80 Microcomputing

8/81

#20

August 1981
US $2.95

80 microcomputing™

the magazine for TRS-80* users

All work and no play . . .

ANNUAL GAMES ISSUE

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.

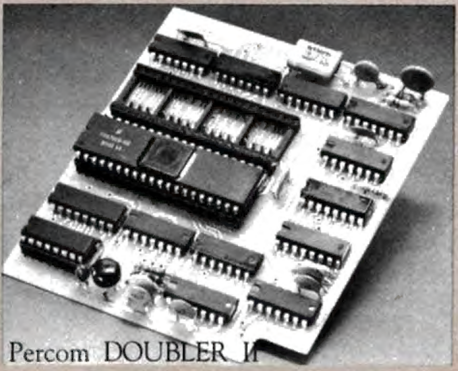
With a DOUBLER II installed, over four times more formatted data — as much as 364 Kbytes — can be stored on one side of a five-inch diskette than can be stored using a standard Tandy Model I drive system.

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~~19.9~~5, including the DBLDOS diskette. Now $169.95!

429

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 1-800-527-1592.

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

Circle 258 on inquiry card.

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 1-800-527-1592.

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

Circle 508 on inquiry card.

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.

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

PERCOM DATA COMPANY, INC. 211 N. Kirby Street Garland, Texas 75042 (214) 272-3421

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

80 Contents

PUBLISHER
Wayne Green

EXECUTIVE VICE PRESIDENT
Sherry Smythe

ASSISTANT TO PRESIDENT
Matt Smith

CORPORATE CONTROLLER
Charles Garniss, Jr.

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

COORDINATION
Penny Brooks

OFFICE MANAGER
Marcia Stone

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 $18 for one year and $45 for three years. In Canada, $20—one year only, U.S. funds. Canadian distributor: Micro Distributing, 409 Queen St. West, Toronto, Ontario, Canada M5V 2A5. Foreign subscriptions (surface mail), $28—one year only, U.S. funds. Foreign subscriptions (air mail), $60—one year only, U.S. funds. 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. Australian Distributor: Electronic Concepts, Attention: Rudi Hoess, 55 Clarence Street, Sidney 2000, Australia. 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.

A Walk on the Monster Side 85
by Bert Latamore
Since forgotten times, man has been a gamester. It is no wonder he has started using microcomputers to flavor his age-old fascination.

The Green Register 214
by Glynn Paul LaBorde
LaBorde's students at the state penitentiary in Angola, LA, may not be your typical scholars, but his program for keeping track of their programs will work—even in less confined academic settings.

Alternate Course 180
by Michael A. Duffin
After finding "Introduction to Data Processing" the dullest, most useless course he ever took, when the call came for this author to design his own intro, he was determined to make it more than "garbage in/garbage out."

Being of Sound Algorithm 136
by Carol S. Weinberg
Have you ever thought who would inherit your '80 when you pass away? This researcher shows you how to write a program for your last will and testament.

Epson's MX-80 126
by Tony Gitt
In our June issue, we compared 57 printers. Here's the 58th. And from its price to its print quality, this Japanese import will be making American printer manufacturers take notice.

Morse Resource—Part I 150
by Allan S. Joffe
W3KBM, Dresher, PA, gives us his first installment on programming a TRS-80 to help ham radio operators on the Morse Code circuit.

APPLICATIONS

192 Family Relationships Alex Haley, eat your heart out. *Sil Horwitz*
248 Hold That Pose Helping photographers find their photos. *Doug and Pam Ebert*
268 Disk Index Make it easy to keep everything up to date in those directories. *Robert A. Schilling*
294 Solar Altitude Plotter Tracking where the sun shines in. *Edward H. Rea III*

CONSTRUCTION

298 **The Wandering 80** Build a home for your computer. *Don Hubert*

DATA MANAGEMENT

194 **Vital Statistics** Remember the good old bell curve? *C. Brian Honess*

EDUCATION

188 **Quiz** Buzzers, bells, flashing lights and learning. *Fred Olsen*

GAME

86 **Startrek 4.0** Captain's log: star date: now! *Jake Commander*
92 **Frankenstein** Avoiding the hassle of reloading a game. *Robert A. Nicholas*
102 **Space Potatoes** The 'taters are coming! The 'taters are coming! *Christopher Brumme*
104 **Tank** A combat game for preschoolers. *Dan L. Rice*
106 **Space Empires** Imperialism circa 2501 A.D. *James L. Smith*
112 **Simon** Simulate a popular electronic game. *Dave McGlumphy*
116 **Star Guard** Protect your stargate from enemy jumpships. *John Beringer*
122 **Formula 80** High speed thrills on your tube. *Richard A. Sprague*

GENERAL

156 **A Macro Processor for Basic—Part II** More on making your computer program itself. *J. Alan Olmstead*
286 **Color Computer First Impressions** Initial looks sometimes deceive. *Richard L. Kilmon*

INTERFACE

164 **TRS-80 to RS-232 to TI-810** Tie a Texas Instruments printer to your Tandy. *Morris Herman*

REVIEW

154 **The Microconnection** You're missing a lot without a modem. *Fred Blechman*
222 **Choosing a Data Base** Three data bases compared. *Wynne Keller*
234 **Word** A word processing program that's back to Basic. *D. Wilson Cooke*

TECHNIQUE

230 **RAM Squeeze** Memory conservation. *John D. Adams*
292 **Three's Not a Crowd** Writing programs with pencil, paper and utilities. *David W. Mawdsley*

TUTORIAL

208 **Programming With the Voice Synthesizer** Mellifluous sounds from your computer. *Craig Werner*
254 **Be a Super USR!** Leap the gap between Basic and machine code in a single bound. *Roger C. Alford*
262 **Datacom Decisions** Making your '80 into a data communications device. *Frank J. Derfler, Jr.*
304 **Digits for Fun** 362,880 combinations—but how many fit the bill? *Ross A. Wirth*
308 **All About Sorts—Part I** Finding out which sort is best. *Len Gorney*

UTILITY

168 **Datagen** At last! Effortless data statements from machine language to Basic. *Dan and Cass Lewart*
240 **TRS-80 Disassembler** Your operating system as a puzzle. *Carl Wuebker*
260 **UC/LC Fix** Why are those tables backward? *Dennis Thurlow*
270 **Dirprog** Bringing together many frequently modified disks. *Jack Egbert*
274 **The Date Maker** Keep your calendars straight. *Garry H. Grosse*
278 **Command File Modifier** Spoon feeding commands to your '80. *Donald G. Crawford*
282 **Basic Shorthand** Enter common Basic words in a single keystroke for programming efficiency. *Jared Radin*

DEPARTMENTS

7 **Remarks** *Wayne Green*
8 **Inside 80** *Ed Juge*
12 **80 Input**
26 **Reviews**
24 **Education 80** *Earl R. Savage*
42 **80 Accountant** *Michael Tannenbaum*
44 **Exclusive Oracle** *Dennis Kitsz*
48 **The Assembly Line** *William Barden, Jr.*
52 **80 Applications** *Dennis Kitsz*
74 **Kitchen Table Software** *David Busch*
77 **Calendar**
78 **New Products**
314 **Load 80**
322 **Reader Service**

