT



AMERICAN Small Business COMPUTERS
118 S. MILL ST. PRYOR, OK. 74361

AMERICAN SMALL BUSINESS COMPUTERS STOCKS THE ENTIRE LINE OF EPSON PRODUCTS AND ACCESSORIES

CALL FOR OUR PRICING !

WE PROMISE IT WILL BE WORTH THE CALL !!

529

918-825-4844

RETAIL STORE LOCATION — SOUTHWOOD SHOPPING CENTER — PRYOR, OK.

require modification for your particular printer. I use an IDS-440 Paper Tiger and some of my commands might not be necessary with other printers. The command LPRINTCHR$(31) in line 980 is a control command that adjusts the line length to 132 characters. The command LPRINTCHR$(01) in the same line switches the printer to an enhanced printing mode so the report heading will appear to be in larger and bolder type. The command LPRINTCHR$(02) in line 1000 returns the printer to normal characters.

You would, of course, want to change line 980 so the name of your organization is printed at the top of your reports. You also probably will wish to revise the tabs in lines 980, 990 and 1000 to adjust the heading. Lines 1300-1320 record all your latest data on the disk, and line 1400 ends the program.

I hope you find this program as useful to your credit union as it has been to mine. I would be interested in hearing your comments or questions. ■

Program Listing

```
10 '***************************************************************
15 '*                                                             *
20 '*                    MONTHLY INDICATORS                       *
25 '*                            OF                               *
30 '*                CREDIT UNION PERFORMANCE                     *
35 '*                                                             *
40 '*                            BY                               *
45 '*                       MARK  REID                            *
50 '*                      6519 VELASCO                           *
55 '*                    DALLAS, TX 75214                         *
60 '*                                                             *
65 '***************************************************************
100 CLEAR900:DIMP(12,20):DEFINTX,Y,Z
110 CLS:PRINT"IS THIS THE BEGINNING OF A NEW YEAR (Y/N)?
115 Z$=INKEY$:IFZ$=""THEN115
120 IFZ$="N"THEN180
125 IFZ$="Y"THEN130
127 REM
128 REM * * * * * INPUT FIRST-OF-YEAR DATA * * * * *
129 REM
130 CLS:PRINT"MAKE ALL FINANCIAL ENTRIES IN WHOLE DOLLARS":PRINT
:INPUT"WHAT YEAR ARE YOU BEGINNING";Y
140 INPUT"BEGINNING LOANS";BL
150 INPUT"BEGINNING ASSETS";BA
160 INPUT"BEGINNING SHARES";BS
170 INPUT"BEGINNING NUMBER OF MEMBERS";BM:GOTO200
177 REM
178 REM * * * * * READ DATA FROM DISK * * * * *
179 REM
180 OPEN"I",1,"CU/DAT":INPUT#1,Y,BL,BA,BS,BM:FORX=1TO12:FORZ=1TO
20:INPUT#1,P(X,Z):NEXTZ:NEXTX:CLOSE1
200 CLS:PRINT"REMEMBER TO MAKE ALL FINANCIAL ENTRIES IN WHOLE DO
LLARS":PRINT
207 REM
208 REM * * * * * INPUT BALANCE SHEET DATA * * * * *
209 REM
210 PRINT"INPUT THE FOLLOWING DATA FROM YOUR BALANCE SHEET"
220 INPUT"MONTH NO.";N:E=12/N:F=1200/N
230 INPUT"LOANS OUTSTANDING, ACCT 701";L
240 INPUT"ALLOWANCE FOR LOAN LOSSES, ACCT 719";LL
250 INPUT"CASH, ACCT 730";C
260 INPUT"FEDERAL AGENCY SECURITIES, ACCT 742";FA
270 INPUT"COMMON TRUST INVESTMENTS, ACCT 743";CT
280 INPUT"SHARES, DEPOSITS, CERTS - OTHER CU'S, ACCT 745";CU
290 INPUT"OTHER SHARES, DEPOSITS CERTIFICATES, ACCT 746";MM
300 TI=FA+CT+CU+MM:ST=CT+CU+MM
310 INPUT"TOTAL ASSETS";A
320 INPUT"REGULAR SHARES, ACCT 901";S
330 INPUT"MM SHARE CERTIFICATES, ACCT 908";SC
340 INPUT"REGULAR RESERVES, ACCT 931";RR
350 INPUT"RESERVE FOR CONTINGENCIES, ACCT 933";RC
360 TR=RR+RC
367 REM
368 REM * * * * * INPUT INCOME STATEMENT DATA * * * * *
369 REM
370 CLS:PRINT"INPUT THE FOLLOWING DATA FROM YOUR INCOME STATEMEN
T"
380 PRINT"BE SURE TO USE YEAR-TO-DATE FIGURES":PRINT
390 INPUT"INTEREST ON LOANS, ACCT 111";I
400 INPUT"TOTAL OPERATING INCOME";OI
410 INPUT"TOTAL OPERATING EXPENSES";OE
417 REM
418 REM * * * * * INPUT STATISTICAL DATA * * * * *
419 REM
420 CLS:PRINT"INPUT THE FOLLOWING DATA FROM YOUR"
430 PRINT"STATISTICAL REPORT":PRINT
440 INPUT"NUMBER OF DELINQUENT LOANS";ND
450 INPUT"DELINQUENT AMOUNT";DA
460 INPUT"TOTAL NUMBER OF LOANS";TL
470 INPUT"LOAN AMOUNT, YEAR TO DATE";LA
480 INPUT"NUMBER OF MEMBERS";NM
490 PRINT"YOU WILL PROBABLY HAVE TO SEE YOUR BOOKKEEPER FOR"
500 INPUT"SHARES PLEDGED AS LOAN COLLATERAL";PS
597 REM
598 REM* * * * * COMPUTATIONS * * * * *
599 REM
600 TS=S+SC:AS=(BS+TS)/2:AL=(BL+L)/2:AA=(BA+A)/2
610 P(N,0)=(OI/AA)*F:P(N,1)=(INT(P(N,0)*100+.5))/100
620 P(N,0)=(I/AL)*F:P(N,2)=(INT(P(N,0)*100+.5))/100
630 P(N,0)=OE/OI:P(N,3)=(INT(P(N,0)*100+.5))/100
640 P(N,0)=(L/TS)*100:P(N,4)=(INT(P(N,0)*100+.5))/100
650 P(N,0)=.95*F*(OI-OE)/(.94*TS):P(N,5)=(INT(P(N,0)*100+.5))/10
0
```

Program continues

Program continued

```
660 P(N,Ø)=L/A:P(N,6)=(INT(P(N,Ø)*1ØØ+.5))/1ØØ
670 P(N,Ø)=TR/DA:P(N,7)=(INT(P(N,Ø)*1ØØ+.5))/1ØØ
680 P(N,Ø)=1ØØ*(TR+LL)/L:P(N,8)=(INT(P(N,Ø)*1ØØ+.5))/1ØØ
690 P(N,Ø)=1ØØ*(C+ST)/(TS-PS):P(N,9)=(INT(P(N,Ø)*1ØØ+.5))/1ØØ
700 P(N,Ø)=1ØØ*(TS+TR-LL)/TS:P(N,1Ø)=(INT(P(N,Ø)*1ØØ+.5))/1ØØ
710 P(N,Ø)=1ØØ*DA/L:P(N,11)=(INT(P(N,Ø)*1ØØ+.5))/1ØØ
712 P(N,Ø)=1ØØ*ND/TL:P(N,12)=(INT(P(N,Ø)*1ØØ+.5))/1ØØ
713 P(N,Ø)=1ØØ*TL/NM:P(N,13)=(INT(P(N,Ø)*1ØØ+.5))/1ØØ
714 P(N,Ø)=(LA/AL)*E:P(N,14)=(INT(P(N,Ø)*1ØØ+.5))/1ØØ
715 P(N,Ø)=L/TL:P(N,15)=INT(P(N,Ø))
716 P(N,Ø)=(NM-BM)*E:P(N,16)=INT(P(N,Ø))
717 P(N,Ø)=(TS-BS)*E:P(N,17)=INT(P(N,Ø)/1ØØØ)
718 P(N,Ø)=TS/NM:P(N,18)=INT(P(N,Ø))
719 P(N,Ø)=E*(TS-BS)/((BM+NM)/2):P(N,19)=(INT(P(N,Ø)*1ØØ+.5))/1Ø
Ø
720 P(N,Ø)=SC/TS:P(N,2Ø)=(INT(P(N,Ø)*1ØØ+.5))/1ØØ
730 PRINT"WOULD YOU LIKE TO INPUT DATA FOR"
740 PRINT"ANOTHER MONTH AT THIS TIME?"
750 Z$=INKEY$:IFZ$=""THEN75Ø
760 IFZ$="N"THEN9ØØ
770 IFZ$="Y"THEN2ØØ
897 REM
898 REM * * * * * OUTPUT FORMATS * * * * *
899 REM
900 P1$="%                              %    ###.##  ###.##  #
##.##  ###.##  ###.##  ###.##  ###.##  ###.##  ###.##  ###.##  #
##.##  ###.##"
910 P2$="%                              %    ######  ######  #
#####  ######  ######  ######  ######  ######  ######  ######  #
#####  ######"
915 P3$="%                              %   ####.## ####.## ##
##.## ####.## ####.## ####.## ####.## ####.## ####.## ####.## ##
##.## ####.##"
977 REM
978 REM * * * * * FORM HEADING * * * * *
979 REM
980 LPRINTCHR$(31):LPRINTCHR$(Ø1):LPRINTTAB(18)"E-SYSTEMS GARLAN
D FEDERAL CREDIT UNION
990 LPRINTTAB(28)"MONTHLY INDICATORS
1000 LPRINTTAB(34)Y:LPRINTCHR$(Ø2)
1010 LPRINT"                                            JAN       FEB
   MAR      APR      MAY      JUNE      JULY      AUG      SEPT      OCT
    NOV      DEC"
1020 LPRINTSTRING$(131,"-")
1027 REM
1028 REM * * * * * PRINT ROUTINE * * * * *
1029 REM
1030 LPRINTUSINGP1$;"ANNUAL RATE OF RETURN ON";P(1,1);P(2,1);P(3
,1);P(4,1);P(5,1);P(6,1);P(7,1);P(8,1);P(9,1);P(1Ø,1);P(11,1);P(
12,1)
1040 LPRINT"     TOTAL ASSETS"
1050 LPRINTUSINGP1$;"ANNUAL RATE OF RETURN ON";P(1,2);P(2,2);P(3
,2);P(4,2);P(5,2);P(6,2);P(7,2);P(8,2);P(9,2);P(1Ø,2);P(11,2);P(
12,2)
1060 LPRINT"     LOANS TO MEMBERS"
1070 LPRINTUSINGP1$;"ACTUAL EXPENSE/INCOME RATIO";P(1,3);P(2,3);
P(3,3);P(4,3);P(5,3);P(6,3);P(7,3);P(8,3);P(9,3);P(1Ø,3);P(11,3)
;P(12,3):LPRINT"     TO DATE"
1080 LPRINTUSINGP1$;"PERCENT LOANS TO SHARES";P(1,4);P(2,4);P(3,
4);P(4,4);P(5,4);P(6,4);P(7,4);P(8,4);P(9,4);P(1Ø,4);P(11,4);P(1
2,4):LPRINT
1090 LPRINTUSINGP1$;"MAXIMUM ALLOWABLE DIVIDEND RATE";P(1,5);P(2
,5);P(3,5);P(4,5);P(5,5);P(6,5);P(7,5);P(8,5);P(9,5);P(1Ø,5);P(1
1,5);P(12,5)
1100 LPRINT"     YEAR TO DATE"
1110 LPRINTUSINGP1$;"LOANS TO ASSETS RATIO";P(1,6);P(2,6);P(3,6)
;P(4,6);P(5,6);P(6,6);P(7,6);P(8,6);P(9,6);P(1Ø,6);P(11,6);P(12,
6):LPRINT
1120 LPRINTUSINGP1$;"RESERVES TO DELINQUENCY RATIO";P(1,7);P(2,7
);P(3,7);P(4,7);P(5,7);P(6,7);P(7,7);P(8,7);P(9,7);P(1Ø,7);P(11,
7);P(12,7):LPRINT
1130 LPRINTUSINGP1$;"PERCENT RESERVES TO LOANS";P(1,8);P(2,8);P(
3,8);P(4,8);P(5,8);P(6,8);P(7,8);P(8,8);P(9,8);P(1Ø,8);P(11,8);P
(12,8):LPRINT
1140 LPRINTUSINGP1$;"PERCENT LIQUIDITY RESERVE (5% MIN)";P(1,9);
P(2,9);P(3,9);P(4,9);P(5,9);P(6,9);P(7,9);P(8,9);P(9,9);P(1Ø,9);
P(11,9);P(12,9):LPRINT
1150 LPRINTUSINGP1$;"LIQUIDATION VALUE OF MEMBERS'";P(1,1Ø);P(2,
1Ø);P(3,1Ø);P(4,1Ø);P(5,1Ø);P(6,1Ø);P(7,1Ø);P(8,1Ø);P(9,1Ø);P(1Ø
,1Ø);P(11,1Ø);P(12,1Ø):LPRINT"     SHARES"
1160 LPRINTUSINGP1$;"PERCENT DELINQUENCIES TO LOANS ($)";P(1,11)
;P(2,11);P(3,11);P(4,11);P(5,11);P(6,11);P(7,11);P(8,11);P(9,11)
;P(1Ø,11);P(11,11);P(12,11):LPRINT
1170 LPRINTUSINGP1$;"PERCENT DELINQUENCIES TO LOANS";P(1,12);P(2
,12);P(3,12);P(4,12);P(5,12);P(6,12);P(7,12);P(8,12);P(9,12);P(1
Ø,12);P(11,12);P(12,12):LPRINT"     (QTY)"
1180 LPRINTUSINGP1$;"PERCENT OF MEMBERS BORROWING";P(1,13);P(2,1
3);P(3,13);P(4,13);P(5,13);P(6,13);P(7,13);P(8,13);P(9,13);P(1Ø,
13);P(11,13);P(12,13):LPRINT
1190 LPRINTUSINGP1$;"LOAN TURNOVER RATIO (YEARS)";P(1,14);P(2,14
);P(3,14);P(4,14);P(5,14);P(6,14);P(7,14);P(8,14);P(9,14);P(1Ø,1
4);P(11,14);P(12,14):LPRINT
1200 LPRINTUSINGP2$;"AVERAGE LOAN ACCOUNT BALANCE";P(1,15);P(2,1
5);P(3,15);P(4,15);P(5,15);P(6,15);P(7,15);P(8,15);P(9,15);P(1Ø,
15);P(11,15);P(12,15):LPRINT
1210 LPRINTUSINGP2$;"ANNUAL MEMBERSHIP GROWTH";P(1,16);P(2,16);P
(3,16);P(4,16);P(5,16);P(6,16);P(7,16);P(8,16);P(9,16);P(1Ø,16);
P(11,16);P(12,16):LPRINT
1220 LPRINTUSINGP2$;"ANNUAL SAVINGS GROWTH ($ X 1ØØØ)";P(1,17);P
(2,17);P(3,17);P(4,17);P(5,17);P(6,17);P(7,17);P(8,17);P(9,17);P
(1Ø,17);P(11,17);P(12,17):LPRINT
1230 LPRINTUSINGP2$;"AVERAGE SHARE BALANCE";P(1,18);P(2,18);P(3,
18);P(4,18);P(5,18);P(6,18);P(7,18);P(8,18);P(9,18);P(1Ø,18);P(1
1,18);P(12,18):LPRINT
1240 LPRINTUSINGP3$;"AVERAGE ANNUAL SAVINGS PER MEMBER";P(1,19);
P(2,19);P(3,19);P(4,19);P(5,19);P(6,19);P(7,19);P(8,19);P(9,19);
P(1Ø,19);P(11,19);P(12,19):LPRINT
1250 LPRINTUSINGP1$;"CERTIFICATES/TOTAL SHARES";P(1,2Ø);P(2,2Ø);
P(3,2Ø);P(4,2Ø);P(5,2Ø);P(6,2Ø);P(7,2Ø);P(8,2Ø);P(9,2Ø);P(1Ø,2Ø)
;P(11,2Ø);P(12,2Ø)
1297 REM
1298 REM * * * * * SAVE DATA ON DISK * * * * *
1299 REM
1300 OPEN"O",1,"CU/DAT":PRINT#1,Y,BL,BA,BS,BM
1310 FORX=1TO12:FORZ=1TO2Ø
1320 PRINT#1,P(X,Z);",";:NEXTZ:NEXTX:CLOSE1
1400 END
```

Outmaneuver a friend with your Color Computer.

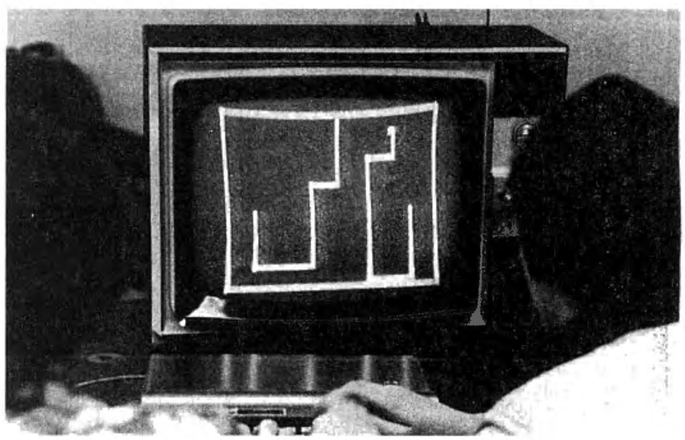
Colorful Maneuvers

James W. Wood
424 N. Missouri, Box 507
Atwood, IL 61913

Tired of running games that involve no decisions or games in which you learn to beat the program? This game uses the computer to generate the playing field and rapidly lengthens two player's lines till one runs into something. You cannot predict your opponent's strategy. Will he attack and try to surround you, or will he just try to stay in the open with lots of running room?

I bought one version of this game in a book, but it didn't work. One player could permanently block the other by holding down a key. In my program if pushing a letter doesn't change your direction, immediately release and push again. The second push will work. Many hours of study by students in my computer class have shown that the program is functional. ■

Contact the author for information on purchasing the program.



G$	player on left
H$	player on right
S	determines initial direction
W$	right players input
R$	left players input
Q$	determines direction
U	right players starting X coordinate
V	right players starting Y coordinate
X	left players starting X coordinate
Y	left players starting Y coordinate
Z	color of line
N	directs program from left to right player
A	number of games left player won
B	number of games right player won
Q	used in drawing border

Table 1. Variable List.

1-5	determine starting direction
8	determines starting position
9	draws border
10,13,101	makes it impossible for one player to permanently block keyboard
20-50 and 120-150	turns line
52,152	determines if game over
300-410	shows where game lost
1000-1170	instructions

Table 2. Line Description.

Program Listing 1. 4K Color Basic.

```
0 CLS:INPUT"PLAYER ON LEFT ";G$:INPUT"PLAYER ON RIGHT ";H$:GOSUB
1000
1 CLS:S=RND(4)
2 IFS=1THENW$="J":R$="A":GOTO8
3 IFS=2THENW$="L":R$="D":GOTO8
4 IFS=3THENW$="I":R$="E":GOTO8
5 W$="N":R$="C"
8 U=8:V=15:X=55:Y=15:N=1
9 CLS0:Z=RND(8):FORQ=0TO63:SET(Q,0,Z):SET(Q,31,Z):NEXTQ:FORQ=0TO
31:SET(0,Q,Z):SET(63,Q,Z):NEXTQ
10 IFN=1THEN12ELSE100
12 Q$=INKEY$:IFQ$=""THENQ$=W$
13 IFASC(Q$)<70THEN102
15 W$=Q$
20 IFQ$="J"THENX=X-1:GOTO52
30 IFQ$="L"THENX=X+1:GOTO52
40 IFQ$="I"THENY=Y-1:GOTO52
50 IFQ$="N"THENY=Y+1
52 IFPOINT(X,Y)<>0THEN300
```

Program continues

✓See List of Advertisers on page 402

Program continued

```
60 SET(X,Y,Z):N=2:GOTO10
100 Q$=INKEY$:IFQ$=""THENQ$=R$
101 IFASC(Q$)>70THEN15
102 R$=Q$
120 IFQ$="A"THENU=U-1:GOTO152
130 IFQ$="D"THENU=U+1:GOTO152
140 IFQ$="E"THENV=V-1:GOTO152
150 IFQ$="C"THENV=V+1
152 IFPOINT(U,V)<>0THEN400
160 SET(U,V,Z):N=1:GOTO10
300 SOUND200,30:FORE=1TO300:SET(X,Y,Z):RESET(X,Y):NEXTE
310 A=A+1:CLS:PRINTG$;" WINS":GOTO500
400 SOUND100,30:FORE=1TO300:SET(U,V,Z):RESET(U,V):NEXTE
410 B=B+1:CLS:PRINTH$;" WINS"
500 PRINT"TOTAL":PRINTG$;" WON";A:PRINTH$;" WON";B:FORQ=1TO600:N
EXTQ:GOTO1
1000 CLS:PRINT"HOW TO TURN YOUR LINE"
1010 PRINT"PRESS THESE LETTERS FOR THE     DIRECTION TO GO"
1106 PRINT
1110 PRINTG$;"          ";H$
1120 PRINT"    E              I"
1130 PRINT"  A  D           J  L"
1140 PRINT"    C              N"
1142 PRINT"         UP"
1143 PRINT"     LEFT   RIGHT"
1144 PRINT"        DOWN"
1150 PRINT:PRINT"PRESS ANY KEY TO CONTINUE"
1160 Y$=INKEY$:IFY$=""THEN1160
1170 RETURN
```

```
8 U=15:V=24:X=113:Y=24:N=1
9 CLS:FOR Q=0 TO
127:SET(Q,0):SET(Q,47):NEXTQ:FORQ=0TO47:SET(0,Q):SET(127,Q):NEXTQ
60 SET(X,Y):N=2:GOTO10
160 SET(U,V)):N=1;GOTO10
300 FOR E=1TO300:SET(X,Y):RESET(X,Y):NEXTE
400 FOR E=1TO300:SET(U,V):RESET(U,V):NEXTE
```

Program Listing 2. Listing Change for Model I and III.

AUTO-DIAL/ AUTO-ANSWER

MODEM FOR $224.95!*

The MICROCONNECTION™ can dial and answer the telephone, run printers, decode busses, transcribe transmissions, answer/originate and direct-connect to the telephone system. Lots of terminal and host software too. For more information write or phone:

the microperipheral corporation
2643 - 151st Pl. N.E.
Redmond, WA 98052
(206) 881-7544

master charge — VISA

*RS232 MICROCONNECTION™

✓306

SCRIPLUS

Scriplus is a modification to Scripsit(c) which enables you to take advantage of the special functions, features, and print formats of your printer while your document is being printed. Allows you to:

change **expanded** print
change no. of characters per inch
or <u>underline</u> in mid-line!

All in accordance with the capabilities of your printer. You can change your print size at will! Features:

1) The user can send commands to the printer to activate special formats
2) Scriplus will not crash programs protected in high memory.
3) "END" returns to DOS READY instead of re-booting.
4) The initial line-feed is changed to a carriage return to empty the text buffer.
5) The user can get an ALPHABETIZED directory from within scriplus.
6) Optionally select automatic line feed after carriage return.
7) Supports custom printer drivers (not included)
8) Modifies Scripsit/LC or /UC. (MOD I)
9) Works with MOD I and MOD III! (Including MOD III 3.1 Scripsit!)
10) Specifically written for the MX-80, but will work on ANY Printer that accepts CHR$ codes for control.

Tape or disk — $24.95
(specify)

— DEALER INQUIRIES INVITED —

QUALITY SOFTWARE DISTRIBUTORS

11500 STEMMONS EXPRESSWAY, SUITE 104
DALLAS, TEXAS 75229
PHONE (214) 484-2976
MICRONET 70130,203 ✓27

TRS-80 & Scripsit are Trademarks of Tandy Corp.

Price Does Not Include Postage

VISA

RELIABLE, SIMPLE MEMORY EXPANSION YOU CAN AFFORD!

Shown above is the **IM-2 48K INTERNAL MEMORY**™ and a phillips screwdriver — everything you need to **ADD** up to **32K RAM** to your **TRS-80*** Model **I** keyboard unit (for a total of **48K RAM**). No soldering or trace cutting — just open the case, remove your **RAM** chips, plug the **INTERNAL MEMORY (IM)** into the **RAM** sockets, plug your chips back in, connect 4 solderless clips (they work!) and close the case (Detailed, illustrated step-by-step instructions show you how to do this). You now have a 48K keyboard unit — and you didn't have to buy an expansion interface! If you don't need 48K, a 32K version is available.

Time and experience have shown that the **TRS-80*** expansion buss cannot reliably drive memory systems. Because the **IM** does not use the buss, it may be used with **ANY** expansion device for reliable memory operation. Some of our customers use an **IM INSTEAD OF** the **RAM** in their expansion interface to eliminate **EI** memory problems!

Because the **IM** draws most of its power from the 12 volt supply, power supply loading is not a problem. Some additional heat is generated, but the **TRS-80*** runs so hot anyway you probably won't notice the increase!

The **INTERNAL MEMORY** is completely tested and assembled using only the finest materials. You have 15 days to return it for a refund if you don't like it, and it is guaranteed for 1 full year. Installation is available for a small fee.

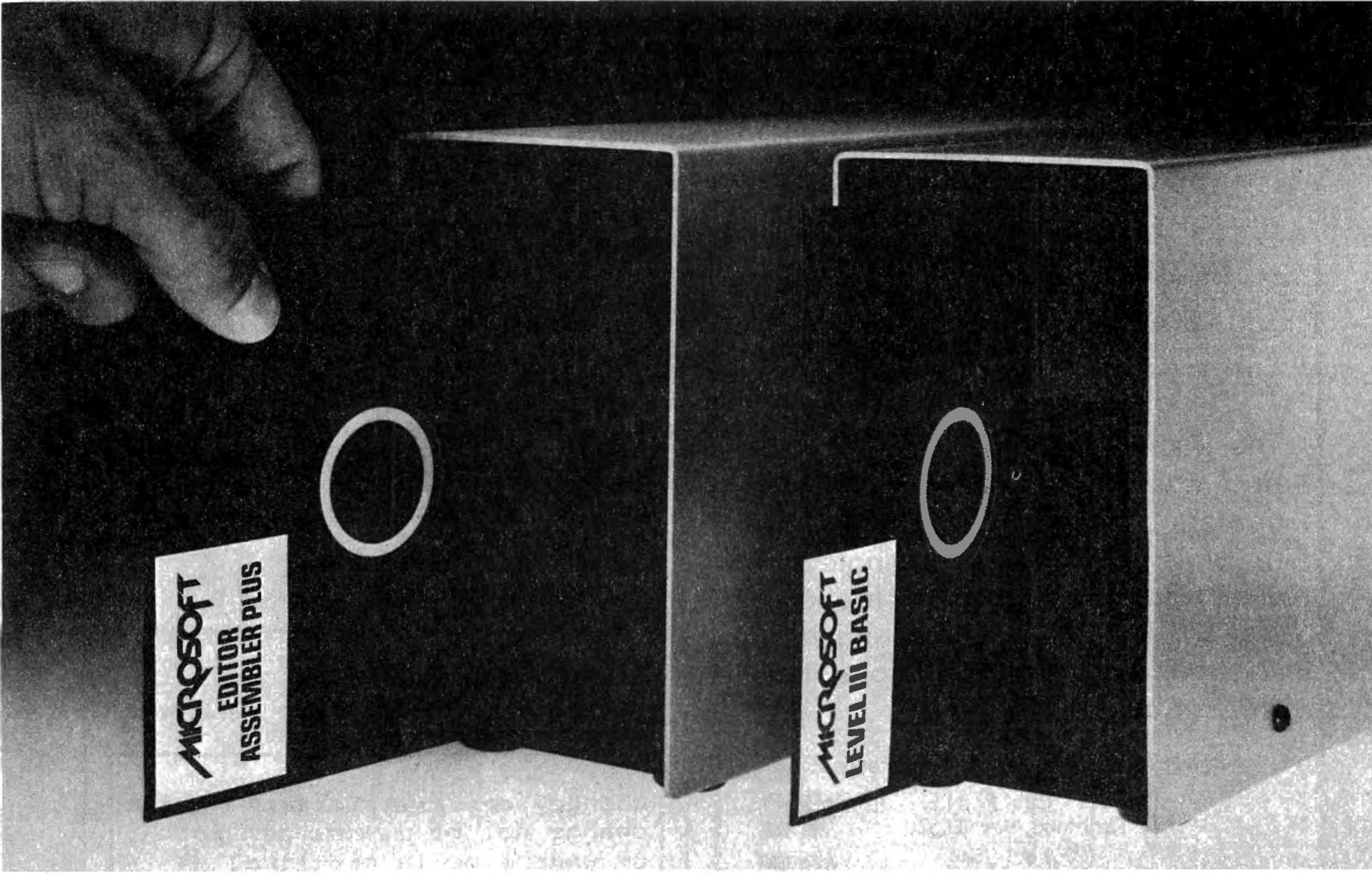
Send a **S.A.S.E.** for more information.

PRICES — IM-1B (32K) $85.50 (w/16K RAM)
IM-2 (48K) $140.00 (w/32K RAM)

Prices include **UPS** shipping in U.S.
Utah residents add 5% tax; FOREIGN ORDERS add 15%.

master charge — *Holmes Engineering* ✓401 — VISA

6246 West 3705 South Salt Lake City, UT 84120 (801) 967-2324



More powerful programming tools for the TRS-80.™ Now on disk.

Better results. Microsoft's Level III BASIC and Editor/Assembler-Plus are programming tools that help you write complex programs in less time, with less effort and utilizing less memory. Better programs. No matter what your programming skill. And for the first time these tools are available on disk.

Editor/Assembler-Plus. A powerful editing, assembly and debugging tool with many sophisticated features that make writing TRS-80 assembly language programs easier, faster and more efficient.

- Full disk capabilities.
- Assembly directly into memory. No need to save object code then reload for execution.
- Macro capability that allows you to define macros for commonly used sequences of instructions.
- Conditional assembly that allows you to generate more than one version of a program.
- Eight breakpoints at a time for program debugging.
- The INCLUDE statement that allows you to call additional disk files for assembly.
- Other features include extensive operators, automatic origin, symbol table printout, quash command, hex, decimal and octal constants, single step-through instructions in memory, five type-out modes, four type-in radices, plus extensive edit commands.

A more powerful BASIC. With Level III BASIC, you get power to perform tasks in BASIC that used to require assembly language. Plus, new ease-of-use features for your TRS-80.

- Advanced graphics. Develop charts, graphs, even animation in Level III BASIC. Draw a line, an outline box or a solid box by specifying just two points. Then save and recall it with BASIC commands.
- MENU. One command that allows you to construct an entire menu.
- CHAIN and COMMON commands allow you to call another program and pass variables to it.
- Powerful editing commands such as COPY/TRANSFER, FIND and CHANGE.
- DUMP command that makes debugging easier.
- Time-limit response. New INPUT # LEN and LINE INPUT # LEN commands allow you to set a time limit on response.
- RS-232 output from BASIC. With a single command.
- More. Level III gives you automatic line numbering, 26 user-definable single stroke instructions, and more.

Disk or cassette. Disk versions of Level III and Editor/Assembler-Plus are brand new. Cassette versions are also available with many of the same capabilities.

Talk to your Microsoft™ dealer. Ask for a demonstration of two of the most powerful tools you can get for your TRS-80: Level III BASIC and Editor/Assembler-Plus. On disk or cassette. From Microsoft.

579

MICROSOFT
CONSUMER PRODUCTS

400 108th Ave. N.E., Bellevue, WA 98004. (206) 454-1315

For readable Basic programs.

Superlist

Morris Jones
3395 Balboa St., Apt. 8
San Francisco, CA 94121

Basic, the world's most popular computer language, is designed to be easy to read and learn. But have you tried reading a fairly complex Level II program lately? To a real beginner, it might as well be machine language. At least with assembly code there is only one instruction per line.

Much can be learned about programming by reading successful programs. Unfortunately, the advantages provided by modern Basic interpreters for microprocessors are disadvantages to the reader of Basic programs. An efficient program that takes advantage of Level II's best features is nearly impossible to read. Micros are limited in memory available, so a compact Level II program is written without spaces between words, using as many statements on a line as possible. Each space character uses one byte; line numbers use five bytes each (that goes for line 65000 as well as for line one).

I wrote my first Basic programs without the aid of a printer. I soon found myself wanting some kind of permanent record of my programs that wasn't susceptible to stray magnetism. A friend kindly offered to provide printouts in exchange for copies of my best programs. He had a disk system that would take a Basic program stored in an ASCII file and print it with each statement on a separate line. From that came the inspiration for a listing program that did not require Disk Basic.

Without a disk, the problem is more complex. Basic does not store its programs in ASCII, but in a very compact coded version using one byte "tokens" for Basic keywords. After obtaining my own printer, I quickly set about writing a program to give me readable listings using what was available in Level II.

```
                00100 ;  HERE IS A SUGGESTED PRINT ROUTINE TO USE WITH
                00110 ;  SUPERLIST FOR PAGED OUTPUT. THIS ROUTINE IS FOR
                00120 ;  THE RADIO SHACK PARALLEL PRINTER INTERFACE, AND IS SIMILAR
                00130 ;  TO THE ROUTINE IN LEVEL II ROM. THIS ROUTINE WILL
                00140 ;  GIVE YOU TWELVE BLANK LINES BETWEEN EACH PAGE OF
                00150 ;  54 LINES.
                00160 ;
                00170 ;  SET BUFF EQU 7DFEH IN MAIN SOURCE FILE FOR SUPERLIST
                00180 ;  OR INSERT THIS CODE FOR PRINTER ROUTINE.
                00190 ;
7DB4            00200         ORG     7DB4H
7DB4 79         00210 PRINT3  LD      A,C             ;CHAR STORED IN C
7DB5 FE0C       00220         CP      12              ;FORM FEED?
7DB7 2018       00230         JR      NZ,PR1          ;NO, PRINT IT
7DB9 DD7E03     00240         LD      A,(IX+3)        ;GET LINES PER PAGE
7DBC DD9604     00250         SUB     (IX+4)          ;SUBTRACT CURRENT COUNT
7DBF 3E0C       00260 PR3     LD      A,12            ;ADD TWELVE LINES
7DC1 47         00270         LD      B,A             ;SAVE IN COUNTER
7DC2 CDEC7D     00280 PR2     CALL    READY           ;WAIT FOR PRINTER
7DC5 3E0A       00290         LD      A,10            ;LOAD A LINE FEED
7DC7 32E837     00300         LD      (37E8H),A       ;SEND IT
7DCA 10F6       00310         DJNZ    PR2             ;UNTIL THROUGH
7DCC DD360400   00320         LD      (IX+4),0        ;ZERO COUNT
7DD0 C9         00330         RET                     ;THROUGH
                00340 ;
7DD1 CDEC7D     00350 PR1     CALL    READY           ;HERE IF NOT FF
7DD4 79         00360         LD      A,C             ;GET BACK CHARACTER
7DD5 32E837     00370         LD      (37E8H),A       ;SEND IT
7DD8 FE0D       00380         CP      13              ;IS IT CR?
7DDA C0         00390         RET     NZ              ;NO, QUIT
7DDB DD3404     00400         INC     (IX+4)          ;BUMP LINE COUNT
7DDE DD7E04     00410         LD      A,(IX+4)        ;GET LINE COUNT
7DE1 DDBE03     00420         CP      (IX+3)          ;PAGE FULL?
7DE4 79         00425         LD      A,C             ;RESTORE CHAR
7DE5 C0         00430         RET     NZ              ;NO, QUIT
7DE6 AF         00440         XOR     A               ;YES, ZERO A
7DE7 CDBF7D     00450         CALL    PR3             ;SEND TWELVE LINES
7DEA 79         00454         LD      A,C             ;RESTORE CHAR
7DEB C9         00458         RET                     ;& QUIT
                00460 ;
7DEC 3AE837     00470 READY   LD      A,(37E8H)       ;GET STATUS
7DEF E6F0       00480         AND     0F0H            ;LOW BITS DON'T MATTER
7DF1 FE30       00490         CP      30H             ;30 IF READY
7DF3 C8         00500         RET     Z               ;GO SEND CHARS
7DF4 3A4038     00503         LD      A,(3840H)       ;STROBE CNTRL KEYS
7DF7 CB57       00506         BIT     2,A             ;BREAK PRESSED?
7DF9 C27200     00509         JP      NZ,0072H        ;QUIT IF SO
7DFC 18EE       00510         JR      READY           ;CHECK AGAIN
                00520 ;
0000            00530         END
00000 TOTAL ERRORS
```

Example

Enter Superlist

These are the features I specified for Superlist when I first began to research Level II:

- Each line of Basic should be split at the colons, with each statement starting a new printed line.
- The printout should have margins on the left and right sides of the paper, so a listing will fit neatly in a binder.
- Jammed-together lines of Basic code should be spread out by putting a space on either side of any Basic keyword.
- At the end of the listings, the program should report its size in number of lines, statements and bytes of memory occupied by the program text.

Each of these features was realized in Superlist, along with two I hadn't expected:

- Lines containing "If. . .Then . . .Else" statements are split at the "Else" by inserting a colon before it.
- Lines with apostrophe-type remarks are split before the apostrophe with a colon.

These were a result of the way Basic stores Else statements and remarks made with an apostrophe.

How to Use Superlist

To use Superlist, you need a Level II with at least 16K of memory, a printer that uses a Radio Shack parallel interface or any serial interface, and a

```
10 CLS
20 CLEAR200
30 DIM N1$(32),B%(8,8),N2$(16),G%(8,8),N3$(16)
40 DEFSTRN,Y:DEFINTA-M,O-X,Z
50 VI=15360:SC=0:IM=1:W=0
55 FORI=1TO8:FORJ=1TO8:B(I,J)=0:G(I,J)=0:NEXTJ,I
60 FORI=1TO32:N1(I)=RIGHT$(STR$(I),2):NEXT
70 FORI=1TO7:N2(I)=STRING$(2,CHR$(64+I)):NEXT:FORI=8TO16:N2(I)=STRING$(2,CHR$(65
+I)):NEXT
80 FORI=1TO8:N3(I)=STR$(I):NEXT:FORI=9TO16:N3(I)=CHR$(56+I)+" ":NEXT
85 IFCTHEN110
90 CLS:PRINT:PRINT"BLACK BOX VERSION 2.0":PRINT"BY MORRIS JONES":PRINT:PRINT"DO
YOU NEED INSTRUCTIONS? ";
95 Y=INKEY$:IFY=""THEN95
100 PRINTY;:FORI=1TO200:NEXT:IFY="Y"THENC=-1:GOTO8000:ELSEIFY<>"N"PRINTCHR$(08);
:GOTO95
110 C=0:CLS
500 FORVV=1TO8:FORHV=1TO8:D=VI+65+64*VV+3*HV:POKED,143:POKED+1,143:NEXTHV,VV
510 FORI=1TO8:PRINT@65+3*I,N1(I);:NEXT
520 FORI=9TO16:PRINT@156+64*(I-9),N1(I);:NEXT
530 FORI=17TO24:PRINT@665-3*(I-17),N1(I);:NEXT
540 FORI=25TO32:PRINT@577-64*(I-25),N1$(I);:NEXT
550 PRINT@99,"CHOOSE:";:PRINT@163,"DEGREE OF DIFFICULTY,";:PRINT@227,"NUMBER OF
BALLS (3-5)?";
560 H=VAL(INKEY$):IFH<3ORH>5THEN560ELSEPRINT@250,H;
570 FORI=1TOH
580 HV=RND(8):VV=RND(8)
590 IFB(HV,VV)THEN580ELSEB(HV,VV)=-1:NEXT
600 FORI=1TO700:NEXT
610 GOSUB8410
620 PRINT@99,"CHOOSE:";:PRINT@163,"1) STOP AND SCORE";:PRINT@227,"2) LAUNCH RAY"
;:PRINT@291,"3) PLACE BALL GUESS";
630 PRINT@355,"4) REMOVE BALL GUESS";:PRINT@419,"5) REDRAW BOX";:PRINT@773,"POIN
TS USED:";SC;
640 PRINT@803,STRING$(34," "):PRINT@803,"CHOICE?";
650 X0=VAL(INKEY$):IFX0<1ORX0>5THEN650ELSEPRINTX0;:FORI=1TO200:NEXT
660 ONX0GOTO1000,2000,3000,4000,5000
1000 FORI=1TO8:FORJ=1TO8:IFG(I,J)THENW=W+1:NEXTJ,IELSENEXTJ,I
1010 IFW>HPRINT@803,"TOO MANY GUESSES";:W=0:FORI=1TO700:NEXT:GOTO610
1020 IFW<HPRINT@803,"NOT ENOUGH GUESSES";:W=0:FORI=1TO700:NEXT:GOTO610
1025 GOSUB8410:GI=-1
```

Program Listing 1

copy of Editor/Assembler to assemble the source code into machine language. If you are using a 16K machine, assemble the code with an origin of 7B00H (31488 decimal), for 32K try BB00H (−17664 decimal), and for 48K try FB00H (−1280 decimal).

If you use a Radio Shack interface to drive your printer then your task is almost over. Delete lines 250 through 270 and lines 2980 through 3600; Superlist will automatically use the printer routine in Level II. You may still wish to use a routine that will format your copy in pages. The Level II driver counts printed lines, but does not send a form feed or a set of line feeds when a page is full.

I used the Heathkit H-14 printer with Small System Software's popular RS-232 interface. If you have one of these insert their machine language after line 2980. At the beginning of Superlist, the program will

✓See List of Advertisers on page 402

change Basic's line printer Device Control Block to reflect this routine location. In fact, once you have run Superlist you can continue to use its printer driver routine to LLIST and LPRINT. I borrowed the RS-232 driver routine used by Electric Pencil to drive my H-14 and modified it to count lines and form feed after a full page. Pencil's driver is a bit shorter than the relocatable code provided by Small Systems.

If you use Radio Shack's RS-232 interface to drive your serial printer, use the source code provided by Radio Shack in the book included with the interface.

In each of these cases the printer driver can be assembled separately if there is not room in the Editor/Assembler buffer for the additional text. Use an origin of 7DB4H. BUFF should be after the printer routine, so set it in the main source listing to the last address of the driver plus one byte. Try not to get carried away writing a long printer driver. BUFF should not end up above 7EC0H.

About Basic RAM

Before you can understand what Superlist is doing, you should know the format Basic uses to store its programs. This task would have been close to impossible were it not for Fuller Software's "Supermap" Level II ROM documentation. Although there seemed to be a few crucial things missing from the map, Fuller provided most of the information I needed.

The first line of any Basic program begins at 42E9H. The first two bytes of a line contain the starting address of the next line (stored by Superlist in NXTLIN). This is followed by the line number (stored in binary as a two-byte unsigned integer) then the line of text ending with a zero byte.

The big problem becomes decoding the Basic keyword tokens. The ASCII for each keyword is stored in ROM; all of the keywords are stored back to back in a list beginning at 1650H. The ASCII keywords are stored in the numerical order of their tokens, starting with token 128. The beginning of each keyword is marked by setting the highest bit (bit seven) in the first byte, which is the same as adding 128 to the ASCII code for the letter. In order to locate the ASCII keyword corresponding to a particular token, the program subtracts 128 from the token (by zeroing bit seven) and uses the remainder to count through the table. The count is decremented each time a new keyword is encountered (a byte with bit seven set) until the correct keyword is found.

Fuller referred to this subroutine in Level II ROM by saying "Call Mr. Spock for his opinion of this message," so I called my decoding routine "Spock." I had fun trying to find out how to decode the tokens, since the format of this table was one of the few things Fuller left out of "Supermap."

The surprises came after I had a working program to test. Basic stores an Else with a colon immediately before it. This had the desirable effect of breaking the line of text at each Else. But if the Basic programmer had already put a colon

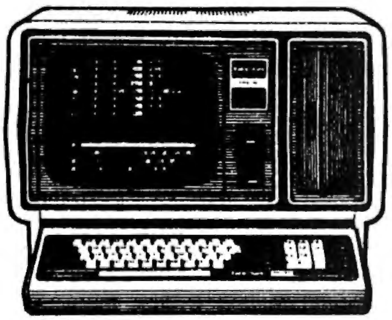
```
  10 CLS
  20 CLEAR 200
  30 DIM N1$(32),B%(8,8),N2$(16),G%(8,8),N3$(16)
  40 DEFSTR N,Y:
     DEFINT A - M,O - X,Z
  50 VI = 15360:
     SC = 0:
     IM = 1:
     W = 0
  55 FOR I = 1 TO 8:
     FOR J = 1 TO 8:
     B(I,J) = 0:
     G(I,J) = 0:
     NEXT J,I
  60 FOR I = 1 TO 32:
     N1(I) = RIGHT$ ( STR$ (I),2):
     NEXT
  70 FOR I = 1 TO 7:
     N2(I) = STRING$ (2, CHR$ (64 + I)):
     NEXT :
     FOR I = 8 TO 16:
     N2(I) = STRING$ (2, CHR$ (65 + I)):
     NEXT
  80 FOR I = 1 TO 8:
     N3(I) = STR$ (I):
     NEXT :
     FOR I = 9 TO 16:
     N3(I) = CHR$ (56 + I) + " ":
     NEXT
  85 IF C THEN 110
  90 CLS :
     PRINT :
     PRINT "BLACK BOX VERSION 2.0":
     PRINT "BY MORRIS JONES":
     PRINT :
     PRINT "DO YOU NEED INSTRUCTIONS? ";
  95 Y = INKEY$ :
     IF Y = "" THEN 95
 100 PRINT Y;:
     FOR I = 1 TO 200:
     NEXT :
     IF Y = "Y" THEN C = - 1:
     GOTO 8000:
     ELSE IF Y < > "N" PRINT CHR$ (08);:
     GOTO 95
 110 C = 0:
     CLS
 500 FOR VV = 1 TO 8:
     FOR HV = 1 TO 8:
     D = VI + 65 + 64 * VV + 3 * HV:
     POKE D,143:
     POKE D + 1,143:
     NEXT HV,VV
 510 FOR I = 1 TO 8:
     PRINT @65 + 3 * I,N1(I);:
     NEXT
 520 FOR I = 9 TO 16:
     PRINT @156 + 64 * (I - 9),N1(I);:
     NEXT
 530 FOR I = 17 TO 24:
     PRINT @665 - 3 * (I - 17),N1(I);:
     NEXT
 540 FOR I = 25 TO 32:
     PRINT @577 - 64 * (I - 25),N1$(I);:
     NEXT
 550 PRINT @99,"CHOOSE:";:
     PRINT @163,"DEGREE OF DIFFICULTY,";:
     PRINT @227,"NUMBER OF BALLS (3-5)?";
 560 H = VAL ( INKEY$ ):
     IF H < 3 OR H > 5 THEN 560:
     ELSE PRINT @250,H;
 570 FOR I = 1 TO H
 580 HV = RND (8):
     VV = RND (8)
 590 IF B(HV,VV) THEN 580:
     ELSE B(HV,VV) = - 1:
     NEXT
 600 FOR I = 1 TO 700:
     NEXT
 610 GOSUB 8410
 620 PRINT @99,"CHOOSE:";:
     PRINT @163,"1) STOP AND SCORE";:
     PRINT @227,"2) LAUNCH RAY";:
     PRINT @291,"3) PLACE BALL GUESS";
 630 PRINT @355,"4) REMOVE BALL GUESS";:
     PRINT @419,"5) REDRAW BOX";:
     PRINT @773,"POINTS USED:";SC;
 640 PRINT @803, STRING$ (34," "):
     PRINT @803,"CHOICE?";
 650 X0 = VAL ( INKEY$ ):
     IF X0 < 1 OR X0 > 5 THEN 650:
     ELSE PRINT X0;:
     FOR I = 1 TO 200:
     NEXT
 660 ON X0 GOTO 1000,2000,3000,4000,5000
1000 FOR I = 1 TO 8:
     FOR J = 1 TO 8:
     IF G(I,J) THEN W = W + 1:
     NEXT J,I:
     ELSE NEXT J,I
1010 IF W > H PRINT @803,"TOO MANY GUESSES";:
     W = 0:
     FOR I = 1 TO 700:
     NEXT :
     GOTO 610
1020 IF W < H PRINT @803,"NOT ENOUGH GUESSES";:
     W = 0:
     FOR I = 1 TO 700:
     NEXT :
     GOTO 610
1025 GOSUB 8410:
     GI = - 1

  35 BASIC LINES IN PROGRAM
 112 BASIC STATEMENTS IN PROGRAM
1348 BYTES USED BY PROGRAM TEXT
```

Program Listing 2

SAVE $$$ **TRS-80**® SAVE $$$

MICROCOMPUTERS

CALL US...
SAVE MONEY



Model II
64K
$3325.00
Full 3
Month
Warranty

CALL COLLECT:
512 - 689-5536
Master Electronics, Inc. ✓84
154 N. 5th, Raymondville, Tx. 78580

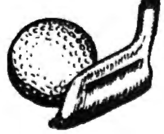
master charge

Form F48 Provided
Standard Warranty On Merchandise

VISA

Authorized TRS-80 Dealer, Store #F-723

GOLFER'S CHALLENGE



By Charles Butler
Take on the formidable "Liberty Links" golf course by yourself or with a friend. The course changes each time you play, offering 18 holes in a mixture of straight and dogleg configurations. They range from three to five par and are replete with trees, water hazards, sand traps, and out of bounds areas.

You judge the distance, choose your club from the full set provided, and indicate the direction you wish the ball to go. But watch out for the breeze or a poor hit -- it can play havoc with your shot! Just like in real golf, it takes practice to hit a sub-par round. Available on tape for TRS-80* models I & III with a minimum of 16K RAM. Order now for only $13.95.

MAIL ORDERS: Include $2.00 shipping & handling (D.C. residents add 6% sales tax).
CHARGE CARD CUSTOMERS: Call (202) 544-6674 for fastest service.

THE LIBERTY SOFTWARE Co
635 Independence Avenue, S.E.
Washington, D.C. 20003
(202) 544-6674 ✓61

ON SALE NOW!

MAKE BASIC PERFORM LIKE A CHAMPION

BASIC is not nearly as slow as most programmers think.

BASIC Faster and Better shows you how to supercharge your BASIC with 300 pages of fast, functions and subroutines.

You won't find any trivial poorly designed 'check-book balancing' programs in this book – it's packed with *useful* programs.

Tutorial for the beginner, instructive for the advanced, and invaluable for the professional, this book doesn't just talk. . . it shows how! All routines are also available on disk, so that you can save hours of keyboarding and debugging.

The # 1 disk *BFBDEM* contains all the demonstration programs, and # 2 *BFBLIB* has all the library functions.

BASIC Faster & Better is $29.95, and the two program disks are $19.95 each.
Get the book and/or disks from your local IJG dealer and B. Dalton bookstores.

IJG

1260 W. Foothill Blvd.
Upland, California 91786

(714) 946-5805

✓See List of Advertisers on page 402

before the Else I ended up with double colons, and a blank line of text. When Basic does a list to video, it suppresses the colon preceding an Else. I decided that was desirable, so Superlist looks only for the case of double colons and skips over one of them, so you won't produce a blank line. Let it be known you need never put a colon before an Else. Basic will do it for you, and you will never see it unless you use Superlist.

The other surprise was the way Basic stores remark lines. An apostrophe is interchangeable with a REM statement, but Basic does not store them the same way. If you put the keyword REM in a line, Basic will store REM as a one byte token. But if you put an apostrophe in a line Basic first stores a colon, then a REM token, then the apostrophe token. What looks like a one byte apostrophe actually takes three bytes of memory to store.

Superlist will look for an apostrophe token each time it encounters a REM token and skip past the REM if there is an apostrophe.

If you wish for Superlist to suppress these Basic extras in the same fashion an ordinary list does, see the modifications below.

Superlist Routines

The only way to tackle any large programming problem is to break it down into a group of smaller problems. Superlist was written in this "block" fashion. Here are *descriptions* of the routines used by Superlist (keep the source listing handy as you read these):

•From the beginning to Start2 the program initializes the memory and variable space it will be using. Superlist can be left in high memory and used over and over. Each time it is restarted it must clear the various flags and counters, which is done by lines 150 through 230. Lines 240 through 290 initialize Basic's line printer Device Control Block at 4025H.

Start2 is the starting point

Program Listing 3

```
               00100 ;SOURCE FILE FOR SUPERLIST.  MEMSIZE=31487  /31488
               00110 ;COPYRIGHT MORRIS JONES  MAY 14, 1980
7B00           00120         ORG     7B00H
7B00 ED73FE7F  00130         LD      (7FFEH),SP      ;SAVE SP
7B04 31FE7F    00140         LD      SP,7FFEH        ;NEW STACK
7B07 DD21AD7D  00150         LD      IX,LINES        ;INDEX TO CLEAR VARS
7B0B AF        00160         XOR     A
7B0C DD7700    00170         LD      (IX),A          ;CLEAR LINES
7B0F DD7701    00180         LD      (IX+1),A
7B12 DD7702    00190         LD      (IX+2),A        ;CLEAR STMNTS
7B15 DD7703    00200         LD      (IX+3),A
7B18 DD7704    00210         LD      (IX+4),A        ;CLEAR QUOTES
7B1B DD7705    00220         LD      (IX+5),A        ;CLEAR QUITFL
7B1E DD7706    00230         LD      (IX+6),A        ;CLEAR CHARS
7B21 DD212540  00240         LD      IX,4025H        ;PRINTER DCB
7B25 21B47D    00250         LD      HL,PRINT3       ;USE 058DH FOR
7B28 DD7501    00260         LD      (IX+1),L        ;   BASIC'S PRINTER
7B2B DD7402    00270         LD      (IX+2),H        ;   DRIVER.
7B2E DD360336  00280         LD      (IX+3),54       ;# OF LINES
7B32 DD360400  00290         LD      (IX+4),0        ;LINE COUNTER
7B36 FD21E942  00300         LD      IY,42E9H        ;INIT BASIC POINTER
7B3A FD22AB7D  00310         LD      (NXTLIN),IY     ;STARTING POINT
7B3E FD2AAB7D  00320 START2  LD      IY,(NXTLIN)     ;NEW LINE START
7B42 DD21197E  00330         LD      IX,BUFF         ;INIT OUTPUT BUFFER
7B46 FD4E00    00340         LD      C,(IY)          ;NEXT LINE LSB
7B49 FD23      00350         INC     IY
7B4B FD4600    00360         LD      B,(IY)          ;NEXT LINE MSB
7B4E FD23      00370         INC     IY
```

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 780 FARMINGDALE NY 11737

POSTAGE WILL BE PAID BY ADDRESSEE

80 microcomputing™

Subscription Dept.
POB 981
Farmingdale NY 11737

Program continued

```
7B50 78       00380          LD     A,B           ;TO TEST
7B51 B1       00390          OR     C             ;FOR ZERO, END OF FILE
7B52 CA087D   00400          JP     Z,QUIT        ;QUIT IF SO
7B55 ED43AB7D 00410          LD     (NXTLIN),BC   ;SAVE IF NOT
7B59 FD6E00   00420          LD     L,(IY)        ;LOW BYTE LINE #
7B5C FD23     00430          INC    IY
7B5E FD6600   00440          LD     H,(IY)        ;HIGH BYTE LINE #
7B61 FD23     00450          INC    IY
7B63 AF       00460          XOR    A             ;CLEAR ACCUMULATOR
7B64 111027   00470          LD     DE,10000      ;FIRST DIVISOR
7B67 CDDC7B   00480          CALL   SUBTR         ;LEAVES DIGIT IN A
7B6A 11E803   00490          LD     DE,1000       ;THOUSANDS DIGIT
7B6D CDDC7B   00500          CALL   SUBTR
7B70 116400   00510          LD     DE,100        ;HUNDREDS DIGIT
7B73 CDDC7B   00520          CALL   SUBTR
7B76 110A00   00530          LD     DE,10         ;TENS DIGIT
7B79 CDDC7B   00540          CALL   SUBTR
7B7C 7D       00550          LD     A,L           ;ONES DIGIT IS LEFT
7B7D C630     00560          ADD    A,30H
7B7F DD7700   00570          LD     (IX),A
7B82 DD23     00580          INC    IX
7B84 3E20     00590          LD     A,20H         ;SPACE AFTER LINE #
7B86 DD7700   00600          LD     (IX),A        ;SAVE IT
7B89 DD23     00610          INC    IX
7B8B FD7E00   00620 A2       LD     A,(IY)        ;GET NEXT CHAR
7B8E FD23     00630          INC    IY            ;INC BASIC PNTR
7B90 CB7F     00640          BIT    7,A           ;CHECK HIGH BIT
7B92 2835     00650          JR     Z,A3          ;GO IF NOT
7B94 FE93     00660          CP     93H           ;CHECK FOR REM
7B96 201C     00670          JR     NZ,A13        ;NO,GO TO ELSE CHECK
7B98 FD7E00   00680          LD     A,(IY)        ;LOOK AT NEXT FOR '
7B9B FEFB     00690          CP     0FBH          ;APOSTROPHE TOKEN
7B9D 2805     00700          JR     Z,A14         ;YES, GO TO : CHECK
7B9F FD7EFF   00710          LD     A,(IY-1)      ;NO, GET BACK REM
7BA2 1820     00720          JR     A15           ;GO TO SPOCK
7BA4 FD23     00730 A14      INC    IY            ;SKIP PAST REM IF '
7BA6 FD7EFC   00740          LD     A,(IY-4)      ;LOOK BACK FOR :
7BA9 FE3A     00750          CP     3AH           ;IN CASE OF 2 COLONS
7BAB 2002     00760          JR     NZ,A16
7BAD DD2B     00770          DEC    IX            ;WRITE OVER IT
7BAF FD7EFF   00780 A16      LD     A,(IY-1)      ;GET BACK APOSTROPHE
7BB2 1810     00790          JR     A15           ;GO TO SPOCK
7BB4 FE95     00800 A13      CP     95H           ;CHECK FOR 'ELSE'
7BB6 200C     00810          JR     NZ,A15
7BB8 FD7EFD   00820          LD     A,(IY-3)      ;LOOK FOR '::'
7BBB FE3A     00830          CP     3AH
7BBD 2002     00840          JR     NZ,A17
7BBF DD2B     00850          DEC    IX            ;YES, WRITE OVER SECOND :
7BC1 FD7EFF   00860 A17      LD     A,(IY-1)      ;GET BACK 'ELSE' TOKEN
7BC4 CDF07B   00870 A15      CALL   SPOCK         ;INTERPRET IF BIT7 SET
7BC7 18C2     00880          JR     A2
7BC9 B7       00890 A3       OR     A             ;TEST FOR ZERO
7BCA 2009     00900          JR     NZ,A4         ;GO FORWARD
7BCC DD7700   00910          LD     (IX),A        ;SAVE THE ZERO
7BCF CD367C   00920          CALL   PRINT1        ;PRINT THE LINE
7BD2 C33E7B   00930          JP     START2        ;TRUCK ON
7BD5 DD7700   00940 A4       LD     (IX),A        ;JUMP HERE IF
7BD8 DD23     00950          INC    IX            ; ORDINARY ASCII
7BDA 18AF     00960          JR     A2            ;GET ANOTHER UN
              00970 ;
7BDC AF       00980 SUBTR    XOR    A             ;CLEAR A
7BDD ED52     00990 B1       SBC    HL,DE         ;SUBTRACT DE
7BDF FAE57B   01000          JP     M,C1          ;JUMP IF NEGATIVE
7BE2 3C       01010          INC    A             ;COUNT IF POSITIVE
7BE3 18F8     01020          JR     B1
7BE5 B7       01030 C1       OR     A             ;TO RESET CARRY FLAG
7BE6 ED5A     01040          ADC    HL,DE         ;ADD THE DIVISOR BACK
7BE8 C630     01050          ADD    A,30H         ;CONVERT TO ASCII
7BEA DD7700   01060          LD     (IX),A
7BED DD23     01070          INC    IX            ;SAVE CHAR
7BEF C9       01080          RET                  ;COUNT THROUGH
              01090 ;
7BF0 F5       01100 SPOCK    PUSH   AF            ;SAVE CODED CHAR
7BF1 DD7EFF   01110          LD     A,(IX-1)      ;LOOK BACK AND SEE
7BF4 FE20     01120          CP     20H           ;IF IT'S A SPACE
7BF6 280B     01130          JR     Z,SP1         ;IF IT IS FORGET IT
7BF8 FE3A     01140          CP     3AH           ;OR IF IT'S A COLON
7BFA 2807     01150          JR     Z,SP1         ;FORGET IT, TOO
7BFC 3E20     01160          LD     A,20H         ;BUT IF IT ISN'T
7BFE DD7700   01170          LD     (IX),A        ;PUT ONE THER
7C01 DD23     01180          INC    IX
7C03 F1       01190 SP1      POP    AF            ;GET BACK TOKEN
7C04 E67F     01200          AND    7FH           ;WASTE THE HIGH BIT
7C06 47       01210          LD     B,A           ;MAKES IT A COUNTER
7C07 04       01220          INC    B
7C08 215016   01230          LD     HL,1650H      ;TABLE START
7C0B 7E       01240 SP2      LD     A,(HL)        ;GET A SAMPLE
7C0C CB7F     01250          BIT    7,A           ;CHECK THE HIGH BIT
7C0E 23       01260          INC    HL            ;MOVE AHEAD ANYWAY
7C0F 28FA     01270          JR     Z,SP2         ;NO GOOD, GET ANOTHER
7C11 10F8     01280          DJNZ   SP2           ;DEC COUNTER IF IT IS
7C13 2B       01290          DEC    HL            ;WE FOUND IT!
7C14 E67F     01300          AND    7FH           ;AND IT'S IN A
7C16 DD7700   01310          LD     (IX),A        ;STORE THE SUCKER
7C19 DD23     01320          INC    IX
7C1B 23       01330 SP3      INC    HL            ;LOOK AT THE
7C1C 7E       01340          LD     A,(HL)        ; NEXT ONE
7C1D CB7F     01350          BIT    7,A           ;NEW WORD?
7C1F 2007     01360          JR     NZ,SP4        ;YES, QUIT
7C21 DD7700   01370          LD     (IX),A        ;NO, SAVE IT
7C24 DD23     01380          INC    IX
7C26 18F3     01390          JR     SP3           ;GO LOOK AGAIN
7C28 FD7E00   01400 SP4      LD     A,(IY)        ;LOOK AT NEXT CHAR
7C2B FE20     01410          CP     20H           ;  FOR A SPACE
7C2D C8       01420          RET    Z             ;QUIT IF IT IS
7C2E 3E20     01430          LD     A,20H         ;ELSE
7C30 DD7700   01440          LD     (IX),A        ;PUT ONE IN
7C33 DD23     01450          INC    IX
7C35 C9       01460          RET                  ;SPLIT
              01470 ;
7C36 DD21197E 01480 PRINT1   LD     IX,BUFF       ;RESET BUFFER PNTR
7C3A ED5BAD7D 01490          LD     DE,(LINES)    ;GET LINE COUNTER
7C3E 13       01500          INC    DE            ;ADD ONE
7C3F ED53AD7D 01510          LD     (LINES),DE    ;SAVE IT
7C43 ED5BAF7D 01520          LD     DE,(STMNTS)   ;GET STATEMENTS COUNT
7C47 13       01530          INC    DE            ;ADD ONE
7C48 ED53AF7D 01540          LD     (STMNTS),DE   ;SAVE IT
7C4C CDDF7C   01550          CALL   INDENT        ;SET LEFT MARGIN
7C4F DD4E00   01560 A11      LD     C,(IX)        ;GET FIRST CHARACTER
7C52 DD23     01570          INC    IX            ;POINT TO NEXT
```

Program continues

for each new line of Basic text read by Superlist. It begins by reading the address of the next line of program and storing it. Then it reads the two-byte line number. Converting this number into decimal digits is a matter of dividing the line number by 10000 to get the first digit, dividing the remainder by 1000 for the next digit and so on. The line number is stored in HL, the divisor in DE. Upon returning from SUBTR (described below) the ASCII digit has been stored in the output buffer, and the remainder is left in HL.

Each character of the Basic line is then put through a series of tests in lines 620 through 960. If bit seven of the character is set, then it is a Basic token. If not, it is merely placed in the output buffer as an ASCII character. A zero byte indicates the end of a line, and the program calls Print1 to print the buffer and goes back to Start2 to set the next line. If the character is a Basic token, it is tested to see if it is an Else or a REM. In the case of an Else, it checks to see if it was preceded by two colons and deletes one from the listing. If the token is a REM followed by an apostrophe, it deletes the REM. The program also looks for two colons preceding the REM and deletes one of those. After testing, the token is sent to Spock for decoding.

•SUBTR is the division routine for the line number. DE contains the divisor and HL the dividend. Each time DE can be subtracted from HL with a positive result, register A is incremented. When the subtraction produces a negative result, DE is added back to HL. Register A then contains the quotient and HL the remainder. Adding 30H to A converts it into an ASCII digit which is then stored in the output buffer.

Upon encountering a Basic token, the program calls Spock. One of the aims of Superlist is to spread out the compacted lines. Spock does this by checking for a space before and after the keyword and putting in spaces if they don't exist. To decode the token, the high bit is set to zero by ANDing the A register with 7FH. The number remaining is how far down the table you must look to find the keyword. (For instance, after zeroing bit seven, if 34 is left, the word is 34th in the list.)

•Print1 is called after a complete line of Basic has been interpreted, spread out and stored in the output buffer. First the counters for lines and statements are incremented, then the program looks at the first characters of the line number. In order to prevent leading zeros from being printed, the routine substitutes a space for a zero until a nonzero character is encountered.

Print1 also looks for colons and if it encounters one, increments the statement counter, prints a carriage return and indents six spaces. However, if the colon is enclosed in quotation marks as part of a string constant, we do not wish to split the line. Quotes is a flag set when the routine encounters a quotation mark. If Quotes is set, the colon check is skipped and reset if Print1 encounters a zero byte signifying the end of the line.

Subroutines

The program drops into four levels of subroutines before printing a character.

•The main routine calls Print1 when the output buffer is ready to be printed. Print1 calls Print2 each time a character is ready to be printed.

•Print2 counts the characters to provide a right margin before finally calling Print. If a line is full, Print2 sends a carriage return and indents 11 spaces before continuing. If the character sent to Print2 is a carriage return, Print2 resets the character counter.

•Indent is optional. It merely prints eight spaces to provide a left margin.

•Print begins by checking to see if you are pressing Break and jumps to Quit if you are. Print is written as if the location of the printer routine is unknown. Index register IX is loaded with the start address of Basic's line printer Device Control Block which contains the

Program continued

```
7C54 79        01580         LD    A,C          ;TEST FOR ASCII ZERO
7C55 FE30      01590         CP    30H
7C57 200D      01600         JR    NZ,A12       ;OUTPUT LINE IF OK
7C59 0E20      01610         LD    C,20H        ;SEND A SPACE IF NOT
7C5B CDAD7C    01620         CALL  PRINT2
7C5E 18EF      01630         JR    A11          ;GO GET NEXT ONE
7C60 DD4E00    01640 A6      LD    C,(IX)       ;GET FIRST CHAR
7C63 DD23      01650         INC   IX           ;POINT TO NEXT ONE
7C65 79        01660         LD    A,C          ;TO TEST IT
7C66 FE22      01670 A12     CP    22H          ; FOR QUOTES
7C68 2007      01680         JR    NZ,A5        ;NO, GO AHEAD
7C6A 3AB17D    01690         LD    A,(QUOTES)   ;YES,
7C6D 2F        01700         CPL                ;CHANGE THE FLAG
7C6E 32B17D    01710         LD    (QUOTES),A
7C71 3AB17D    01720 A5      LD    A,(QUOTES)   ;WELL IT WASN'T THERE
7C74 B7        01730         OR    A            ; IF WE JUMPED HERE
7C75 79        01740         LD    A,C          ;GET BACK CHAR
7C76 2004      01750         JR    NZ,A7        ;IF FLAG SKIP COLON TEST
7C78 FE3A      01760         CP    3AH          ;CHECK FOR COLON
7C7A 2812      01770         JR    Z,A8         ;YES, GO FORWARD
7C7C B7        01780 A7      OR    A            ;NO, TEST FOR ZERO
7C7D 200A      01790         JR    NZ,A10       ;NO, JUMP TO PRINT
7C7F AF        01800         XOR   A
7C80 32B17D    01810         LD    (QUOTES),A   ;CLEAR QUOTES FLAG
7C83 0E0D      01820         LD    C,0DH        ;YES, SEND
7C85 CDAD7C    01830         CALL  PRINT2       ; A CRLF
7C88 C9        01840         RET
7C89 CDAD7C    01850 A10     CALL  PRINT2       ;PRINT THE CHAR IF<>0
7C8C 18D2      01860         JR    A6           ;GET MORE CHARACTERS
7C8E ED5BAF7D  01870 A8      LD    DE,(STMNTS)  ;HERE IF COLON
7C92 13        01880         INC   DE           ;ADD ONE TO
7C93 ED53AF7D  01890         LD    (STMNTS),DE  ; STATEMENT COUNTER
7C97 CDAD7C    01900         CALL  PRINT2       ;PRINT THE COLON
7C9A 0E0D      01910         LD    C,0DH        ;AND SEND
7C9C CDAD7C    01920         CALL  PRINT2       ;A CRLF
7C9F CDDF7C    01930         CALL  INDENT       ;SET LEFT MARGIN
7CA2 0606      01940         LD    B,6          ;COUNT SIX
7CA4 0E20      01950         LD    C,20H        ; SPACES
7CA6 CDAD7C    01960 A9      CALL  PRINT2       ;AND SEND
7CA9 10FB      01970         DJNZ  A9           ; EACH ONE
7CAB 18B3      01980         JR    A6           ;GET MORE CHARS
               01990 ;
7CAD 79        02000 PRINT2  LD    A,C          ;ROUTINE TO COUNT CHARS
7CAE FE0D      02010         CP    0DH          ;TEST FOR CR
7CB0 2825      02020         JR    Z,P1         ;GO CLEAR AND SEND
7CB2 3AB37D    02030         LD    A,(CHARS)    ;GET CHAR COUNTER
7CB5 3C        02040         INC   A            ;ADD ONE
7CB6 32B37D    02050         LD    (CHARS),A    ;SAVE IT
7CB9 FE41      02060         CP    65           ;65 IS FULL LINE
7CBB 201E      02070         JR    NZ,P2        ;PRINT IT IF NOT 65 YET
7CBD C5        02080         PUSH  BC           ;SAVE THE CHAR
7CBE 0E0D      02090         LD    C,0DH        ;LOAD A CR
7CC0 CDEB7C    02100         CALL  PRINT        ;AND SEND IT
7CC3 CDDF7C    02110         CALL  INDENT       ;LEFT MARGIN
7CC6 3E0C      02120         LD    A,12         ;RESET CHARS
7CC8 32B37D    02130         LD    (CHARS),A    ;TO INDENTED VALUE
7CCB 060B      02140         LD    B,11         ;FOR ELEVEN SPACES
7CCD 0E20      02150         LD    C,20H        ;SPACE ASCII
7CCF CDEB7C    02160 P3      CALL  PRINT        ;PRINT A SPACE
7CD2 10FB      02170         DJNZ  P3           ;ELEVEN TIMES
7CD4 C1        02180         POP   BC           ;GET BACK CHAR
7CD5 1804      02190         JR    P2           ;GO PRINT IT
7CD7 AF        02200 P1      XOR   A            ;CLEAR CHARS
7CD8 32B37D    02210         LD    (CHARS),A
7CDB CDEB7C    02220 P2      CALL  PRINT
7CDE C9        02230         RET                ;BACK TO PRINT1
               02240 ;
7CDF C5        02250 INDENT  PUSH  BC
7CE0 0608      02260         LD    B,8          ;INDENT FOR LEFT MARGIN
7CE2 0E20      02270         LD    C,20H        ;SPACE
7CE4 CDEB7C    02280 IND1    CALL  PRINT
7CE7 10FB      02290         DJNZ  IND1
7CE9 C1        02300         POP   BC
7CEA C9        02310         RET
               02320 ;
7CEB E5        02330 PRINT   PUSH  HL
7CEC DDE5      02340         PUSH  IX
7CEE 214038    02350         LD    HL,3840H     ;LOOK AT BREAK KEY
7CF1 CB56      02360         BIT   2,(HL)
7CF3 2023      02370         JR    NZ,Q2        ;QUIT IF PRESSED
7CF5 DD212540  02380         LD    IX,4025H     ;PRINTER DCB GOES IN IX
7CF9 21047D    02390         LD    HL,PRTRET
7CFC E5        02400         PUSH  HL           ;PUSH RETURN ADDRESS
7CFD DD6E01    02410         LD    L,(IX+1)     ;WE DO THIS SINCE
7D00 DD6602    02420         LD    H,(IX+2)     ; WE CAN'T CALL(HL)
7D03 E9        02430         JP    (HL)
7D04 DDE1      02440 PRTRET  POP   IX           ;RETURN HERE AND GET
7D06 E1        02450         POP   HL           ;  BACK REGS
7D07 C9        02460         RET
               02470 ;
7D08 3AB27D    02480 QUIT    LD    A,(QUITFL)   ;GET FLAG
7D0B B7        02490         OR    A            ;TEST IT
7D0C 2811      02500         JR    Z,Q1         ;IF IT'S NOT SET GO UP
7D0E 3E00      02510         LD    A,0          ;CLEAR IT IF IT IS
7D10 32B27D    02520         LD    (QUITFL),A
7D13 0E0C      02530         LD    C,12         ;FORM FEED
7D15 CDEB7C    02540         CALL  PRINT
7D18 ED7BFE7F  02550 Q2      LD    SP,(7FFEH)   ;RESTORE STACK
7D1C C37200    02560         JP    72H          ;AND GO BACK TO BASIC
7D1F ED4BAD7D  02570 Q1      LD    BC,(LINES)   ;PUT LINES
7D23 ED43517D  02580         LD    (M2A),BC     ;IN MESSAGE
7D27 ED4BAF7D  02590         LD    BC,(STMNTS)  ;PUT STATEMENTS
7D2B ED436C7D  02600         LD    (M3A),BC     ;IN MESSAGE
7D2F FDE5      02610         PUSH  IY           ;END OF BASIC
7D31 E1        02620         POP   HL           ;INTO HL
7D32 11E942    02630         LD    DE,42E9H     ;START OF BASIC
7D35 B7        02640         OR    A            ;CLEAR CARRY
7D36 ED52      02650         SBC   HL,DE        ;GET THE DIFFERENCE
7D38 228C7D    02660         LD    (M4A),HL     ;PUT IN MESSAGE
7D3B 014A7D    02670         LD    BC,M1        ;GET LINE ADDRESS
7D3E ED43AB7D  02680         LD    (NXTLIN),BC  ;AND PUT IN NXTLIN
7D42 3EFF      02690         LD    A,0FFH       ;TO SET QUIT FLAG
7D44 32B27D    02700         LD    (QUITFL),A
7D47 C33E7B    02710         JP    START2       ;GO PRINT IT!
               02720 ;
7D4A 4F7D      02730 M1      DEFW  M2
7D4C 0000      02740 M1A     DEFW  0
7D4E 00        02750         DEFB  0
7D4F 6A7D      02760 M2      DEFW  M3
7D51 0000      02770 M2A     DEFW  0
7D53 42        02780         DEFM  'BASIC LINES IN PROGRAM'
```

Program continues

✓See List of Advertisers on page 402

Program continued

```
7D54 41
7D55 53
7D56 49
7D57 43
7D58 20
7D59 4C
7D5A 49
7D5B 4E
7D5C 45
7D5D 53
7D5E 20
7D5F 49
7D60 4E
7D61 20
7D62 50
7D63 52
7D64 4F
7D65 47
7D66 52
7D67 41
7D68 4D
7D69 00          02790           DEFB    0
7D6A 8A7D        02800 M3        DEFW    M4
7D6C 0000        02810 M3A       DEFW    0
7D6E 42          02820           DEFM    'BASIC STATEMENTS IN PROGRAM'
7D6F 41
7D70 53
7D71 49
7D72 43
7D73 20
7D74 53
7D75 54
7D76 41
7D77 54
7D78 45
7D79 4D
7D7A 45
7D7B 4E
7D7C 54
7D7D 53
7D7E 20
7D7F 49
7D80 4E
7D81 20
7D82 50
7D83 52
7D84 4F
7D85 47
7D86 52
7D87 41
7D88 4D
7D89 00          02830           DEFB    0
7D8A A97D        02840 M4        DEFW    M5
7D8C 0000        02850 M4A       DEFW    0
```

Program continues

address of the printer driver, and this address can be loaded into HL. The driver routine ends with a RET, but there is no Z80 instruction to Call (HL). When the RET is executed it will pop a return address from the stack; the program must put the return address in the stack by PUSHing it before loading the routine address in HL and jumping to it.

When the main routine encounters 0000 as the NXTLIN address, Superlist has reached the end of the Basic program and jumps to the Quit routine.