PUBLISHER/EDITOR
Wayne Green

MANAGING EDITOR
Debra Marshall

SENIOR EDITOR
Pamela Petrakos

TECHNICAL CONSULTANT
Jake Commander

PRODUCTION EDITOR
Susan Gross

NEWS EDITOR
Bert Latamore

REVIEW EDITOR
Michael Nadeau

NEW PRODUCTS EDITOR
Janet Fiderio

TECHNICAL EDITORS
Chris Brown
G. Michael Vose

EDITORS
John P. Mello Jr.
Betty Thayer

LAYOUT EDITORS
Joan Ahern
Bob Dukette
Sharon Phinney

TECHNICAL CONTRIBUTING EDITOR
Dennis Kitsz

EDITORIAL ADMINISTRATION
Pat Graham
Nancy Noyd

DESIGN ASSOCIATE
Diana Shonk

PRODUCTION MANAGER
Nancy Salmon

ASST. PRODUCTION MANAGER
Michael Murphy

AD GRAPHICS MANAGER
Robert Drew

AD COORDINATOR
Sue Symonds

ADVERTISING PRODUCTION
Steve Baldwin, Bruce Hedin, Jane Preston, Linda Drew

PRODUCTION DEPT.
Fiona Davies, Dianne Riston, Linda Drew, Kenneth Jackson, Ross Kenyon, Theresa Ostebo, Deborah Stone, Thomas Villeneuve, Gary Graham, Donna Wohlfarth, Sandra Dukette

PHOTOGRAPHY
William Heydolph, Terrie Anderson, Bill Suttenfield, Paul Babich

TYPESETTING
Barbara Latti, Sara Bedell, Michele Desrocher, Stephen Jewett, Luann Keddy, Mary Kinzel, Kelly Smith, Karen Stewart

Cover by Jay Connelly

Take this hundred

...if you can

This could be yours, but probably never will. This $100 bill is the reward we offer for the discovery of a legitimate bug in a DOSPLUS DISK Operating System. To qualify you must be:

1. A registered DOSPLUS owner
2. Able to demonstrate and document a real bug, not simply a design shortcoming.

We're convinced that DOSPLUS is the fastest, most powerful, and easiest to operate system on the market. DOSPLUS works! And works right! For the businessman and hobbyists, the speed and simplicity cannot be beat. For the BASIC programmer, NO-ONE can offer you more than we do. And for software authors, compatible kernel systems are available for distribution.

NO-ONE ELSE can give you!

- **The same high level of Model I/Model III diskette and program compatibility.**
- **True, complete BASIC program chaining with files open and variables saved.**

We have it all. Features galore AND error-free operation. If we're wrong, and you can prove it, we've got a brand new $100 bill for you.

We're not conceited, we're convinced! Order yours today, and soon you will be too!!

MICRO SYSTEMS SOFTWARE
5846 Funston Street, Hollywood, Florida 33023
Phone: (305) 983-3390

384

DOSPLUS

VISA master charge

CALL TOLL FREE FOR FAST SERVICE (800) 824-7888, OPERATOR 193FOR 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

80 REMARKS

by Wayne Green

"Even more exciting. . . is the new Casio VL-1. It also has two and one-half octaves, but is only about two-thirds the size of a regular piano keyboard, and will fit in your pocket or briefcase."

Music, Music, Music

While only a small percentage of you combine music with your computers, I'd like to encourage this application. This is a request for more articles on generating music with your TRS-80 systems.

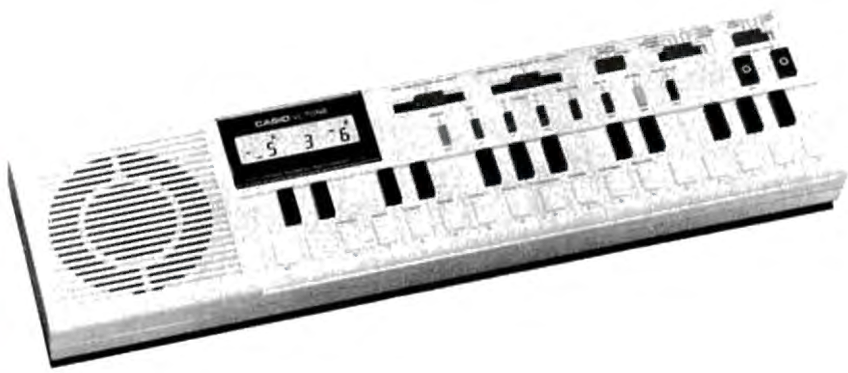
Paia has a wide variety of great gadgets which can work with your computer to allow you to simply play music, or compose. You might want to check their latest catalog.

The M-10, one of the inexpensive Casio keyboards, sells for around $150 and has two and one-half octaves. It is a full-sized keyboard, yet it is small enough to carry around. It can handle up to eight notes simultaneously. This unit has no memory, but looks as if it could be easily connected to a computer as a keyboard input for synthesized composing. I'd sure like to see some articles on this combination.

Even more exciting, perhaps, is the new Casio VL-1. It also has two and one-half octaves, but is only about two-thirds the size of a regular piano keyboard, and will fit in your pocket or briefcase. It can handle only one note at a time, but it has a memory for 100 notes and will play them back with the same rhythm with which you enter them. The unit also has ten different rhythms built in, which you can program and vary in tempo over a wide range. To cap it, there is a built-in song, complete with the rhythm accompaniment.

The VL-1 also has three octaves which are switchable, a mix of rhythm and melody which is variable, and six different instrument generators. The number six generator is programmable so you can create any type of synthesized tone you want. The tone has eight parameters which can be programmed.

The batteries are built-in, as is a small speaker and an LCD readout of the notes you are playing. This shows the current note being played, plus the last two previously played. For around $70, this has to be one of the best adult toys yet, and ideal for a musical computer system.■



Casio VL-1.

80 Subscriptions

The subscription ad for *80* has raised some eyebrows at Radio Shack. For a big outfit as tough on suppliers as they are, they seem to have suprisingly thin skins. I put the Radio Shack catalog prices alongside those advertised in a typical recent issue of *80*, and noted that by patronizing *80* advertisers, you could save $986 on the purchase of a Model III system. That seemed to me like a very good reason to buy the magazine.

Since writing that ad I've had some second thoughts; I wish now that I'd thrown in an offer of $14 worth of Instant Software free for anyone buying the package, making their total savings $1,000. That would pay for a lot of subscriptions.

The typical issue of *80*, in addition to being able to save you a kilobuck on buying a system, includes several hundred dollars worth of software. Now, if we could just get you to mention this to your friends—even those who have not yet bought their TRS-80—heck, particularly those who have not yet bought. We can save them a bundle. You'll have to save yours on your second system, or your accessories.

Load-80

The idea of this project was to provide Load-80 subscribers with the programs in each issue of *80* on a cassette. Having typed my fingers to the bone entering some long programs, only to find that I'd made errors, I thought you might appreciate this service. The time and aggravation it takes to type one program will more than pay for a Load-80.

Interest has been rising monthly so we'll be offering it as a yearly subscription. We didn't rush to make this offer because we weren't sure how well the project would take off. Well—let's say *some* people around here were worried; I felt it *had* to be a winner. We had a few problems getting all the programs on magnetic media, but it seems to be running smoothly now.

The '80 Encyclopedia

First we planned to reprint some of the *80* articles in book form for possible sale through Radio Shack. But as we looked at the situation, we realized we had enough important original material to put out a monthly book of around 300 pages. This material covers information that Radio Shack doesn't provide about the TRS-80, and programs for the system. Each issue includes hundreds of dollars worth of programs.

Trying to keep up with developments concerning the TRS-80 is initiating a growing number of publications. Remember that the more information you have about a computer, the more you can get it to do.

Just to give you an idea, in a recent call from an employee of a firm which writes software for the IBM 5120 system, I was told that his firm had recently adapted their bookkeeping software for use on the

Continues to page 10

INSIDE 80

by Ed Juge, director of computer merchandising, Tandy Radio Shack

"We have run numerous checks . . . waits of two to six minutes are much more common than 20 to 30 minute waits."

This past week, I received a couple of phone calls from TRS-80 owners who have unsuccessfully tried calling our 800-number. As I said last month, we have expanded both our staff and the number of incoming WATS lines.