The messages that tell how big the program is must still be taken care of. To print these messages I decided to set up a block of memory with the same format as Basic RAM. The lines and statements have been counted and saved in LINES and STMNTS. These are two-byte unsigned integers, just like the line numbers of Basic. Quit stores these numbers where the line numbers would appear in Basic. When the program

See List of Advertisers on page 402

Program continued

```
7D8E 42        02860         DEFM    'BYTES USED BY PROGRAM TEXT'
7D8F 59
7D90 54
7D91 45
7D92 53
7D93 20
7D94 55
7D95 53
7D96 45
7D97 44
7D98 20
7D99 42
7D9A 59
7D9B 20
7D9C 50
7D9D 52
7D9E 4F
7D9F 47
7DA0 52
7DA1 41
7DA2 4D
7DA3 20
7DA4 54
7DA5 45
7DA6 58
7DA7 54
7DA8 00        02870         DEFB    0
7DA9 0000      02880 M5      DEFW    0
7DAB 0000      02890 NXTLIN  DEFW    0
7DAD 0000      02900 LINES   DEFW    0
7DAF 0000      02910 STMNTS  DEFW    0
7DB1 00        02920 QUOTES  DEFB    0
7DB2 00        02930 QUITFL  DEFB    0
7DB3 00        02940 CHARS   DEFB    0
               02950 ;
               02960 ;    INSERT YOUR PRINTER ROUTINE HERE
               02970 ;
7DB4           02980 PRINT3  EQU     $
               02990 *LIST OFF                         ;PROTECT TRS232 DRIVER
               03600 *LIST ON
7E19           03610 BUFF    EQU     $
7B00           03620         END     7B00H
00000 TOTAL ERRORS

DELAY     7E13
OUT       7E0E
OUT0      7E0C
PR1       7DF8
OUT1      7E08
PRINT4    7DEF
ST4       7DDF
ST3       7DE9
ST2       7DC8
```

Program continues

reaches Quit the last address of the Basic program is in IY. The start of Basic is subtracted from this address to set the length of the program in bytes, and this number is stored with the last message. Quit then sets a flag (Quitfl), stores the start of the message block in NXTLIN and jumps back to Start2. The program decodes the text and prints it just as if it were Basic. The next time control goes to Quit and the flag has been set, Quit prints a form feed, restores the stack pointer and jumps back to Basic.

Modifications

Superlist can be customized to suit whatever needs or desires you may have for Basic listings.

The left margin of the listing is set by the Indent routine. To change the left margin, change the value in line 2260. If you want no left margin, leave out the Indent routine and delete the calls to Indent in lines 1550, 1930 and 2110.

The total line length is set in line 2060 to 65 characters. You can change this value to whatever right margin you prefer.

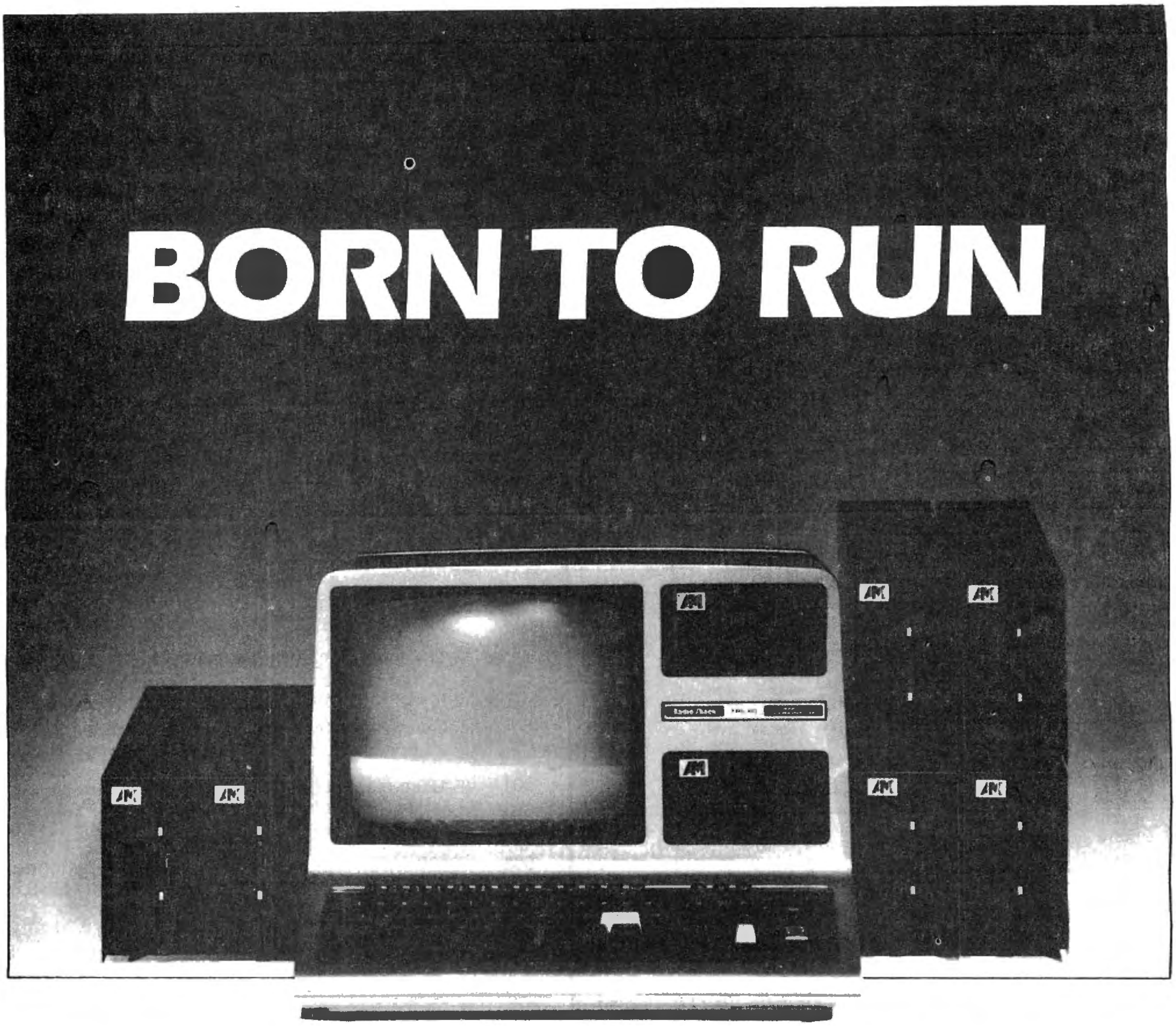
Line 280 intializes the printer Device Control Block for 54 lines per page. If you want more or less, change that value. You must have a printer routine that keeps track of the line count to get paged output.

If you do not wish the automatic colon before Else statements, delete lines 820-840 and 860. To suppress the automatic colon before an apostrophe remark statement, delete lines 740-760 and 780.

Superlist was written before the recent changes in Level II Basic. The only changes affecting Superlist are the start of Basic program storage and the possible relocation of the token look-up table. I understand a new edition of "Supermap" includes all the changes to Level II. "Supermap" is available from Fuller Software, Grand Prairie, TX, or your local computer store. ■

Program continued

```
ST1      7DB6
M5       7DA9
M4       7D8A
M3       7D6A
M1A      7D4C
M2       7D4F
M1       7D4A
M4A      7D8C
M3A      7D6C
M2A      7D51
Q1       7D1F
QUITFL   7DB2
PRTRET   7D04
Q2       7D18
IND1     7CE4
P3       7CCF
PRINT    7CEB
P2       7CDB
CHARS    7DB3
P1       7CD7
A9       7CA6
A10      7C89
A8       7C8E
A7       7C7C
QUOTES   7DB1
A5       7C71
A6       7C60
PRINT2   7CAD
A12      7C66
A11      7C4F
INDENT   7CDF
STMNTS   7DAF
SP4      7C28
SP3      7C1B
SP2      7C0B
SP1      7C03
C1       7BE5
B1       7BDD
PRINT1   7C36
A4       7BD5
SPOCK    7BF0
A17      7BC1
A16      7BAF
A15      7BC4
A14      7BA4
A13      7BB4
A3       7BC9
A2       7BB8
SUBTR    7BDC
QUIT     7D08
BUFF     7E19
START2   7B3E
NXTLIN   7DAB
PRINT3   7DB4
LINES    7DAD
```


BORN TO RUN

See List of Advertisers on page 402

An op-amp screening program for those times when you can't get the one you really want.

Second Sourcing

David Tinis
National Controls Inc.
4500 John Young Parkway
Orlando, FL 32804

Have you ever been enthusiastic about an electronics project only to find one or two parts were hard or impossible to get? When it comes to op-amps this seems especially true. While many bear standard part numbers, it's not uncommon for equivalent parts to have a different identification.

Since one facet of my job is to screen vendor-submitted components for compliance with specifications established for current parts in usage, it seemed desirable to let the computer do the preliminary screening. Op-amp is a program I wrote that allows me to compare 14 selected parameters between a currently used part and one being submitted as a possible second source. The program was originally written for use on an INTEL MDS230 and later adapted to my TRS-80.

The Parts File

There are actually two parts to this program. The first is used to create a file named Parts containing the specifications of known and used parts. The second is to compare the part under consideration with a known part in the previously created Parts file.

The Parts file is created as a sequential access file under VTOS 4.0. One feature of VTOS 4.0 is an enhancement to Disk Basic which allows a sequential file to be opened and positioned to its end. New data can then be added without the necessity of reading the entire file into an array in memory, adding the new data to the array and then rewriting the file. Non-VTOS users will have to modify the program for this type of operation.

The Evaluation

The evaluation function of the program is straight forward. The program first queries for the known op-amp in the Parts file to be used as the standard of comparison. It then asks for parameter information about the part under consideration and, after all information has been entered, lists any specifications that compare unfavorably. A printout or display on your video of all the parameters point by point can be obtained.

One thing to remember when entering parameter information is to avoid using typical data. Instead, use a bad example for the parameter being evaluated.

The fourteen parameters considered by the program are listed below:

- Power Supply Voltage: Most op-amps operate from equal voltages of opposite polarity. The value to be used here is the maximum allowable power supply voltage of either polarity referenced to zero volts.
- Differential Input Voltage: The maximum allowable difference in input voltage between the two op-amp input terminals.
- Input Voltage: Maximum allowable input voltage of either polarity that can be applied to an input with the other held to zero volts.
- Input Offset Voltage: The voltage that must be applied to the inputs to achieve a zero output voltage. This generally indicates how closely matched the inputs are. The lower this value, the better.
- Input Offset Current: The difference in input bias currents which causes a voltage difference across the input resistors. If this value is large, unacceptable input voltages can be generated.
- Input Bias Current: Average of the two input bias currents which cause voltage drop across the input resistors that must be overcome to the input signal.
- Input Noise Voltage: A measure of internally generated noise at the inputs of the op-amp.
- Differential Input Resistance: The impedance seen by a source looking into the inputs of the op-amp. This figure is used when matching the op-amp to driving device. A mismatch will cause excessive loss of input signal.
- Common Mode Rejection Ratio: This indicates how well balanced the differential stages of the op-amp are.
- Large Signal Voltage Gain: The ratio of the change in output voltage to a change in input voltage measured without feedback or compensation. This figure is frequency-dependent.
- Output Voltage Swing: Peak voltage output that can be obtained without clipping. This is measured with reference to zero volts.
- Slew Rate: The maximum rate of change of output voltage with respect to time that can be obtained while maintaining linearity.
- Power Supply Rejection Ratio: A measure of sensitivity to changes in power supply voltage and noise on power supply lines.
- Power Dissipation: The maximum power the device can safely dissipate at a given temperature and under specified conditions of power supply voltage and load.

Most op-amp data sheets list other parameters but these 14 were chosen for their overall importance to device operation. Once a preliminary screening of these parameters has been done, a closer look can be given the part if warranted. In any event this program can be a real aid in finding new sources of replacement parts. ■

Program Listing

```
10 CLS
20 CLEAR500:DIM A(16),K(16),U(16),S$(16),U$(16)
21 FORX=1TO16:READU$(X):NEXTX
22 DATA "VOLTS"," ","VOLTS","VOLTS","MICROVOLTS","NANOAMPS","NAN
OAMPS","MICROVOLTS","MEGOHMS"," ","DB","V/MV","VOLTS","V/MICROSE
COND","DB","MW"
25 FORX=1TO16:READS$(X):NEXTX
```

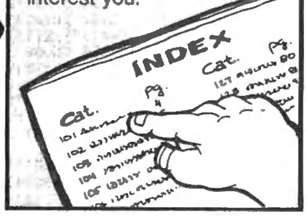
Program continues

How to Buy or Sell Computer Equipment and Software

and come out a winner

START HERE WITH...

Begin your search in the index of Computer Shopper's bargain filled pages. Locate the category and page number of items that interest you.



Computer Shopper's easy-to-read format makes your search quick and simple with classifications at the top of each column in bold black headings.

Start or add to your computer system by finding money saving bargains in each month's issue from individuals who no longer need their personal equipment.

You've got your computer hardware, but what about the software? Use a Computer Shopper ad to find what you need. Someone advertising in Computer Shopper probably has what you want.

You have the convenience of shopping nationwide without ever leaving your home or office. And if you buy out of state you'll save on sales tax.

If you need help with any computer related problem whether it's an interface problem or advice on the right peripheral for a home-brew system, use the free HELP column especially designed for that purpose.

As you outgrow your system or want to trade up (most dealers won't take trade-ins), use Computer Shopper ads to sell your items to 20,000 readers nationwide for the low cost of 10 cents per word.

Congratulations, you're a winner whether you buy, sell or trade in the computer marketplace by subscribing and using Computer Shopper's action yellow pages. And for your first bargain, take advantage of a FREE issue of Computer Shopper as a "Special Get Acquainted Offer."

Act today and receive your first issue free in addition to the $10, one year/12 issue regular subscription. Of course, if you are not 100% satisfied, you may cancel and still keep the free issue just for trying Computer Shopper.

BONUS: New subscribers are entitled to a complimentary classified ad. Use it for pre-owned equipment or software or as a want-to-buy ad. Just print or type your ad on a plain piece of paper and send it along with your subscription. Your phone will probably start ringing before you receive your first issue.

For MasterCard and VISA Subscriptions only,
CALL TOLL FREE 1-800-327-9920

For MasterCard and VISA Classified Ad orders only,
CALL TOLL FREE 1-800-327-9926

COMPUTER SHOPPER ✓212
P.O. Box F138 • Titusville, FL 32780
Telephone 305-269-3211

Program continued

```
26 DATA "POWER SUPPLY VOLTAGE"," ","DIFFERENTIAL INPUT VOLTAGE",
"INPUT VOLTAGE","INPUT OFFSET VOLTAGE","INPUT OFFSET CURRENT","I
NPUT BIAS CURRENT","INPUT NOISE VOLTAGE","DIFFERENTIAL INPUT RES
ISTANCE"," "
27 DATA "COMMON MODE REJECTION RATIO","LARGE SIGNAL VOLTAGE GAIN
","OUTPUT VOLTAGE SWING","SLEW RATE","POWER SUPPLY REJECTION RAT
IO","POWER DISSIPATION"
3Ø PRINTTAB(15);"OPERATIONAL AMPLIFIER EVALUATION":PRINT
4Ø PRINT"THIS PROGRAM CREATES AND UPDATES AN OP-AMP DATA FILE NA
MED":PRINT"'PARTS'.   THIS FILE PROVIDES A DATA BASE FOR PARAMET
ER":PRINT"COMPARISONS WHICH WILL DETERMINE COMPONENT INTERCHAGEA
BILITY.":PRINT
7Ø INPUT"OPTIONS (1) PARTS FILE UPDATE (2) EVALUATE A PART (3) Q
UIT";A
8Ø IFA<1ORA>3THENPRINT:GOTO7Ø
9Ø ONAGOTO2ØØØ,4ØØØ,5ØØØ
2ØØØ CLS:PRINT:PRINT"PARTS FILE UPDATE. ENTER SPECIFICATIONS AS
REQUESTED.":PRINT"IF UNABLE TO DETERMINE A VALUE, ENTER Ø."
2ØØ5 PRINT
2Ø1Ø INPUT"WHAT IS THE PART ID ";F$
2Ø11 ONERRORGOTO2Ø15:OPEN "I",#1,"PARTS"
2Ø12 INPUT#1,A$,K(1),K(2),K(3),K(4),K(5),K(6),K(7),K(8),K(9),K(1
Ø),K(11),K(12),K(13),K(14),K(15),K(16):IFF$=A$THENPRINT:PRINTF$;
" IS ALREADY IN THE PARTS FILE.":PRINT:CLOSE 1:GOTO7Ø:ELSEIFEOF(
1)THENCLOSE1:GOTO2Ø15:ELSE2Ø12
2Ø15 PRINT
2Ø2Ø FORX=1TO16
2Ø3Ø IFX=2ORX=1ØGOTO2Ø5Ø
2Ø4Ø PRINTS$(X);" (";U$(X);") ";:INPUTA(X)
2Ø5Ø NEXTX
218Ø FORX=1TO16:K(X)=A(X):NEXTX
219Ø OPEN "E",#1,"PARTS"
22ØØ PRINT#1,F$;",";K(1);K(2);K(3);K(4);K(5);K(6);K(7);K(8);K(9)
;K(1Ø);K(11);K(12);K(13);K(14);K(15);K(16)
221Ø CLOSE 1
222Ø PRINT:PRINT"PARTS FILE HAS BEEN UPDATED TO INCLUDE ";F$
223Ø PRINT:GOTO7Ø
224Ø PRINT"DISK ERROR - RESTART NECESSARY":RESUME7Ø
4ØØØ CLS:PRINT:PRINT"OP-AMP COMPONENT EVALUATION."
4Ø1Ø PRINT:INPUT"WHICH OP-AMP DO YOU WISH TO EVALUATE AGAINST";A
$
4Ø2Ø OPEN "I",#1,"PARTS"
4Ø3Ø INPUT#1,F$,K(1),K(2),K(3),K(4),K(5),K(6),K(7),K(8),K(9),K(1
Ø),K(11),K(12),K(13),K(14),K(15),K(16)
4Ø31 IFA$=F$THEN4Ø38
4Ø32 IF EOF(1)THEN4Ø34ELSEGOTO4Ø3Ø
4Ø34 CLOSE 1:PRINT:PRINT"UNABLE TO FIND ";A$;". TRY AGAIN. ":PRI
NT:GOTO7Ø
4Ø38 CLOSE 1
4Ø4Ø PRINT:PRINT"ENTER THE SPECS OF THE OP-AMP BEING EVALUATED A
S REQUESTED."
4Ø45 PRINT:INPUT"WHAT IS THE ID OF THE OP-AMP BEING EVALUATED ";
X$
4Ø5Ø FORX=1TO16
4Ø55 IFX=2ORX=1ØGOTO4Ø65
4Ø6Ø PRINTS$(X);" (";U$(X);") ";:INPUTU(X)
4Ø65 NEXTX:PRINT
4Ø75 IFK(1)=ØTHENPRINT"NO DATA FOR ";S$(1):GOTO4Ø95
4Ø8Ø IFU(1)<K(1)THENPRINT"FAILS ";S$(1)
4Ø95 IFK(3)=ØTHENPRINT"NO DATA FOR ";S$(3):GOTO41Ø5
41ØØ IFU(3)<K(3)THENPRINT"FAILS ";S$(3)
41Ø5 IFK(4)=ØTHENPRINT"NO DATA FOR ";S$(4):GOTO4115
411Ø IFU(4)<K(4)THENPRINT"FAILS ";S$(4)
4115 IFK(5)=ØTHENPRINT"NO DATA FOR ";S$(5):GOTO4125
412Ø IFU(5)>K(5)THENPRINT"FAILS ";S$(5)
4125 IFK(6)=ØTHENPRINT"NO DATA FOR ";S$(6):GOTO4135
413Ø IFU(6)>K(6)THENPRINT"FAILS ";S$(6)
4135 IFK(7)=ØTHENPRINT"NO DATA FOR ";S$(7):GOTO4145
414Ø IFU(7)>K(7)THENPRINT"FAILS ";S$(7)
4145 IFK(8)=ØTHENPRINT"NO DATA FOR ";S$(8):GOTO4155
415Ø IFU(8)>K(8)THENPRINT"FAILS ";S$(8)
4155 IFK(9)=ØTHENPRINT"NO DATA FOR ";S$(9):GOTO4175
416Ø IFU(9)<K(9)THENPRINT"FAILS ";S$(9)
4175 IFK(11)=ØTHENPRINT"NO DATA FOR ";S$(11):GOTO4185
418Ø IFU(11)<K(11)THENPRINT"FAILS ";S$(11)
4185 IFK(12)=ØTHENPRINT"NO DATA FOR ";S$(12):GOTO4195
419Ø IFU(12)<K(12)THENPRINT"FAILS ";S$(12)
4195 IFK(13)=ØTHENPRINT"NO DATA FOR ";S$(13):GOTO42Ø5
42ØØ IFU(13)<K(13)THENPRINT"FAILS ";S$(13)
42Ø5 IFK(14)=ØTHENPRINT"NO DATA FOR ";S$(14):GOTO4215
421Ø IFU(14)<K(14)THENPRINT"FAILS ";S$(14)
4215 IFK(15)=ØTHENPRINT"NO DATA FOR ";S$(15):GOTO4225
422Ø IFU(15)<K(15)THENPRINT"FAILS ";S$(15)
4225 IFK(16)=ØTHENPRINT"NO DATA FOR ";S$(16):GOTO4235
423Ø IFU(16)<K(16)THENPRINT"FAILS ";S$(16)
4235 PRINT:PRINT"EVALUATION COMPLETED. DISCREPANCIES AS NOTED."
424Ø INPUT"DO YOU WANT A COMPARISON PRINTOUT (Y/N) ";A$
425Ø IFA$="N"GOTO431Ø
426Ø IFA$<>"Y"THEN424Ø
4261 PRINT:INPUT"DO YOU WANT THE PRINTOUT ON (1) VIDEO (2) PRINT
ER";A:IFA=1THEN4265ELSE432Ø
4265 PRINT"PARAMETER";:PRINTTAB(3Ø);F$;:PRINTTAB(39);X$;:PRINTTA
B(5Ø);"UNITS"
427Ø FORX=1TO16
428Ø IFX=2ORX=1ØGOTO43ØØ
429Ø PRINTS$(X);:PRINTTAB(31);K(X);:PRINTTAB(38);U(X);:PRINTTAB(
48);U$(X)
43ØØ NEXTX
431Ø GOTO7Ø
432Ø A=PEEK(14312):IFA>128THEN438ØELSELPRINT"PARAMETER";:LPRINTT
AB(3Ø);F$;:LPRINTTAB(39);X$;:LPRINTTAB(5Ø);"UNITS"
433Ø FORX=1TO16
434Ø IFX=2ORX=1ØGOTO436Ø
435Ø LPRINTS$(X);:LPRINTTAB(31);K(X);:LPRINTTAB(38);U(X);:LPRINT
TAB(48);U$(X)
436Ø NEXTX
437Ø GOTO7Ø
438Ø INPUT"PRINTER NOT READY. ABORT? (Y/N)";A$:IFA$<>"N"THEN439Ø
ELSEINPUT"READY PRINTER AND PRESS <ENTER>";A$:GOTO432Ø
439Ø INPUT"PRINT TO VIDEO INSTEAD (Y/N)";A$:IFA$="Y"THEN4265ELSE
7Ø
5ØØØ CLS:END
```

✓See List of Advertisers on page 402

. . . of things that go bump in your keyboard.

Keyboard Incantations

Allen V. Robnett
Gallatin High School
Green Wave Drive
Gallatin, TN 37066

Ghost stories are usually just for thrills, but here's one which offers insights, too. A good ghost story depends on setting the scene, and this scene requires a TRS-80 keyboard.

So sit ye down, and turn the lamp down low.

The incantation must be carried out exactly or the spell will not work. Certain keys must be held down while others are struck. The held keys are indicated by a numeral below the key name. The number indicates which finger must be used for everything to come out right. The little finger of the left hand is numbered 1, and the little finger of the right hand, 10. An asterisk means the key is not to be held.

As the warlock of the story, I will make intermittent utterances. You must not release any held key until instructed to do so.

The spell involves keyboard-video I/O, so do this:

I	O	Space
(8)	(9)	(*)

Hmmm. Very strange. Maybe you ought to do that again. This time don't release the keys.

Let's take a bit from machine language and make a CD call to the wizard of Halloween:

C	D	H	Space
(3)	(2)	(7)	(*)

Don't let go! That's creepy—a syntax error and you never came close to the Enter key.

Don't turn back now. Strike:

T	Q
(*)	(*)

Release your right hand (I,O, and H) and press onward:

O	O	L	T
(*)	(*)	(*)	(*)

How's that for spelling?

Release and clear everything. We'll try a few more.

When you and your keyboard play a duet, you should let it do more of the work. Press (in succession) and hold: D U T

See what I mean? What's behind this mystery? Clear again and release. We'll ask the keyboard.

Press and hold the letters: C L U S

OK, release and relax.

Breaking the Spell

Of course, the parlor tricks we've done here are just for fun, but they reveal something about the way the keyboard works. Multiple key presses can be useful. Shifted characters are a good example, as are the 123, DFG, or JKL commands used in some well-known utilities.

Many sources refer to addresses 3800H through 3BFFH (14336 through 15359) as keyboard memory. Maybe it should be called pseudo memory. The first byte of the address (38, 39, 3A, or 3B) simply connects the keyboard output lines to the microprocessor. The second byte of the address, which ranges between 00 and FF, selects the key row to be read. For discussion or scanning purposes, the keys are arranged in an 8 × 8 matrix (or map) as shown in Fig. 1. It is convenient to call one matrix index R (for row) and the other C (for column). This description reflects the electrical connections but is not related to the physical rows of the keyboard.

Each key is simply a switch which, when pressed, will connect its row conductor to its column conductor. Each row is addressed by one bit of the low byte of the memory address. Hence, 3801H has bit 0 set ($2^0 = 1$), and thus addresses row 0.

The row addresses, are as given in Fig. 2.

In scanning the keyboard for input, ROM simply addresses each row in turn, thereby sending a signal to the row, and looks to see which column (if any) is activated. The eight columns become the eight bits of the output data byte. When a single key is pressed, the data byte will have only one bit set. The decimal value of the byte must be either 1,2,4,8,16,32,64, or 128 (decimal). It is a simple matter for ROM to figure out which

	C0	C1	C2	C3	C4	C5	C6	C7
R0	@	A	B	C	D	E	F	G
R1	H	I	J	K	L	M	N	O
R2	P	Q	R	S	T	U	V	W
R3	X	Y	Z					
R4	0	1	2	3	4	5	6	7
R5	8	9	:	;	,	-	.	/
R6	ENT	CLR	BRK	↑	↓	←	→	SPC
R7	SHIFT							

Fig. 1 Keyboard Matrix

Keyboard Row	Keyboard Address	
	Hex	Decimal
0	3801	14337
1	3802	14338
2	3804	14340
3	3808	14344
4	3810	14352
5	3820	14368
6	3840	14400
7	3880	14464

Fig. 2. Keyboard Memory Addresses

See List of Advertisers on page 402

"This bit of magic served as an incentive."

key it was. You try it.

Suppose you PEEK(3804H) and get a reply of 8 (decimal). Which key was being held down at the time? Address 3804H means ROM read row 2 ($2^2 = 4$). Data byte 8 ($= 2^3$) means the column of the key was three which makes the key "S".

So much for single key presses. Now try an experiment. Type in PRINT PEEK(14337), but before hitting Enter, press and hold A and B simultaneously while you backspace twice and enter (still holding "AB"). Why does the computer respond 6? Did it add the 2 for A to the 4 for B? Not exactly; it OR'd them.

```
          00000010   2
    OR    00000100   4
Yields    00000110   6
```

Isn't that the same as adding? No, look at:

```
          00000010   2
    OR    00000110   6
Yields    00000110   6
```

For our present purpose since each key in a row sets only one bit in the data byte, the effect will be the same as adding. If the row 0 data byte is 6 (for AB), how does ROM know it is supposed to print either A or B?

Glad you asked. ROM uses seven memory locations (4036H through 403CH) to store the data byte for each of the rows except the last (which contains only Shift). With each scan of the keyboard, the new byte for each row is compared with the byte from the previous scan. No action is taken if they are the same. If they are different, the previous byte is masked out and the result is output.

Note that Shift is treated differently since there is no reason to separate it from keys pressed simultaneously. But that's another story.

If you really manage to press the two keys close enough in time for ROM to see the events as simultaneous, then ROM will not be able to strip the extra code properly. It will happily take the lower of the bits remaining after the attempt.

Letter Substitutions

This still doesn't explain the weird letter substitutions or what happened to the addresses in between 3801H, 3802H, 3804H, 3808H, and so forth. The missing addresses represent combinations of rows, which might conceivably be useful if you wanted to use some exotic password or command.

The substitution sleight of hand stems from a sneak path in the electronic circuit. If you hold keys from a given column but in different rows, all of the columns activated in *any* row appear to be activated in *all* the involved rows.

For example, if you own a utility that is activated by holding 123, try holding ABC2 instead. It works. Looking at Fig. 1 you can see that ABC1 or ABC3 would have worked just as well. Believe it or not, even "Clear/Break/↑1" do the job.

A set of keys in one row defines the columns involved, and one more key links the rows together. Now you see why "I 0 Space" clears the screen. I and 0 activate columns C1 and C7. Space makes sure that when row 6 is scanned, it too appears to have C1 and C7 activated.

ROM knows the previous data byte for row 6 was 00, so it doesn't strip anything. It is quite content to take the lower of the two activated columns, which (for row 6) is Clear.

More Mysteries

This bit of magic served as an incentive to wend our way through the keyboard mechanism. It gives some insight into how wizards use multiple key codes, and it provides a little Halloween fun.

If you are interested in PEEKing at the keyboard memory, you must arrange it so that Enter is not the last key hit. Program Listing 1 will do this, formatting the display in lines 16-bytes wide and identifying the first address of each line.

You must press a key and hold it while the display is printing, or press several keys together.

If you are curious about the remainder of the keyboard memory (3900H through 3BFFH), insert lines

```
35 For K = 1 TO 4 : PRINT
105 NEXT K
```

You will find that the next three pages are identical to the first page. They exist only because of a shortcut in the row-addressing scheme.

If you are intrigued by the ROM mechanism in stripping off previously held keys, try disassembling 03E3H through 03F9 in ROM. It's only 23 bytes.

Ask your computer what turns it on. Hold down in this order: I F M N Space 0.■

```
10 DEFINT A-Z
20 K$ = INKEY$ : IF K$ = " " THEN 20
30 M = 14336: I = M: N = 14351
40 FOR J = 1 TO 16 : PRINT
50 PRINT I;
60 FOR I = M TO N
70 PRINT PEEK (I);
80 NEXT I
100 M = M + 16: N = N + 16
110 NEXT J
120 GOTO 20
```

Program Listing. Active Keyboard Display

A machine-language disk fixer.

Purezzap

Purezzap
Computer Consultants
Downey, CA

W.R. Stanley
City Rte. 5
204 Avery Lane
La Grange, GA 30240

Just as there are monitor programs available for cassette users, there are several monitors that allow the disk user to analyze and modify programs on disk.

Two disk-based monitors that come to mind are Superzap by Apparat, and Purezzap by Computer Consultants.

Superzap is loaded and run under Disk Basic, and some of its functions are slowed by dependence upon the Basic interpretive process. This in no way detracts from the technical accuracy and wide capabilities of Superzap.

Purezzap is a machine language program that takes much less time to bring up, and its overall operation seems to be smoother because of the smaller time lags encountered.

For the one-drive owner, the entire program should be initially loaded into RAM so that a program of interest on another disk can be manipulated by inserting that disk into the same single drive.

The monitor program should allow the user to specify any desired track and sector location to be examined, instead of always starting at track 0, sector 0 for all functions. This will save a lot of wear and tear on both the disk drive mechanism and the user's patience.

The monitor should also permit examination of an entire sector at a time, with a pleasing and functional format displayed on the video screen. Cursor control should be versatile so that a particular byte displayed can be specifically designated for modification as necessary.

The user should be able to step the drive (and display) in either direction from the current sector location to an adjacent sector without returning to a menu and reentering a whole string of commands and options. This saves considerable time in examining the many sectors storing a lengthy program.

Since we might also be interested in making minor changes to Basic programs on disk, an ASCII-equivalent dump of the 256 hexadecimal bytes in the sector display is of great help in locating specific program elements.

This ASCII display assists in analyzing and locating certain subroutines in a lengthy machine language program because many complicated programs use ASCII phrases for menu and error message displays.

We want to not only transfer disk data to the screen (and RAM), but also have the capability of modifying specific bytes and writing the modified sector back to exactly the same disk location from where the sector data was read out. There would be no point otherwise in attempting to modify bytes in a sector.

Purezzap

Purezzap comes on a formatted disk which contains only the monitor program itself, a directory track and a boot. No DOS is on the disk; you must transfer Purezzap to a disk containing at least a minimal disk operating system. The monitor program is furnished in this fashion to protect the copyrights of others regarding DOS programs.

The documentation furnished with Purezzap consists of a single page that explains the functions of the commands and options included in the menu displayed on the monitor screen.

When Purezzap is loaded from disk, it takes control of the computer. That disk may then be removed to make room for the disk to be examined. This makes it nice for those of us who have only one drive. The program is fully capable of functioning with two drives, however, with the active drive designated by the user.

When the program is brought up, a comprehensive menu is displayed. You can then choose any of several options such as reading a sector from disk, comparing two designated sectors, purifying the directory, zeroing unallocated sectors, killing all system files to derive a formatted disk, computing hash codes, or jumping from Purezzap back to DOS.

Read and Write

Several choices from the master menu will bring up sub-menus that get down to specifics in configuring program operations to arrive at desired results. For instance, pressing the R key (Read Disk Sector) will replace the master menu with a smaller menu asking you to specify the drive, track and sector to be dumped from disk into memory and be displayed on the video monitor screen.

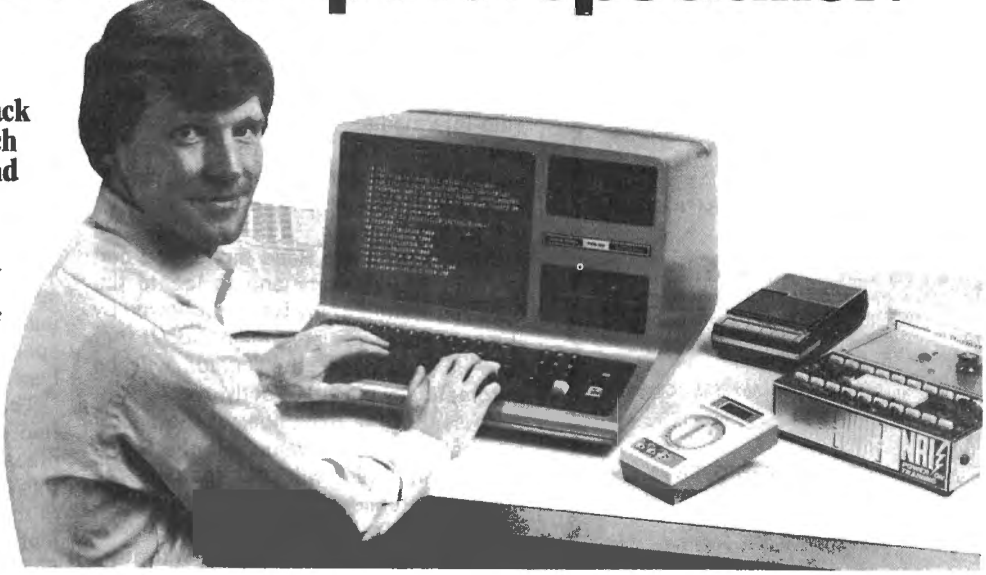
After the Read function has been completed, the full 256 bytes of the specified sector will be displayed in 16 lines of 16 bytes each. The ASCII equivalent of each hex byte in that line will be to the right of each line.

Along the left side of the screen you will see the current drive, track and sector numbers. Also displayed is the relative byte number corresponding to the current cursor location.

In addition, you will see a non-destructive cursor blinking at the zero relative byte in the current sector. You will also see a blinking cursor in the ASCII equivalent segment of the display. As the main cursor is moved around on the screen, the ASCII cursor is also moved to the corresponding relative location.

Let's say that you have examined the dump of this sector and see the byte that you wish to change. Simply use the arrow keys on the keyboard to position the cursor over that byte. Then press the M (Modify) key. The size of the cursor will change, indicating that Purezzap is now in

Now NRI takes you inside the new TRS-80 Model III microcomputer to train you at home as the new breed of computer specialist!



NRI teams up with Radio Shack advanced technology to teach you how to use, program and service state-of-the-art microcomputers...

It's no longer enough to be just a programmer or a technician. With microcomputers moving into the fabric of our lives (over 250,000 of the TRS-80™ alone have been sold), interdisciplinary skills are demanded. And NRI can prepare you with the first course of its kind, covering the complete world of the microcomputer.

Learn At Home in Your Spare Time

With NRI training, the programmer gains practical knowledge of hardware, enabling him to design simpler, more effective programs. And, with advanced programming skills, the technician can test and debug systems quickly and easily.

Only NRI gives you both kinds of training with the convenience of home study. No classroom pressures, no night school, no gasoline wasted. You learn at your convenience, at your own pace. Yet you're always backed by the NRI staff and your instructor, answering questions, giving you guidance, and available for special help if you need it.

Training includes new TRS-80 Model III microcomputer, 6-function LCD Beckman multimeter, and the NRI Discovery Lab with hundreds of tests and experiments.

(TRS-80 is a trademark of the Radio Shack division of Tandy Corp.)

You Get Your Own Computer to Learn On and Keep

NRI training is hands-on training, with practical experiments and demonstrations as the very foundation of your knowledge. You don't just program your computer, you go inside it...watch how circuits interact...interface with other systems...gain a real insight into its nature.

You also work with an advanced liquid crystal display hand-held multimeter and the NRI Discovery Lab, performing over 60 separate experiments. You learn troubleshooting procedures and gain greater understanding of the information. Both microcomputer and equipment come as part of your training for you to use and keep.

Send for Free Catalog... No Salesman Will Call

Get all the details on this exciting course in NRI's free, 100-page catalog. It shows all equipment, lesson outlines, and facts on other electronics courses such as Complete Communications with CB, TV/Audio and Video, Digital Electronics, and more. Send today, no salesman will ever bother you. Keep up with the latest technology as you learn on the latest model of the world's most popular computer. If coupon has been used, write to NRI Schools, 3939 Wisconsin Ave., Washington, D.C. 20016.

NRI Schools
McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue
Washington, D.C. 20016

All career courses approved under GI bill.
☐ Check for details

NO SALESMAN WILL CALL.
Please check for one free catalog only.

- ☐ Computer Electronics including Microcomputers
- ☐ Color TV, Audio, and Video System Servicing
- ☐ Electronics Design Technology
- ☐ Digital Electronics
- ☐ Communications Electronics • FCC Licenses • Mobile CB • Aircraft • Marine
- ☐ Basic Electronics
- ☐ Small Engine Servicing
- ☐ Appliance Servicing
- ☐ Automotive Servicing
- ☐ Auto Air Conditioning
- ☐ Air Conditioning, Heating, Refrigeration, & Solar Technology
- ☐ Building Construction

Name (Please Print) Age

Street

City/State/Zip

Accredited by the Accrediting Commission of the National Home Study Council #179-091

"You can get in trouble very fast by zapping bytes on a disk without knowing what you are doing."

the Modify mode.

Type in your desired byte value at the current cursor position. The byte display at the current cursor location will change as you enter the new value. If this is the only change to be made to data in this sector, press W (Write).

Another sub-menu will now appear, offering you, among other things, the option of writing the edited sector back to disk. This intermediate step reduces errors in writing a modified sector to disk because you must request a Write operation on two separate occasions before that operation will actually occur.