In April, when the present phone system was being installed, there were some known glitches which allowed an occassional incoming call to circumvent the queue of waiting callers. As far as Ma Bell can determine, that is no longer possible.

In May, our Computer Services group answered over 21,000 telephone calls. Even with such a large number of calls, if you use the numbers I gave you last month (which are published monthly in our newsletter), you could end up in a queue of as many as 11 callers at the worst. We have run numerous checks and find that waits of two to six minutes are much more common than 20 to 30 minute waits.

So, to minimize delays in getting help or information, try your local store first. Step two is to call your nearest Radio Shack Computer Center or store with an expanded computer department and computer marketing representative. We've tried to staff these locations with knowledgeable people, who have received specialized training in the computer hardware and software lines. Whether or not you bought your equipment from them, they will help when you need it.

A call to Fort Worth should be your last resort, when help in the field isn't close enough, or the answer simply isn't available there. The phones are manned by a full staff from 8 a.m. until 5 p.m., and by a reduced staff until 7 p.m., all Texas times. The least busy time is early morning. The busiest is midday. The last few days of one month and the first few of the following month are the worst times, as business people doing month-end closeouts call then with questions.

Radio Shack Newsletter Expands

The *TRS-80 Microcomputer News* has grown by leaps and bounds. Throughout the second quarter of this year, it's been 24 pages. The advertising insert is gone, and we've been getting lots of compliments on the content. Almost half of the May issue was devoted to an in-depth story of Radio Shack's printer codes, responses, graphics modes, and even a description of the printer drivers in Models I, II and III. Each issue also carries a fair number of user-submitted programs. If you do get the newsletter, and have any suggestions, by all means, let us hear them!

"A call to Fort Worth should be your last resort, when help in the field . . . just isn't available."

Level II Basic Tutorial

We have a new tutorial manual available for Level II and Model III Basic owners called *Getting Started with TRS-80 Basic*. It should be available in your local stores by the time you read this. It is included with new (non-Level I) Model IIIs. Those of you who bought your Model III before it was available, and sent in your cards, will be sent a copy automatically. It is $6.95, and is available under Catalog No. 26-2107. The style is very much like the original Level I tutorial manual, and it assumes you are a beginner.

Plug 'N Power Controller

Some of you may be familiar with Radio Shack's Plug 'N Power light and appliance controller system. Modules connected between lights or appliances and the ac line can be controlled remotely from a small controller box. The signals from the control box to the modules are carried on the ac line itself. You can turn things on or off, and you can dim lights, all from the control box which could be placed anywhere in your house.

The system includes the controller, and software to control up to 256 possible modules. Connection is via the cassette port. Brightness of lights is adjustable to any of 10 levels under computer control. However, this is a one-way communication system. An errant module cannot be detected, nor can a module communicate in any way with the computer. The software allows either direct command from the keyboard, or the execution of instructions based on a specified schedule and the real-time clock. The software (minus RTC functions) can even be called as a subroutine by other Basic programs.

Videotex Update

A lot of folks got quite wound up when we announced Videotex a year or so ago. Others weren't overly enthusiastic. As you know, we announced both a Videotex terminal, and software to turn your personal computer into a terminal. Initially, we included a free hour on the CompuServe Information Service (the MicroNet folks). Today, you still get that hour, plus an hour on the Dow Jones Information Service. But what do the services actually offer?

CompuServe offers news from eight major newspapers (more coming) including the *New York Times, Columbus Dispatch, Virginian-Pilot and Ledger Star, Washington Post, San Francisco Chronicle, San Francisco Examiner, Los Angeles Times* and *Minneapolis Star & Tribune* plus the complete AP newswire. You can get national, international, regional and local news, weather, sports and commentary.

Then there's family information on subjects including food, nutrition, personal health, etc., and an encyclopedia-like bank of general reference material on history, geography, world events and famous people. Of course there is also a wide selection of games.

Several kinds of electronic communication are available, including mail (send and receive messages), a bulletin board for posting general information, and a CB simulation which allows on-line chatting between subscribers on a number of available channels.

CompuServe's MicroNet service lets

you program a mainframe computer in Extended Basic, Fortran, APL, Pascal, BLIS 10, Snobol and Assembly. There's a MicroNet software exchange, and you order computer-generated art suitable for framing or papering a small wall! (Programs can be downloaded only into computers, not into a Videotex terminal.)

Dow Jones Information Service

This database contains exclusive current and past news stories from the DJ News Service, the *Wall Street Journal*, and *Barron's National Business and Financial Weekly*. You'll have access to current price quotations (15-minute delayed) on more than 6,000 stocks and other securities listed on the New York, American, Midwest and Pacific exchanges. Dow Jones also allows access to revenues, earnings, dividends, price/earnings ratios and stock price performance relative to market indicators, on 3,200 companies and 180 industries, through an agreement with Media General, Inc. You'll get the latest breaking business and financial news within 90 seconds, selecting only the exact information you want to know more about. News within the past 90 days is also available.

There are enough new and exciting things happening with the TRS-80 Videotex, that you'll surely want to try it out soon if you haven't yet. It's all happening so fast that I can almost guarantee you that this information will be outdated by the time it reaches you. And it won't be long before more local databases begin springing up for you to access with your Videotex software. We have installed several Videotex host computer systems—Model II systems with eight or 16-line Radio Shack multiplexers.

This month I had plans to do a special project for the column (I mentioned it in closing last month in fact.) Unfortunately, we've been in the process of putting the 1982 Radio Shack catalog to bed, and preparing a bigger and better Computer Catalog you'll see late in the fall. We have at least another month's intensive work on the computer book, so I may not even make it in September, but I'll try. Given a choice, though, you'll find the new catalog a lot more exciting than my ramblings.■

Errata

I must apologize for two errors in last month's column, both prices. The new price on the Pocket Computer is $229.95, not $229.00, and the price on the new printer/cassette interface is $149.95, not $149.00.

While I'm at it, I'll tell you PC owners to be watchful when you run your Radio Shack programs with the new printer. PC Basic responds to Print commands by displaying the requested information, and waiting until you press Enter to continue. Well, when you've initialized the printer, Print commands do not pause, resulting in a continuous printout of the menu until you press Break.

The reason is that when most of those programs were coded, the printer was still only a gleam in our eye. Many of the existing programs are memory tight, and wouldn't lend themselves to easy modification.

80 REMARKS

from page 7

TRS-80 and found the system to be better and faster than on the IBM—and at a fraction of the hardware price.

Desktop Computing

In the past I have briefly mentioned a new publication we're getting ready for this fall. I wanted a magazine with articles written from the user's point of view, and written completely in plain English. That's right, no computerese or buzzwords.

Enthusiasm is growing rapidly. It's a badly needed magazine, one which will tell the businessman about computer systems which are actually doing jobs he wants done.■

Tandy Growth Retarded

A recent report by International Resource Development, Inc. pointed out that Radio Shack growth is being held back by the long-term company policy of selling *only* Radio Shack products. With computers by far the most important aspect of Radio Shack sales these days, the report concluded that unless Radio Shack starts handling products of other firms in their stores and computer centers, they are in a position to be hurt by the Japanese and larger U.S. firms entering the market.

Indeed, it is possible that the Radio Shack management has developed an over-confident attitude as a result of their success—mostly over poorly financed and inadequately managed competitors. Now, with the big guns getting ready for the battle for the small computer market of the 1990's, the old Radio Shack tactics may not be the best to continue.

In my discussions with Radio Shack personnel I've run up against a hardened attitude toward "the way we've always done it." Well, that may have worked in the low-priced electronics market, but Radio Shack is selling to a new group today: middle income people and businessmen.

For instance, there is little argument that this publication has helped the sale of TRS-80 systems. There is, however, a fear of selling it in Radio Shack stores because there are non-Radio Shack products advertised. They are continuing the rule of keeping out non-Radio Shack merchandise—and that includes *80*.

When Radio Shack talks software with us, they are only interested in a few of the "best sellers." This is one of the key problems which could have long term repercussions and give IBM and some of the Japanese firms the edge they need to sink Tandy. In order to fit computers into business—and to people—it is necessary to carry a wide range of software, not just a few best sellers.

Of course, it's possible that every major firm coming into the small computer field is going to be as myopic and hidebound as Radio Shack, in which case your investment in equipment and programs could be safe. Even an investment in Tandy stock might be safe. But the likelihood of *all* firms pulling the same self-destructive stunt is minimal, particularly when there are a growing number of software firms pressing for sanity in the field. If any one firm wakes up, they could have an unbeatable edge over the others—and that includes the 6,000 Radio Shack store selling system.

Back in the early days of the TRS-80 the chap who got Tandy involved with the TRS-80 also got them to try out a Tandy Computer Center. It was set up in the Tandy building Ft. Worth, in their enormous lobby/skating rink/mall. The store did well, but it quickly got away from selling Tandy equipment and started going strong for equipment from other manufacturers. The Tandy management blew the store right out of the water, including manager Don French.

Tandy wanted to sell Tandy equipment, using the age old Radio Shack formula. I don't know who shot down the experiment, but I believe that if it had gone ahead, they might have a couple of thousand Tandy Computer Centers around the country, instead of the Byte Shops and Computerlands we now see. And Tandy would have had a lot more control over Apple and other upstarts.

The IRD report gives the warning. The story is not much different from what I have been trying to tell 'em in Ft. Worth.■

80 INPUT

"Not all schools are armed camps with frustrated teachers and chaotic classrooms."

A Novice Replies

Mr. Pennington, you are absolutely correct—I am a naive computer novice ("Bad-image and Discipline," March). Had I been a pseudo-expert, I too might be writing books, but as a novice who is anxious to learn, I just read them.

You also may be correct about my lack of understanding of English because I apparently read meaning into the title of your book that you hadn't intended. And, as I reread the title "TRS-80 DISK & Other Mysteries," it still suggests you are going to tell us all about how the TRS-80 disk system works. One or your advertisements describes your book as "The Definitive Work on the TRS-80 Disk System" —wow—definitive. And, your table of contents lists "Operating Systems," so you too were fooled by your book's title.

John Grass
Portola Valley, CA

Cabin Fever

I am working at an isolated site at Yanbu, Saudi Arabia where my company is managing a project to complete an entire industrial city in the next 25 years. Our present population includes over 20,000 people of at least 35 nationalities, as well as 23 Apples and three TRS-80s, two of which I own.

I brought an old Model I Level I 4K unit with me, then was lucky enough to find a good deal on a 32K two disk unit with the famous Quick Printer. I had no idea what it was all about, but I was sure I needed help in fighting cabin fever. I had only the March 1980 issue of *80* to give me a hint of the potential uses of the machine, and during the next month's vacation trip home to New Orleans, I grabbed all the back issues I could lay my hands on, including the splendid May issue with the handy word processor that I have modified somewhat to fit into aluminum paper. Soon I ordered my subscription, even though I was intimidated by the $60 overseas air mail price.

I must say that, for all its flaws, your publication has been of incalculable use to me here. It is much better presented than anything my Applecomputing friends have shown me, both in format and in content. Some readers have complained about the advertising; another explained that he would have to go 120 miles for computer items. Well, I can go 185 miles of desert away, if I can get transportation not normally available for unofficial use, only to find sales people who have trouble with Level I English. Your magazine is my Sears & Roebuck Catalog. I have ordered a popular dot matrix printer, and word processing and mailing list software through your advertisers.

Shipping is horribly expensive, and there is no guarantee that customs will allow anything in. Transmitting devices are not allowed, so electronics are suspect. My printer is coming in with a friend from the states on a business trip, if he can get one off the shelf in the Los Angeles area. At least he will be able to explain what it is if there are any questions. Also, the days of the retail computer shop with sales staff to hold your hand are numbered, and the mail order houses are holding the keys to the marketplace. I hope my software can get in by mail.

Charles B. Dupre
Saudi Arabian Parsons Ltd.
Jeddah, Saudi Arabia

S Y Z Y G Y Interface Module

Congratulations to Dr. James Nestor for documenting his difficulty in interfacing a Xerox 1740 and Model II (June 1981). He provided a clear explanation of a common problem that arises when a computer RS-232 serial port is connected directly to a peripheral device, and even greater frustration when that device is a letter quality printer.

The problem is not with the Model II or with Xerox 1740, but occurs whenever a direct connection is used instead of a modem. The solution consists of fooling the port and printer into thinking that intelligent handshaking is occurring by use of a device called a null modem. He correctly pointed out one configuration. A more useful configuration is one that also permits a four conductor cable to be used (pins 1, 2, 3, 7); all handshaking is absent. But shame on him for proposing that the changes be made on the Xerox cable connector. The Xerox as well as the other letter quality printers he mentioned are expensive, complex devices. The cord is permanently attached, and to cut and splice is butchery. If and when he sells it or uses it with a modem, it will not work. Without the manual, he or the next person will not be able to reconfigure it correctly.

There is a family of interface modules manufactured by S Y Z Y G Y designed to overcome the obstacles faced by Dr. Nestor. They are less than two inches long and have a connector on each end and are simply plugged onto the end of the Xerox cord and then plugged into the RS232 computer port. All the necessary connections are done by the module. A similar unit is available to change the sex of the connector on the end of your RS-232 cable. Additionally, S Y Z Y G Y makes an RS-232 Patch Set that would have reduced Dr. Nestor's time for diagnosis and treatment to a few minutes.

Richard L. Rohde, president
S Y Z Y G Y
Covina, CA

In Defense of Teachers

I am a public school library media specialist in Jefferson County, Colorado, and president of Local 900 of the American Federation of Teachers. As such I have strong objections to comments made in Mr. Green's *80 Remarks* in the May 1981 issue. I agree there are lots of problems with public education today. However, public schools are not nearly as grim as he describes. Not all schools are armed camps with frustrated teachers and chaotic classrooms. Many teachers are doing an excellent job. These students are only a "captive audience" in that they are required by law to be there. Because education is compulsory the teacher needs to make it *more* interesting (than it

Continued on page 16

80 AID

Producing Musical Script

Help! Having recently acquired an MX-80 type II printer for my 48K twin disk TRS-80, I am keen to use the excellent dot-matrix ability to produce reproduction quality musical script suitable for school and/or church use, but the total scope of the project defeats me.

Is there anyone out there with the same need who would be prepared to cooperate in a joint effort to do this? My Basic is a little above average (I am a full-time user in another field—yacht design) and my musical knowledge is the same as my machine language programming ability, just adequate to the task.

If there are enough starters the effort could be agreeably shared. So far I have never seen any published material on this topic and any references known to a reader would be appreciated.

Mike Tattersall
Box 14588
Panmure
Auckland, New Zealand

Converting T-Bug

Does anyone know how to change T-Bug's tape loading and writing procedure so it works on a Model III? I have been trying to convert the tape I/O to ROM routines but have so far been unsuccessful. Any help would be greatly appreciated.

Jonathon Yarden
3408 Fliothaven Rd.
Louisville, KY 40222

Freeing Memory

When I added the extended Basic ROM to my TRS-80 Color Computer I was very disappointed by the drastic reduction in free memory. Ed Rutledge, a colleague in another department who also owns a color computer, stopped by recently to see our new Epson MX-80. He mentioned in passing a way to free up most of the memory reserved for graphics by Extended Basic: PMODE 0,1 PCLEAR 1

These commands may be entered in direct mode or included in a program. PRINT MEM now returns 13095, clearing 4500 of the 6000 bytes automatically reserved. If anyone has figured out how to free up the final 1.5K, please pass the information along.