While still in the Read mode, you can use the Search command to locate the byte you want. Simply type S, followed by the byte value you are looking for. If that byte is displayed, the cursor will be automatically placed directly over it.

To search the remainder of the sector for that same byte value, hit the S key again. The cursor will then indicate the next occurrence of the byte. When the search function turns up no further locations for that byte, the cursor will automatically return to the zero-relative byte location.

Directory Cleanup

Purifying the directory consists of zeroing out all killed directory entries. This makes it easier to find the entry data for the program of interest.

Zeroing unallocated sectors is helpful for transferring programs back and forth between locations on one or two disks in the modification process. You usually end up with the same sector data in two or more locations on the same disk, but the data in only one sector is valid. The other sector data is still accessible by the monitor because only the directory entry is modified when a kill is made; the sector data itself is unchanged.

This can be misleading to the unwary because, if Purezzap is used to hunt all over the disk for sector data, you have a good chance of seizing upon a dead sector. The data in that sector looks just as you expect it to, so you go to great lengths to modify it.

After that sector is written back to disk and the main program is run again, it will appear that your modifications were unsuccessful. This minor tragedy can be alleviated if all unused sectors on that disk were zeroed.

Hash code computation is a mystery to most disk zappers, but it is necessary to derive or verify the proper hash code when reconstructing the directory for a clobbered file. This code is based on the values of the hex numbers representing the file name in a given directory entry, and it is stored in a specific position in the Hit table in the directory.

The hash code is a pointer to both the location and validity of a specific entry in one of the following sectors in the directory track. If the code is improper or missing, the computer will not be able to find the file you request it to load. Purezzap will generate and display the proper hash code for any filespec you type in.

Other Functions

Sector comparison using Purezzap is also very useful, especially when you are using two drives to reconstruct damaged sectors. This function still works just fine using one drive, though, because you can specify the drive, track and sector numbers for the comparison operation.

This is initiated by Reading the specified sector as before. Then the C key is pressed, bringing up another sub-menu. At this point you designate the exact location of the sector to be compared with the one just displayed. The data from the second sector is loaded into memory, and the comparison starts.

The relative byte number of the first occurrence of a byte value difference between the two sectors being compared is displayed, along with the values of those two bytes in their respective sectors. Hitting the C key again will continue the comparison until the next difference is encountered.

The kill function in Purezzap will change a system diskette to a formatted data diskette by killing all system files such as the SYSx/SYS, Basic and Format programs that are usually on a full DOS disk. The directory track and the boot sector are still available for use. Unchanged are all user programs and files present at the time of kill.

Pressing J will jump from the Purezzap master menu to the DOS currently in the computer, returning you to the DOS Ready prompt.

Return from any function of Purezzap back to the master menu is done simply by typing X. This is not only for recovery from an incorrectly entered command or parameter, but returns you to familiar ground in case you and the progam get into foreign territory.

Further Thoughts

If you don't already realize it, you can get in trouble very fast by zapping bytes on a disk without having at least a general idea of what you are doing. A publication of immense value to the neophyte is Harv Pennington's *TRS-80 Disk and Other Mysteries.* ■

✓See List of Advertisers on page 402

Say, disk users, here's a sexy sector inspector.

Simplezap

David Safford
887 Linda Lee Circle
Lemoore, CA 93245

This simple Basic program allows direct sector input, examination, modification and output. To understand how the program works, you need some elementary disk theory.

A floppy disk is organized into 35 concentric tracks, and each track is divided into 10 sectors. Each sector contains a header which tells which track and sector it is, followed by 256 bytes of data. Any DOS must have routines that read and write these 256 byte sectors directly. The higher level DOS functions take these physical records and translate them to logical records or files as necessary.

All current Model 1 versions of TRSDOS locate these routines on memory page 4600–46FF (hex). Memory page 4600H is loaded from track 0 sector 7 during bootstrap, which is why any change must be made to this critical sector. Page 4600H contains all the subroutines to select a disk and start its motor, seek the required track, then read or write a specified sector. In assembly language, accessing these routines is extremely easy. The format is:

C = Drive # (00-03 H)
D = Track # (00-23 H)
E = Sector # (00-09 H)
HL→256 byte RAM buffer
CALL 46DD for read sector to buffer
CALL 46E6 for write sector from buffer

My Basic program simply POKEs the machine language routine into memory, calls it with the USR function, and then calls Debug to examine and modify the sectors in memory. After returning to Basic from Debug, the program can use the machine language routine to write the sector back to disk.

How to Use It

- It is not necessary to reserve memory for the buffer—the program uses so little memory that judicious selection of the buffer address (i.e., 7500H-7A00H) will keep it clear of the Basic program and the processor stack even in a 16K machine.
- The program first prompts for the 256-byte buffer address. The program will always locate the buffer at the start of a page address; if you input 8180H, the buffer will actually be located at 8100H. This allows Debug to display the entire sector as one full page of memory.
- The "Drive, Track, Sector" prompt asks for decimal inputs—drive (0-4), track (0-35), sector (0-9). Once entered, the computer loads the sector to the buffer location and calls Debug.
- After examining and modifying the sector with standard Debug commands, type "G" (enter) to return to Basic.
- The program will then ask if you want to write the buffer contents to disk. Entering "E" will stop the program and return to Command Level Basic. This is the only way to exit since the break key will only execute Debug. Entering "Y" will write the sector; any other track sector prompt. Any entry other than "Y" will jump the program back to the drive, track, sector prompt without writing the sector.
- Once the program and Debug have been called into memory, any disk can be substituted in drive 0 for examination or modification as long as they are formatted. Therefore, even single drive systems can use the program to restore bad directories or otherwise defective disks. The only irreparable defect is one in which the disk interface chip cannot find or read the sector header. If the interface cannot positively identify the requested sector by its header, it will lock up.

This program makes sector I/O on a TRSDOS system easy. Since the program is in Basic, modifications are relatively simple to make. ■

```
10 DATA14,0,22,0,30,0,33,0,126,195,221,70
20 H$="0123456789ABCDEF"
30 CLS:PRINT"SECTOR I/O PROGRAM"
40 INPUT"ENTER BUFFER LOCATION (HEX)";A$
50 A=0:FORI=1TO16:FORJ=1TO4
60 IFMID$(A$,J,1) =MID$ (H$,I,1) THENA=A+(I-1)*16[(4-J)
70 NEXTJ:NEXTI:A=INT(A+.5):C=INT(A/256+.5):IFA>32767THENA=A-6553
6
80 FORI=0TO11:READB:POKE(A+256+I),B:NEXTI
90 DEFUSR1=A+256:POKEA+264,C
100 CLS:INPUT"ENTER DRIVE(0-4),TRACK(0-34),SECTOR(0-9)";D,T,S
110 POKEA+257,D:POKEA+259,T:POKEA+261,S
120 PRINTUSR1(0):CLS
130 CMD"D"
140 CLS:"INPUT"OUTPUT UPDATED SECTOR(Y/N/E)";A$="E" THEN STOP
150 IFA$<>"Y"THEN100 ELSE IF T=17 THEN I=239 ELSE I=230
160 POKEA+266,I:PRINTUSR1(0) :CLS:POKEA+266,221:GOTO100
```

Program Listing

"Sometimes a product is marketed that is of such good quality and value that it soon becomes the standard in its class. Orchestra-80 is this kind of program.
Jim Heid
80 Microcomputing
May 1981, p. 30

Orchestra-80™
MUSIC SYNTHESIZER
$79.95
plus $2.00 postage and handling
CA residents add 6% sales tax
MasterCard and Visa accepted

Free Downloading of Orchestra-80/85 Files Now Available!

Photo-80	(201) 790-6795	Connection-80	(516) 588-5836
Forum-80	(201) 688-7117	Communique-80	(201) 992-4847
Bullet-80	(203) 744-4644	Forum-80	(303) 341-0636
Bullet-80	(216) 724-1963	Innov-80	(516) 482-8491
Connection-80	(415) 651-4147	Bullet-80	(408) 241-0769
Bullet-80	(714) 359-3189	Micronet R ORCH80<70110,503>	

"The Orchestra-80 and 85 systems are outstanding achievements in cost vs. performance in the music synthesizer area. They produce a fine sound, and program so easily that novices have little trouble creating music with their first efforts."
George Somers
Pianist, and former Juilliard instructor

SYNTHESIZE MUSIC AND PERCUSSION IN STEREO!

- STEREO—Separation by instrument! For example, play trumpet and oboe through channel A, clarinet and organ through channel B. You can switch instruments from channel to channel at any time!
- PERCUSSION—Now you can add a wide range of percussive sounds and special effects to your music!
- Existing Orchestra-80 files load and play in stereo automatically!
- Optional 5th Voice for use with speed-up mods!
- New editing features and commands!
- Plugs into any 16k Model I keyboard or expansion interface without voiding warranty!
- Includes tape and disk versions on cassette, 4 sample music files, manual, and fully assembled and tested printed circuit board!

Orchestra-85™
$129.95
plus $2.00 postage and handling
($5 for overseas airmail)
CA residents add 6% sales tax
MasterCard and Visa accepted

WRITE FOR INFORMATION ON:
- **Upgrading**—Orchestra-80 to Orchestra-85 Switch to Stereo!
- **Greatest Hits**—Volumes I and II. All new music! Ready to load and play!
- **Orchestra-90**™—Model III version of Orchestra-85 Available soon!
- **Orchestra-80**™—The Original! $79.95

Software Affair
858 Rubis Drive, Sunnyvale, CA 94087
(408) 295-9195
✓356

This mod rolls 16K of RAM over your ROM operating system.

ROM Roll-Over

Geary Kelch
23 Mapledale Avenue
Glen Burnie, MD 21061

If you are concerned with machine-language programming or would like to add 16K or Random Access Memory to your TRS-80 without buying an expansion interface, you should be interested in this hardware modification. It is straightforward, inexpensive, and takes less than eight hours to build and debug. And it may give you a new perspective on your TRS-80's basic operating system, whether you have a Level I or a Level II machine.

By adding four chips and a 16K bank of 4116 chips whose address can be externally relocated anywhere in memory, you can replace your TRS-80's ROM with more versatile RAM memory. With this modification you can overlay the TRS-80's ROM operating system with your modified version, or with a totally rewritten one contained in the Random Access Memory. The existing ROM is taken out of the circuit when this overlay is performed, and system control passes to the program contained in the added 16K memory block. Then you are able to dynamically change or replace part or all of the TRS-80's ROM-driven operating system.

Let me explain how this operation works. First, the TRS-80 has its operating system and Basic interpreter in Read Only Memory occupying the lowest memory addresses. The Level I takes up the first 4K of memory space from address 0000 to 0FFF (hex). Since this is ROM-type memory, its contents are unchangeable. The same is true for the Level II system, except it occupies the first 12K or memory address range 0000 to 2FFF(hex). This modification can replace the inflexible Read Only Memory with the preferable Random Access Memory. The simplified logic diagram in Fig. 1 shows how the chip select circuit can make this change. The circuit involves two 7400 chips, one 7406, one 7442, a single-pole-single-throw switch, and a bank of eight 4116 RAM chips, at a cost of less than $40.

The rest of the circuit is used to generate two chip select signals—one for low memory (0000–3FFF), the other for high memory. The high memory address can be any of three remaining 16K blocks, but the circuit shown in Fig. 3 is addressing locations 8000–BFFF (hex) as high memory. Also, since the entire layout can be wired in the keyboard, it is not necessary to have an expansion box. That is what makes it a great way to add an extra 16K to your machine at about the cost of the memory chips alone. You will, however, void any warranty on the system.

The trick to making this change easily is in adding the eight memory chips to the circuit with as few lines as possible. It can be done by soldering the new 4116 chips pin-for-pin directly to the existing memory, piggyback style as in Fig. 2. This should be done using a low-power iron and soldering only a few pins at a time. The memory chips must be removed from the sockets before any soldering is started. Pin 15 of the upper chip in the pair should not be soldered to the corresponding pin of the lower chip, since this input will be used as the chip select for the 16K bank. These pins will be connected to the chip select logic generated by the four gates comprising the rest of the circuit. Once all the memory pairs have been soldered, with the exception of each pin 15, they can be replaced in the sockets. Make sure they are in the proper orientation.

The chip select circuitry that will feed pin 15 of the added chips is shown in Fig. 3. The pins used are noted, but if you have a Transistor-transistor Logic handbook, you may want to change the wiring slightly to make it easier for yourself. The logic, however, is in its simplest form and the gate count cannot be reduced. Location of the

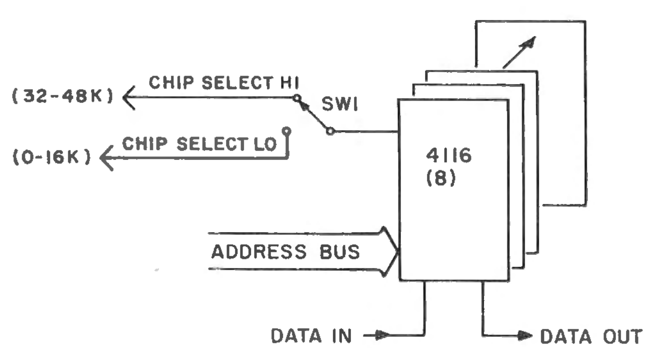


Fig. 1. Chip select circuitry for relocatable 16K bank of memory. Switch SW1 selects one of two chip select signals determining the address of the RAM memory block.

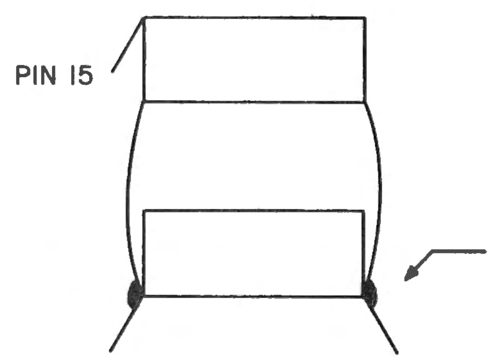
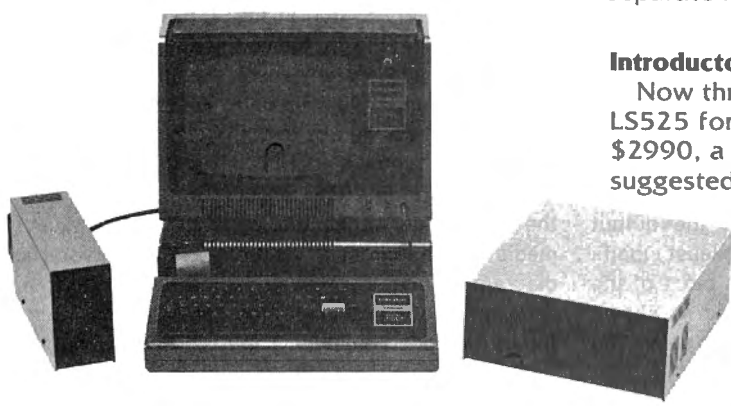


Fig. 2. Pin 15 used for chip select of upper bank, bent away from others to avoid shorting to pin 15 of lower chip.

THE NEW LAREDO LS525. WHEN YOU'RE READY TO GET DOWN TO BUSINESS.

Laredo Systems, Inc. introduces the LS525 five megabyte rigid disk memory system for TRS80 Models I and III. Laredo's LS525 rounds up a Seagate ST506 Drive, LDOS Operating System by Logical Systems, and Laredo's own LSI



500 Series Controller into a single-board controller that emulates the famed IBM 3370 disk system, complete with:

• Improved read/write/seek access time through full block buffering and variably tuned interleave.

• On-board data separator, micro-code diagnostics and write-precompensation.

• Increased disk media reliability and data integrity with automatic defect block allocation and extended ID fields.

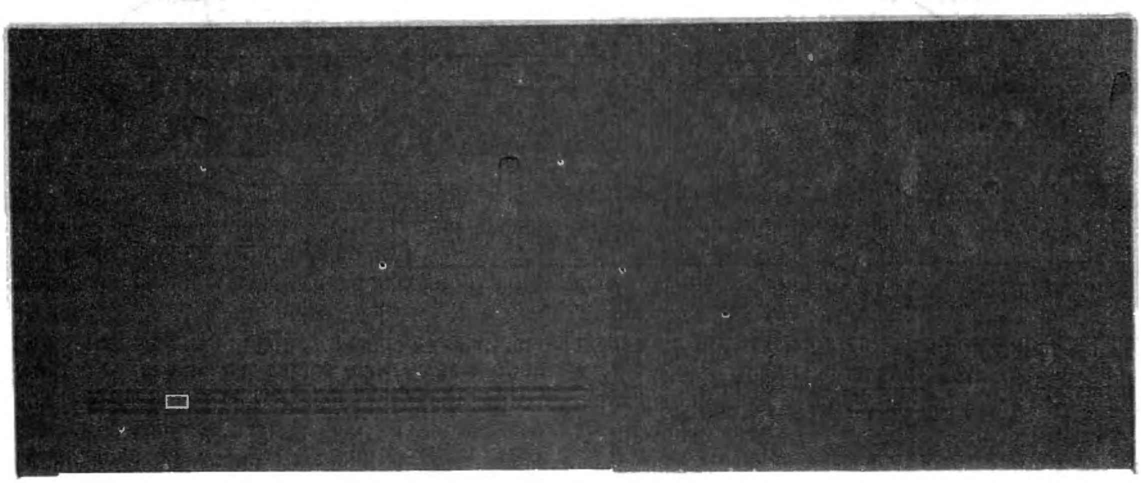
• Versatile CPU compatibility provided by separate host adaptor.

Introductory Offer To TRS80 Users.

Now through October 1, 1981 buy the LS525 for the OEM/dealer price of $2990, a savings of $760 off the usual suggested retail price of $3750.

LDOS Operating System with manual and diskette available for $149.

For more information about the LS525 Memory System and the LSI 500 Series Controller, contact: **Laredo Systems, Inc.**
2264 Calle de Luna, ✓26
Santa Clara, CA 95050
(408) 980-1888



laredo systems inc.

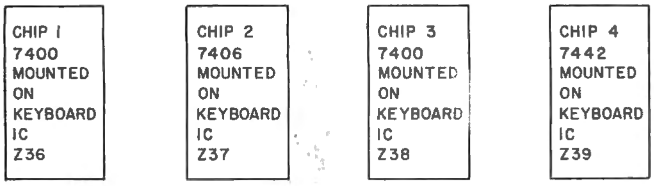


Fig. 3. By mounting the 74s on the back of integrated circuits Z36<->Z39, wiring complexity and lead length will be kept to a minimum.

chips and the method of wiring them is up to you. In my system everything was done as simply as possible to keep cost and lead length down. By inverting the chips and mounting them upside down on the backs of Z36–Z39 as shown in Fig. 4, your leads may be soldered directly to the integrated circuit pins. Mounting the chips in the position minimizes lead length and eliminates crossing wires which helps to keep down noise. If desired, a small circuit board may be mounted to hold the four integrated circuits.

Trouble Shooting

To make trouble-shooting easier, there are a few functions of the circuit that should be discussed in detail. The first deals with the memory address selection. As mentioned before, the add-on memory may be switched from a low to a high address via an external switch, SW1. The 7442 gate does the address decoding enabling chip selection. It is a 4-in-10-out binary decoder. By running A14 and A15 of the TRS-80 address bus to its two LSB inputs, you get 16K bank select signals at the output pins. The output's least significant bit, which is pin 1 of the chip, must be used as shown because it is used to select the RAM when it is in the low position. However, any of the other three chip select signals (pins 2, 3 or 4) may be used to select RAM when SW1 is in the high RAM position. Pin 2 of the 7442 would cause position 4000–7FFF to be used. Pin 3 selects addresses 8000–BFFF and pin 4 places the memory block at the top of storage or addresses C000–FFFF in hex. In the example, pin 3 was used, since it does not interfere with the existing memory in a 16K machine, and is contiguous with the original 16K. If your memory size is not the same you may want to move the add on memory address up or down. A 4-position rotary switch in the place of SW1 would enable you to move the address anywhere on a 16K boundary by running all four select lines out to the externally mounted switch.

There is another function of the chip select logic not immediately apparent in the diagram. This involves the deselection of a 4K block at the top of the added memory. This deselection takes place when the memory is located at address 0000–3FFF (when the switch is in the low position). The deselection is accomplished by ANDing A12, A13 and CS Lo. When A12 and A13 are both high and CS Lo is active, a blocking of the chip select signal occurs. This is to prevent the processor from accessing the address range 3000–3FFF of the add-on memory when it is occupying low memory. Hence, the effective range is the first 12K (the range of Level II ROMs), rather than the full 16K block. It is necessary to block the upper 4K selection from 3000–3FFF since this is the area used by the video display and the keyboard input. This chip select blocking is not implemented when SW1 is in the high position; therefore the full 16K range is accessible in the upper addresses. Finally, be sure the jumper from pin 5 of Z74 to five volts is in place. This activates the memory Data In buffers for addresses above 7FFF. Without this jumper, you

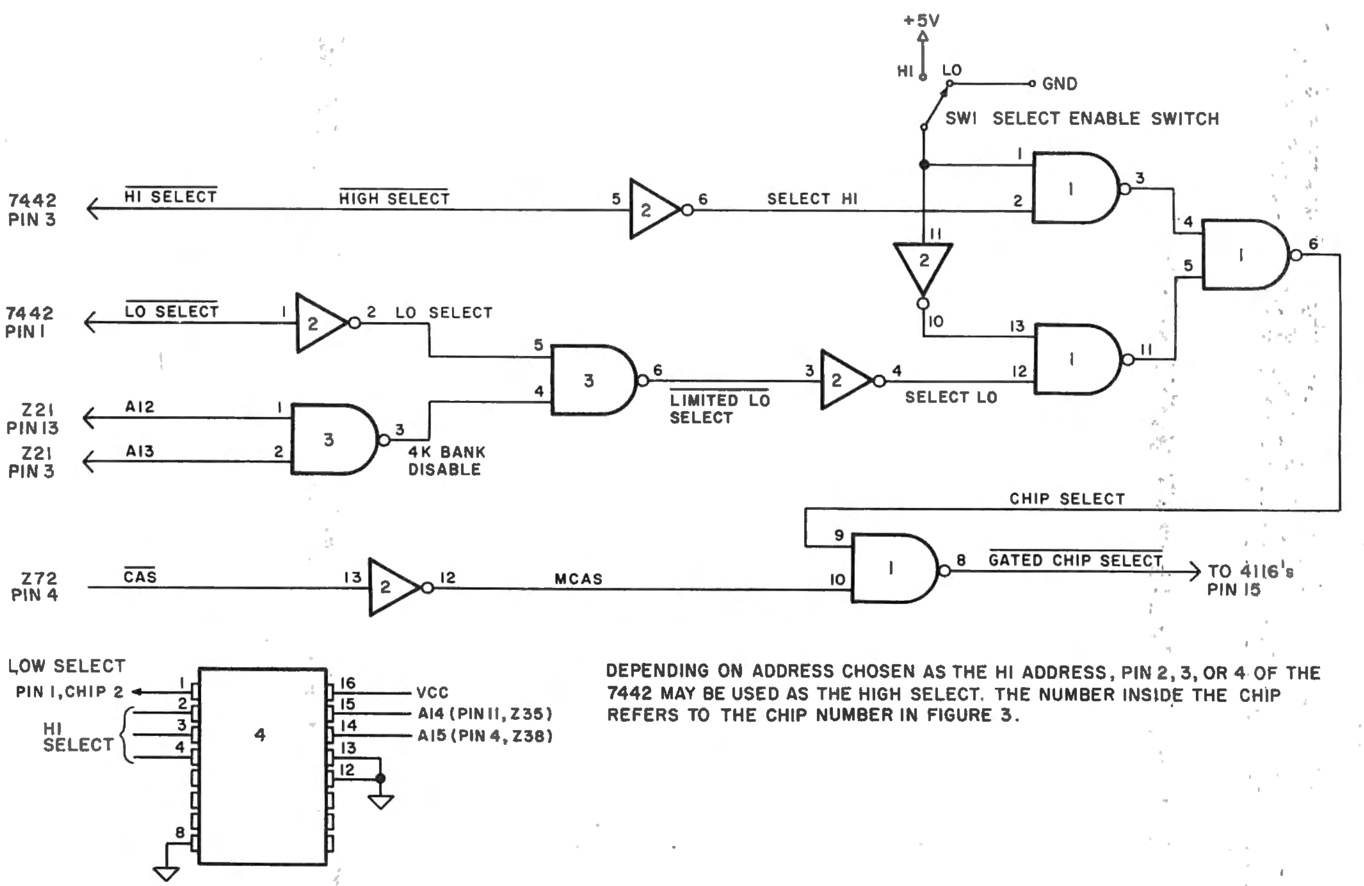


Fig. 4. Depending on what address is chosen as the HI address pin, two, three or four of the 7442 may be chosen as the high select. The number inside the chip refers to the chip number in Fig. 3.

✓See List of Advertisers on page 402

cannot read memory above the first 16K of useable RAM.

There is a short procedure to bring the system up for testing. With SW1 in the high RAM position, switch on the keyboard. The normal power on system response should be apparent. If there is no response, hit the Reset key. If there still is no correct response, toggle SW1 and try again. If this fails in resetting the system, there is an error in the chip-select circuit.

Address	Data
7000	21
7001	00
7002	80
7003	01
7004	00
7005	00
7006	0A
7007	77
7008	23
7009	03
700A	3E
700B	10
700C	B8
700D	CA
700E	91
700F	40
7010	C3
7011	06
7012	70

Table 1

Once the circuit is functioning, the T-Bug program or some other monitor routine allowing the writing of a machine-language loop must be loaded. When the prompt appears, manually load the program listed in Table 1, using the M verb of T-Bug. This short machine loop does a block memory move. It copies the contents of ROM into the added 16K RAM. If you are not using memory address 8000–BFFF as your high RAM position, addresses 7001 and 7002 of the program must be changed to reflect the starting address of the relocatable RAM (the least significant byte in address 7001 and the most significant byte in 7002). When the transfer is completed, control will pass back to T-Bug. Using the M verb again, examine locations 8000 to BFFF randomly to determine if the transfer has been made. What should appear is a copy of the ROM data starting at address 8000. Check this by comparing data at Oxyz to data at 8xyz. The data should be the same. In general, data at address Axyz should be the same as data at address Axyz + 8000 for the range of 0–2. Once this has been verified, switch SW1 to the low RAM position. Your operating system is now in place. Again, with the M verb, try altering location 0000 in memory to 00(hex). If this works, you have verified that the RAM resident operating system has replaced the ROM. You now have the ability to alter the entire operating system.

Here is a short example of what can be done with this new capability. With a Level 1 machine, after running a program from T-Bug, once the machine loop hangs there is no way to return to the T-Bug. The only way to regain control of the system is to reset and reload the T-Bug tape. This can be changed by altering two locations of the operating system. Place the entry point of T-Bug at address 0010,0011. Press the Reset key and the TRS-80 reset jump address control will return to T-Bug. Now if your machine program gets lost, you can Reset and return to T-Bug immediately. Any address may be placed at this restart location so control may be passed to a desired program with the Reset key.

With a little imagination, and some machine-language or assembly program software ability, you should find many uses for this modified operating system.■

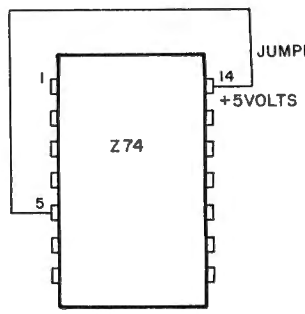


Fig. 5. This jumper must be installed to enable the data in buffers for memory addresses above 7FFF (Hex.).

MISOSYS MISOSYS MISOSYS MISOSYS MISOSYS MISOSYS MISOSYS MISOSYS MISOSYS MISOSYS MISOSYS MISOSYS

EDAS 3.5

EDAS is a sophisticated Editor and Assembler for TRS-80 I & III

- All text may be input in upper or lower case.
- Assemble directly from one or more disk files via *GET.
- Assemble to disk or memory
- Binary, octal, decimal, hex, and string constants; Multiple constants may be input on a single line.
- CMDFILE utility included.
- Conditional assembly support.
- Cross-Reference utility.
- DOS functions DIR, FREE, KILL, and LIST are supported.
- 14-character labels including special chars. "@", "$", ".", "?".
- Editor includes block move, global change, renumber, find.
- EDTASM & M-80 source files can be read or written.
- Expression evaluator supports +, −, *, /, MOD plus logical AND, OR, and XOR.
- Paged & titled listings with page numbers and date/time.
- PAGE, TITLE, SUBTTL, SPACE, & COM pseudo-ops supported.
- Set memory size, page prompt, JCL execution, Abort option.
- EDAS is supported with TRSDOS compatible DOSs (LDOS, VTOS).
- Price is $79.00 + $4.00. S&H.

MISOSYS SERIOUS SOFTWARE

MISOSYS MISOSYS MISOSYS MISOSYS MISOSYS

LDOS,Version 5.1 is the Ultimate in Operating Systems for the TRS-80 Models I and III. MISOSYS is your East Coast Headquarters for LDOS, the documented system! Version 5.1 is priced at $169.00 + $5.00 S&H per system. Deduct $50 if ordering both.

LDOS 5.1

DSMBLR

Complement your assembly language tools with this Z-80 disassembler which produces screen, printer, cassette, or disk file output. A two pass process provides SYMBOLS for 16-bit address and 8-bit relative references. EQUates & ORG are generated. Read SYSTEM programs & display load address range. $20 (DSMBLR I for non-disk use is $15).

EDAS, CMDFILE, DSMBLR Specify Mod I or III. Add $1 S&H+$.50 per program. VA residents add 4%.

VISA

MISOSYS - Dept. MU 3
5904 Edgehill Drive,
Alexandria, Virginia 22303
703-960-2998 MicroNET 70140,310
Dealers Inquiry Invited ✓221

MasterCard

DISKMOD

Turn your Editor Assembler into a disk package. This 32K patch modifies EDTASM for DOS operation. Features? Add full disk I/O, block move, global change, printer pagination, sorted symbol table, print memory utilization, enhance DEFM expansion, protect memory, and recover after BOOT. From within the EDTASM you will have DIR, KILL, & FREE. Upgrade your EDTASM! $20.

THE BØØKS

THE BOOKs must be a part of your tools. Volume I gives you access to all math operations in your Level II ROM. A symbol table of the entire machine noting over 500 addresses is included. Volume II tells you everything you wanted to know about the level II I/O—printer, keyboard, video, and cassette routines are fully explained. Each volume has a fully commented listing of all the routines discussed. THE BOOKS will save you hours of assembler programming. Each volume is priced at $14.95 or buy both for $24.95. Add $1.50 S&H per book.

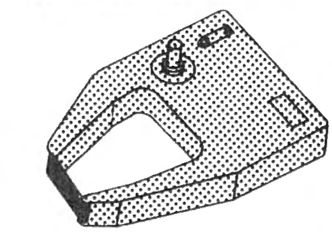
DISCOUNT PRINTER RIBBONS

40% OFF!! OR MORE!

Brand New, Top Quality, Exact Replacement Ribbons & Cartridges. These Ribbons Produce **Super Jet** Black Impressions and Ultra Reliable Print Life. They Are Delivered to Your Door Promptly for Much Less Than Most Retail Stores

★SPECIAL! BUY 10 and GET ONE FREE!

YOUR PRINTER	PACK SIZE	RETAIL LIST**	YOUR WHOLESALE PRICE		SIZE	COMMENTS	CAT ORDER #
ANADEX 9000 Series	1/pk	14.00 ea	14.00	(14.00 ea)	500"	Nylon Jet Blk	C-777
CENTRONICS 700–703, 737, 779	3/pk	18.95/3 pk	11.95/3 pk	(3.98 ea)	563" x 45'	Nylon Jet Blk	C-700
CENTRONICS 100, 101A, 102, 103, 300, 301, 306, 308, 330, 358, 398, 500, 501, 503, 508, 588, 620, 820.	3/pk	26.33/3 pk	17.55/3 pk	(5.85 ea)	1" x 108'	Nylon Jet Blk 5 mil High Speed	C-100
CENTRONICS 704-705	1/pk	16.95 ea	13.95/Giant Cart	(13.95 ea)	5 16" x 210'	Giant Cart	C-7045
DEC ½ x 40YD.	3/pk	17.77/3 pk	12.95/3 pk	(4.32 ea)	1 2" x 120'	Double Spools	R-600
DEC ½ x 60YD.	3/pk	20.12/3 pk	14.25/3 pk	(4.75 ea)	1 2" x 180'	Double Spools	R-644
DIABLO HYTYPE II (M/S BLK) HI YIELD. FITS 70 PRINTERS!	1/pk	9.31 ea	6.87 ea	(6.87 ea)	5 16" x 'High Yield'	300,000 plus imp	C-511
EPSON MX70/80	1/pk	16.00 ea	13.95/Cart	(13.95 ea)	500" x60	Nylon Jet Blk	C-522
IBM -"SILVER DOLLAR" Sys. 34, Sys. 32 MDLA, Series IMDL4974, 5256, 3287, 3770, 3771-3774, 4974, 5100, 5103, 5110, 5228, 5256, 5320MDLA	5/pk	5.80 ea	14.90/5 pk	(2.98 ea)	9 16" x 30'	Nylon Jet Blk	R-300
IBM - HARMONICA ½". SERIES I. MOD 4973/II, 3200, 3289, MOD 2.	3/pk	9.42 ea	20.85/3 pk	(6.95 ea)	1 2" x 108'	Nylon Jet Blk	C-350
NEC SPINWRITER	4/pk	23.40/3 cart	23.60/4 pk rb reload	(5.90 ea)	1 2" x 51'	Nylon/Ex Lng Life	R-400
QUME (FITS 80 PRINTER MODS)	3/pk	18.00/3 pk	13.95/3 pk	(4.65 ea)	1 4" x 310'	Multistrike Film	C-525
RADIO SHACK DAISY WHEEL II	1 pk	24.95/3pk	8.25	(8.25 ea)	250"	Mylar Multistrike	C-789
RADIO SHACK LPIII, LPV	one/pk	13.95/cart	8.95/Reload rib. only	(8.95 ea)	500" x 45'	Nylon Incl Instr	R-T3
RADIO SHACK LPII, LPIV	3/pk	18.95/3 pk	11.95/3 pk	(3.98 ea)	563" x 45'	Nylon Jet Blk	C-700
TELETYPE MOD 33, 28, 35, 37, 38, 88	10/pk	2.40 ea	13.90/10 pk	(1.39 ea)	1 2" x 36'	Nylon Jet Blk	R-450
WANG M/S. 5541W, WC. 5581, WD, 6581W, 2281W	1/pk	6.85 ea	5.95 ea	(5.95 ea)	5 16" x 393'	Multistrike Film	C-550



TERMS:

MINIMUM PURCHASE - **$20**
PAYMENT BY: C.O.D. (UPS), CHECK, MASTER CARD, OR VISA CHARGE CARD.
VOLUME DISCOUNTS:
20 - 50 PACKS **10%**
51 - 100 PACKS **15%**
***UNDER $20, ADD $5 HANDLING.**
****APPROX. RETAIL. PRICE VARIES.**

ANCIE LABORATORIES ✓461
9204 Baltimore Blvd. College Park, MD 20740
301-345-6000 (Wash. D.C Local)
301-792-2060 (Balto. MD Local)
800-638-0987 (National)

NAME______________________
ADDRESS______________________
CITY__________ STATE__________ ZIP__________

ANCIE Laboratories
9204 Baltimore Blvd.
College Park, MD 20740

301-345-6000 (Wash. D.C. Local)
301-792-2060 (Balto. MD Local)
800-638-0987 (National)

QTY ______ CAT.# ______ AMT. ______

TOTAL ______

☐ Check Enclosed
☐ C.O.D.
☐ VISA
☐ MASTER CHARGE
ACCT. # ______________
EXP. DATE ______________
MIN. ORDER $20
PRICES SUBJECT TO CHANGE

Fifteen new commands for serious T-Bug aficionados.

Newbug

Ken Marks Jr.
913 W. 62nd Street
Downers Grove, IL 60516

Soon after I had Radio Shack's T-Bug, I realized that its nine commands did not provide enough programming power for the serious machine-code programmer. Mr. Curtis' article "T-Bug for II" in the April 1980 issue of *80 Microcomputing* started me thinking that others may find my modifications useful. Because I added 15 new commands to T-Bug and for the sake of clarity in this article, I have named the new version NEWBUG. The following source code listing may be assembled using Radio Shack's Editor/Assembler.

NEWBUG

Once you have the object code of the NEWBUG patch, simply load T-Bug II (the version of T-Bug modified with Mr. Curtis' enhancements), and with the L command, load the patch. When the tape stops, the first difference you will notice is that the # prompt has become an *. This tells you that this is no longer T-Bug, but NEWBUG. Before you go any further, it would be wise to punch a copy of NEWBUG to tape. To do this, type P 4380 4C5C 4762 NEWBUG. If you use the V command to verify the dump, you will get an error message. This is because the input buffer is written to tape also, and when you go to verify the dump, the contents of this buffer will have changed causing the apparent error. You don't have to fully understand this, but there is usually no need to worry. If you are using T-Bug to enter the patch, type in the jump in line 27 of the listing last. This is the line that converts T-Bug into NEWBUG. If you type it in before the rest of the patch is in place, it will wreak havoc with the system.

The following is a short description of some of the uses of the new commands. Keep in mind that these are only examples and you should feel free to experiment with the commands where possible, but take care to heed any warnings given.