David M. McCord
Psychology Dept.
University of Alabama
University, AL 35486

Printing Astrology Program

Help! When Radio Shack had its last sale I bought the Astrology program for my TRS-80 Model I. I have a GE Terminet serial printer connected to it and when I bought the program I knew I would have to make some changes for the printout. Unknown to me, this program bypasses the Level II ROM driver and it uses its own to print. This made things a lot harder. I found out where the Basic printing source code is, 689AH to 68D4H. This will print 95 percent of the printout, but at the end it runs off the end of the paper and stops. If anyone has solved this problem, please contact me.

Tarus Paul Balog
1655 Plantation Circle
Asheboro, NC 27203

Parts List Wanted

I enjoyed the article in your September 1980 issue on Selectric Hard Copy by Mr. Michael W. Bickerton. To follow through with a similar project of my own, I wrote to Mr. Bickerton requesting a parts listing; I have yet to receive a response. Since I am still interested in getting my Selectric to work with a Heath H-89 with serial, ASCII RS232 interface, I would appreciate a parts list, specifically the IC identification, plus any other hints or recommendations regarding construction and operation of this interesting project.

W. C. Pfingstag
939 Cedar Ave.
Mare Island, CA 94590

We also tried writing Mr. Bickerton with no luck. If you're out there, Michael, send us your new address. —Eds.

Game Confusion

I have appended seven games and made a menu to play whichever game I choose and that's where the trouble starts. I am getting data statements from other games that are creating havoc and discontent. How can I get the stack printer to read data from the game being played and not from the first data statement of the first game? In other words, I would like to have each game separate, even though the games are on one load with consecutive line numbers.

These were appended using "Boss," but there is nothing mentioned about this data problem.

Dennis Banik
E. 29 Euclid
Spokane, WA 99207

Calling All Fortran Users

The Radio Shack Fortran system for the Model I TRS-80 (item no. 26-2201) has been around for many months now. There must be many regular users among the readers of *80 Microcomputing*. This reader would like to hear form those Fortran users for the purpose of comparing notes and comments. If enough response and interest appears, one or more articles will be prepared for possible publication summarizing the results.

Gerald A. Sabin
6022 Sage Drive
Orlando, FL 32807

80 DEBUg

The Cathode Connection

In the October 1980 issue I found an article that was just what I needed. What I didn't need was the trouble I had putting it to use.

James D. Kunzman described how to interface an NEC Spinwriter (serial) to the TRS-80. I have been using a Spinwriter for over a year but with the same problems he described when trying to run at 1200 baud.

His cure was very simple except for one thing. In column 3 on page 146 he said to connect the anode of the 1N914 diode to U3 pin 11. He meant to say cathode. Naturally I checked everything in the system before I thought to check that. It works fine now.

Since his address is just down the road from me I tried to call him. I guess Uncle Sam's Air Force has forced him to move on. He is unknown at his listed number.

Russell L. Pyke
9770 S.W. 159th St.
Miami, FL 33157

Confused Variable Names

Several of your readers have asked me about my Car Mentor program in the May 1981 *80 Microcomputing*.

After close study and with considerable help from your readers, I found the following errors in the program:

```
Line 1210 should read DR = X(20)/12/100
Line 1220 should read X1 = DR-(X(21)/12/100)
Line 1230 should read X2 = DR-(X(22)/12/100)
Line 1300 should read L = X(19)/12/100
Line 1730 should read TH(J) = IP(J) + XH(J) + IH(J)
+ RH(J) - TD + DP(J)
```

The problem came about because I used the array X(J) to handle data input. Somewhere along the line I lost track of what value was where (even though I had a list in the remarks). The moral of the story is: Use unique variable names for each of the input variables. Using unique variable names will make the program somewhat longer, but you won't forget (or are less likely to forget) where you put what. I wouldn't have made the above errors had I used unique variable names instead of the array.

Note that these changes will change the answers. I suggest that anyone using the program type it in as is. Get the results shown in the example; then make the change.

Leslie E. Sparks
1014 Evergreen Drive
Durham, NC 27712

Apology to RS Dealers

This concerns the article in the March 1981 issue titled "Opinion = Peek (Mail)," or "Dear Radio Shack."

I fully expect that in 1981 we will see a full network-type database management system released for (not by) the Shack to revolutionize information management.

My apologies to any Shack dealers who might have been bugged by inquiries about the database manager. And a special apology to Micro Data Base Systems, Inc., of Lafayette, IN that did release this remarkable piece of programming.

Jim Glosser
Harley-Davidson Co.
1925 Eden Rd.
York, PA 17402

Missing Listing

As nice as it is to see one's own name in print in your magazine, I must tell you that in your May issue, it was only half nice.

My article, "Talk to your TRS-80" which appears on page 296, makes reference to the accompanying program listing, that is nowhere to be seen.

Since it has been some time since you originally had this article, I guess the listing has been lost, strayed or erased, therefore I am enclosing another for those who might be interested.

I have also adapted the original TRS-80 program of "Blackjack" to op-

Continued on page 19

used to be) to: 1) encourage student learning (remember the adage "you can lead a horse to the water, but you can't make him drink"?); and 2) compete with TV.

Most students are there to learn and are learning far more now than they were in the past, mostly because today's educator realizes not all students learn at the same pace nor from the same methods and materials. Public education is better today at meeting the learner's special needs. Most problems in today's schools are from a lack of adequate funding and public support. Today's public seems to think that today's educators are supposed to teach students the values and self-discipline that should be learned at home. It is so easy to blame the schools for all that's wrong.

As a union teacher I especially resent your last paragraph, which insinuates that unions are the reason for a lack of extra-curricular activities for students. Teachers are perhaps the lowest-paid professionals in existence. Why is it that people like Mr. Green still view teachers as public slaves who should devote all of their time and expertise to students for nothing? Many teachers, both female and male, have to hold second jobs to support their families. Many are taking post-graduate classes—at their own expense, unlike their counterparts in the business world—to either maintain teaching certificates or to advance on the salary schedule to keep up with inflation (there's no such thing as merit pay for teachers). Teachers also have families with whom they want and need to spend time. I hate to think what my job and salary would be like today if it hadn't been for the teacher's union!

Marty Krenek
Lakewood, CO

Instant Block Cursor

I enjoyed Ron Balewski's article, "Block that Cursor," which appeared in April's *80 Microcomputing*. I did, however, find two problems. First of all, I run a 48K system with two disk drives. Ron's program does not work with TRSDOS at all (TRSDOS crashes). It does, however, work well with Disk Basic. Converting the program for a 48K system takes more effort than it would seem. I added a top-of-memory indicator so that you do not have to answer the Memory Size question. The program is first saved as a disk file (via Debug). It is then loaded but not executed. Enter Basic and POKE 16414,04 and 16415,254 and voila, instant block cursor! Warning—do not issue any CMD commands after using

this. Also, you may have to turn off the expansion interface to get it out of memory.

Andrew Shecktor
Centre Square, PA

Not a Tinkerer

I found your review of the TC-8 very disappointing (May 1981). I think your review and their advertising should make clear that, to be usable, the CTR-41 must be modified. As one who does not know how to solder nor cares to know anything about hardware, I think that others should be so informed. It is not safe to assume that all your readers are tinkerers.

Phil Humbaugh
Hartsdale, NY

Notes on Games

Below is a simple listing which I used to use in order to defeat Radio Shack's Microchess. I now have an excellent chess game from Hayden Software in Sargon II.

I would also like to comment on Big Five software. It is the *best* in the market for arcade-type games. I highly recommend these games.

David Yuen
Farmingdale, NY 11735

	Player (W)	Computer (B)
1	E2-E4	E7-E5
2	D1-F3	F8-B4
3	F1-C4	D7-D5
4	C4-D5	C7-C6
5	F3-F7	mate

The Bus Contention

I would like to offer a caution to readers who may build Frank DiNunzio's joystick interface unit (June 1981, page 157) for the TRS-80 Model I. There is a theoretical bus contention problem that could eventually cause damage to the computer if the joysticks are turned on when a CLOAD is done or any other INP-based device is used.

Bus contention results when two devices place information into the computer simultaneously. All computers are designed in such a way that only one device is permitted to respond to a request by the Central Processing Unit (CPU). That is the reason for numbered memory addresses and input/output ports. When the computer executes an INP(225), an assembly command IN(FF) or a machine code D3 FF, internal circuits are decoded that allow the cassette input to talk to the CPU.

Mr. DiNunzio specifies that an INP(1) statement will activate the joysticks. In fact, any INP statement will cause his joystick interface to respond. Due to a certain measure of luck, some timing overlap present in the cassette circuit and the tough integrated circuits in Mr. DiNunzio's computer, the cassette input is overriding the joystick input and he receives correct data to the computer. How long the ICs directing the cassette operation (or those in his joysticks) will last is a matter of question, especially if any of the joysticks are moved all the way to the zero position on the same line as the cassette input.

Furthermore, Mr. DiNunzio uses four AA cells to power the interface. Again, luck has been with him. LS TTL integrated circuits (the kind used in the joysticks) are rated to run properly from 4.75 to 5.25 volts. Four AA cells normally produce between 5.9 and 6.1 volts. Any failure in the joystick integrated circuits may cause 6 volts to reach the computer bus—a virtual prescription for damage.

It isn't my intention to be hard on Mr. DiNunzio. He found a design that worked for him, and he used it. It's that kind of thinking that has led to advances in the electronics field. But I feel it is my responsibility to *80* readers to present both these dilemmas and simple solutions to them.

First, the addition of a low-power 5-volt regulator, type 78L05, in series with the AA cells will provide safe, correct power (see Fig. 1). Second, the addition of a simple address decoder consisting of two ICs will completely decode INP(1) to provide a longer-lasting TRS-80 (see Fig. 2).

Dennis Bathory Kitsz
Technical Contributing Editor

Incompatibility

After deciding to develop a word processing ability for my TRS-80, I decided to look around at my options. Being a "play it safe with hardware" type I decided to go with the Radio Shack lowercase modification. I figured that it would be compatible with everything I wanted to use and therefore would give me no problems with Ra-

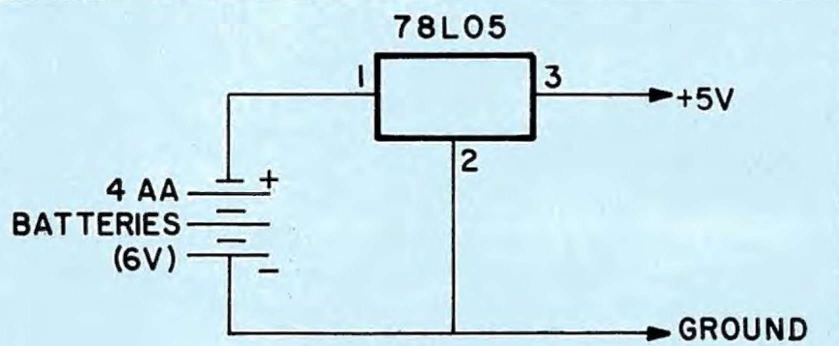


Fig. 1

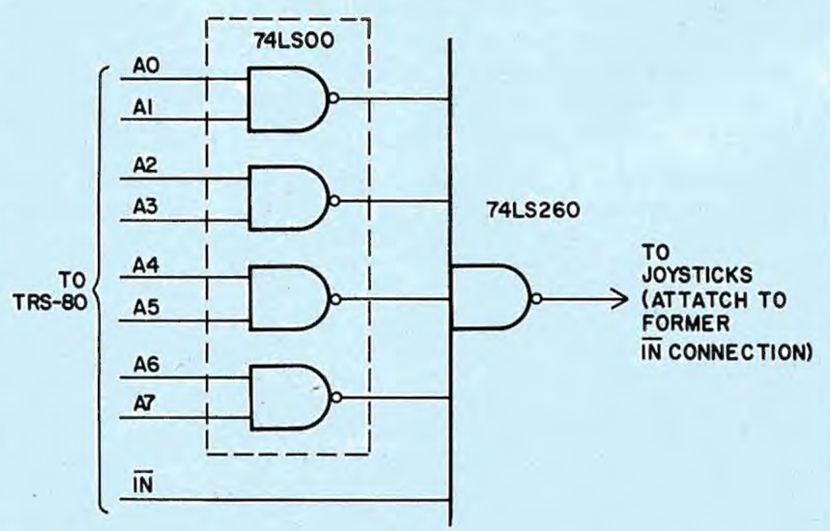


Fig. 2

80 DEBUg

erate via VoxBox. Anyone who would like a listing may have one if they will send me a self-addressed stamped envelope.

Art Welcher
360 S. Wetherly Drive
Beverly Hills, CA 90211

```
80 DIM W$(32)
90 INPUT"1.TRAIN  2.LISTEN  3.ENTER VOCAB  4.SAVE VOCAB.";Q
100 ON Q GOTO 20000,990,19000,19100
990 GOSUB1600
1000 REM GET VOICE INPUT
1010 DEFUSR2(0)=&HF003
1020 PRINT@50,"SPEAK"
1030 X=USR2(0)
1040 W=PEEK(-3078)
1050 IFW>31 THEN1500
1060 PRINTTAB(15)"YOU SAID ";W$(W)
1070 FOR I=1TO300:NEXT I
1080 GOTO1000
1500 PRINT"PLEASE REPEAT THAT":GOTO1070
1600 CLS
1610 PRINT"WORKING VOCABULARY":PRINT
1620 O=256:FPR V=11TO21
1630 READ W$(V):PRINT@O,V;".  ";W$(V);:O=O+64
1640 NEXT
1650 O=275:FOR V=11TO21
1660 READW$(V):PRINT@O,V;".  ";W$(V);:O=O+64
1670 NEXT
1680 O=296:FOR V=22TO30
1690 READW$(V):PRINT@O,V;".  ";W$(V);:O=O+64
1700 NEXT
1710 RETURN
19000 REM CALL UP VOCAB
19010 CMD"LOAD VOCAB1/CMD"
19020 CLOSE:PRINT"VOCABULARY STORED":GOTO20
19030 REM SAVE VOCAB
19040 CMD"DUMP VOCAB1/CMD (START=X'F000',END=X'FFFF',TRA=X'402D'
)"
19050 CLOSE:PRINT"VOCABULARY STORED":GOTO20
20000 REM TRAINING INPUT
20010 A=PEEK(-4095):IF A<>12 THEN CMD"LOAD VDR"
20020 REM TR INDEX
20030 REM TRAIN DIGIT ZERO
20040 FOR TR=0TO31
20050 READ W$(TR)
20060 PRINT"SAY -----";W$(TR)
20070 GOSUB21000
20080 NEXT TR:PRINT"32 WORD VOCABULARY ENTERED"
20090 GOTO20
20100 DATA"0","1","2","3","4","5","6","7","8","9"
20110 DATA"POINT","COMMA","ZIP","CHECK","AMOUNT"
20120 DATA"ENTER","OK","START","SAVE","DOLLARS"
20130 DATA"CENTS","ZERO","BACKUP","CANCEL","STOP"
20140 DATA"SCRATCH","SCRUB IT","HOLD IT","GO BACK","OVER","BUDGE
T"
20150 GOTO30
21000 REM CALL TRAINING
21010 TA=1258
21020 IF TR>32767 THEN TA=TA-65536
21030 DEFUSR1=&HF000
21040 POKE-3078,TR
21050 X=USR1(0)
21060 IF PEEK(-3078)<>0 THEN GOTO21090
21070 RETURN
21090 PRINT"ERROR - PLEASE REPEAT"
21100 GOTO21030
```

Program Listing

dio Shack's software, hardware, and probably most everybody else's software and hardware.