Clear

Occasionally the CRT screen gets cluttered up with alphanumeric or graphic characters. This can happen while debugging programs which display data or text on the screen. One way to clear the screen is to jump to Basic by executing a J 0072, hit the Clear key, and jump back to T-Bug. There is now a quicker and

Program Listing

```
                                  "NEWBUG" a patch to modify T-BUG
                                  written by Ken Marks Jr.   final copy 1/11/81
                00001 ;
                00002 ;
                00003 ;
                00004 ;
43A5            00005 BACK2     EQU     43A5H
43CD            00006 BACK      EQU     43CDH
4401            00007 NONE      EQU     4401H
4522            00008 OUTA      EQU     4522H
4532            00009 ASPC      EQU     4532H
453C            00010 CRS       EQU     453CH
457B            00011 SPC       EQU     457BH
4589            00012 INA       EQU     4589H
```

Program continues

Program continued

```
45A7           LST     EQU   45A7H
45C8           INKY    EQU   45C8H
4762           NEWBUG  EQU   4762H
4825           REGS    EQU   4825H
482D           FLAG    EQU   482DH
483D           CURSR   EQU   483DH
43DE                   ORG   43DEH         ;ADDRESS OF PROMPT
43DE 2A                DEFB  2AH           ;CHANGE TO *
4782                   ORG   4782H
4782 C28D48            JP    NZ,CLEAR      ;INSERT JUMP TO PATCH
4851                   ORG   4851H         ;START OF PATCH
4851 D9        HLIN    EXX                 ;THIS SUBROUTINE
4852 CD8945            CALL  INA           ;LOADS THE HL
4855 D9                EXX                 ;REGISTER WITH
4856 67                LD    H,A           ;AN INPUT FROM
4857 D9                EXX                 ;THE KEYBOARD
4858 CD8945            CALL  INA
485B D9                EXX
485C 6F                LD    L,A
485D C9                RET
485E C5        HLOUT   PUSH  BC            ;THIS SUBROUTINE
485F 7C                LD    A,H           ;OUTPUTS THE
4860 CD6C48            CALL  OUT           ;CONTENTS OF
4863 7D                LD    A,L           ;THE HL REGISTER
4864 CD6C48            CALL  OUT           ;& A SPACE
4867 CD7B45            CALL  SPC           ;AT THE CURRENT
486A C1                POP   BC            ;CURSOR POSITION
486B C9                RET
486C 47        OUT     LD    B,A           ;THIS SUBROUTINE
486D CB3F              SRL   A             ;OUTPUTS THE
486F CB3F              SRL   A             ;CONTENTS OF THE
4871 CB3F              SRL   A             ;A REGISTER
4873 CB3F              SRL   A             ;AT THE CURRENT
4875 D9                EXX                 ;CURSOR POSITION
4876 CD2245            CALL  OUTA
4879 D9                EXX
487A 78                LD    A,B
487B E60F              AND   0FH
487D D9                EXX
487E CD2245            CALL  OUTA
4881 D9                EXX
4882 C9                RET
4883 DF        ERR     RST   18H           ;COMPARE START AND END
4884 D8                RET   C             ;RETURN IF ALL OKAY
4885 2A3D48            LD    HL,(CURSR)    ;ELSE GET CURSOR POS.
4888 3645              LD    (HL),'E'      ;OUTPUT AN "E"
488A C3A543            JP    BACK2         ;AND ABORT
488D FE63      CLEAR   CP    63H           ;CHECK FOR <CLEAR>
488F 200A              JR    NZ,ARITHM     ;GO IF NOT PRESSED
4891 AF                XOR   A             ;CLEAR A
```

Program continues

neater approach. The Clear key works just as it did in Basic, clearing the full screen and returning the * prompt to the upper left-hand corner.

Hexadecimal Arithmetic

Have you ever wanted to calculate an absolute address from a jump relative displacement value? Have you ever found the need to add or subtract two hex numbers for one reason or another? Well, my hexadecimal math has never been great, and the fact that even slight mathematical errors are enough to "bomb" a machine-code program should point out the usefulness of this command. It has saved me much needless grief. Type A (for arithmetic) followed by any four digit hexadecimal number. Now specify either addition or subtraction by typing + or −. You must now give the computer a number to be added to or subtracted from the first, so enter a second four digit number. An = and the resulting hex value will be displayed.

Selective Block Change

While working on a program, I found the need for a quick way to search for all the occurrences of a particular byte and then change them to another value in a specified block of memory. So, out of necessity rose the block change com-

⸝See List of Advertisers on page 402

Program continued

```
4892 322240          LD    (4022H),A      ;TURN CURSOR OFF
4895 CDC901          CALL  1C9H           ;CLEAR SCREEN
4898 C3CD43          JP    BACK           ;AND RETURN
489B FE41    ARITHM  CP    'A'            ;CHECK FOR "A"
489D 203A            JR    NZ,CHANGE      ;GO IF NOT PRESSED
489F CD3245          CALL  ASPC           ;OUTPUT (A) + SPACE
48A2 CD5148          CALL  HLIN           ;GET 1st OPERAND
48A5 E5              PUSH  HL             ;SAVE IT
48A6 CD7B45          CALL  SPC            ;SPACE
48A9 CDC845  B1      CALL  INKY           ;LOAD A FROM KEYBOARD
48AC FE2B            CP    '+'            ;ADDITION?
48AE 2806            JR    Z,B2
48B0 FE3D            CP    '='            ;SUBTRACTION?
48B2 20F5            JR    NZ,B1          ;NO. TRY AGAIN
48B4 D610            SUB   10H            ;CORRECT TO "-"
48B6 F5      B2      PUSH  AF             ;SAVE OPERATOR
48B7 CD3245          CALL  ASPC           ;OUTPUT (A) + SPACE
48BA CD5148          CALL  HLIN           ;GET 2nd OPERAND
48BD E5              PUSH  HL             ;SAVE IT
48BE CD7B45          CALL  SPC            ;SPACE
48C1 3E3D            LD    A,'='
48C3 CD3245          CALL  ASPC           ;OUTPUT "=" + SPACE
48C6 D1              POP   DE             ;POP OPERAND #2
48C7 F1              POP   AF             ;POP OPERATOR
48C8 E1              POP   HL             ;POP OPERAND #1
48C9 FE2B            CP    '+'            ;ADD?
48CB 2805            JR    Z,PLUS         ;GO IF (A)="+"
48CD AF              XOR   A              ;CLEAR CARRY
48CE ED52            SBC   HL,DE          ;AND SUBTRACT
48D0 1801            JR    B3
48D2 19      PLUS    ADD   HL,DE          ;ADD
48D3 CD5E48  B3      CALL  HLOUT          ;DISPLAY RESULT
48D6 C3A543          JP    BACK2          ;RETURN
48D9 FE43    CHANGE  CP    'C'            ;CHECK FOR "C"
48DB 2042            JR    NZ,EXPAND      ;GO IF NOT PRESSED
48DD CD3245          CALL  ASPC           ;OUTPUT (A) + SPACE
48E0 CD5148          CALL  HLIN           ;GET START OF CHANGE
48E3 E5              PUSH  HL             ;SAVE IT
48E4 CD7B45          CALL  SPC            ;SPACE
48E7 CD5148          CALL  HLIN           ;GET END OF CHANGE
48EA E5              PUSH  HL             ;SAVE IT
48EB CD7B45          CALL  SPC            ;SPACE
48EE CD8945          CALL  INA            ;GET TARGET BYTE
48F1 F5              PUSH  AF             ;SAVE IT
48F2 CD7B45          CALL  SPC            ;SPACE
48F5 CD8945          CALL  INA            ;GET NEW BYTE
48F8 47              LD    B,A            ;NEW BYTE -> B
48F9 F1              POP   AF             ;POP TARGET BYTE
48FA E1              POP   HL             ;POP END
48FB D1              POP   DE             ;POP START
48FC F5              PUSH  AF             ;SAVE TARGET AGAIN
48FD DF              RST   18H            ;CHECK FOR START>END
48FE 3009            JR    NC,B4          ;GO IF NOT
4900 2A3D48          LD    HL,(CURSR)     ;ELSE OUTPUT
4903 3645            LD    (HL),'E'       ;"E" FOR ERROR
4905 F1              POP   AF             ;RESTORE STACK
4906 C3A543          JP    BACK2          ;RETURN
4909 AF      B4      XOR   A              ;CLEAR CARRY
490A ED52            SBC   HL,DE          ;COMPUTE COUNT
490C E5              PUSH  HL             ;SAVE IT
490D D5              PUSH  DE             ;PUSH ALL ONTO STACK
490E C5              PUSH  BC             ;TO SWITCH VALUES
490F D1              POP   DE             ;NEW BYTE -> D
4910 E1              POP   HL             ;START ->  HL
4911 C1              POP   BC             ;BYTE COUNT -> BC
4912 F1              POP   AF             ;TARGET BYTE -> A
4913 03              INC   BC             ;CORRECT BYTE COUNT
4914 03              INC   BC             ;BY ADDING 2
4915 EDB1    L1      CPIR                 ;SEARCH FOR TARGET BYTE
4917 E2A543          JP    PO,BACK2       ;RETURN IF SEARCH OVER
491A 2B              DEC   HL             ;NOW HL POINTS TO TARGET
491B 72              LD    (HL),D         ;CHANGE TARGET INTO NEW
491C 23              INC   HL             ;RESTORE TO NORMAL
491D 18F6            JR    L1             ;CONTINUE 'TILL DONE
491F FE45    EXPAND  CP    'E'            ;CHECK FOR "E"
4921 2044            JR    NZ,HEXDMP      ;GO IF NOT PRESSED
4923 214049          LD    HL,MES         ;POINT TO MESSAGE
4926 CD5349          CALL  OUTMES         ;OUTPUT
4929 0608            LD    B,8            ;(B) = BIT COUNT
492B 212D48  L2      LD    HL,FLAG        ;ADDRESS OF T-BUG FLAG
492E 3E30            LD    A,'0'          ;BIT (B) ASSUMED = 0
4930 CB06            RLC   (HL)           ;LEFTMOST BIT -> CARRY
4932 3001            JR    NC,B5          ;BRANCH IF BIT = 0
4934 3C              INC   A              ;NOW (A) IS ASCII "1"
```

Program continues

mand. As a quick and visible display of this command's operation, try the following example: type C followed by the starting and ending addresses of the block of memory you wish to change. For this demonstration, type 3C00 and 3FFF. These are the starting and ending addresses of the video display memory. Choose the addresses carefully or you may find (the hard way) that writing over the vector restarts (immediately following the screen memory), the area of memory in which NEWBUG resides, or any other dedicated block of memory, can be hazardous to the well being of any program in memory.

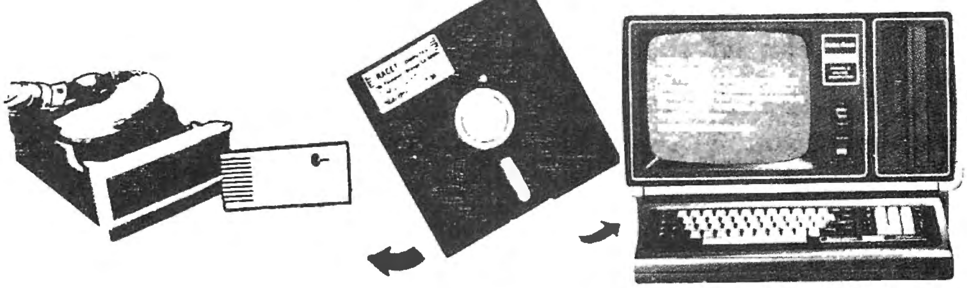
So far the command has done nothing. That is because we have not told it what byte to look for. Enter a two digit "target" byte followed by the byte that will replace it wherever it is found. For this example type 20 and BF. If you didn't follow what happened, the computer went searching through the video memory for all occurrences of 20 which is the ASCII code for a space and, upon finding them, changed each into BF which is the graphic code for all pixels (blocks) on. This is, of course, only an example and any values can be used, but take care in choosing safe values so that you don't "bomb" the system.

Expand Flag Register

The need to know the status of each individual flag bit in the flag register is important when debugging a program. The flag register can be displayed with the R command, but this is of little value unless you have memorized the position of each separate flag bit and have paper and pencil nearby to convert the hexadecimal digits into their binary equivalents. Press the R key and find the F register. Now press the E key. This will expand the register into its binary form. By comparing the hexadecimal output with its binary counterpart, you should find the command's operation quite reliable.

Hex/ASCII Dump

My need to display a full



FIELD PROVEN!!

10 MEGABYTES and MORE for the TRS-80* Model II plus SHARED ACCESS to HARD DISK DRIVE

Hard/Soft Disk System (HSDS) Software allows access as single drive. You can have that 10 Megabyte continuous file - that 50,000 name maillist or inventory! Or a directory with 1000 entries! All completely compatible with TRSDOS 2.0 BASIC. You can mix floppy and hard disk drives. Includes special utilities including HPURGE, DCS Directory Catalog System, HZAP Hard Disk Superzap, and many special formatting options. Three to eight times faster than floppy! RACET quality

HARD DISK DRIVE & CONTROLLER $5995. Second User $595. HSDS Software $400. (Note: HSDS now also available for CORVUS drives!!)

KFS-80 (1-drive 32K Min - Mod II 64K) **Mod I and III $100, Mod II $175**
Assembly language ISAM facility! Interfaces with BASIC. Very fast access to records in large files. Access/insert times 1-4 seconds. B-Tree directory structure eliminates record search. No file reorganization required. Sector buffering reduces media and drive wear while increasing performance. Can be used for primary and secondary file database applications. Records are kept in sorted order on disk by alpha key. Deferred write option. Comprehensive file maintenance and debugging utilities are included. **Attention applications programmers** - Save time in implementing your custom database application software.

MAILLIST (1-drive 32K Min - Mod II 64K) **Mod I and III $75, Mod II $150**
This ISAM-based maillist minimizes disk access times. Four keys — no separate sorting. Supports 9-digit zip code and 3-digit state code. Up to 30 attributes. Mask and query selection. Record access/update time 1-4 seconds!!

DISCAT (32K 1-drive Min) **Mod I and III $50**
This comprehensive Diskette Cataloguing/Indexing utility allows the user to keep track of thousands of programs in a categorized library. Machine language program works with all TRSDOS and NEWDOS versions. Files include program names and extensions, program length, diskette numbers, front and back, and diskette free space. RS232 drivers and other features.

LPSPOOL (32K 1-drive Min) **Mod I $75**
LPSPOOL — Add multi-tasking to permit concurrent printing while running your application program. The spooler and despooler obtain print jobs from queues maintained by the system as print files are generated. LPSPOOL supports both parallel and serial printers.

BASIC LINK FACILITY 'BLINK' (Mod I Min 32K 1-disk) **Mod I $25; Mod II $50; Mod III $30**
Link from one BASIC program to another saving all variables! The new program can be smaller or larger than the original program in memory. The chained program may either replace the original program, or can be **merged** by statement number. The statement number where the chained program execution is to begin may be specified!

INFINITE BASIC (Mod I & Mod III Tape or Disk) **Mod I $50; Mod III $60**
Extends Level II BASIC with complete MATRIX functions and 50 more string functions. Includes RACET machine language sorts! Sort 1000 elements in 9 seconds!! Select only functions you want to optimize memory usage.

INFINITE BUSINESS (Requires Infinite BASIC) **Mod I & III $30**
Complete printer pagination controls — auto headers, footers, page numbers. Packed decimal arithmetic — 127 digit accuracy +, −, *, /. Binary search of sorted and unsorted arrays. Hash codes.

COMPROC (Mod I & Mod III — Disk only) **Mod I $20; Mod III $30**
Command Processor. Auto your disk to perform any sequence of instructions that you can give from the keyboard. DIR, FREE, pause, wait for user input, BASIC, No. of FILES and MEM SIZE, RUN program, respond to input statements, BREAK, return to DOS, etc. Includes lowercase driver software, debounce and screenprint!

GSF (Mod I & III Tape or Disk - Specify Memory Size) **Mod I $25; Mod II $50; Mod III $30**
Generalized Subroutine Facilities. The STANDARD against which all other sorts are compared! And then compare prices! Machine language — fast and powerful! Multi-key multi-variable and multi-key character string. Zero and move arrays. Mod II includes USR PEEKS and POKES. Includes sample programs.

DSM (Mod I Min 32K 2-drive system. Mod II 64K 1-drive. Mod III Min 32K 1-drive) **Mod I $75; Mod II $150; Mod III $90**
Disk Sort/Merge for RANDOM files. All machine language stand-alone package for sorting speed. Establish sort specification in simple BASIC command File. Execute from DOS. Only operator action to sort is to change diskettes when requested! Handles multiple diskette files! Super fast sort times — improved disk I/O times make this the fastest Disk Sort/Merge available on the TRS.

UTILITY PACKAGE (Mod II 64K) **$150**
Important enhancements to the Mod II. The file recovery capabilities alone will pay for the package in even one application! Fully documented in 124 page manual! XHIT, XGAT, XCOPY and SUPERZAP are used to reconstruct or recover data from bad diskettes! XCOPY provides multi-file copies, 'wild-card' mask select, absolute sector mode and other features. SUPERZAP allows examine/change any sector on diskette including track-0, and absolute disk backup/copy with I/O recovery. DCS builds consolidated directories from multiple diskettes into a single display or listing sorted by disk name or file name plus more. Change Disk ID with DISKID. XCREATE preallocates files and sets 'LOF' to end to speed disk accesses. DEBUG!! adds single step, trace, subroutine calling, program looping, dynamic disassembly and more!!

BASIC CROSS REFERENCE UTILITY (Mod II 64K) **$50**
SEEK and FIND functions for Variables, Line Numbers, Strings, Keywords. 'All' options available for line numbers and variables. Load from BASIC — Call with 'CTRL'R. Output to screen or printer!

DEVELOPMENT PACKAGE (Mod II 64K) **$125**
Includes RACET machine language SUPERZAP, Apparat Disassembler, and Model II interface to the Microsoft 'Editor Assembler Plus' software package including uploading services and patches for Disk I/O. Purchase price includes complete copy of Editor Assembler + and documentation for Mod I. Assemble directly into memory. MACRO facility, save all or portions of source to disk, dynamic debug facility (ZBUG), extended editor commands.

DEALER inquiries invited. System Houses - check for special pricing on hard drives and software.

CIRCLE READER REQUEST FOR FREE 24-PAGE CATALOG
*TRS-80 IS A TRADEMARK OF TANDY CORPORATION

CHECK, VISA, M/C, C.O.D., PURCHASE ORDER
TELEPHONE ORDERS ACCEPTED (714) 997-4950

RACET COMPUTES
1330 N. GLASSELL, SUITE M, ORANGE, CA 92667

41

Program continued

```
4935 CD3245   00147 B5      CALL    ASPC              ;OUTPUT (A) + SPACE
4938 10F1     00148         DJNZ    L2                ;CONTIUE FOR ALL BITS
493A CD3C45   00149         CALL    CRS               ;OUTPUT A C/R
493D C3A543   00150         JP      BACK2             ;RETURN
4940 18       00151 MES     DEFB    18H               ;BACKSPACE
4941 53       00152         DEFM    'S Z - H - P N C'       ;FLAG MESSAGE
4950 0D20     00153         DEFW    200DH             ;C/R AND SPACE
4952 00       00154         DEFB    0                 ;0 MARKS THE END
4953 ED4B3D48 00155 OUTMES  LD      BC,(CURSR)        ;OUTPUTS MESSAGE
4957 ED432040 00156         LD      (4020H),BC        ;AT CURRENT CURSOR
495B CDA728   00157         CALL    28A7H             ;POSITION USING
495E ED4B2040 00158         LD      BC,(4020H)        ;ROM OUTPUT
4962 ED433D48 00159         LD      (CURSR),BC        ;ROUTINE
4966 C9       00160         RET
4967 FE48     00161 HEXDMP  CP      'H'               ;CHECK FOR "H"
4969 2066     00162         JR      NZ,INITIA         ;GO IF NOT PRESSED
496B CD3245   00163         CALL    ASPC              ;OUTPUT (A) + SPACE
496E CD5148   00164         CALL    HLIN              ;GET START ADDRESS
4971 EB       00165         EX      DE,HL             ;MOVE TO DE
4972 0E10     00166 HEXOUT  LD      C,10H             ;LINE COUNT
4974 2A3D48   00167         LD      HL,(CURSR)        ;GET T-BUG CURSOR
4977 222040   00168         LD      (4020H),HL        ;PUT IN BASIC CURSOR
497A EB       00169         EX      DE,HL             ;SAVE DE
497B 3E0D     00170         LD      A,0DH             ;0DH = C/R
497D CD3300   00171         CALL    33H               ;OUTPUT A C/R
4980 EB       00172         EX      DE,HL             ;RESTORE DE
4981 2A2040   00173         LD      HL,(4020H)        ;BASIC'S CURSOR
4984 223D48   00174         LD      (CURSR),HL        ;BACK TO T-BUG'S
4987 62       00175         LD      H,D               ;PUT DE INTO HL
4988 6B       00176         LD      L,E
4989 CD5E48   00177         CALL    HLOUT             ;OUTPUT MEM POINTER
498C 3E5E     00178         LD      A,5EH             ;5EH = RIGHT ARROW
498E CD3245   00179         CALL    ASPC              ;OUTPUT ARROW
4991 D5       00180         PUSH    DE                ;SAVE MEM POINTER
```

Program continues

page of 256 continuous bytes of memory in hexadecimal along with their associated ASCII symbols prompted this next command. If you type H 0000 you will see the first 256 bytes of ROM. Each line contains the starting address of that line followed by sixteen hexadecimal numbers grouped in twos. The ASCII symbols for these are printed to the right on each line. The cursor character is displayed in place of graphic and control characters. Press Enter to look at the next 256 bytes of memory. You should now see the second page of ROM which starts at 0100. If you look in the ASCII section, you will notice the data for the Radio Shack Level II Basic message. Press 1 if you want to look line by line through memory instead. By pressing any key 1–9, that many more lines will be displayed. Pressing 0, you will find, scans over the contents of the next 1024 bytes of memory. The scan feature can come in handy for

✓ See List of Advertisers on page 402

searching large sections of memory. To exit this command, press X. I will mention here that the X key can be used to exit most of the new commands.

Initialize

When you first load NEWBUG and get ready to start debugging, or whenever you wish to clear all the workable registers, simply press I. This initializes NEWBUG by zeroing the register sets, both prime and non-prime, and the IX and IY index registers. The stack pointer (SP) and program counter (PC) are not zeroed because this would have disastrous effects. To convince yourself that the registers are in fact cleared, press I and then R and observe for yourself.

Block Checksum

Checksums are useful for comparing blocks of memory and for other special cases when you need to know the sum of a block of memory. Most checksums add together the values of all the bytes in the specified block of memory and display only the least significant byte of the resulting checksum. For most cases this is fine, but there are times when the total value of the checksum is needed. Type K for checksum (C was already used for the change command) followed by the starting and ending addresses of the block of memory of which you wish to compute the checksum. The resulting eight digit hexadecimal number is the computed checksum of the chosen block of memory. Because the checksum command is non-destructive (that is, it doesn't change any memory locations), it can be used anywhere in RAM or ROM. In all commands dealing with a block of memory the starting and ending memory locations are included in the operation and are said to be inclusive commands. You may also notice that if the starting address is larger than the ending address, an E will be printed and the command will be aborted. This is because the commands are not capable of operating on imaginary blocks of memory which end before

Program continued

```
4992 0608     00181        LD     B,8            ;FOR 8 PAIRS OF BYTES
4994 1A       00182 L3     LD     A,(DE)
4995 67       00183        LD     H,A            ;H=FIRST BYTE
4996 13       00184        INC    DE             ;INC POINTER
4997 1A       00185        LD     A,(DE)
4998 6F       00186        LD     L,A            ;L=SECOND BYTE
4999 13       00187        INC    DE             ;INC POINTER
499A CD5E48   00188        CALL   HLOUT          ;OUTPUT A PAIR OF BYTES
499D 10F5     00189        DJNZ   L3             ;CONTINUE 8 TIMES
499F D1       00190        POP    DE             ;GET MEM POINTER AGAIN
49A0 2A3D48   00191        LD     HL,(CURSR)     ;CURSOR -> HL
49A3 0610     00192        LD     B,10H          ;FOR 16 CHARACTERS
49A5 1A       00193 L4     LD     A,(DE)         ;GET CHAR
49A6 FE20     00194        CP     ' '            ;TEST FOR LESS THAN 20H
49A8 3804     00195        JR     C,B6           ;(CONTROL CHARACTER)
49AA FE81     00196        CP     81H            ;TEST FOR GRAPHIC CODE
49AC 3802     00197        JR     C,B7
49AE 3E5F     00198 B6     LD     A,5FH          ;SUBSTITUTE CHAR
49B0 77       00199 B7     LD     (HL),A         ;STORE CHAR
49B1 23       00200        INC    HL             ;INC SCREEN POINTER
49B2 13       00201        INC    DE             ;INC MEMORY POINTER
49B3 10F0     00202        DJNZ   L4             ;CONTINUE FOR 16 CHARS
49B5 0D       00203        DEC    C              ;DEC LINE COUNTER
49B6 20BC     00204        JR     NZ,HEXOUT+2    ;GO IF NOT DONE
49B8 D9       00205 B8     EXX                   ;SAVE REGISTERS
49B9 CDC845   00206        CALL   INKY           ;GET (A) FROM KEYBOARD
49BC D9       00207        EXX                   ;RESTORE REGS
49BD FE58     00208        CP     'X'            ;CHECK FOR EXIT
49BF CA9148   00209        JP     Z,CLEAR+4      ;CLS AND RETURN
49C2 FE0D     00210        CP     0DH            ;CHECK FOR C/R
49C4 28AC     00211        JR     Z,HEXOUT       ;OUTPUT 16 MORE LINES
49C6 D630     00212        SUB    30H            ;CORRECT ASCII
49C8 38EE     00213        JR     C,B8           ;IF LESS THAN 1
49CA FE0A     00214        CP     0AH
49CC 30EA     00215        JR     NC,B8          ;IF GREATER THAN 9
49CE 4F       00216        LD     C,A            ;COUNT -> C
49CF 18A3     00217        JR     HEXOUT+2       ;CONINUE
49D1 FE49     00218 INITIA CP     'I'            ;CHECK FOR "I"
49D3 200D     00219        JR     NZ,CHKSUM      ;GO IF NOT PRESSED
49D5 212548   00220        LD     HL,REGS        ;(HL)=START OF REGISTERS
49D8 0614     00221        LD     B,14H          ;BYTE COUNT -> B
49DA 3600     00222 L5     LD     (HL),0         ;ZERO A REGISTER
49DC 23       00223        INC    HL             ;POINT TO NEXT
49DD 10FB     00224        DJNZ   L5             ;CONTINUE 'TILL DONE
49DF C39148   00225        JP     CLEAR+4        ;CLEAR SCREEN & RETURN
49E2 FE4B     00226 CHKSUM CP     'K'            ;CHECK FOR "K"
49E4 203C     00227        JR     NZ,NPUT        ;GO IF NOT PRESSED
49E6 CD3245   00228        CALL   ASPC           ;OUTPUT (A) + SPACE
49E9 CD5148   00229        CALL   HLIN           ;GET START OF CHECKSUM
49EC E5       00230        PUSH   HL             ;SAVE START ADDRESS
49ED CD7B45   00231        CALL   SPC            ;OUTPUT SPACE
49F0 CD5148   00232        CALL   HLIN           ;GET END ADDRESS
49F3 E5       00233        PUSH   HL             ;SAVE END
49F4 CD7B45   00234        CALL   SPC            ;OUTPUT SPACE
49F7 D1       00235        POP    DE             ;GET END
49F8 13       00236        INC    DE             ;CORRECT FOR COUNT
49F9 E1       00237        POP    HL             ;GET START
49FA CD8348   00238        CALL   ERR            ;CHECK FOR START>END
49FD D9       00239        EXX                   ;SWITCH REGS
49FE 210000   00240        LD     HL,0           ;CLEAR LSB OF CHECKSUM
4A01 E5       00241        PUSH   HL
4A02 D1       00242        POP    DE             ;CLEAR MSB OF CHECKSUB
4A03 E5       00243        PUSH   HL
4A04 C1       00244        POP    BC             ;CLEAR BC
4A05 D9       00245        EXX                   ;SWITCH REGS
4A06 7E       00246 B9     LD     A,(HL)         ;GET BYTE
4A07 D9       00247        EXX                   ;SWITCH REGS
4A08 4F       00248        LD     C,A            ;BYTE INTO BC
4A09 09       00249        ADD    HL,BC          ;ADD TO CHECKSUM
4A0A 3001     00250        JR     NC,B10         ;JUMP IF NO CARRY
4A0C 13       00251        INC    DE             ;INC MSB OF CHECKSUM
4A0D D9       00252 B10    EXX                   ;SWITCH REGS
4A0E 23       00253        INC    HL             ;INC POINTER
4A0F DF       00254        RST    18H            ;CHECK IF DONE
4A10 38F4     00255        JR     C,B9           ;CONTINUE
4A12 D9       00256        EXX                   ;SWITCH REGS
4A13 EB       00257        EX     DE,HL          ;PUT MSB IN HL
4A14 CD5E48   00258        CALL   HLOUT          ;OUTPUT IT
4A17 213D48   00259        LD     HL,CURSR       ;GET CURSOR
4A1A 35       00260        DEC    (HL)           ;REMOVE SPACE
4A1B EB       00261        EX     DE,HL          ;GET LSB IN HL
4A1C CD5E48   00262        CALL   HLOUT          ;OUTPUT IT
4A1F C3A543   00263        JP     BACK2          ;RETURN
```

Program continues

✓See List of Advertisers on page 402

Program continued

```
4A22 FE4E     00264 NPUT    CP      'N'            ;CHECK IF "N"
4A24 2054     00265         JR      NZ,OUTPT       ;GO IF NOT PRESSED
4A26 CD3245   00266         CALL    ASPC           ;OUTPUT (A) + SPACE
4A29 CD5148   00267         CALL    HLIN           ;GET START OF INPUT
4A2C EB       00268         EX      DE,HL          ;START -> DE
4A2D CD7B45   00269         CALL    SPC            ;SPACE
4A30 215E4C   00270         LD      HL,END+1       ;LOWEST ALLOWED -> HL
4A33 CD8348   00271         CALL    ERR            ;CHECK IF START IS LOW
4A36 EB       00272         EX      DE,HL          ;DE <--> HL
4A37 CD9302   00273         CALL    293H           ;READ LEADER
4A3A 0608     00274         LD      B,8            ;COUNT -> B
4A3C CD684A   00275         CALL    BLOK           ;INPUT (B) BYTES
4A3F FE3C     00276 L7      CP      3CH            ;DATA HEADER?
4A41 2014     00277         JR      NZ,ENTRY       ;NO. MUST BE ENTRY PT.
4A43 CD2C02   00278         CALL    22CH           ;FLIP/FLOP "*"
4A46 CD3502   00279         CALL    235H           ;READ BYTE
4A49 77       00280         LD      (HL),A         ;STORE IT
4A4A 23       00281         INC     HL
4A4B 47       00282         LD      B,A            ;LOAD B WITH COUNT
4A4C CD684A   00283         CALL    BLOK           ;READ IN (B) BYTES
4A4F CD704A   00284         CALL    IN2            ;GET 2 MORE
4A52 CD704A   00285         CALL    IN2            ;AND ANOTHER 2
4A55 18E8     00286         JR      L7             ;CONTINUE LOADING
4A57 CD704A   00287 ENTRY   CALL    IN2            ;GET ENTRY POINT
4A5A 2B       00288         DEC     HL             ;HL = END OF INPUT
4A5B EB       00289         EX      DE,HL          ;SAVE HL
4A5C 3E2D     00290         LD      A,'-'
4A5E CD3245   00291         CALL    ASPC           ;OUTPUT "-" + SPACE
4A61 EB       00292         EX      DE,HL          ;GET END AGAIN
4A62 CD5E48   00293         CALL    HLOUT          ;OUTPUT END ADDRESS
4A65 C3A543   00294         JP      BACK2          ;RETURN
4A68 CD3502   00295 BLOK    CALL    235H           ;SUBROUTINE TO INPUT
4A6B 77       00296         LD      (HL),A         ;(B) BYTES FROM TAPE
4A6C 23       00297         INC     HL             ;AND STORE AT (HL)
4A6D 10F9     00298         DJNZ    BLOK
4A6F C9       00299         RET
4A70 EB       00300 IN2     EX      DE,HL          ;SUBROUTINE TO INPUT
4A71 CD1403   00301         CALL    314H           ;2 BYTES AND
4A74 EB       00302         EX      DE,HL          ;STORE IN (HL)
4A75 73       00303         LD      (HL),E
4A76 23       00304         INC     HL
4A77 72       00305         LD      (HL),D
4A78 23       00306         INC     HL
4A79 C9       00307         RET
4A7A FE4F     00308 OUTPT   CP      'O'            ;CHECK FOR "O"
4A7C 2025     00309         JR      NZ,QUE         ;GO IF NOT PRESSED
4A7E CD3245   00310         CALL    ASPC           ;OUTPUT (A) + SPACE
4A81 CD5148   00311         CALL    HLIN           ;GET START OF OUTPUT
4A84 E5       00312         PUSH    HL             ;SAVE IT
4A85 CD7B45   00313         CALL    SPC            ;SPACE
4A88 CD5148   00314         CALL    HLIN           ;GET END OF OUTPUT
```

Program continues

they start.

Tape Input

If you look at the source code listing of the NEWBUG patch, you will see that the code for the patch loads into three separate blocks of memory starting at 43DE, 4782, and 4851. Because the patch is non-continuous, it cannot be punched to tape to make a backup copy with the P command. The input command will read in any system formatted machine-code program and store a byte-for-byte image of the taped program in a memory buffer which you select. If you have assembled NEWBUG with an editor/assembler and have the tape with the object code nearby, load the tape and type N 5000. The cassette drive should start, the tape should be read in. and the * should blink as usual. When the program is finished loading, the ending byte of the input buffer is displayed. Save the starting and ending addresses for use with the next command. You may examine the contents of the buffer with the H command if you wish.

Tape Output

Until now, the input command may seem to be of little use because the program stored in the buffer cannot be correctly copied to tape with the P command. Well, that's what the output command is

for. Position the tape to where you wish to dump a backup copy, press the record and play buttons, and type O followed by the starting and ending addresses you should have saved from the last command. The tape will start rolling and a copy of the original program will be written to tape. If you want to make another copy of the program, just use the output command again using the same addresses. Unless you wipe out the copy of the program in the buffer, there is no need to load it for each copy you make. If you try to input into a buffer less than 4C5C, the computer will respond with an E. This is to protect NEWBUG from being written over accidentally.

Cue Tape

While positioning tapes for reading or writing, the small grey remote plug on the cassette recorder is constantly being pulled out. This leads to bent and worn out plugs. In an effort to save the plug, the Q command transfers control of the cassette to you. Control will be returned to the computer when you hit the X key.

Block Search

Some machine-code monitors have commands to search for occurrences of one or two bytes. I found the need to search for a string of three or more bytes at a time, so the variable length block search command became part of NEWBUG's menu of commands.

Suppose we wanted to know from where in ROM the subroutine for clearing the screen (located at 01C9) was called. To do this, we have to search locations 0000 to 3000 (the block of memory containing the ROM) for occurrences of CD C9 01 which is the machine code for the instruction call 01C9. Type S 0000 3000 CD C9 01 and press Enter. The addresses are found and displayed eight to a line. Go ahead and use the M command to verify that the code is actually at these addresses. The uniqueness of this command is that the search key (in this case CD C9 01) may be

Program continued

```
4A8B E5        00315        PUSH  HL          ;SAVE IT
4A8C CD7B45    00316        CALL  SPC         ;SPACE
4A8F D1        00317        POP   DE          ;POP END INTO DE
4A90 E1        00318        POP   HL          ;POP START INTO HL
4A91 13        00319        INC   DE          ;INC TO INCLUDE LAST BYTE
4A92 CD8348    00320        CALL  ERR         ;CHECK FOR START>END
4A95 CD8402    00321        CALL  284H        ;WRITE LEADER
4A98 7E        00322 L8     LD    A,(HL)      ;DATA -> A
4A99 CD6402    00323        CALL  264H        ;OUTPUT A
4A9C 23        00324        INC   HL          ;POINT TO NEXT
4A9D DF        00325        RST   18H         ;CHECK IF DONE
4A9E 38F8      00326        JR    C,L8        ;CONINUE IF NOT
4AA0 C3A543    00327        JP    BACK2       ;RETURN
4AA3 FE51      00328 QUE    CP    'Q'         ;CHECK FOR "Q"
4AA5 2011      00329        JR    NZ,SEARCH   ;GO IF NOT PRESSED
4AA7 CD3245    00330        CALL  ASPC        ;OUTPUT (A) + SPACE
4AAA 3E04      00331        LD    A,4         ;SET BIT 2
4AAC D3FF      00332        OUT   (0FFH),A    ;TURN ON CASSETTE
4AAE CDC845    00333 B11    CALL  INKY        ;SCAN KEYBOARD
4AB1 FE58      00334        CP    'X'         ;STOP?
4AB3 20F9      00335        JR    NZ,B11      ;CONTINUE IF NOT
4AB5 C3A543    00336        JP    BACK2       ;RETURN
4AB8 FE53      00337 SEARCH CP    'S'         ;CHECK FOR "S"
4ABA 206F      00338        JR    NZ,TRANSF   ;GO IF NOT PRESSED
4ABC CD3245    00339        CALL  ASPC        ;OUTPUT (A) + SPACE
4ABF CD5148    00340        CALL  HLIN        ;GET START OF SEARCH
4AC2 E5        00341        PUSH  HL          ;SAVE IT
4AC3 CD7B45    00342        CALL  SPC         ;SPACE
4AC6 CD5148    00343        CALL  HLIN        ;GET END OF SEARCH
4AC9 E5        00344        PUSH  HL          ;SAVE IT
4ACA CD7B45    00345        CALL  SPC         ;SPACE
4ACD D1        00346        POP   DE          ;POP END
4ACE E1        00347        POP   HL          ;POP START
4ACF 13        00348        INC   DE          ;CORRECT COUNT
4AD0 CD8348    00349        CALL  ERR         ;CHECK FOR START>END
4AD3 E5        00350        PUSH  HL          ;SAVE START AGAIN
4AD4 D5        00351        PUSH  DE          ;SAVE END ALSO
4AD5 CD8945    00352        CALL  INA         ;GET 1st BYTE OF KEY
4AD8 210042    00353        LD    HL,4200H    ;START OF KEY BUFFER
4ADB 0E00      00354        LD    C,0         ;COUNT = 0
4ADD 77        00355        LD    (HL),A      ;SAVE BYTE
4ADE D9        00356 L9     EXX               ;SAVE REGISTERS
4ADF CD7B45    00357        CALL  SPC         ;SPACE
4AE2 D9        00358        EXX               ;GET REGISTERS
4AE3 2B        00359        DEC   HL          ;DEC BUFFER ADDRESS
4AE4 0C        00360        INC   C           ;INC COUNT
4AE5 D9        00361        EXX               ;SAVE REGS
4AE6 CDC845    00362        CALL  INKY        ;GET 1 INPUT
4AE9 FE0D      00363        CP    0DH         ;C/R?
4AEB 280F      00364        JR    Z,B12       ;IF SO THEN START
4AED CDA745    00365        CALL  LST         ;ELSE GET OTHER HALF
4AF0 CD8C45    00366        CALL  INA+3       ;OF THE INPUT
4AF3 D9        00367        EXX               ;GET REGS
4AF4 77        00368        LD    (HL),A      ;STORE BYTE
4AF5 3E0F      00369        LD    A,0FH
4AF7 B9        00370        CP    C           ;16 INPUTS?
4AF8 20E4      00371        JR    NZ,L9       ;GO IF NOT
4AFA 0C        00372        INC   C           ;CORRECT COUNT
4AFB D9        00373        EXX               ;SAVE REGS
4AFC CD3C45    00374 B12    CALL  CRS         ;OUTPUT A C/R
4AFF D9        00375        EXX               ;GET REGS
4B00 D1        00376        POP   DE          ;END OF SEARCH
4B01 E1        00377        POP   HL          ;START OF SEARCH
4B02 41        00378 B13    LD    B,C         ;COUNT -> B
4B03 DD210042  00379        LD    IX,4200H    ;START OF LOOKUP KEY
4B07 DD7E00    00380 B14    LD    A,(IX)      ;GET KEY BYTE
4B0A DD2B      00381        DEC   IX          ;DEC KEY POINTER
4B0C 23        00382        INC   HL          ;INC MEMORY POINTER
4B0D BE        00383        CP    (HL)        ;MATCH?
4B0E 2015      00384        JR    NZ,B15      ;GO IF NOT
4B10 10F5      00385        DJNZ  B14         ;LOOP B TIMES
4B12 E5        00386        PUSH  HL          ;SAVE
4B13 C5        00387        PUSH  BC          ;SAVE
4B14 AF        00388        XOR   A           ;CLEAR CARRY
4B15 47        00389        LD    B,A         ;B=0
4B16 ED42      00390        SBC   HL,BC       ;(HL)=START OF FIND-1
4B18 23        00391        INC   HL          ;CORRECT (HL)
4B19 CD5E48    00392        CALL  HLOUT       ;OUTPUT (HL) + SPACE
4B1C 0603      00393        LD    B,3         ;OUTPUT 3 MORE SPACES
4B1E CD7B45    00394 SPA3   CALL  SPC
4B21 10FB      00395        DJNZ  SPA3
4B23 C1        00396        POP   BC          ;POP
4B24 E1        00397        POP   HL          ;POP
4B25 DF        00398 B15    RST   18H         ;IS SEARCH OVER?
```

Program continues

Program continued

```
4B26 38DA            JR     C,B13         ;GO IF NOT
4B28 C3A543          JP     BACK2         ;RETURN IF FINISHED
4B2B FE54     TRANSF CP     'T'           ;CHECK FOR "T"
4B2D 2046            JR     NZ,WRITE      ;GO IF NOT PRESSED
4B2F CD3245          CALL   ASPC          ;OUTPUT (A) + SPACE
4B32 CD5148          CALL   HLIN          ;GET START OF SOURCE
4B35 E5              PUSH   HL            ;SAVE IT
4B36 CD7B45          CALL   SPC           ;SPACE
4B39 CD5148          CALL   HLIN          ;GET END OF SOURCE
4B3C E5              PUSH   HL            ;SAVE IT
4B3D CD7B45          CALL   SPC           ;SPACE
4B40 CD5148          CALL   HLIN          ;GET START OF DESTINATION
4B43 E5              PUSH   HL            ;SAVE IT
4B44 CD7B45          CALL   SPC           ;SPACE
4B47 C1              POP    BC            ;DESTINATION -> BC
4B48 D1              POP    DE            ;SOURCE END -> DE
4B49 E1              POP    HL            ;SOURCE START -> HL
4B4A CD8348          CALL   ERR           ;CHECK IF START>END
4B4D EB              EX     DE,HL         ;DE <--> HL
4B4E AF              XOR    A             ;CLEAR CARRY
4B4F ED52            SBC    HL,DE         ;GET BYTE COUNT-1
4B51 23              INC    HL            ;CORRECT IT
4B52 D5              PUSH   DE            ;PUSH ALL ONTO STACK
4B53 C5              PUSH   BC            ;TO SWITCH
4B54 E5              PUSH   HL
4B55 C1              POP    BC            ;BYTE COUNT
4B56 D1              POP    DE            ;DESTINATION
4B57 E1              POP    HL            ;SOURCE
4B58 E5              PUSH   HL            ;SAVE AGAIN
4B59 B7              OR     A             ;CLEAR CARRY
4B5A ED52            SBC    HL,DE         ;FIND MOVE DIRECTION
4B5C E1              POP    HL            ;GET SOURCE AGAIN
4B5D 3806            JR     C,B16         ;GO IF MOVE BACKWARDS
4B5F EDB0            LDIR                 ;MOVE IT
4B61 EB              EX     DE,HL         ;DE <--> HL
4B62 2B              DEC    HL            ;CORRECT TO END
4B63 180A            JR     B17
4B65 09       B16    ADD    HL,BC         ;FIND START OF MOVE+1
4B66 2B              DEC    HL            ;CORRECT START
4B67 EB              EX     DE,HL
4B68 09              ADD    HL,BC         ;FIND END OF MOVE+1
4B69 2B              DEC    HL            ;CORRECT ALSO
4B6A EB              EX     DE,HL
4B6B D5              PUSH   DE            ;SAVE END
4B6C EDB8            LDDR                 ;MOVE BACKWARDS
4B6E E1              POP    HL            ;POP END FROM STACK
4B6F CD5E48   B17    CALL   HLOUT         ;DISPLAY END
4B72 C3A543          JP     BACK2         ;RETURN
4B75 FE57     WRITE  CP     'W'           ;CHECK FOR "W"
4B77 2074            JR     NZ,ZERO       ;GO IF NOT PRESSED
```

Program continues

from one to sixteen bytes long. If the key is sixteen bytes long, execution will begin automatically. However, if the key is less than sixteen bytes long, press Enter to start execution. Take care not to search for a byte or bytes which may occur at a great number of places because the command will print page after page of addresses without stopping until it has listed them all. If you don't have a full 48K of memory, typing the command S C000 FFFE FF Enter will show you what I am referring to.

Block Transfer

Transferring blocks of data from one place to another in memory requires writing a block move program and executing it. I found myself doing this so frequently that the transfer command was created to save time. As an example of this command's use, let's move the first page of ROM (0000 to 00FF) into RAM starting at 5000. Press T followed by the source block addresses 0000 and 00FF. Follow this with the starting address of the destination block which in this case is 5000. In no time at all the ending address of the destination block is displayed and the * prompt returns ready for your next command. Type H 5000 to have a glimpse of the data that was just transferred. You may wish to perform a checksum on the two blocks 0000 to 00FF and 5000 to 50FF to see that they are identical. The source and destination blocks were not overlapping but the command can handle them without losing data if they do overlap.

Direct Register Write

Did you ever wish that you could step in and write values directly into the active registers to alter the flow of a program that you were debugging? If so, you will enjoy this next command. Press W and you will see the contents of the AF register pair followed by a /. If you want to place a 1234 in the BC register pair and a 5678 in the IY index register, press Enter to advance to the BC pair and type 1234. Now press Enter until the IY register is displayed and type 5678. It's that easy. Are you convinced that they were changed? Don't take my word for it. Press R and see for yourself.

Zero Memory

This command will clear (by writing in 00s) any block of RAM memory. By typing Z 4C5C BFFF, the programming workspace of a 32K machine will be cleared. Users with a 48K machine will notice that Z 4C5C FFFF yields little more than an E being displayed. This is because the computer interprets FFFF as a −1. However, Z 4C5C FFFE works fine. This is a destructive command. Choose values wisely.

Memory Test

As a finish, press * and wait a moment. The resulting output tells you that the memory has been checked and is valid to the displayed address. If this first value is not your machine's top of memory, the cell at this address plus one may be faulty or intermittent. Press * a few times to confirm the results of the test. The test starts with the RAM immediately following NEWBUG. The test is nondestructive and does not change any of the memory that is tested. The second line of this output tells you at what size you set the current memory, in case you do not remember. ■

Program continued

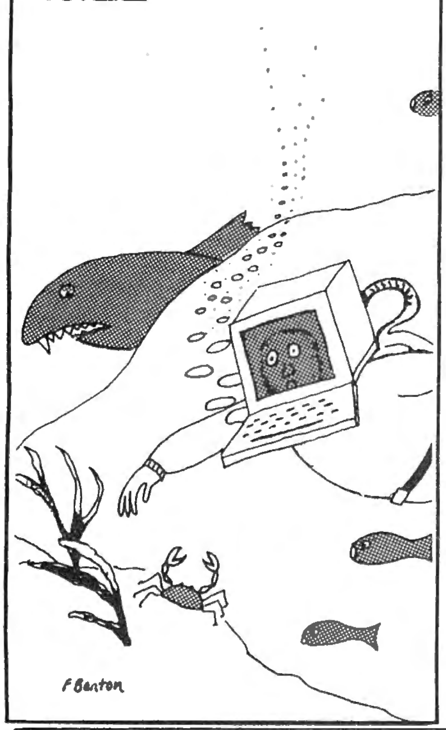
```
4B79 CD3C45          CALL  CRS          ;OUTPUT A C/R
4B7C FD212D48        LD    IY,REGS+8    ;POINTS TO T-BUG'S AF REG
4B80 DD21CE4B        LD    IX,MESREG    ;START OF MESSAGE
4B84 ED5B3D48  B18   LD    DE,(CURSR)   ;CURSOR -> DE
4B88 0605            LD    B,5          ;CHAR COUNT = 5
4B8A DD7E00    L10   LD    A,(IX+0)     ;GET CHAR
4B8D B7              OR    A            ;TEST FOR ZERO
4B8E CAA543          JP    Z,BACK2      ;RETURN IF DONE
4B91 12              LD    (DE),A       ;STORE ON SCREEN
4B92 DD23            INC   IX           ;INC MESSAGE PTR
4B94 13              INC   DE           ;INC SCREEN PTR
4B95 10F3            DJNZ  L10          ;LOOP 5 TIMES
4B97 ED533D48        LD    (CURSR),DE   ;SAVE NEW SCREEN POSITION
4B9B FD6601          LD    H,(IY+1)     ;GET VALUE OF CURRENT
4B9E FD6E00          LD    L,(IY+0)     ;REGISTER INTO HL
4BA1 FD23            INC   IY           ;UPDATE REG PTR
4BA3 FD23            INC   IY
4BA5 CD5E48          CALL  HLOUT        ;OUTPUT OLD REG CONTENTS
4BA8 3E2F            LD    A,'/'        ;OUTPUT A "/"
4BAA CD3245          CALL  ASPC
4BAD CDC845          CALL  INKY         ;GET INPUT FROM KEYBOARD
4BB0 FE0D            CP    0DH          ;CHECK FOR <ENTER>
4BB2 2811            JR    Z,B19        ;GO TO NEXT IF PRESSED
4BB4 CDA745          CALL  LST          ;GET SECOND HALF OF A
4BB7 CD8C45          CALL  INA+3
4BBA CD5548          CALL  HLIN+4       ;CALL HERE TO COMPLETE
4BBD EB              EX    DE,HL        ;PUT NEW VALUE IN DE
4BBE FDE5            PUSH  IY           ;GET REGISTER PTR
4BC0 E1              POP   HL           ;IN HL
4BC1 2B              DEC   HL
4BC2 72              LD    (HL),D       ;STORE NEW MSB
4BC3 2B              DEC   HL
4BC4 73              LD    (HL),E       ;STORE NEW LSB
4BC5 DDE5      B19   PUSH  IX           ;SAVE MESSAGE PTR
4BC7 CD3C45          CALL  CRS          ;OUTPUT C/R
4BCA DDE1            POP   IX           ;RESTORE MESSAGE PTR
4BCC 18B6            JR    B18          ;CONTINUE
4BCE 41        MESREG DEFM 'AF = BC = DE = HL = IX = IY = '
4BEC 00              DEFB  0            ;0 MARKS THE END
4BED FE5A      ZERO  CP    'Z'          ;CHECK FOR "Z"
4BEF 201C            JR    NZ,TEST      ;GO IF NOT PRESSED
4BF1 CD3245          CALL  ASPC         ;OUTPUT (A) + SPACE
4BF4 CD5148          CALL  HLIN         ;GET START
4BF7 E5              PUSH  HL           ;SAVE IT
4BF8 CD7B45          CALL  SPC          ;SPACE
4BFB CD5148          CALL  HLIN         ;GET END
4BFE D1              POP   DE           ;POP START
4BFF EB              EX    DE,HL        ;SWITCH THEM
4C00 13              INC   DE           ;INCLUDE LAST BYTE
4C01 CD8348          CALL  ERR          ;CHECK FOR START>END
4C04 3600      B20   LD    (HL),0       ;"ZERO" (HL)
4C06 23              INC   HL           ;POINT TO NEXT
4C07 DF              RST   18H          ;CHECK IF DONE
4C08 38FA            JR    C,B20        ;GO IF NOT
4C0A C3A543          JP    BACK2        ;RETURN
4C0D FE2A      TEST  CP    '*'          ;CHECK FOR "*"
4C0F C20144          JP    NZ,NONE      ;BACK TO T-BUG IF NOT
4C12 213A4C          LD    HL,MES2      ;GET MESSAGE
4C15 CD5349          CALL  OUTMES       ;OUTPUT IT
4C18 215D4C          LD    HL,END       ;START OF USER MEMORY
4C1B 23        L11   INC   HL           ;NEXT BYTE
4C1C 7E              LD    A,(HL)       ;GET CONTENTS
4C1D 47              LD    B,A          ;SAVE IN B
4C1E 2F              CPL                ;COMPLEMENT (A)
4C1F 77              LD    (HL),A       ;PUT BACK IN (HL)
4C20 BE              CP    (HL)         ;TEST MEMORY CELL
4C21 70              LD    (HL),B       ;RESTORE TO ORIGINAL
4C22 28F7            JR    Z,L11        ;CONTINUE IF ZERO
4C24 2B              DEC   HL           ;BACK UP TO LAST GOOD
4C25 CD5E48          CALL  HLOUT        ;OUTPUT LAST GOOD BYTE
4C28 214C4C          LD    HL,MES3      ;GET NEXT MESSAGE
4C2B CD5349          CALL  OUTMES       ;OUTPUT IT
4C2E 2AB140          LD    HL,(40B1H)   ;GET MEMORY SIZE
4C31 CD5E48          CALL  HLOUT        ;OUTPUT ALSO
4C34 CD3C45          CALL  CRS          ;OUTPUT C/R
4C37 C3A543          JP    BACK2        ;RETURN
4C3A 0D        MES2  DEFB  0DH          ;C/R
4C3B 4D              DEFM  'MEMORY VALID TO '
4C4B 00              DEFB  0            ;0 MARKS THE END
4C4C 0D        MES3  DEFB  0DH          ;C/R
4C4D 4D              DEFM  'MEMORY SET AT   '
4C5D 00        END   DEFB  0            ;0 MARKS THE END
4762                 END   NEWBUG
```

Program continues

Program continued

```
ARITHM 489B 00066   00061
ASPC   4532 00009   00068 00079 00084 00098 00147 00163 00179
                    00228 00266 00291 00310 00330 00339 00403
                    00468 00490
B1     48A9 00072   00076
B10    4A0D 00252   00250
B11    4AAE 00333   00335
B12    4AFC 00374   00364
B13    4B02 00378   00399
B14    4B07 00380   00385
B15    4B25 00398   00384
B16    4B65 00436   00431
B17    4B6F 00445   00435
B18    4B84 00452   00485
B19    4BC5 00482   00471
B2     48B6 00078   00074
B20    4C04 00499   00502
B3     48D3 00094   00092
B4     4909 00120   00115
B5     4935 00147   00145
B6     49AE 00198   00195
B7     49B0 00199   00197
B8     49B8 00205   00213 00215
B9     4A06 00246   00255
BACK   43CD 00006   00065
BACK2  43A5 00005   00059 00095 00119 00132 00150 00263 00294
                    00327 00336 00400 00446 00456 00503 00524
BLOK   4A68 00295   00275 00283 00298
CHANGE 48D9 00096   00067
CHKSUM 49E2 00226   00219
CLEAR  488D 00060   00022 00209 00225
CRS    453C 00010   00149 00374 00449 00483 00523
CURSR  483D 00018   00057 00116 00155 00159 00167 00174 00191
                    00259 00452 00461
END    4C5D 00530   00270 00508
ENTRY  4A57 00287   00277
ERR    4883 00055   00238 00271 00320 00349 00416 00498
EXPAND 491F 00137   00097
FLAG   482D 00017   00142
HEXDMP 4967 00161   00138
HEXOUT 4972 00166   00204 00211 00217
HLIN   4851 00024   00069 00080 00099 00102 00164 00229 00232
                    00267 00311 00314 00340 00343 00404 00407
                    00410 00474 00491 00494
HLOUT  485E 00033   00094 00177 00188 00258 00262 00293 00392
                    00445 00466 00518 00522
IN2    4A70 00300   00284 00285 00287
INA    4589 00012   00025 00029 00105 00108 00352 00366 00473
INITIA 49D1 00218   00162
INKY   45C8 00014   00072 00206 00333 00362 00469
L1     4915 00131   00136
L10    4B8A 00454   00460
L11    4C1B 00509   00516
L2     492B 00142   00148
L3     4994 00182   00189
L4     49A5 00193   00202
L5     49DA 00222   00224
L7     4A3F 00276   00286
L8     4A98 00322   00326
L9     4ADE 00356   00371
LST    45A7 00013   00365 00472
MES    4940 00151   00139
MES2   4C3A 00525   00506
MES3   4C4C 00528   00519
MESREG 4BCE 00486   00451
NEWBUG 4762 00015   00531
NONE   4401 00007   00505
NPUT   4A22 00264   00227
OUT    486C 00041   00035 00037
OUTA   4522 00008   00047 00052
OUTMES 4953 00155   00140 00507 00520
OUTPT  4A7A 00308   00265
PLUS   48D2 00093   00089
QUE    4AA3 00328   00309
REGS   4825 00016   00220 00450
SEARCH 4AB8 00337   00329
SPA3   4B1E 00394   00395
SPC    457B 00011   00038 00071 00082 00101 00104 00107 00231
                    00234 00269 00313 00316 00342 00345 00357
                    00394 00406 00409 00412 00493
TEST   4C0D 00504   00489
TRANSF 4B2B 00401   00338
WRITE  4B75 00447   00402
ZERO   4BED 00488   00448
```

101 USES FOR
A DEAD
MICROCOMPUTER
no.22



Disc Controller for TRS-80 Model III

ONLY $375.00

Includes controller board, mounting brackets, cables, and power supply for 2 internal drives. Controls up to 4 drives which may be any combination of internal or external. This controller has been **tested for compatibility with TRSDOS[1], LDOS[2], and NEWDOS[3].** Using TRSDOS[1] you may use either Tandon or MPI 40 or 80 track drives. With LDOS[2] or NEWDOS[3] you may use any combination or 5¼" drives. The controller supports any mix of 5¼" drives including single or double density, single or double sided, 35, 40, 77 or 80 tracks.

TERMS: Cashier's Check, VISA, MASTERCARD, or C.O.D. Personal checks accepted, but will require 2-3 weeks to process. C.O.D. add 3%.

To Order, Call (505) 265-5072
OR Mail order with payment to address below:

J&M Systems, Ltd.

✓369

605½ San Mateo Blvd., N.E.
Albuquerque, N.M. 87108

[1] TRSDOS is a Trademark of Radio Shack Corporation
[2] LDOS is a Trademark of Logical Systems, Inc
[3] NEWDOS is a Trademark of APPARAT

FREE!

A SASE gets you a free program listing and a list of our low prices. Here are a few:

AXIOM IMP2 Printer $579
IMP2PAA1 With tractor, single or roll feed, .5K buffer, includes full graphics software.

SEIKO Printer $359
CP 80 M

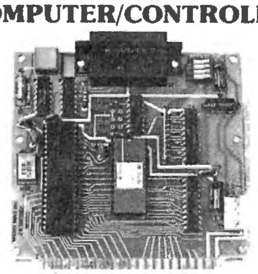
PERTEC Drive $325
FD 200 complete "flippy" (with supply & case) capable of 40 track, double density operation.

Also cassettes, diskettes, cables, memory, power supplies and cases
Software: auto minder, graphics monitor, mailbox, talking keyboard & more

TERMS: Check, Mastercharge, Visa or money order

Micro Mnemonics
(801) 825-9317
403 W 2250N
SUNSET, UTAH 84015 ✓485

Z8 BASIC COMPUTER/CONTROLLER



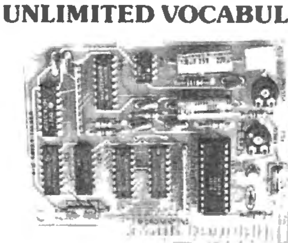
As featured in Byte Magazine, July and August 1981

- On board tiny BASIC Interpreter.
- 2 on board parallel ports.
- Serial I/O port
- 6 interrupts.
- Just attach a CRT terminal and immediately write control programs in BASIC.
- BAUD RATES 110-9600 BPS.
- Data and address buses available for 124K memory and I/O expansion.
- 4K RAM, 2716 or 2732 EPROM operation.
- Consumes only 1½ WATTS

Z8 Basic Microcomputer/Controller
Assembled & Tested........$195.00
Complete Kit..............$165.00
Universal Power Supply
(+5, +12, & -12v)..........$ 35.00

Z8 is a trademark of Zilog Inc.

SWEET-TALKER, IT GIVES YOUR COMPUTER AN UNLIMITED VOCABULARY.



As Featured In Byte Magazine, September 1981

- Utilizes VORTRAX SC-01A speech synthesizer chip.
- Unlimited vocabulary.
- Contains 64 different phonemes which are accessed by an 8-bit code.
- Text is automatically translated into electrically synthesized speech.
- Parallel port driven or Plug-in compatible with APPLE II.
- On board audio amplifier.
- Sample Program for APPLE II on cassette

SWEET-TALKER
Assembled and Tested
Parallel Port Circuit Card.......$139
APPLE II Plug-in Card.........$149

VORTRAX is a trademark of Federal Screw Works

DISK-80 EXPANSION INTERFACE FOR THE TRS-80 MODEL I

As Featured In Byte Magazine, March 1981

- Disk controller (4 drives)
- Hardware data separator
- Buffered TRS-bus connector
- Real-time clock
- Printer port (optional)

DISK 80-ASSEMBLED & TESTED
with 32K RAM.............$329.95
Centronics Printer Port add..................$ 50.00
DISK-80 pc board...........$ 48.00
Printer/Power Supply pc board.................$ 16.00
Complete Kit with 32K RAM and Printer Port.......$275.00

TRS-80 is trademark of Tandy Corp.

To Order: Call Toll Free - 1-800-645-3479
(In N.Y. State Call: 1-516-374-6793)
For Information Call: 1-516-374-6793

MICROMINT INC.
917 Midway
Woodmere, N.Y. 11598

MICRO MINT ✓310

Twelve new commands for good old Level II Basic.

Basic—Enhanced Again

Mark Goodwin
S.R. 79, Box 103
Orland, ME 04472

When Radio Shack marketed their Model I they wanted it to appeal to a wide variety of users. The TRS-80 is a versatile machine but it does lack flexibility in more specific applications.

When I first acquired my computer, I was very excited about the graphics capabilities. But with my first attempts at creating games, I found the TRS-80 is limited in graphics capabilities. Many programs can be enhanced by the use of high speed graphics, but with Level II Basic fast graphics are just not possible.

I realized that if I wanted high speed graphics, I would have to take on a serious study of machine language. Once I learned to create faster graphics by using machine language, I quickly found disadvantages to this method. The biggest disadvantage is the amount of time needed to develop a program. Furthermore, accessing machine language subroutines from Level II Basic can be frustrating.

So I began studying, to see if there was a way to add commands to Level II Basic. After a couple of years of experience with machine language, I finally found various ways to link up new commands with Level II Basic. From there it was a simple matter to write the machine language routines necessary to add virtually any command I wanted. Thus I created Enhanced Basic V1.0 (see Program Listing 1).

Enhanced Basic adds 12 new commands and modifies one existing command. It allows the user much more flexibility in writing graphic programs. Furthermore, it allows easier access to machine language subroutines in applications where even greater flexibility is required.

I do not intend to go into the details about how Enhanced Basic is written. The purpose of this article is not to instruct you on machine language programming, but to supply you with a useful modification to Level II Basic. Program Listings 2, 3, 4, 5 and 6 contain some brief examples of Enhanced Basic's versatility.■

The program is available from the author.

Table 1. Command Descriptions

USRaddress,(x)
Calls a machine language subroutine at the address specified. Execution is the same as Level II USR(x) command in every other respect. This enables the program to contain an unlimited number of machine language subroutines, without the need of POKEing the entry address into memory.

Example:
100 X = USR27000,(N)
is the same as:
100 POKE16526,120:POKE16527,105:X = USR(N)

X = &#1,(x,y)
This command allows for setting a double width graphics point.
x must> = 0 and<64
y must> = 0 and<48

Example:
100 X = &#1,(32,23)
is the same as:
100SET(64,23):SET(65,23)

X = &#2,(x,y)
This command allows resetting a double width graphics point.
x must> = 0 and<64
y must> = 0 and<48

Example:
100 X = &#2,(32,23)
is the same as:

Table continues

Table continued

100 RESET(64,23):RESET(65,23)

X = &#3,(x,y)
This command allows the testing of a double width graphics point. It will return (−1 or true) if the point is set, otherwise returns (0 or false) if the point is reset.

Example:
100 IF&#3,(32,23)THENX = &#2,(32,23)ELSEX = &#1,(32,23)
Tests to see if the point is set. If true then the point will be reset, else the point will be set.

X = &#4,n
This command will display byte n at all video locations.

Example:
100 X = &#4,191
is the same as:
100 FORN = 15360TO16383:POKEN,191:NEXT

X = &#5
This command will invert all video locations containing characters greater than 127.

X = &#6,address,n,x,x,x,etc.
This command allows POKEing multiple bytes into memory.
address = starting address
n = number of bytes to be POKEd
X = byte to be POKEd

Example:
100 X = &#6,27000,7,205,127,10,41,195,15,10
is the same as:
100 POKE27000,205:POKE27001,127:POKE27002,10:POKE27003,41:POKE27004,195:POKE27005,15:POKE27006,10

X = &#7,address
This command allows fetching from memory a 16 bit word.

Example:
100 X = &#7,27000
is the same as:
100 X = PEEK(27000) + PEEK(27001)*256

X = &#8,address,(value)
This command allows POKEing a 16 bit word into memory
address = address to be POKEd
value = 16 bit word to be POKEd

Example:
100 X = &#8,27000,(27000)
is the same as:
100 POKE27000,120:POKE27001,105

X = &#9,x,y,length
This command allows the drawing of a horizontal line.
x = horizontal starting position
y = vertical position of the line
length = length of the line

Example:
100 X = &#9,0,0,128
is the same as:
100 FORX = 0TO127:SET(X,0):NEXT

X = &#10,x,y,length
This command allows the drawing of a vertical line.
x = horizontal position of the line
y = vertical starting position
length = length of the line

Example:
100 X = &#10,0,0,48
is the same as:
100 FORY = 0TO47:SET(0,Y):NEXT

X = &#11,x,y,length
This command allows the erasing of a horizontal line.
x = horizontal starting position
y = vertical position of the line
length = length of the line

Example:
100 X = &#11,0,0,128
is the same as:
100 FORX = 0TO127:RESET(X,0):NEXT

X = &#12,x,y,length

Table continues

Table continued

This command allows the erasing of a vertical line.
x = horizontal position of the line.
y = vertical starting position
length = length of the line

Example:
100 X = &#12,0,0,48
is the same as:
100 FORY = 0TO47:RESET(0,Y)NEXT

Program Listing 1.

```
42E9                      ORG    42E9H
42E9 CDC901     START     CALL   1C9H         ;CLEARS SCRE
EN
42EC 21D244               LD     HL,M1        ;POINT HL TO
 MESSAGE
42EF CDA728               CALL   28A7H        ;DISPLAY MES
SAGE
42F2 21F744               LD     HL,BASIC     ;HL=START OF
 BASIC-1
42F5 3600                 LD     (HL),0
42F7 23                   INC    HL
42F8 22A440               LD     (40A4H),HL   ;START OF BA
SIC PROGRAM POINTER=HL
42FB 2AB140               LD     HL,(40B1H)   ;HL=TOP OF B
ASIC MEMORY POINTER
42FE 11CEFF               LD     DE,0FFCEH
4301 19                   ADD    HL,DE        ;HL=HL-50
4302 22A040               LD     (40A0H),HL   ;START OF ST
RING SPACE POINTER=HL
4305 CD4D1B               CALL   1B4DH        ;RESET POINT
ERS
4308 CD7644               CALL   LEV3         ;CALL LINKUP
 ROUTINE
430B C3191A               JP     1A19H        ;RETURN TO B
ASIC
430E CD9D0A     DUMMY     CALL   0A9DH        ;SET NTF TO
INTEGER
4311 210000               LD     HL,0
4314 CD9A0A               CALL   0A9AH        ;LOAD ACCUM
WITH ZERO
4317 2AE640     RET1      LD     HL,(40E6H)   ;HL=ENCODED
STATEMUNT POINTER
431A C9                   RET
431B F5         GRAPH     PUSH   AF           ;SAVE GRAPHI
CS MODE TO STACK
431C 7A                   LD     A,D          ;A=X VALUE
431D F5                   PUSH   AF           ;SAVE IT
431E 7B                   LD     A,E          ;A=Y VALUE
431F 218D01               LD     HL,18DH      ;POINT TO DU
MMY STRING
4322 C35001               JP     150H         ;GOTO GRAPHI
CS ROUTINE
4325 CF         SUB1      RST    08H          ;CHECK SYNTA
X
4326 2C                   DEFB   ','          ;MUST BE A ,

4327 CF                   RST    08H          ;CHECK SYNTA
X
4328 28                   DEFB   '('          ;MUST BE A (

4329 CD1C2B               CALL   2B1CH        ;EVALUATE EX
PRESSION
432C C9                   RET
432D CD2543     SUB       CALL   SUB1
4330 FE40                 CP     40H          ;SEE IF X>63

4332 D24A1E               JP     NC,1E4AH     ;IF>63 THEN
FC ERROR
4335 F5                   PUSH   AF           ;SAVE X VALU
E
4336 CD172B               CALL   2B17H        ;EVALUATE EX
PRESSION
4339 FE30                 CP     30H          ;SEE IF Y>47

433B D24A1E               JP     NC,1E4AH     ;IF>47 THEN
FC ERROR
433E F5                   PUSH   AF           ;SAVE Y VALU
E
433F CD8C01               CALL   18CH         ;CHECK SYNTA
X
4342 22E640               LD     (40E6H),HL   ;ENCODED STA
TEMENT POINTER=HL
4345 F1                   POP    AF           ;GET Y VALUE

4346 5F                   LD     E,A          ;SAVE IN E
4347 F1                   POP    AF           ;GET X VALUE

4348 87                   ADD    A,A          ;X=X*2
4349 57                   LD     D,A          ;SAVE IN D
434A C9                   RET
434B CD2D43     SET2      CALL   SUB
434E D5                   PUSH   DE           ;SAVE X,Y VA
LUE
434F 3E80                 LD     A,80H        ;A=SET GRAPH
ICS MODE
4351 CD1B43               CALL   GRAPH
4354 D1                   POP    DE           ;GET X,Y VAL
UE
4355 14                   INC    D            ;X=X+1
4356 3E80                 LD     A,80H        ;A=SET GRAPH
```

Program continues

✓See List of Advertisers on page 402

Program continued

```
ICS MODE
4358 CD1B43     00650          CALL    GRAPH
435B C30E43     00660          JP      DUMMY           ;RETURN TO B
ASIC PROGRAM
435E CD2D43     00670 RES2     CALL    SUB
4361 D5         00680          PUSH    DE              ;SAVE X,Y VA
LUE
4362 3E01       00690          LD      A,1             ;A=RESET GRA
PHICS MODE
4364 CD1B43     00700          CALL    GRAPH
4367 D1         00710          POP     DE              ;GET X,Y VAL
UE
4368 14         00720          INC     D               ;X=X+1
4369 3E01       00730          LD      A,1             ;A=RESET GRA
PHICS MODE
436B CD1B43     00740          CALL    GRAPH
436E C30E43     00750          JP      DUMMY           ;RETURN TO B
ASIC PROGRAM
4371 CD2D43     00760 POI2     CALL    SUB
4374 AF         00770          XOR     A               ;A=POINT GRA
PHICS MODE
4375 CD1B43     00780          CALL    GRAPH
4378 CD7F0A     00790          CALL    0A7FH           ;CONVERT ACC
UM TO INTEGER
437B CD9A0A     00800          CALL    0A9AH
437E C31743     00810          JP      RET1            ;RETURN TO B
ASIC PROGRAM WITH VALUE
4381 CD172B     00820 WHITE    CALL    2B17H           ;EVALUATE EX
PRESSION
4384 22E640     00830          LD      (40E6H),HL      ;ENCODED STA
TEMENT POINTER=HL
4387 21003C     00840          LD      HL,3C00H        ;HL=START OF
 VIDEO MEMORY
438A 010004     00850          LD      BC,1024         ;BC=LENGTH O
F VIDEO MEMORY
438D 77         00860 WHI1     LD      (HL),A          ;DISPLAY BYT
E AT (HL)
438E 23         00870          INC     HL
438F 0B         00880          DEC     BC
4390 57         00890          LD      D,A             ;SAVE BYTE T
O BE DISPLAYED
4391 78         00900          LD      A,B             ;CHECK
4392 B1         00910          OR      C               ;  FOR END O
F MEMORY
4393 7A         00920          LD      A,D
4394 20F7       00930          JR      NZ,WHI1         ;CONTINUE UN
TIL END OF VIDEO MEMORY
4396 C30E43     00940          JP      DUMMY
4399 22E640     00950 REV      LD      (40E6H),HL
439C 21003C     00960          LD      HL,3C00H        ;HL=START OF
 VIDEO MEMORY
439F 010004     00970          LD      BC,0400H        ;BC=LENGTH O
F VIDEO MEMORY
43A2 7E         00980 REV2     LD      A,(HL)          ;A=VIDEO MEM
ORY (HL)
43A3 FE80       00990          CP      80H             ;CHECK FOR G
RAPHICS CODE
43A5 3806       01000          JR      C,REV1          ;IF NOT GOTO
 REV1
43A7 2F         01010          CPL                     ;INVERT A
43A8 CBFF       01020          SET     7,A             ;  SET BIT 7

43AA CBB7       01030          RES     6,A             ;   RESET B
IT 6
43AC 77         01040          LD      (HL),A          ;DISPLAY BYT
E
43AD 23         01050 REV1     INC     HL
43AE 0B         01060          DEC     BC
43AF 78         01070          LD      A,B             ;CHECK
43B0 B1         01080          OR      C               ;  FOR END O
F VIDEO MEMORY
43B1 20EF       01090          JR      NZ,REV2         ;    IF NOT
JUMP
43B3 C30E43     01100          JP      DUMMY
43B6 CF         01110 POKE     RST     08H             ;CHECK SYNTA
X
43B7 2C         01120          DEFB    ','             ;MUST BE A ,

43B8 CD022B     01130          CALL    2B02H           ;GET STARTIN
G ADDRESS
43BB D5         01140          PUSH    DE              ;SAVE IT
43BC CD172B     01150          CALL    2B17H           ;GET NUMBER
OF BYTES
43BF F5         01160          PUSH    AF              ;SAVE IT
43C0 22E640     01170          LD      (40E6H),HL
43C3 1801       01180          JR      POK1
43C5 C5         01190 POK2     PUSH    BC
43C6 2AE640     01200 POK1     LD      HL,(40E6H)
43C9 CD172B     01210          CALL    2B17H           ;GET BYTE TO
 BE POKED
43CC 22E640     01220          LD      (40E6H),HL
43CF C1         01230          POP     BC
43D0 E1         01240          POP     HL
43D1 77         01250          LD      (HL),A          ;POKE BYTE
43D2 23         01260          INC     HL
43D3 E5         01270          PUSH    HL
43D4 10EF       01280          DJNZ    POK2            ;IF BYTES LE
FT JUMP
43D6 E1         01290          POP     HL
43D7 C30E43     01300          JP      DUMMY
43DA CD012B     01310 USR      CALL    2B01H           ;GET ADDRESS

43DD ED538E40   01320          LD      (408EH),DE      ;SAVE IN ENT
RY POINT LOCATTION
43E1 CF         01330          RST     08H             ;CHECK SYNTA
X
43E2 2C         01340          DEFB    ','             ;MUST BE A ,

43E3 2B         01350          DEC     HL
43E4 C9         01360          RET
43E5 CF         01370 FETCH    RST     08H             ;CHECK SYNTA
X
```

Program continues

Program continued

```
43E6 2C         01380          DEFB    ','             ;MUST BE A ,

43E7 CD022B     01390          CALL    2B02H           ;GET ADDRESS

43EA 22E640     01400          LD      (40E6H),HL
43ED D5         01410          PUSH    DE
43EE E1         01420          POP     HL
43EF 7E         01430          LD      A,(HL)
43F0 4F         01440          LD      C,A             ;C=LSB
43F1 23         01450          INC     HL
43F2 7E         01460          LD      A,(HL)
43F3 47         01470          LD      B,A             ;B=MSB
43F4 C5         01480          PUSH    BC
43F5 E1         01490          POP     HL
43F6 CD9A0A     01500          CALL    0A9AH           ;SAVE VALUE
43F9 C31743     01510          JP      RET1            ;RETURN WITH
 VALUE
43FC CF         01520 STORE    RST     08H             ;CHECK SYNTA
X
43FD 2C         01530          DEFB    ','             ;MUST BE A ,

43FE CD022B     01540          CALL    2B02H           ;GET ADDRESS

4401 D5         01550          PUSH    DE
4402 CF         01560          RST     08H             ;CHECK SYNTA
X
4403 2C         01570          DEFB    ','             ;MUST BE A ,

4404 CF         01580          RST     08H             ;CHECK SYNTA
X
4405 28         01590          DEFB    '('             ;MUST BE A (

4406 CD022B     01600          CALL    2B02H           ;GET NUMBER
4409 D5         01610          PUSH    DE
440A CD8C01     01620          CALL    18CH            ;CHECK SYNTA
X
440D 22E640     01630          LD      (40E6H),HL
4410 D1         01640          POP     DE
4411 E1         01650          POP     HL
4412 7B         01660          LD      A,E
4413 77         01670          LD      (HL),A          ;POKE LSB
4414 23         01680          INC     HL
4415 7A         01690          LD      A,D
4416 77         01700          LD      (HL),A          ;POKE MSB
4417 C30E43     01710          JP      DUMMY
441A CD172B     01720 LINES    CALL    2B17H           ;GET X VALUE

441D F5         01730          PUSH    AF
441E CD172B     01740          CALL    2B17H           ;GET Y VALUE

4421 F5         01750          PUSH    AF
4422 CD172B     01760          CALL    2B17H           ;GET LENGTH
4425 22E640     01770          LD      (40E6H),HL
4428 47         01780          LD      B,A             ;B=LENGTH
4429 F1         01790          POP     AF
442A 5F         01800          LD      E,A             ;E=Y VALUE
442B F1         01810          POP     AF
442C 57         01820          LD      D,A             ;D=X VALUE
442D C9         01830          RET
442E CD1A44     01840 HLINS    CALL    LINES
4431 3E80       01850 HLI1     LD      A,80H           ;A=SET GRAPH
ICS MODE
4433 C5         01860          PUSH    BC
4434 D5         01870          PUSH    DE
4435 CD1B43     01880          CALL    GRAPH
4438 D1         01890          POP     DE
4439 C1         01900          POP     BC
443A 14         01910          INC     D               ;X=X+1
443B 10F4       01920          DJNZ    HLI1            ;LOOP TILL L
INE SET
443D C30E43     01930          JP      DUMMY
4440 CD1A44     01940 VLINS    CALL    LINES
4443 3E80       01950 VLI1     LD      A,80H           ;A=SET GRAPH
ICS MODE
4445 C5         01960          PUSH    BC
4446 D5         01970          PUSH    DE
4447 CD1B43     01980          CALL    GRAPH
444A D1         01990          POP     DE
444B C1         02000          POP     BC
444C 1C         02010          INC     E               ;Y=Y+1
444D 10F4       02020          DJNZ    VLI1            ;LOOP TILL L
INE SET
444F C30E43     02030          JP      DUMMY
4452 CD1A44     02040 HLINR    CALL    LINES
4455 3E01       02050 HLI2     LD      A,1             ;A=RESET GRA
PHICS MODE
4457 C5         02060          PUSH    BC
4458 D5         02070          PUSH    DE
4459 CD1B43     02080          CALL    GRAPH
445C D1         02090          POP     DE
445D C1         02100          POP     BC
445E 14         02110          INC     D               ;X=X+1
445F 10F4       02120          DJNZ    HLI2            ;LOOP TILL L
INE RESET
4461 C30E43     02130          JP      DUMMY
4464 CD1A44     02140 VLINR    CALL    LINES
4467 3E01       02150 VLI2     LD      A,1             ;A=RESET GRA
PHICS MODE
4469 C5         02160          PUSH    BC
446A D5         02170          PUSH    DE
446B CD1B43     02180          CALL    GRAPH
446E D1         02190          POP     DE
446F C1         02200          POP     BC
4470 1C         02210          INC     E               ;Y=Y+1
4471 10F4       02220          DJNZ    VLI2            ;LOOP TILL L
INE RESET
4473 C30E43     02230          JP      DUMMY
4476 3EC3       02240 LEV3     LD      A,0C3H
4478 329441     02250          LD      (4194H),A       ;SET UP
447B 218B44     02260          LD      HL,PT           ;  &
447E 229541     02270          LD      (4195H),HL      ;     LINK
```