Eighty dollars later (is there significance to this number 80?) I find that I have to load a software driver from cassette (I have no disk ability yet) every time I want to use the lowercase mod in Basic. Well, things could be worse, I thought, I could have lost compatibility. Hah hah. The joke is on me. I did. With Radio Shack Level II Basic. Every time I try to use the String$ command the computer will lock up. The break key will do nothing. The reset key is my only hope. It proceeds to erase not only my entire Basic program but also to disable the lowercase driver that was so grudgingly put in.

A call to my friendly Radio Shack dealer who sold me the mod brought the "I haven't heard about that one yet, why don't you call the 800 number hotline in Fort Worth" response. So I tried. And I tried. And I tried. One would think that the phone company would try to sell them additional lines so that 99 percent of their customers wouldn't get busy signals.

As of this writing I have still not been able to reach Fort Worth on the hotline. If someone out there knows of a solution to this problem, I would really appreciate learning of it. By the time this letter is published, I still my not have been able to get through to the hotline.

At least the lowercase mod will work with Scripsit so that I can write this letter.

William R. Patterson
Somerdale, NJ

Modifying DOS

I just recently got around to reading my December issue when I was stopped by Ed Juge's column.

He makes brief mention that the newer drives for the Model I are 40-track, double density capable. But he fails to tell us if the DOS will be modified to take advantage of the extra tracks. I am sure many of us who stuck by the Shack would appreciate being able to use them.

Mr. Juge also states that the TRS-80 will not work correctly in double density. Why not? There must be some way to make the computer handle double density.

Gerald DeConto
Williams Lake, British Columbia
Canada

EDTASM Keybounce

My problem is keybounce while using EDTASM (26-2002). I have bene discouraged from learning assembly language because of this. My questions are:

- Can EDTASM be modified to correct keybounce?
- Is Radio Shack working on this problem?
- When will a solution be provided?

I have a 16K, Level II and I do not accept key removal, etc. as a solution.

B.H. Cathey
Jamestown, NC

Radio Shack Answers

We don't anticipate making a new release of TRSDOS for the TRS-80 Model I at this time. Even if we did, I doubt that it

Continued on page 20

would be practical for us to provide 40-track operation. There are just too many 35-track disk drives on Model I systems. Plus, Model I owners then lose the ability to easily transport disks between 35- and 40-track systems.

Mr. DeConto is correct about double density for the Model I—there is a way to do it. It requires a new disk controller chip capable of double density such as the WD 1793 used in the Model III and a new TRSDOS. However, the WD 1771 single density disk controller chip is not plug-in compatible with any double density controller chip. It's not a matter of ". . . pull one chip out and replace it with another."

I'm sorry that Mr. Patterson wasn't able to get through to our Customer Services department. We've expanded the number of people and WATS lines again but I suspect that we'll continue to have some problems during peak calling hours. The problem with Basic "freezing" occurred with early tapes of ULCBAS. There are two remedies. After loading ULCBAS and returning to Basic, make two POKEs prior to any other commands—POKE 28829,242 and POKE 28830,X. The value of X depends on the amount of memory available. For 16K X = 125, 32K X = 189, and 48K X = 253. The second is to return the ULCBAS tape to a local Radio Shack store and ask that it be sent in for exchange. The current version of ULCBAS already incorporates the changes so that the two POKEs aren't necessary.

Mr. Cathey doesn't give many specifics about his particular problem. From what he does say, I assume he is using a very early Model I keyboard with the KBFIX utility (EDTASM 26-2002 requires all available memory and overwrites KBFIX). It's an unfortunate fact of life that most electro-mechanical devices require some degree of maintenance and cleaning.

We changed to a capacitance-type keyboard when our original vendor couldn't deliver, and ended up with a bonus. It eliminated the need for contact cleaning, provided better feel and a by-product was a reduction in keybounce. If Mr. Cathey was using a typewriter he would experience the need for some routine cleaning. We do have two new packages, Series I Editor Assembler for cassette, and for disk systems. They will be available very soon, and they use their own keyboard driver routines to reduce keybounce. This should take care of the problem Mr. Cathey is experiencing.

Ed Juge, director
Computer Merchandising
Radio Shack/Tandy
Fort Worth, TX

Garbage Sorts

I have a program that requires my Model II to sort a list of 30-character strings. During the sort the computer hangs up several times (sometimes for 20-30 seconds) and then continues sorting.

The Radio Shack representative in the Language Support Group tells me that at those times the computer is doing a Garbage Sort in which it reorganizes the string variable table to reclaim lost space. If one clears more string space, then the Garbage Sorts occur less frequently but take longer.

If would be useful for you to run an article on Garbage Sorts, exactly what is happening, and how to program around them to minimize impact on run time.

William E. Jones, M.D.
Austin, TX

Model I or III?

When I recieved your June issue, I was disappointed to see that you are still not indicating in your (otherwise valuable) articles whether the information is pertinent to the Model I or Model III or both. Shape up.

Daniel M. Long
Englewood, OH

Cleaning Cable Connectors

I was plagued with spontaneous resets, erratic printing mistakes, loss of keyboard control, loss of Basic program in memory and Scripsit text turning to garbage with my TRS-80 Model I 32K one disk system. Some friends assured me my problem most likely was dirt in the ribbon cable connectors and/or a loose chip or chips.

Following directions, I removed all cable connectors from the EI, printer and keyboard and cleaned the female connectors with a squirt of color TV tuner cleaner. I rubbed all male card edge connectors with a clean pencil eraser and wiped them off with cleaner on a cotton-tip applicator. I removed the six screws on the bottom of the keyboard, flipped it and removed the cover. Supporting the circuit board I pressed each of the chips into place completely. The EI was similarly treated (leave it upside down to reach the chips, and be particularly careful to completely seat the big chip on the side). Following reassembly of my hardware I found to my delight that those hardware problems seem to be solved.

I wanted to share my problem and solution with any readers experiencing similar frustrations. My friends also said that power spikes and drops (not a sag) could cause my hardware problems. If cleaning doesn't solve the problems, a possible power spike/drop fix could be suggested.

Spencer Weersing
Montague, MI

File Folder Labels

While working on my system I wrote a very short program that should help around the office or even with the home filing system.

It is based on the Line-printer II, but can be used on any printer that allows expanded letters. The file folder labels that are typed are usually hard to read in the normal type size. Using this utility will allow for two lines to be entered and printed, followed by three line feeds to advance the standard Avery 200 self-adhesive file folder labels.

Because these labels are only printed once, there is no storage and the program recycles after each label. If the label only requires one line input then a simple <enter> will omit the second line but still allow for label advancement.