Program continues

Program continued

```
4481 32A941    02280          LD      (41A9H),A        ;SET UP
4484 21DA43    02290          LD      HL,USR           ;  USR
4487 22AA41    02300          LD      (41AAH),HL       ;    LINK
448A C9        02310          RET
448B D7        02320 PT       RST     10H
448C CF        02330          RST     08H              ;CHECK SYNTA
X
448D 23        02340          DEFB    '#'              ;MUST BE A #

448E CD1C2B    02350          CALL    2B1CH            ;GET COMMAND
 NUMBER
4491 FE0D      02360          CP      13
4493 D24A1E    02370          JP      NC,1E4AH         ;IF COMMAND
#>12 THEN FC ERROR
4496 FE01      02380          CP      1
4498 CA4B43    02390          JP      Z,SET2
449B FE02      02400          CP      2
449D CA5E43    02410          JP      Z,RES2
44A0 FE03      02420          CP      3
44A2 CA7143    02430          JP      Z,POI2
44A5 FE04      02440          CP      4
44A7 CA8143    02450          JP      Z,WHITE
44AA FE05      02460          CP      5
44AC CA9943    02470          JP      Z,REV
44AF FE06      02480          CP      6
44B1 CAB643    02490          JP      Z,POKE
44B4 FE07      02500          CP      7
44B6 CAE543    02510          JP      Z,FETCH
44B9 FE08      02520          CP      8
44BB CAFC43    02530          JP      Z,STORE
44BE FE09      02540          CP      9
44C0 CA2E44    02550          JP      Z,HLINS
44C3 FE0A      02560          CP      10
44C5 CA4044    02570          JP      Z,VLINS
44C8 FE0B      02580          CP      11
44CA CA5244    02590          JP      Z,HLINR
44CD FE0C      02600          CP      12
44CF CA6444    02610          JP      Z,VLINR
44D2 45        02620 M1       DEFM    'ENHANCED BASIC V1.0'
44E5 0D        02630          DEFB    0DH
44E6 42        02640          DEFM    'BY MARK GOODWIN'
44F5 0D00      02650          DEFW    0DH
44F7 00        02660 BASIC    DEFB    0
42E9           02670          END     START
```

```
10 CLS
20 X=&#6,27000,7,205,127,10,41,195,154,10
30 FORN=1TO10:PRINTN;"*2=";USR27000,(N):NEXT
```

Program Listing 2

```
10 CLS
20 FORN=0TO63
30 FORN1=0TO47STEP2
40 X=&#1,(N,N1)
50 NEXTN1,N
60 FORN=0TO63
70 FORN1=0TO47
80 IF&#3,(N,N1)THENX=&#2,(N,N1)ELSEX=&#1,(N,N1)
90 NEXTN1,N
100 GOTO60
```

Program Listing 3

```
10 CLS
20 X=&#4,190:PRINT@0,"HELLO";
30 IFINKEY$=""THEN30
40 X=&#5
50 GOTO50
```

Program Listing 4

```
10 CLS
20 X=&#8,27000,(12345)
30 PRINT&#7,27000
```

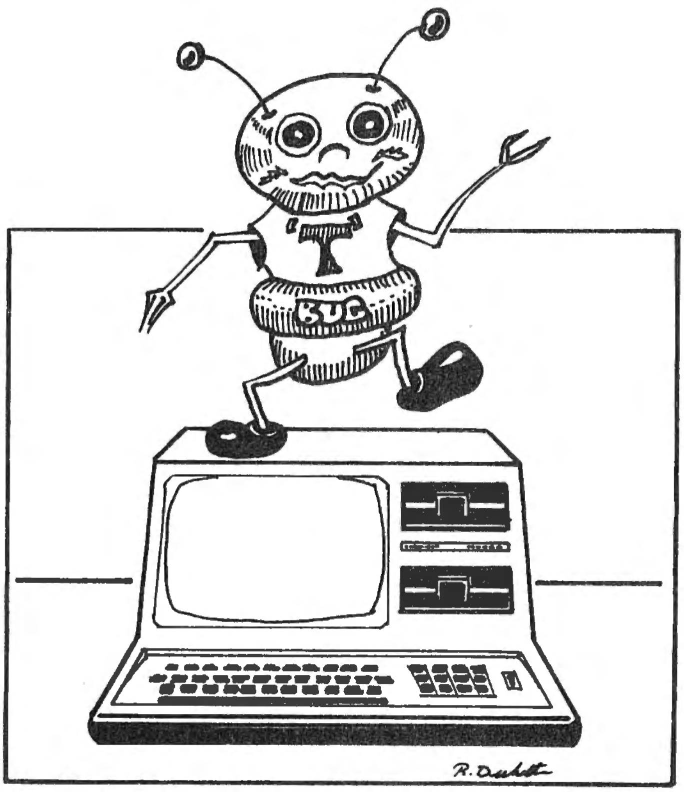
Program Listing 5

```
10 CLS
20 FORN=0TO47:X=&#9,0,N,128:NEXT
30 FORN=47TO0STEP-1:X=&#11,0,N,128:NEXT
40 FORN=0TO127:X=&#10,N,0,48:NEXT
50 FORN=127TO0STEP-1:X=&#12,N,0,48:NEXT
```

Program Listing 6

With a mite of effort, it'll work on your Model III!

T-Bug III



Kevin Kleinfelter
3337 Sevier Ave, Apt. 3
Knoxville, TN 37920

When you bought your Model III you may have been told, "And we'll have T-Bug for Model III very soon." Well, after three months of waiting, I finally gave up and fixed it myself. This article will show how easily you can make T-Bug run on your Model III.

A First Attempt

Unwilling to believe Radio Shack, I loaded my Level II T-Bug into my Model III and tried it out. Voila...all the commands worked except the tape routines. Upon studying dumps of T-Bug and ROM, I learned that the tape I/O performed by T-Bug is completely incompatible with Model III. The solution? Replace T-Bug's tape routines with Model III ROM routines.

Step 1

Load T-Bug at 500 baud. The punch command's tape routines are replaced by ROM calls here. In place of T-Bug's routines, put ROM's write header, write byte, and cassette off routines. Use T-Bug's memory command to make the changes shown in Fig. 1.

ADDR	INSERT
46DD	000000
46EB	CD8702
46EE	180B
46FE	6402
4706	6402
4713	6402
4717	6402
4725	6402
4729	6402
4731	6402
4737	6402
473D	6402
4740	F801
4746	6402
474A	6402
4750	6402
4759	6402

Fig. 1

ADDR	INSERT
4643	000000
4647	9602
4649	1807
464B	AF
464C	00000000
4653	3502
465C	3502
4661	3F3C
4666	3F3C
4669	3502
4674	3502
467D	3502
4686	3502
4699	F801
469D	3502
46A1	3502

Fig. 2

Steps 2 and 3

Now the T-Bug routines for the Load command are replaced by calls to ROM's read header, read byte, and cassette off. Once again use the memory command (see Fig. 2).

Finally, use the memory command to change location 4211 (hex) to anything but 00. This sets the cassette speed to 1500 baud. Set up the tape recorder and type "P 4380 4980 43A0 TBUG" and hit Enter. While you're at it, why not punch out several copies so you won't have to rewind after each load?

As with all new programs, test your new T-Bug thoroughly. Also, you probably will want to look up one of those articles about relocating T-Bug to high memory.

Another Choice

Alternatively, you can use an assembler (perhaps EDTASM modified to run on Model III) to make a patch tape. Load EDTASM, then the patch tape, and then take T-Bug and punch out several copies. Now you have your own T-Bug III! ■

✓See List of Advertisers on page 402

A median-length statistics program for those who have deviated from big computers.

Pocket Stats

Dr. Walter J. Atkins, Jr.
Qtrs. 4410A
USAF Academy, CO 80840

As an educator and scientist, I frequently need to find statistics for a set of data. I have found my TRS-80 Pocket Computer ideal for making these calculations.

This program will find and display a great deal of statistical information. It will accept a data set of up to approximately sixty values, and will find the average value, the variance, the standard deviation, the median value, the high and low data values, and the midrange of the set.

I had just finished writing and testing this program when I read Len Gorney's article, "On the Average," in the May 1981 issue of *80*. That prompted me to include the midrange calculation in this program. His article is also a good reference for explanations of the mean, midrange, and median.

The variance and the standard deviation are statistics that indicate how much data values deviate from the mean. The standard deviation is the square root of the variance. One use of the standard deviation is in assigning grades to test scores. The mean value of the students' scores is assigned the grade of C. The scores in the B range lie one standard deviation above the mean. The scores in the A range lie two standard deviations above the mean. Likewise, scores assigned D and F are one and two standard deviations below the mean.

The following numbers represent the scores that a group of students received on a test; we'll use the Pocket Computer to find the statistics for these grades. The students' grades are 47, 52, 57, 65, 72, 77, 55, 61, 85, 87, 66, 75, 98, 88, and 73.

The mean value of the fifteen scores is 70.533. The variance is 219.552. The standard deviation is 14.817. The highest set value is 98. The lowest value is 47. Thus, the midrange value is 72.5. To calculate the median value, the computer first sorts the values into ascending order, which takes about one minute and thirteen seconds for the 15 data values used here. Since there is an odd number of values, the median is the value in the middle of the sorted list. In this case that is 72.

The portability of the Pocket Computer facilitates data analysis almost anywhere, I use my computer so much that it is never further away than my briefcase.

The Program

The program is 53 lines long. It is capable of storing a data set and its associated statistics so the data can be entered at one time and the statistic read out at another.

Line 150 allows the display of a previously stored data set and its associated statistics. Line 170 initializes program variables. Lines 180–240 enter and store the values. Lines 250–330 calculate and display all statistics except for the median value. Line 340 determines if the median is to be calculated. Lines 370–400 sort the values into ascending order. Lines 410–440 calculate and display the median value. Line 450 allows the data entered to be stored in the Pocket Computer's permanent memory for later recall. Lines 500–530 display a stored data set.

The variables used in the program are shown in Table 1. ■

A	
B	
C	
D	— Data value
E	
F	
G	— Sum of data values
H	— High data value
I	— Sum of squared data values
J	
K	— Sort flag
L	— Low data value
M	
N	— Number of data points
O	
P	— Mean value
Q	— Median value
R	
S	— Standard Deviation value
T	— Store data flag
U	— Midrange value
V	— Variance value
W	— Median routine pointer
X	— Loop counter
Y	— Storage array pointer
Z	— Median calculated flag

Table 1.

Statistics Program Variables

```
100  REM *STATISTICS*
110  REM *W.J.ATKINS*
120  REM *COPYRIGHTED*
130  REM *MAY81*
140  PAUSE "* STATISTICS *"
150  IF T=1 INPUT "DSPLY STORED DATA (Y/N)";R$:IF R$="Y"
     THEN 500
160  IF T=1 INPUT"CLEAR DATA (Y/N)?";R$:IF R$="N" THEN 150
170  D=0:G=0:H=0:I=0:N=0:P=0:Q=0:S=0:T=0:U=0:Z=0
180  INPUT "HOW MANY DATA VALUES? ";N
190  Y=27:L=999999
200  FOR X=1 TO N
210  INPUT "DATA VALUE=> ";D:A(Y)=D:Y=Y+1
220  G=G+D:IF D>H LET H=D
230  I=I+D*D:IF D<L LET L=D
240  NEXT X
250  P=G/N
260  V=(I-G*G/N)/(N-1):S= √ V
270  USING "####.###"
280  PRINT "AVERAGE (MEAN)=";P
290  PRINT "VARIANCE=";V
300  PRINT "STANDARD DEV.=";S
310  PRINT "HIGH DATA VALUE=";H
320  PRINT "LOW DATA VALUE=";L
330  U=L+((H-L)/2):PRINT "MIDRANGE VALUE=";U
340  IF Z=0 INPUT "COMPUTE MEDIAN (Y/N)?";R$:W=INT(N/2):
     IF R$="N" THEN 450
350  IF Z=1 THEN 440
360  PAUSE "PLEASE WAIT"
370  K=0:FOR X=27 TO N+26
380  IF A(X)>A(X+1) LET A=A(X):B=A(X+1):A(X)=B:A(X+1)=A:K=1
390  NEXT X
400  IF K=1 THEN 370
410  IF W*2=N LET Q=(A(W+26)+A(W+27))/2:GOTO 430
420  Q=A(W+27)
430  BEEP 2
440  PRINT "MEDIAN=";Q:Z=1
450  INPUT "RETAIN DATA (Y/N)?";R$:IF R$="N" LET T=0:
     GOTO 470
410  T=1
420  GOTO 140
500  REM *DATADISPLAY*
505  PRINT "NUMBER OF DATA=";N
510  FOR X=27 TO N+26
520  PRINT "DATA=";A(X):NEXT X
530  GOTO 270
```

Program Listing. Pocket Computer Statistics Program

RELOAD 80

You've noticed those LOAD80 tape files with the note, "Needs EDTASM" in the comments column. These are machine language programs in source code format. Radio Shack's EDiTor/ASseMbler (EDTASM) is required to load and use these files. If you have ignored them because you didn't want to take the time to learn how to use the Editor/Assembler here is a quick tutorial to help you get started.

The Radio Shack Editor/Assembler for the Model I and the newly released package for the Model III both use the same command structure. There are tape and disk versions for both TRS-80 Models. The machine language programs in *80 Microcomputing* are *always* printed in EDTASM format. These programs are listed as assembly language source code files. These programs must be entered into the computer using the Editor/Assembler, either by hand or from tape, and then assembled and recorded on a separate tape (or disk) before they can be run. Remember, a source code file will *not* run—it must be assembled first.

To load a LOAD80 source code file, first load your EDTASM program. Once loaded, you should see the asterisk (*) prompt. Now, type L (or LT or LD with a disk based version), followed by an equals (=) sign, and a filename. (If you press the space bar five or six times in place of the filename, EDTASM will load the first file it encounters—for tape files only.) Then press PLAY on your tape recorder to load your file. The familiar asterisk pair will appear and one will flash on and off as the file loads.

When the source code file is loaded, enter P#:* to list the file on the screen. Inspect this listing as you would a Basic program to ensure that you have a good load. If the load is bad, reload the file. Data will be in the buffer and EDTASM will ask if you are concatenating. Concatenating means combining two files to make one. In most cases, you will answer the concatenating question, N for 'no.'

Now that your file is loaded and verified, you can assemble the program by answering 'A' to the * prompt. After the 'A,' add a filename (six characters or less). The file will then be assembled and EDTASM will prompt you to ready your cassette for the object code file. Your EDTASM documentation contains some optional parameters that you can specify, using the slash (/) as a delimiter. These optional parameters include: NO for no object code, NS for no symbol table, LP for output to the lineprinter, and so on.

Once you have made an object code tape, you can load the tape using the Basic SYSTEM command to load machine language programs. If the program fails to perform as you desire, reload the original source code file and use EDTASM's edit function to modify the program to suit you. Then assemble and run the new version and see how it goes!

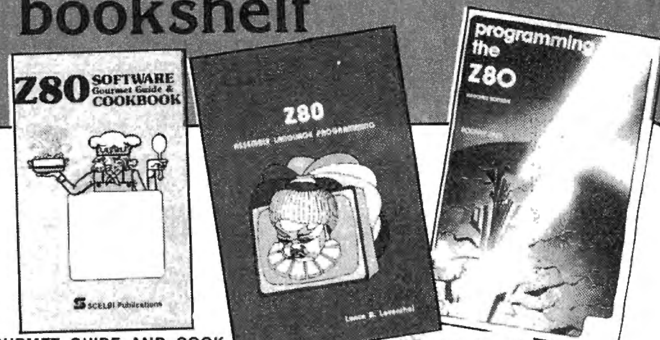
Our Circulation Department reports that the custom software written to handle the LOAD80 subscription program is nearly ready. The new year will probably see a subscription offer for LOAD80. Details forthcoming in December.

Next month, Holiday goodies for LOAD80 users.■

November LOAD80 Directory

Program	Title	Page #	Comments
1	TENANT	138	None
2	INVEST	144	None
3	HOUSE	160	None
4	GRAPHS	176	None
5	PRODTIMER	220	None
6	CHAIN	250	Needs EDTASM
7	ALTER	274	None
8	CREDUN	321	None
9	SPRLIST	333	None
10	SECSRC	348	None
11	NEWBUG	368	Needs EDTASM
12	ENHANCE	384	None

80 microcomputing™ bookshelf



• ***MICROCOMPUTING* CODING SHEETS** *Microcomputing*'s dozen or so programmers wouldn't try to work without these handy scratch pads, which help prevent the little errors that can cost hours and hours of programming time. Available for programming is Assembly/Machine Language (PD1001), which has columns for address, instruction (3 bytes), source code (label, op code, operand) and comments; and for BASIC (PD1002) which is 72 columns wide. 50 sheets to a pad. $2.39.*

PROGRAMMING —Z-80—

• **TRS-80 ASSEMBLY LANGUAGE**—BK1217—by Hubert S. Howe, Jr. This book incorporates into a single volume all the pertinent facts and information you need to know to program and enjoy the TRS-80. Included are clear presentations of all introductory concepts, completely tested practical programs and subroutines, details of ROM and RAM and disk operating systems, plus comprehensive tables, charts and appendices. Suitable for the first time user or more experienced users. $9.95.*

• **INSIDE LEVEL II**—BK1183—For machine language programmers! This is a comprehensive reference guide to the Level II ROMs, allowing easy utilization of the sophisticated routines they contain. It concisely explains set-ups, calling sequences, variable passage and I/O routines. Part II presents an entirely new composite program structure which unloads under the SYSTEM command and executes in both BASIC and machine code with the speed and efficiency of a compiler. Special consideration is given to disk systems $15.95.*

• **PROGRAMMING THE Z-80**—BK1122—by Rodnay Zaks. Here is assembly language programming for the Z-80 presented as a progressive, step-by-step course. This book is both an educational text *and* a self-contained reference book, useful to both the beginning and the experienced programmer who wish to learn about the Z-80. Exercises to test the reader are included. $14.95 *

• **Z-80 SOFTWARE GOURMET GUIDE AND COOKBOOK**—BK1045—by Nat Wadsworth. Scelbi's newest cookbook! This book contains a complete description of the powerful Z-80 instruction set and a wide variety of programming information. Use the author's ingredients including routines, subroutines and short programs, choose a time-tested recipe and start cooking! $16.99.*

• **Z-80 ASSEMBLY LANGUAGE PROGRAMMING**—BK1177—by Lance A. Leventhal. This book thoroughly covers the Z-80 instruction set, abounding in simple programming examples which illustrate software development concepts and actual assembly language usage. Features include Z-80 I/O devices and interfacing methods, assembler conventions, and comparisons with 8080A/8085 instruction sets and interrupt structure. $16.99.*

6809

• **6809 MICROCOMPUTER PROGRAMMING AND INTERFACING**—BK1215—by Andrew C. Staugaard, Jr. Getting involved with Tandy's new Color Computer? If so, this new book from the Blacksburg Group will allow you to exploit the awesome power of the machine's 6809 microprocessor. Detailed information on processor architecture, addressing modes, register operation, data movement, arithmetic logic operations, I/O and interfacing is provided, as well as a review section at the end of each chapter. Four appendices are included covering the 6809 instruction set, specification sheets of the 6809 family of processors, other 6800 series equipment and the 6809/6821 Peripheral Interface Adapter. This book is a must for the serious Color Computer owner. $13.95 *

A NEW PROGRAMMING SERIES

The Microprocessor Software Engineering Series by John Zarrella provides common sense descriptions of advanced computer system topics for engineers, programmers and development managers. Each volume is a self-contained review of a software engineering topic, explaining fundamental concepts in easy-to-understand language and describing sophisticated software tools and techniques. Detailed glossary of technical jargon is included in each volume. This series will help you find the solutions to your software problems.

• **WORD PROCESSING AND TEXT EDITING**—BK1194—Provides a firm basis for understanding word processing terminology and for comparing systems $7.95 *

• **SYSTEM ARCHITECTURE**—BK1195 —Presents a detailed overview of advanced computer system design including object architecture and capability-based addressing. $9.95.*

• **OPERATING SYSTEMS: CONCEPTS AND PRINCIPLES**—BK1193—Presents an overview of the basic operating system types, their components and capabilities $7.95 *

ALL NEW—FOR 80 OWNERS ENCYCLOPEDIA FOR THE TRS-80*

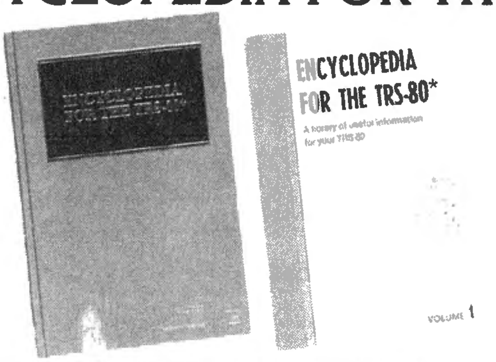
ENCYCLOPEDIA FOR THE TRS-80* is the newest peripheral for your microcomputer. This ten-volume reference series contains programs and articles especially selected to extend the documentation that comes with the TRS-80. Volumes of the **Encyclopedia** are being issued one-at-a-time, approximately six weeks apart. This means that each new volume will reflect the latest developments in TRS-80 microcomputing. **Encyclopedia for the TRS-80** is available in two editions. The deluxe COLLECTOR'S EDITION has a handsome green and black hard cover with a dust jacket. A soft cover edition is also available.

To order single hard cover volumes of the **Encyclopedia** please specify:

Volume 1	EN8101	$19.95*
Volume 2	EN8102	$19.95*
Volume 3 (avail. Nov.)	EN8103	$19.95*
Volume 4 (avail. Jan.)	EN8104	$19.95*
Volume 5 (avail. Feb.)	EN8105	$19.95*

To order single softcover volumes of the **Encyclopedia** please specify:

Volume 1	EN8081	$10.95*
Volume 2	EN8082	$10.95*
Volume 3 (avail. Nov.)	EN8083	$10.95*
Volume 4 (avail. Jan.)	EN8084	$10.95*
Volume 5 (avail. Feb.)	EN8085	$10.95*



SPECIAL INTRODUCTORY OFFER
Order the entire set (the first ten volumes) of the **Encyclopedia for the TRS-80** and receive the tenth volume **FREE!**

10 Volume deluxe hard cover COLLECTOR'S EDITION —a $199.50 value.
EN8100 $164.00 postpaid

10 Volume soft cover edition, a $109.50 value.
EN8080 $83.00 postpaid

ENCYCLOPEDIA LOADER is the software companion to the Encyclopedia for the TRS-80. This special series of cassettes will allow you to load selected programs from the Encyclopedia instantly. Hours of keyboarding time and the aggravating search for typos is eliminated by using the **Encyclopedia Loader.**

Encyclopedia Loader for Volume 1 of the Encyclopedia of the TRS-80 is available at a **Special Introductory** Price of $12.95 (a $14.95 value). To take advantage of this offer specify:
EL800111 $12.95*

Additional Encyclopedia Loaders will be available as each volume of the Encyclopedia for the TRS-80 is issued. To order specify **Encyclopedia Loader:**

Volume 2	**EN8002**	**$14.95***
Volume 3	**EN8003**	**$14.95***
Volume 4	**EN8004**	**$14.95***
Volume 5	**EN8005**	**$14.95***

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to 80 Microcomputing Bookshelf • Peterborough NH 03458 Be sure to include check or detailed credit card information.

No C.O.D. orders accepted. All above add $1.50 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

80 microcomputing™ bookshelf

—INTRODUCTORY—

new

● **UNDERSTANDING AND PROGRAMMING MICROCOMPUTERS**—BK7382—A valuable addition to your computing library. This two-part text includes the best articles that have appeared in 73 and Kilobaud Microcomputing magazines on the hardware and software aspects of microcomputing. Well-known authors and well-structured text helps the reader get involved. $10.95*

● **SOME OF THE BEST FROM KILOBAUD MICROCOMPUTING**—BK7311—A collection of the best articles that have appeared in Kilobaud MICROCOMPUTING. Included is material on the TRS-80 and PET systems, CP/M, the 8080/8085/Z-80 chips, the ASR-33 terminal. Data base management, word processing, text editors and file structures are covered too. Programming techniques and hard-core hardware construction projects for modems, high speed cassette interfaces and TVTs are also included in this large format, 200 plus page edition. $10.95.*

● **YOUR FIRST COMPUTER**—BK1191—by Rodnay Zaks. Whether you are using a computer, thinking about using one or considering purchasing one, this book is indispensable. It explains what a computer system is, what it can do, how it works and how to select various components and peripheral units. It is written in everyday language and contains invaluable information for the novice and the experienced programmer. (The first edition of this book was published under the title "An Introduction to Personal and Business Computing".) $7.95*

● **MICROPROCESSOR INTERFACING TECHNIQUES**—BK1037—by Austin Lesea & Rodnay Zaks—will teach you how to interconnect a complete system and interface it to all the usual peripherals. It covers hardware and software skills and techniques, including the use and design of model buses such as the IEEE 488 or S-100. $15.95.*

● **HOBBY COMPUTERS ARE HERE!**—BK7322—If you want to come up to speed on how computers work... hardware and software... this is an excellent book. It starts with fundamentals and explains the circuits, and the basics of programming, along with a couple of TVT construction projects, ASCII-Baudot, etc. This book has the highest recommendations as a teaching aid. $4.95.*

● **THE NEW HOBBY COMPUTERS**—BK7340—This book takes it from where "HOBBY COMPUTERS ARE HERE!" leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards... and much, much more! Don't miss this tremendous value! Only $4.95.*

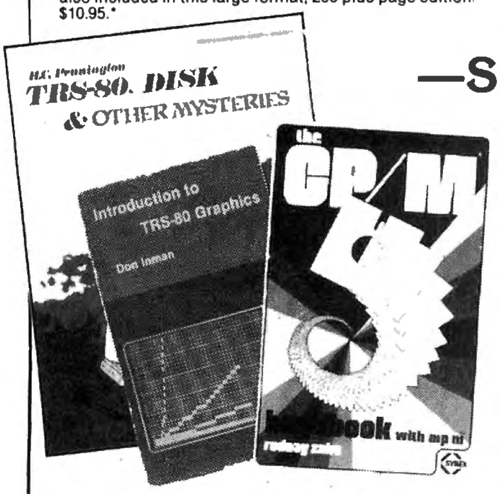
● **AN INTRODUCTION TO MICROCOMPUTERS, VOL. 0**—BK1130—The Beginner's Book—Written for readers who know nothing about computers—for those who have an interest in how to use computers—and for everyone else who must live with computers and should know a little about them. The first in a series of 4 volumes, this book will explain how computers work and what they can do. Computers have become an integral part of life and society. During any given day you are affected by computers, so start learning more about them with Volume 0. $7.95.*

● **VOL. I**—BK1030—**2nd Edition completely revised.** Dedicated to the basic concepts of microcomputers and hardware theory. The purpose of Volume I is to give you a thorough understanding of what microcomputers are. From basic concepts (which are covered in detail), Volume I builds the necessary components of a microcomputer system. This book highlights the difference between minicomputers and microcomputers. $12.99.*

● **VOL. II**—BK1040 (with binder)—Contains descriptions of individual microprocessors and support devices used only with the parent microprocessor. Volume II describes all available chips. $31.99*

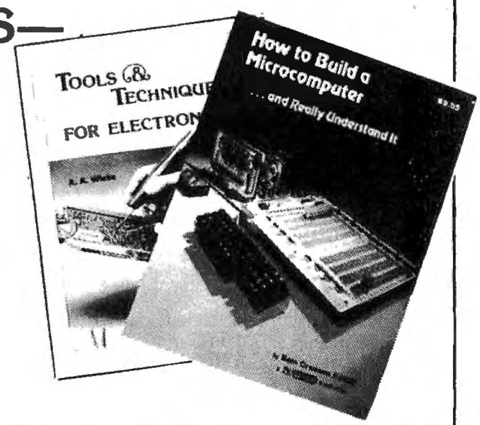
● **VOL. III**—BK1133 (with binder)—Contains descriptions of all support devices that can be used with any microprocessor. $21.99*

—SPECIAL INTERESTS—



● **THE CUSTOM TRS-80 AND OTHER MYSTERIES**—BK1218—by Dennis Kitsz. More than 300 pages of TRS-80 customizing information. With this book you'll be able to explore your computer like never before. Want to turn an 8 track into a mass storage unit? Individual reverse characters? Replace the BASIC ROMs? Make Music? High speed, reverse video, Level I and Level II? Fix it if it breaks down? All this and much, much more. Even if you have never used a soldering iron or read a circuit diagram, this book will teach you how! This is the definitive guide to customizing your 80! $29.95.*

● **BASIC FASTER AND BETTER AND OTHER MYSTERIES**—BK1221—by Lewis Rosenfelder. You don't have to learn assembly language to make your programs run fast. With the dozens of programming tricks and techniques in this book you can sort at high speed, swap screens in the twinkling of an eye, write INKEY routines that people think are in assembly language and add your own commands to BASIC. Find out how to write elegant code that makes your BASIC really hum, and explore the power of USR calls. $29.95.*



● **TRS-80 DISK AND OTHER MYSTERIES**—BK1181—by Harvard C. Pennington. This is the definitive work on the TRS-80 disk system. It is full of detailed "How to" information with examples, samples and in-depth explanations suitable for beginners and professionals alike. The recovery of one lost file is worth the price alone. $22.50.*

● **MICROSOFT BASIC DECODED AND OTHER MYSTERIES**—BK1186—by James Farvour. From the company that brought you *TRS-80 DISK AND OTHER MYSTERIES*! Contains more than 6500 lines of comments for the disassembled Level II ROMs, six additional chapters describing every BASIC subroutine, with assembly language routines showing how to use them. Flowcharts for all major routines give the reader a real insight into how the interpreter works. $29.50.

● **THE CP/M HANDBOOK (with MP/M)**—BK1187—by Rodnay Zaks. A complete guide and reference handbook for CP/M—the industry standard in operating systems. Step-by-step instruction for everything from turning on the system and inserting the diskette to correct user discipline and remedial action for problem situations. This also includes a complete discussion of all versions of CP/M up to and including 2.2, MP/M and CDOS. $14.95*.

● **INTRODUCTION TO TRS-80 GRAPHICS**—BK1180—by Don Inman. Dissatisfied with your Level I or Level II manual's coverage of graphics capabilities? This well-structured book (suitable for classroom use) is ideal for those who want to use all the graphics capabilities built into the TRS-80. A tutorial method is used with many demonstrations. It is based on the Level I, but all material is suitable for Level II use. $8.95.*

● **TOOLS & TECHNIQUES FOR ELECTRONICS**—BK7348—by A. A. Wicks is an easy-to-understand book written for the beginning kit builder as well as the experienced hobbyist. It has numerous pictures and descriptions of the safe and correct ways to use basic and specialized tools for electronic projects as well as specialized metal working tools and the chemical aids which are used in repair shops. $4.95.*

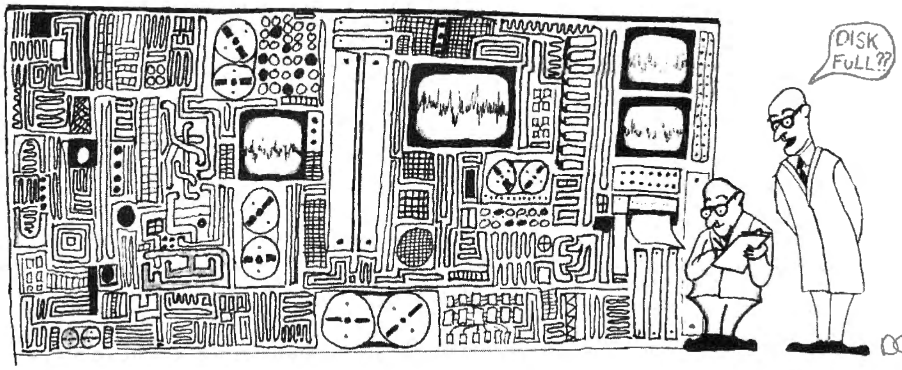
● **HOW TO BUILD A MICROCOMPUTER—AND REALLY UNDERSTAND IT**—BK7325—by Sam Creason. The electronics hobbyist who wants to build his own microcomputer system now has a practical "How-To" guidebook. This book is a combination technical manual and programming guide that takes the hobbyist step-by-step through the design, construction, testing and debugging of a complete microcomputer system. Must reading for anyone desiring a true understanding of small computer systems. $9.95.*

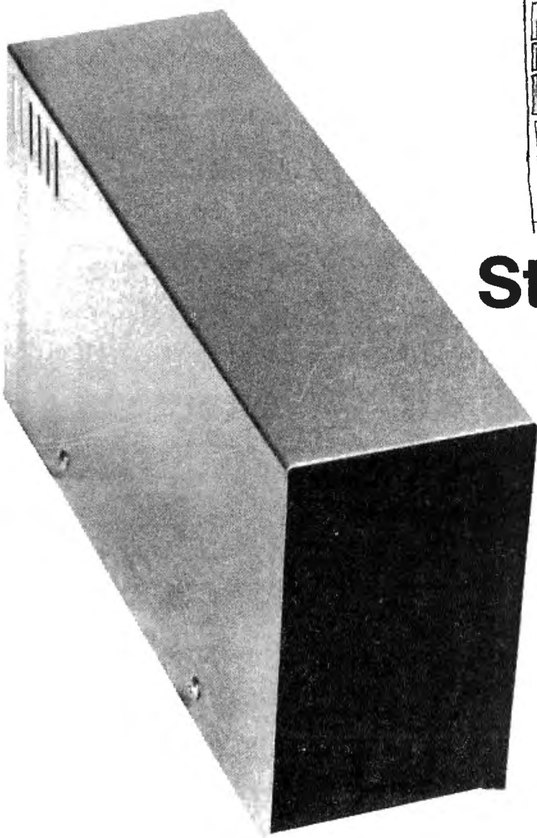
*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to 80 Microcomputing Bookshelf ● Peterborough NH 03458. Be sure to include check or detailed credit card information.

No C.O.D. orders accepted. All above add $1.50 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

PRICES SUBJECT TO CHANGE WITHOUT NOTICE





Still have storage problems?

Fully TRS-80® Compatible

(Model I or III)

Disk Drives

Model 148 250K, 1 Side, 40 Tracks **$321.00**
248 500K, 2 Sides, 40 Tracks per side $439.00
196 500K, 1 side, 80 Tracks $439.00
296 1 Meg, 2 Sides, 80 Tracks per side $590.00

Includes: Case and Power Supply — Fully Tested
10 Day Money Back Guarantee
90 Days Parts and Labor Warranty (Excl. Shpg.)

Standard Features Include

- 40/80 Tracks
- Speed constant ‹1½%
- Hi-Temp stability
- Single/double density
- Flippy Option Available

ADDS MORE POWER TO YOUR SYSTEM

Save time . . . Order by phone

(Orders only: 1-800-621-3229)

Information: 312-987-1024
Tech. assistance: 312-987-1032

Other Money Savings Opportunities Order by Phone or Mail

CABLES
2 Drive Cable **$29.00**
4 Drive Cable **$39.00**

Diskettes
Syncom . . . . 10 for **$35.00**
Plastic File Box **$3.95**

Operating Systems
TRSDOS 2.3 . . . . . . **$14.95**
40 Track Patch **$9.95**
NEWDOS +
40 Track **$99.00**
Newdos 80 2.0 . . . **$139.00**
Super Utility **$49.00**
Make 80 . . . . **$14.00**
Make 80 III **$29.00**

TRS-80
16K Model III **$899.00**
48K Model III
w/2 Disks **$1900.00**

Printers
Epson MX80 . . . . . **$550.00**
MX80 F/T **$675.00**
MX100 . . . . **$850.00**

16K Memory Kits
Prime NEC 200ns dynamic RAM. Comes with complete instructions **$39.00**

MIDWEST COMPUTER PERIPHERALS®

1467 S. MICHIGAN
CHICAGO, ILLINOIS 60605

✔ 8

TRS 80 TANDY CORP.

FREE! UPS GROUND SHIPPING ON ALL ORDERS OVER $100.00

Quantity	Description	$ each	Total
		6% IL Tax	
		TOTAL	

(Minimum order $50.00)

☐ Check enclosed
Bill my ☐ Visa ☐ Am Ex
☐ Master Charge
Acct. No. ____________ Exp. ________
Name ____________
Address ____________
City ____________ State ________ Zip ________

MIDWEST COMPUTER PERIPHERALS
1467 S. MICHIGAN AVE.
CHICAGO, IL 60605

Not Responsible for Typographical Errors. Prices Subject to Change w/o notice.
Drive Door May Not Be As Shown

Why use their flexible discs:

Athana, BASF, Control Data, Dysan, IBM, Maxell, Nashua, Scotch, Shugart, Syncom, 3M, Verbatim or Wabash

when you could be using

MEMOREX

for as low as $1.94 each?

Find the flexible disc you're now using on our cross reference list... then write down the equivalent Memorex part number you should be ordering.

Product Family	Product Description	Memorex Part Number (3201-)	CE quant. 100 price per disc ($)	Athana	BASF	Dysan	IBM	Maxell	Nashua	Scotch 3M	Shugart	Syncom	Verbatim	Wabash	Control Data
	IBM Compatible (128 B/S 26 Sectors)	3060	1.99	473071	53426	800506	2305830	FD1 128	FD-1	740 0	S/A 100	15002	FD34-9000	F111111X	421602
Flexible Disc 1s	IBM Compatible (128 B/S, 26 Sectors) w/ W P N	3067	2.04	—	—	—	—	—	—	740-0	—	—	FD34-9000	—	—
Single-Headed Drives	IBM Compatible (128 B/S 26 Sectors) w/ W P N & Hub Ring	3064	2.39	—	—	—	—	—	—	—	—	—	FD34-9000	—	—
Single-Density Media	IBM Compatible (128 B/S 26 Sectors) REVERSIBLE	1729	3.19	473072	54431	—	—	—	FD-2	740/2-0	—	15150	FF34-9000	F171111X	—
	IBM System 6 Compatible	3066	2.04	473077	54581	800509	1669959	—	—	740-0 086	—	15003	FD60-9000	F118111X	—
	IBM Compatible (256 B/S, 15 Sectors)	3109	1.99	473073	—	800584	2305845	—	—	740-3600	—	15006	FD36-9000	F112111X	—
	IBM Compatible (512 B/S 8 Sectors)	3110	1.99	473074	—	800585	1669954	—	—	—	—	15004	FD60-8000	F113111X	—
	Shugart Compatible 32 Hard Sector	3015	1.99	470801	53802	101/1	—	FH1-32	FD-137	740-32	S/A-101	15025	FD32-9000	—	421332
	Wang Compatible, 32 Hard Sector w/Hub Ring	3067	2.49	—	54491	—	—	—	—	740-32RH	—	—	—	F37A411X	—
	CPT 8000 Compatible	3045	2.69	—	—	—	—	—	—	—	—	15229	—	—	—
Flexible Disc 1d	IBM Compatible (128 B/S 26 Sectors)	3090	2.69	474071	54566	3740/1D	—	FD1-128/M2100	FD-1D	741-0	—	—	FD34-8000	F131111X	423002
Single Headed Drives	Soft Sector (128 B/S 26 Sectors) REVERSIBLE	3093	3.69	—	—	—	—	—	—	—	—	—	—	—	—
Double Density Media	Shugart Compatible, 32 Hard Sector	3091	2.89	470801	54586	101/1D	—	FH1-32D	—	741-32	S/A-103	15075	FD32-8000	F33A411X	423322
	Wang Compatible, 32 Hard Sector w/Hub Ring	3098	3.09	—	—	—	—	—	—	—	—	—	—	—	—
Flexible Disc 2s	Soft Sector (128 B/S, 26 Sectors)	3113	3.09	—	54478	800814	1766870	—	—	—	S/A-150	15153	FD10-4026	F121111X	—
Double-Headed Drives Single-Density Media	Soft Sector (256 B/S 15 Sectors)	3106	3.09	473477	54226	800815	2736700	FD2-256D	—	742-0	—	15154	FD10-4015	F122111X	424612
Flexible Disc 2d	Soft Sector (Unformatted)	3102	3.09	473485	—	DY150	—	FD2-XDM	FD-2D	743-0	—	15103	DD34-4001	—	425002
Double-Headed Drives	Soft Sector (128 B/S, 26 Sectors)	3115	3.09	—	—	—	—	—	—	—	S/A-150	—	—	—	—
Double-Density Media	Soft Sector (256 B/S, 26 Sectors)	3103	3.09	473471	54325	800817	1766872	FD2-256D	—	743-0/256	—	15101	DD34-4026	F144111X	425602
	Soft Sector (512 B/S, 15 Sectors)	3114	3.09	473472	54479	800818	1669044	—	—	743-0/512	—	15100	DD34-4015	F145111X	425612
	Soft Sector (1024 B/S, 8 Sectors)	3104	3.09	473473	54485	800819	1669045	—	—	743-0/1024	—	15102	DD34-4008	F147111X	425622
	32 Hard Sector	3105	3.09	470851	—	101/2D	—	FH2-32D	—	743-32	S/A-151	15125	DD32-4000	F34A411X	425322
	Burroughs B-80 Compatible, 32 Hard Sector	3092	3.09	—	—	—	—	—	—	—	—	—	—	F34A811X	—
	Soft Sector (1024 B/S, 8 Sectors) w/ Hub Ring	3119	3.49	—	—	—	—	—	—	—	—	—	—	—	—
	Shugart Compatible, 32 Hard Sector	3181	3.39	—	—	—	—	—	—	—	—	—	DD32-4000	—	—
Flexible Disc FD Memorex 651 or Equiv Drive Compatible	FD VI (Vinyl Jacket)	30712003	2.89	470851	—	FDIV	—	—	FD-166	611-0	—	15026	FD65-1000	F61A111X	—
Mini Flexible Disc 1s	Soft Sector (Unformatted)	3401	1.94	475001	54256	104/1	—	MD1	MD 1	744-0	S/A-104	15300	MD525-01	M11A211X	441002
5¼" Single-Headed Drives	10 Hard Sector	3403	1.94	475010	54257	107/1	—	—	MD 110	744-10	S/A-107	15325	MD525-10	M41A211X	441102
Single-Density Media	16 Hard Sector	3405	1.94	475016	54258	106/1	—	MH1	MD 116	744-16	S/A-106	15326	MD525-16	M51A211X	441162
	Soft Sector (Unformatted) w/Hub Ring	3431	2.14	—	—	—	—	—	—	—	—	—	MD525-01	—	—
	10 Hard Sector, w/Hub Ring	3433	2.14	—	—	—	—	—	—	—	—	—	MD525-10	—	—
	16 Hard Sector, w/Hub Ring	3435	2.14	—	—	—	—	—	—	—	—	—	MD525-16	—	—
Mini Flexible Disc 1d	Soft Sector (Unformatted)	3417	2.14	—	54846	104/1D	—	—	—	—	—	—	MD525-01	—	—
5¼" Single-Headed Drives	10 Hard Sector	3418	2.14	—	54849	107/1D	—	—	—	—	—	—	MD525-10	—	—
Double-Density Media	16 Hard Sector	3419	2.14	—	54852	106/1D	—	—	—	—	—	—	MD525-16	—	—
	Soft Sector (Unformatted) w/Hub Ring	3481	2.34	—	—	—	—	—	—	—	—	—	MD525-01	—	—
	10 Hard Sector w/Hub Ring	3483	2.34	—	—	—	—	—	—	—	—	—	MD525-10	—	—
	16 Hard Sector w/Hub Ring	3486	2.34	—	—	—	—	—	—	—	—	—	MD525-16	—	—
Mini Flexible Disc 2d	Soft Sector (Unformatted)	3421	2.59	—	54824	104/2D	—	—	—	—	S/A-104	—	MD550-01	—	—
5¼" Double-Headed Drives	10 Hard Sector	3423	2.59	—	54827	107/2D	—	—	—	—	S/A-107	—	MD550-10	—	—
Double Density Media	16 Hard Sector	3425	2.59	—	54830	106/2D	—	—	—	—	S/A-106	—	MD550-16	—	—
	Soft Sector (Unformatted) w/Hub Ring	3491	2.79	—	—	—	—	—	—	—	—	—	MD550-01	—	—
	10 Hard Sector w/Hub Ring	3493	2.79	—	—	—	—	—	—	—	—	—	MD550-10	—	—
	16 Hard Sector w/Hub Ring	3495	2.79	—	—	—	—	—	—	—	—	—	MD550-16	—	—

Memorex Flexible Discs...The Ultimate in Memory Excellence

Quality
Memorex means quality products that you can depend on. Quality control at Memorex means starting with the best materials available. Continual surveillance throughout the entire manufacturing process. The benefit of Memorex's years of experience in magnetic media production, resulting, for instance, in proprietary coating formulations. The most sophisticated testing procedures you'll find anywhere in the business.

100 Percent Error Free
Each and every Memorex Flexible Disc is certified to be 100 percent error free. *Each* track of *each* flexible disc is tested, individually, to Memorex's stringent standards of excellence. They test signal amplitude, resolution, low-pass modulation, overwrite, missing pulse error and extra pulse error. They are torque-tested, and competitively tested on drives available from almost every major drive manufacturer in the industry including drives that Memorex manufactures. Rigid quality audits are built into every step of the manufacturing process and stringent testing result in a standard of excellence that assures you, our customer, of a quality product designed for increased data reliability and consistent top performance.

Customer-Oriented Packaging
Memorex's commitment to excellence does not stop with a quality product. They are proud of their flexible discs and they package them with pride. Both their packaging and their labeling have been designed with your ease of identification and use in mind. The desk-top box containing ten discs is convenient for filing and storage. Both box labels and jacket labels provide full information on compatibility, density, sectoring, and record length. Envelopes with multi-language care and handling instructions and color-coded removable labels are included. A write-protect feature is available to provide data security.

Full One Year Warranty — Your Assurance of Quality
Memorex Flexible Discs will be replaced by Memorex if they are found to be defective in materials or workmanship within one year of the date of purchase. Other than replacement, Memorex will not be responsible for any damages or losses (including consequential damages) caused by the use of Memorex Flexible Discs.

Quantity Discounts Available
Memorex Flexible Discs are packed 10 discs to a carton and 10 cartons to a case. Please order only in increments of 100 units for quantity 100 pricing. We are also willing to accommodate your smaller orders. Quantities less than 100 units are available in increments of 10 units at a 10% surcharge. **Quantity discounts** are also available. Order 500 or more discs at the same time and deduct 1%; 1,000 or more saves you 2%; 2,000 or more saves you 3%; 5,000 or more saves you 4%; 10,000 or more saves you 5%; 25,000 or more saves you 6%; 50,000 or more saves you 7% and 100,000 or more discs earns you an 8% discount off our super low quantity 100 price. Almost all Memorex Flexible Discs are immediately available from **CE**. Our warehouse facilities are equipped to help us get you the quality product you need, when you need it. If you need further assistance to find the flexible disc that's right for you, call the Memorex compatibility hotline. Dial 800-538-8080 and ask for the *flexible disc hotline* extension 0997. In California dial 800-672-3525 extension 0997.

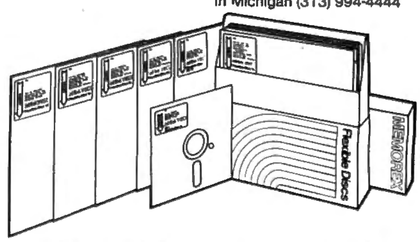
Buy with Confidence
To get the fastest delivery from **CE** of your Memorex Flexible Discs, send or phone your order directly to our Computer Products Division. Be sure to calculate your price using the **CE** prices in this ad. Michigan residents please add 4% sales tax. Written purchase orders are accepted from approved government agencies and most well rated firms at a 10% surcharge for net 10 billing. All sales are subject to availability, acceptance and verification. All sales are final. Prices, terms and specifications are subject to change without notice. Out of stock items will be placed on backorder automatically unless **CE** is instructed differently. Minimum order $50.00. International orders are invited with a $20.00 surcharge for special handling in addition to shipping charges. All shipments are F.O.B. Ann Arbor, Michigan. No COD's please. Non-certified and foreign checks require bank clearance.

Mail orders to: **Communications Electronics**, Box 1002, Ann Arbor, Michigan 48106 U.S.A. Add $8.00 per case or partial-case of 100 8-inch discs or $6.00 per case of 100 5¼-inch mini-discs for U.P.S. *ground* shipping and handling in the continental U.S.A. If you have a Master Card or Visa card, you may call anytime and place a credit card order. **Order toll-free** in the United States. Call anytime 800-521-4414. If you are outside the U.S. or in Michigan, dial 313-994-4444. **Dealer inquiries invited.** All order lines at **Communications Electronics** are staffed 24 hours.

MEMBER

VISA MasterCard

Order Toll-Free!
(800) 521-4414
In Michigan (313) 994-4444



For Data Reliability—Memorex Flexible Discs

263

COMMUNICATIONS ELECTRONICS™

Computer Products Division

854 Phoenix ☐ Box 1002 ☐ Ann Arbor, Michigan 48106 U.S.A.
Call TOLL-FREE (800) 521-4414 or outside U.S.A. (313) 994-4444

We've got it!

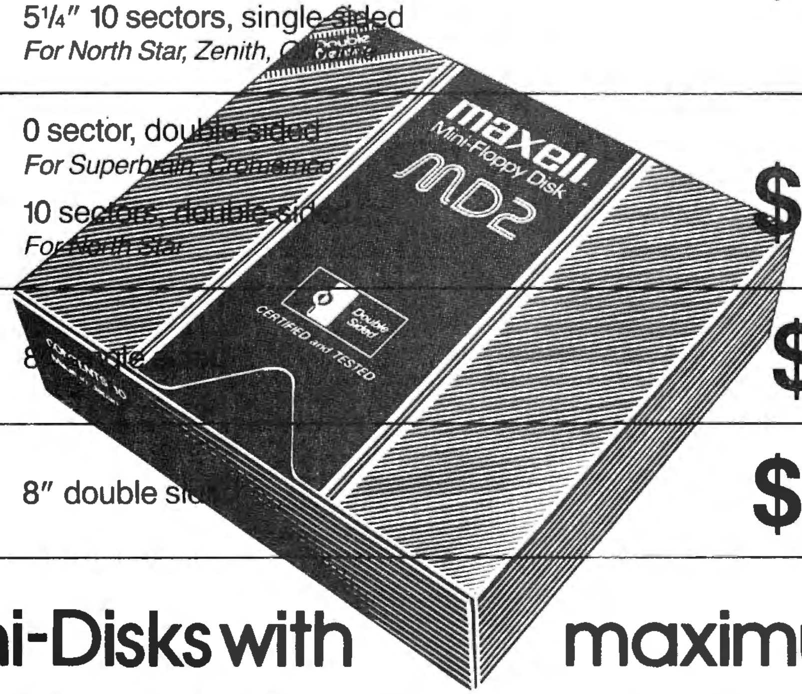
The best disks money can buy . . .

If your data is important use quality MAXELL disks.

Buy our year-end special ('til December 31, 1981).

All MAXELL certified for single or double density. maxell

MD1	5¼" 0 sector, single-sided *For TRS-80, Superbrain, Apple*	
MH1	5¼" 16 sectors, single-sided *For Micropolis Systems*	**$29.95**
MH1-10	5¼" 10 sectors, single-sided *For North Star, Zenith,*	
MD2D	0 sector, double-sided *For Superbrain, Cromemco*	**$44.95**
MD2-10	10 sectors, double-sided *For North Star*	
FD1	8"	**$41.95**
FD2	8" double si	**$54.95**



The Mini-Disks with maximum quality.

Mini Micro Mart, Inc.

943 W. Genesee St. Syracuse, N.Y. 13204 (315)422-4467

✓24

LIST OF ADVERTISERS

RS Number		Page
171	ABS Suppliers	100
580	ABS Suppliers	387
452	A.M. Electronics	346, 347
129	Aardvark 80	142
229	Access Unlimited	170, 171
265	Ace Computer Products	391
34	Acorn Software	108, 109
350	Acorn Software	100
348	Advanced Oper. Sys	106
97	Adventure Int'l.	251
307	Adventure Int'l.	35
222	Adventure Int'l.	47
224	Adventure Int'l.	55
311	Adventure Int'l.	63
387	Aerocomp, Inc.	282, 283
413	Allen Ashley/PPS	280
79	Allen Gelder Software	143, 358
80	All Systems Go	375
69	Alpha Byte Stores, Inc.	CIII, 22, 23
210	Alpha Products Company	25, 27, 29
124	Alphanetics	178
138	The Alternate Source	218, 219
526	American Business Computers	76
527	American Business Computers	76
528	American Business Computers	76
560	American Business Computers	76
561	American Business Computers	76
562	American Business Computers	76
563	American Business Computers	76
520	American Business Computers	77
521	American Business Computers	77
522	American Business Computers	77
523	American Business Computers	77
524	American Business Computers	77
525	American Business Computers	77
564	American Business Computers	317
565	American Business Computers	317
529	American Business Computers	325
461	Ancie Labs	367
264	Apparat, Inc.	101, 103
*	Archbold Electronics	167
491	ARCsoft Publishers	179
449	Aspen Software Co./Soft Tools	281
445	AT-80	258
398	Aton Int'l.	198
345	Automated Simulations	99
48	Automated Simulations	175
122	B.T. Enterprises	304, 305
*	BAP$ Software	279
49	Basics and Beyond, Inc.	93
186	Baysian Investment Services	375
343	Baysian Investment Services	104
237	BCCOMPCO	330
351	Berg Works	199
357	Big Five Software Company	43
335	Bizcomp Corp	100
582	Bonita Computer Software	387
416	Bosen Electronics	342
76	Breeze Computing Inc.	326
459	CMD Micro	381
145	C & S Electronics Mart, Ltd.	252
390	Canty & Associates	316
62	Cecdat, Inc.	199
165	Centec Process Systems Inc.	104
339	Charles Mann & Assoc.	104
347	Charles Mann & Assoc	102
502	Cload Magazine, Inc	246
32	Cload Magazine, Inc	247
346	Clockwork Software	104
588	Color Software Services	391
263	Communications Electronics	400
496	Complete Computer Services	387
100	CompuCover	376
594	Compumax Associates, Inc.	365
107	Computer Applications Unlimited	297
199	Computer Case Company	167
584	Computer Design Labs	168
372	Computer Discount of America	96
22	Computer Information Exchange	146
581	Computer Marketing Services Inc	21
130	Computer Plus	223
111	The Computer Store	53
212	Computer Shopper	349
162	Computerware	99
198	Computer Wit, Inc	358
322	The Computerists Directory	372
392	Computex	188, 189
9	Computronics, Inc.	264-271
204	Comsoft	351
10	Contract Services Associates	287
52	Coolidge & Associates	259
78	Coosol	248
465	The Cornsoft Group	62
378	Cornucopia Software	117
36	Corsair Computer Co.	279
415	Cosmopolitan Electronics Corp.	342
233	Cottage Software	206
*	CPAids	89
298	CPU Shop	329
168	Creative Computing	99
412	Custom Data	279
121	Custom Electronics	255
574	Custom Software	379
169	Custom Software	105
170	Custom Software	105
*	Cybernetics, Inc.	75
464	Dan G. Haney & Associates, Inc.	297
540	Data Resources	17
548	Data Resources	19
407	Datasoft Inc.	259
172	Data Tech. Ind.	106
274	Data Trans	166
570	Delta Computer Systems	382
167	Demi-Software	102
583	Designatron	183
368	Digital Systems Engineering	183
163	DIP, Inc.	106
440	Discount Software Group	217
30	Displayed Video	133
393	D & M Software	257
88	Documan Software	215
318	Dr. Software	351
*	DSC Publishing	127
242	Dynamic Software	387
5	E-Z Software	385
592	E.A.P. Company	391
117	EBG & Assoc.	277
278	ESI Lynx	289
510	Elgen Systems	280
	80 Microcomputing	
	Binders	387
	Books	395, 396, 397
	Bulk	375
	Desktop Computing	115
	Encyclopedia	290, 291, 387
	Load-80	163, 373
	Manuscript	278
	Moving	393
	Subscriptions	169, 255, 338
	University Microfilm	260
33	Eighty-US Journal	320
58	Electronic Specialists	129
569	Engineering Analysis Software	387
326	Engineering Consultants Ltd.	99
404	Epson America	225
3	Exatron	CIV
500	Exatron	207
446	Excellonix	387
225	Fantastic Software	277
141	FEC, Ltd.	255
557	William A. Fink	369
160	Fireside Computing	100
462	Fort worth Computer	379
161	FutureSoft	98
39	G&L Software Enterprises	93
75	Godbout	354
281	Golden Braid	87
218	Good-Lyddon Data Systems	215
284	Gosub International, Inc	299
223	H&H Trading Company	143
*	HW Electronics	399
352	Hayes Microcomputer Products	41
125	Hexagon Systems	135
577	H I.S./Computermation, Inc	174
401	Holmes Engineering	331
103	Howe Software	385
86	ICR Furturesoft	141
161	ICR Futuresoft	98
365	I.D.S.	316
*	IJG, Inc.	209, 327, 337
40	Icrom Enterprises, Ltd.	295
467	Illustrated Memory Banks/IMB	280
305	Insiders Software Consultants Inc.	114
*	Instant Software Inc	
	Catalog	226-242
596	Institute For Scientific Anal. Inc.	260
246	Interface, Inc	381
287	Interlude	39
591	Intersoft Unlimited	345
323	J.C. Datatron	222
331	J.F. Consulting	100
369	J&M Systems	383
193	Joe Computer	376
405	Jerry Medlin & Associates	369
85	Johnson Associates	135
190	JPC Products	245
149	Kogyosha Co Ltd./Nik.	343
26	Laredo Systems, Inc.	363
81	LEDS Publishing Company Inc	277
63	Thomas V. Lenz	310
*	Level IV Products, Inc.	393
61	Liberty Software	337
57	Libra Labs, Inc	197
391	Lindbergh Systems	187
572	LNW Research	49
571	LNW Research	97
53	LNW Research	377
35	Logical Systems Inc.	121
11	LT Data, Inc	253
74	Lynn Computer Services	135
268	Maine Software	354
330	Manhattan Software Inc.	102
334	Manhattan Software Inc.	104
593	Market Management Systems	299
84	Master Electronics	337
328	Matchless Systems	105
585	Maxtek, Inc	254
128	Med Systems Software	201, 202, 203
421	Medfield Computer Software	342
104	Mercer Systems, Inc.	215
20	Meta Technologies Corporation	9, 11, 13
542	Metatronics	15
598	MG Products	142
54	Micro Architect, Inc.	129
173	Micro Architect Inc.	104
205	Micro Blajak Systems, Inc.	224
315	Micro Data Base Systems	185
470	Micro Grip	382
46	Micro Investment Software	299
89	Micro Learningware	224
16	Micro Mainframe	275
72	Micro Management Systems, Inc.	151
423	Micro Management Systems, Inc	293
68	Micro Matrix	350
29	Micro Mega	330
310	Micro Mint	383
485	Micro Mnemonics	383
495	Micro Pro Systems	215
384	Micro Systems Software, Inc.	CII, 261
486	Micro Tax	391
109	Micro Works	344
332	Micro Works	102
333	Micro Works	105
379	Micro-Design	140
476	Micro-80	65
470	Micro-Grip, Ltd	382
359	Micro-Images	334
349	Microbyte Research & Development	105
28	Microcomputer Tech Inc.	302, 303
599	Microcon Software Centers	194
248	MicrOdome	343
*	Microed	359
355	Micromatic Program Company	116
306	Microperipheral Corp.	331, 389
579	Microsoft CPD	332
243	Microsouth District	195
362	Microtek, Inc.	301
8	Midwest Computer Peripherals	398
578	Midwest Data Systems	302
309	Midwest Scientific Instrument	167
112	Miller Microcomputer Services	145
24	Mini Micro Mart Inc	401
98	Minis 'n Micros Inc.	255
221	Misosys	367
451	MTS Enterprises	212
144	Mumford Micro Systems	335
*	NRI Schools	357
135	NDM Designs	324
480	Nelson Software Systems Inc	263
194	New England Business Service	359
336	Occupational Software Co.	98
389	Omega Sales	355
387	Omikron	7
439	OmniSystems	385
105	Omnitek	381
296	Orange Micro Inc.	51
376	Orion Instruments	183
127	P C Newsletter	263
595	P.M.W.	222
66	PSM Inc.	381
370	Pacific Exchanges	195
153	Pacific Office Systems	319
64	Pan American Electronics	87
586	Penner Bros., Inc.	391
207	Pensadyne Computer Services	324
*	Percom Data Co.	32, 33
429	Percom Data Co.	3
508	Percom Data Co.	3
258	Percom Data Co	3
1	Percom Data Co.	3
576	The Peripheral People	369
51	Perry Oil & Gas	120
422	Personal Micro Computers Inc.	249
*	Phase I Systems	191
272	Pickles & Trout	373
474	Picotrin Tech. Inc.	200
131	Plus Computer Tech. Inc.	315
386	Practical Analysis	379
435	Precision Prototypes	309
408	Process Control Tech.	365
340	Professional Business Software	102
575	Program Innovations	257
179	Programmer	105
364	The Programmers Guild	343
179	Programmer	105
17	The Program Store	57,58,59,60
156	The Program Store	58
277	Progressive Electronics	111
327	Prosoft	100
441	Prosoft	73
27	Quality Software Distributors	91,331
329	Quality Software Distributors	105
271	Quant Systems	393
41	Racet Computes'	371
133	Radio Ranch, Inc.	259
337	Radio Shack	99
167	Radio Shack	106
236	Rand's Inc.	162
*	Realty Software Company	224
374	Refware	129
77	REMarkable Software	319
70	Remsoft Inc.	147
276	Richcraft Engineering Ltd.	195
164	Rochester Data	102
143	Sales Data, Inc.	359
589	SCDC	376
291	Scientific Engineering Lab	215
297	Service Technology	345
19	Simutek	353
91	68 Micro Journal	260
397	Small Systems Design	334
*	Snappware	79,81,83,85
341	Snappware	100
342	Snappware	102
515	Soft Sector Marketing Inc.	311
495	Soft Sector Marketing Inc.	312
434	Soft Sector Marketing Inc	313
356	Software Affair	361
299	Software Concepts	125
13	Software House Int'l.	199
*	the Software Plantation, Inc.	56
597	Software Solutions	256
344	Solartek	99
454	Sound Center Radio Shack	391
373	Southern Innovative Design	65
132	Spectral Associates	385
*	Spectronics, Inc/Level IV	211
275	Speedway Electronics	323
317	Standard Microsystems Inc	71
511	Starbuck Data Co.	391
438	Stocking Source	193
82	Sturdivant & Dunn, Inc.	215
150	Sublogic	366
151	Sun Research	365
403	Superior Software	255
321	Synergetic Solutions	297
231	Syngergistic Solar Inc.	394
358	Syracuse R & D Center	375
60	Systemworks	319
148	Tab Sales Company	206
45	Taranto & Associates	341
410	Tesso, Scott	278
25	Texas Computer Systems	69
6	Total Access	213
325	Transformation Technology	98
414	Transformation Technology	278
178	Triton Manufacturing Co.	99
18	Twenty First Century Software	93
43	Universal Software Studios	107
*	V R Data Corporation	158, 159
137	Van Horn Office Supply	326
363	Walonick Associates	373
433	Warlock Warehouse	334
411	Wel Mur of California, Inc.	316
590	Western Operations	75
312	Windham Software	155
373	XYZT Computer Dimensions Inc.	351
482	York 10 Computerware	263
166	York 10 Computerware	106

For further information from our advertisers, please use the Reader Service card.

*This advertiser prefers to be contacted directly

READER SERVICE

This card valid until December 31, 1981

Please help us to bring you a better magazine— by answering these questions.

A. What is your occupation?
1. Professional
2. Engineer
3. Data Processing
4. Business
5. Education
6. Technician
7. Government
8. Student
9. Other ____________

B. What are your primary interests in applications of your TRS-80? Rate all that apply on a scale of 1 (least) to 5 (most).
1. Business
2. Hobby
3. Home Management/Finance
4. Education
5. Science or Technical
6. Other ____________

C. Is your computer use:
1. Connected with your occupation
2. Not connected with your occupation
3. Both of the above apply

D. What of type of TRS-80 do you own?
1. Model I Level I
2. Model I Level II
3. Model II
4. Model III Level I
5. Model III Level II
6. Color
7. Pocket Computer

E. On average, how many of each issue's program listings do you actually type into your microcomputer?
1. 0-2
2. 3-5
3. 6-8
4. 9 or more

F. Are you currently a member of a network?
1. Yes Which one? ____________
2. No

G. To what types of software users groups do you belong?
1. Hardware Exclusive
2. General Club
3. College Organization
4. Other ____________

H. From what companies have you purchased software?
1. Hayden
2. Hewlett-Packard
3. Instant Software
4. Microsoft
5. Personal Software
6. SAMS
7. Other ____________

I. How many people read your copy of *80 Microcomputing*?
1. 1
2. 2
3. 3
4. 4 or more

J. Where did you obtain this copy of *80 Microcomputing*?
1. Subscription copy
2. Newsstand
3. Computer store
4. Friend
5. Other ____________

K. Please enter the month and page number of your favorite *80 Microcomputing* article published during the last 12 months

____________ Month ____________ Page Number

L. On a scale of 0 (no interest) to 5 (great interest) please rate your interest in the following columns:
1. Remarks
2. Inside 80
3. 80 Applications
4. Education 80
5. News
6. Exclusive 80
7. 80 Accountant
8. Kitchen Table Software
9. The Assembly Line
10. Input
11. New Products
12. Reviews

M. If you are not a subscriber please circle 500.

Reader Service: To receive more information from any of the advertisers in this issue of *80 Microcomputing*, circle the number on the postage-paid Reader Service Card that corresponds with the Reader Service number on the ad in which you are interested. You will find numbers, preceded by a ✓, near the logo of each advertiser. Complete the entire card, drop into a mailbox and in 4-6 weeks you will hear from the advertiser directly.

1	6	11	16	21	151	156	161	166	171	301	306	311	316	321	451	456	461	466	471
2	7	12	17	22	152	157	162	167	172	302	307	312	317	322	452	457	462	467	472
3	8	13	18	23	153	158	163	168	173	303	308	313	318	323	453	458	463	468	473
4	9	14	19	24	154	159	164	169	174	304	309	314	319	324	454	459	464	469	474
5	10	15	20	25	155	160	165	170	175	305	310	315	320	325	455	460	465	470	475
26	31	36	41	46	176	181	186	191	196	326	331	336	341	346	476	481	486	491	496
27	32	37	42	47	177	182	187	192	197	327	332	337	342	347	477	482	487	492	497
28	33	38	43	48	178	183	188	193	198	328	333	338	343	348	478	483	488	493	498
29	34	39	44	49	179	184	189	194	199	329	334	339	344	349	479	484	489	494	499
30	35	40	45	50	180	185	190	195	200	330	335	340	345	350	480	485	490	495	500
51	56	61	66	71	201	206	211	216	221	351	356	361	366	371	501	506	511	516	521
52	57	62	67	72	202	207	212	217	222	352	357	362	367	372	502	507	512	517	522
53	58	63	68	73	203	208	213	218	223	353	358	363	368	373	503	508	513	518	523
54	59	64	69	74	204	209	214	219	224	354	359	364	369	374	504	509	514	519	524
55	60	65	70	75	205	210	215	220	225	355	360	365	370	375	505	510	515	520	525
76	81	86	91	96	226	231	236	241	246	376	381	386	391	396	526	531	536	541	546
77	82	87	92	97	227	232	237	242	247	377	382	387	392	397	527	532	537	542	547
78	83	88	93	98	228	233	238	243	248	378	383	388	393	398	528	533	538	543	548
79	84	89	94	99	229	234	239	244	249	379	384	389	394	399	529	534	539	544	549
80	85	90	95	100	230	235	240	245	250	380	385	390	395	400	530	535	540	545	550
101	106	111	116	121	251	256	261	266	271	401	406	411	416	421	551	556	561	566	571
102	107	112	117	122	252	257	262	267	272	402	407	412	417	422	552	557	562	567	572
103	108	113	118	123	253	258	263	268	273	403	408	413	418	423	553	558	563	568	573
104	109	114	119	124	254	259	264	269	274	404	409	414	419	424	554	559	564	569	574
105	110	115	120	125	255	260	265	270	275	405	410	415	420	425	555	560	565	570	575
126	131	136	141	146	276	281	286	291	296	426	431	436	441	446	576	581	586	591	596
127	132	137	142	147	277	282	287	292	297	427	432	437	442	447	577	582	587	592	597
128	133	138	143	148	278	283	288	293	298	428	433	438	443	448	578	583	588	593	598
129	134	139	144	149	279	284	289	294	299	429	434	439	444	449	579	584	589	594	599
130	135	140	145	150	280	285	290	295	300	430	435	440	445	450	580	585	590	595	600

Name ____________

Address ____________

City ____________ State ____________ Zip ____________

80 Microcomputing ● November 1981

BOOKS

80 Microcomputing
Peterborough NH 03458

Please send me the following 80 Microcomputing products:

Qty.	Catalog #	Title	Unit Price	Total

Shipping and handling charges:
$1.50 1st book, $1.00 each additional book
(UPS, use street address)
$10.00 each book overseas airmail

Shipping/handling ____________

Total ____________

Enclosed $ ____________ ☐ Check ☐ M.O.

Bill: ☐ AE ☐ MC ☐ VISA

Card # ____________

Exp date ____________ Interbank # ____________

Signature ____________

Name ____________

Address ____________

City ____________ State ____________ Zip ____________

11-81

Please allow 4–6 weeks for delivery

No C.O.D. orders accepted.

SUBSCRIPTION

80 microcomputing subscribers save $10 off the newsstand price.

☐ New subscription ☐ Renewal
☐ 1 year —$25
☐ 2 years—$38
☐ 3 years—$53

Enclosed $ ____________ ☐ Check ☐ M.O.

Bill: ☐ MC ☐ Visa ☐ AE ☐ me

Card # ____________ Exp. date ____________

Signature ____________ Interbank # ____________

Name ____________

Address ____________

City ____________ State ____________ Zip ____________

Canada—$27 1 year only, US funds
Other foreign—$35 1 year only US funds

31NB89

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 780 FARMINGDALE NY 11737

POSTAGE WILL BE PAID BY ADDRESSEE

80 microcomputing™

Subscription Dept.
POB 981
Farmingdale NY 11737

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 80 PETERBOROUGH NH 03458

POSTAGE WILL BE PAID BY ADDRESSEE

80 microcomputing™

Peterborough NH 03458

Att. Mail Order

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 30 DALTON MA 01226

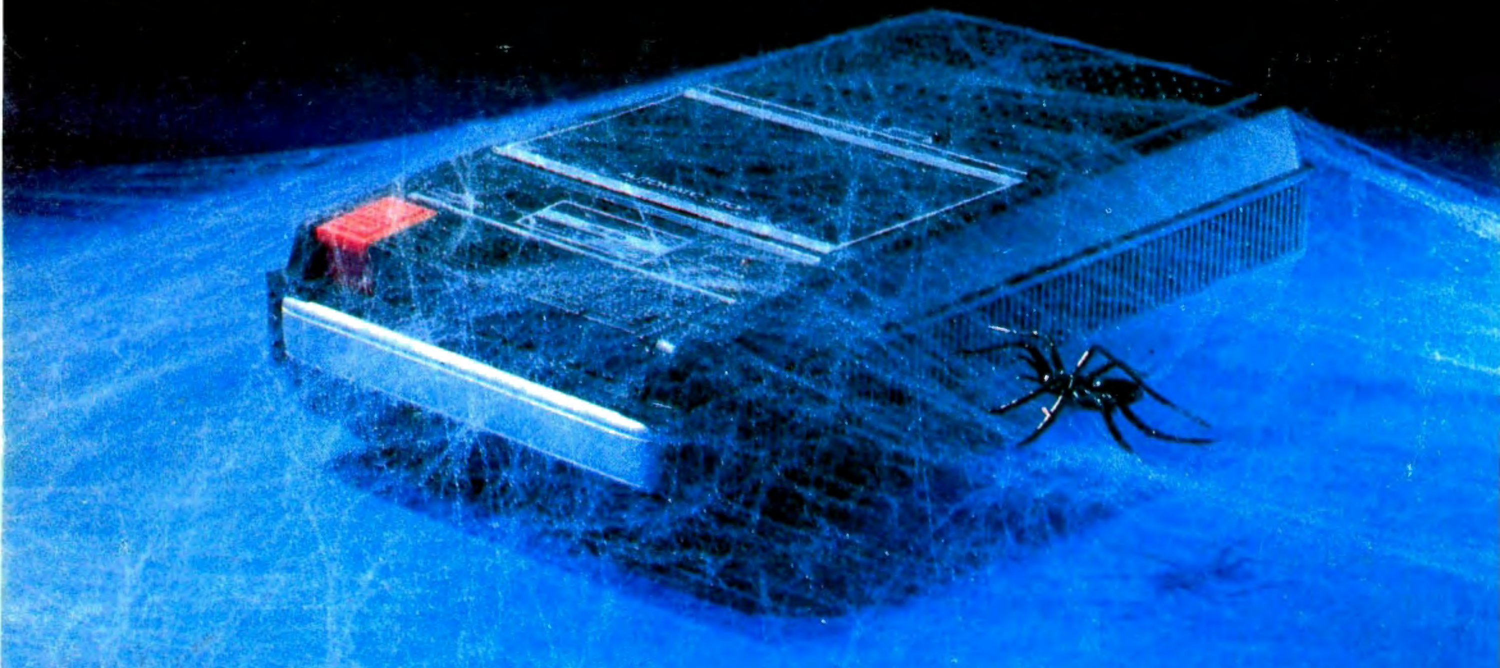
POSTAGE WILL BE PAID BY ADDRESSEE

80 microcomputing T.M.

P. O. BOX 306
DALTON, MA 01226

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

TIRED OF WAITING?



Frustrating isn't it! No matter how much you speed up your program it still seems to take forever to save data onto a cassette. Wouldn't it be great if someone could design a mass storage system with the speed of a disk, but at half the cost? *Exatron* did, the *Exatron Stringy Floppy (ESF)*.

Totally self-contained, the ESF is an extremely fast, reliable, and economical alternative to cassette or disk storage of programs or data. All of the ESF's operations are under the computer's control, with no buttons, switches, knobs or levers to adjust or forget.

The ESF uses a miniature tape cartridge, about the size of a business card, called a wafer. The transport mechanism uses a direct drive motor with only one moving part. Designed to read and write digital data only, the ESF suffers from none of the drawbacks of cassettes - without the expense of disks.

Several versions of the ESF are available, for the *TRS-80, Apple, PET, OSI* and an *RS 232* unit. Even the slowest of the units is 15 times faster than a cassette, and all are as reliable as disk drives - in fact a lot of users say they are *more* reliable!

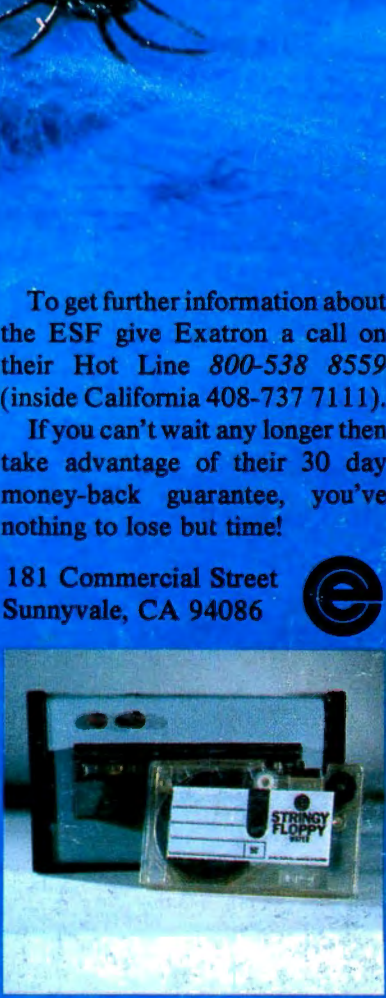
excellence in electronics

exatron

To get further information about the ESF give Exatron a call on their Hot Line *800-538 8559* (inside California 408-737 7111).

If you can't wait any longer then take advantage of their 30 day money-back guarantee, you've nothing to lose but time!

181 Commercial Street
Sunnyvale, CA 94086



TRS-80, Apple and PET are trademarks of Tandy, Apple and Commodore respectively.