The program is simple; lines 10 and 20 clear the screen and prompt the inputs. Lines 30 and 40 set the printer for expanded format, and give leading blanks and output to the printer. Line 50 gives the line feeds to advance the next label and line 60 returns for the next input.

```
10 CLS : LINEINPUT " ENTER FIRST LINE: ";A$
20 LINEINPUT "ENTER SECOND LINE: ";B$
30 LPRINT CHR$(27);CHR$(14);"   ";A$
40 LPRINT CHR$(27);CHR$(14);"   ";B$
50 FOR X = 1 TO 2: LPRINTCHR$(138) : NEXT
60 GOTO 10
```

Bill Bowman
Selkirk, Manitoba
Canada

Debugging T-Bug

I have a Model I, Level II, 16K, and the new ROM. This I understand will not cause any problem with the article "Get T-Bug High" in the January 1980 *80 Microcomputing.*

After successfully moving T-Bug to 7380H the PEEK, POKE program was run. The text states 213 locations should have changed; I had only 198 locations changed.

NEVER UNDER$OLD.

Thats right. If you can find a lower price in this magazine WE WILL BEAT IT! Period.*

SPECIALS

SPECIAL #1 . . . . $44.50
TRS-80 Disk and Other Mysteries Book, Box of Verbatim Diskettes, and a Plastic Library Case.

SPECIAL #3 . . . . . . $149
Apparat NEWDOS/80, Box of Verbatim Diskettes, and a Plastic Library Case.

SPECIAL #4 . . . . . . $190
Microsoft Basic Compiler, Box of Verbatim Diskettes, and a Plastic Library Case.

SPECIAL #5 . . . . $49.95
Microsoft Basic Decoded and Other Mysteries, Box of Verbatim Diskettes, and a Plastic Library Case.

PRINTERS

EPSON MX-80	CALL
MX-80 Graphics ROM	CALL
NEW EPSON MX-80 TF	CALL
Okidata Microline 80	$415
Okidata Microline 82	$625
Okidata Microline 83	$899
IDS Paper Tiger 460G	$1050
Anadex DP-9000,1	$1195
Anadex DP-9500,1	$1295
Centronics 737	$689
NEC Spinwriter	$CALL
NEC w/ Sellum Option	$CALL
Starwriter 25 cps	$1440
Starwriter 45 cps	$1770
Malibu 165	$1995
MPI 88G	$495

TRS-80 HARDWARE

TRS-80's	$CALL
Micropolis 77 track	$399
Percom Doubler	$199
Percom Separator	$27
Orchestra 80	$79
Shugart 40 tr.	$299
Pertec 40 tr.	$299
MPI 40 tr.	$299
16K Memory Kit	$19.95
Cat Modem	$145
D-Cat Modem	$155
Lexicon Modem	$125
Verbatim 525-01	$26.50
Verbatim 8" DD	$44.00
Isolator ISO-2	$49.95

NEW! Double Density Special

A Percom Doubler, Apparat's NEWDOS/80, Double Zap II [to convert the NEWDOS/80 to single and double density], a box of Verbatim Datalife diskettes, and a Plastic Library Case. A $436 Value...for

$359

Send For Our New Free Catalog!

*Include $2.00 shipping and handling on all orders. Visa and Master Card accepted. Never undersold offer applies only to items currently deliverable from other retail vendors at advertised price. All never undersold offers good as supply lasts. Please add $2.00 for all COD orders. Please call for items not listed. We gladly answer any questions on all of our hardware, software, and supply needs. Quanitity discounts available. School purchase orders accepted. Please remember to figure competitors shipping and handling charges when arriving at never undersold price.

POWERFUL SOFTWARE

Racet's New DISCAT $45
This New machine language disk catalog program will help you keep track of your programs in a neat, orderly library. It can catalog up to 1900 programs in each of 9 catagories for a total of 17000 program location records.

Snapp Spooler $90
Fully automatic print spooler for the model I. Increases your computers efficiency by freeing the computer while printing.

TRACKSESS $24
This is the complete disk zap utility, capable of copying standard and non standard formatting. The dynamic editing allows you to change any byte in any sector with amazing ease

SCRINPUT $29
Format your screen displays and input routines with this machine language subroutine. Gives your program a more professional look.

WORD PROCESSING

SPECIAL DELIVERY $118
The best in mailing list programs for the TRS-80 Models I & III. Special Delivery is easy to use & is written in fast machine language. Special Delivery allows you to merge your mailing list file with text files written with the Electric Pencil* or Scripsit* to create personalized form letters.

XTRA SPECIAL DELIVERY $189
Does everything special delivery does and more. Extra allows you to include special keyed messages in your text file, and includes Mailsort which allows sorting on multiple fields.

PROSOFT $79.95
'PROSOFT' Word processing software is written for the TRS-80 and the Centronics 737. Takes full advantage of all the capabilities available on the 737 including true proportional spacing. Comes complete with a 110 page users manual.

213-883-8594

ALPHA BYTE STORES

~69

5115 Douglas Fir Road
Suite B
Calabasas, California
91302

Going to the 22 manual changes, locations 758CH and 7676H had stayed 47H instead of going to 77H; I left these at 47H.

After punching the new T-Bug on tape, power down, power up, and loading the new T-Bug, everthing was fine until an L, P, J or R command was tried. I can look at all the memory I want, but anything else gets a memory size question staring at me.

The original T-Bug had to be upgraded for the new ROM.

At last count, 12 times following letter by letter to the text. Can anybody straighten me out?

Gary Bennett
Norcross, GA

The Author Responds

Your letter indicates that you end up with 15 fewer changes than the article says you should have. Taking a clue from the two locations (758C and 7676 hex) that you said were not changed by your Basic routine, I have found that each one of these locations has successive bytes in the range of 43 to 49 hex (67 to 73 decimal). On the assumption that your PEEKing and POKEing is somehow incrementing the address location twice after each change, I checked to see just how many locations there are where two successive bytes fall into the range we are looking for. There are 15 such places! With the exception of 758C and 7676, all of these places need the second byte changed to the high value and the first byte changed back to its original value. My guess is that, for some reason, your Basic program is skipping over an address each time it does a POKE.

The 15 successive byte locations, addressed in terms of T-BUG High, are:

74E9/A 74EF/74FO 74F8/9 74FE/F 75OA/B 758B/C 7675/6 76A6/7 76B4/5 76DB/C 76E1/2 771A/B 772C/D 7734/5 773A/B

Notice that, with the exceptions of 758C and 7676, all first locations appear in the change back table of the article. Each of the second locations should have been changed, correctly so, by the POKEing, but my guess is that they were not.

Other than 758B/C and 7675/6, assure that the second byte of each of the places shown has the high value, while the first byte retains the low value. That should make T-BUG High operational for you, I'll bet!

The program in the article, after correcting line 30 to read "... THEN 70," will do all PEEK/POKEing properly. If the suggestions above prove to be the correct solution, check the program you used to see if it indeed was doing a double incrementing of the address (P) when making a POKE change.

During the past year and a half or so, I have had a great many inquiries from readers of 80 looking for help with T-BUG High. Virtually each inquiry presented a different problem, and I hope that I have helped all those who have written and phoned from all over the United States. The big help, however, should really come from Radio Shack... by now they ought to have realized that most of us have gone well beyond Level I, 4K... and when they issue a new T-BUG for Model III, I trust they will note that we want our utility programs up and out of the way!

Irwin Rappaport
Upper Saddle River, NJ

Dumpload for Backup

As a response to my article (*80 Microcomputing*, March 1981, p. 250) Mr. William C. Perry sent me a copy of his program Dumpload to try out.

Dumpload is an assembly language program designed to write a disk's contents to cassette, read it back and create a disk. It contains two sets of tape routines, one at the normal 500 baud and one at around 1800 baud. The 1800-baud routines will allow you to record three disks per side on a C-60 cassette. I found the program to be extremely easy to use and quite complete. I highly recommend the program for backing up disk libraries.

Dumpload is available from Complete Computer Services, Newburgh, IN. It will run on a 16K Model I.

Robert J. Hocking
Harris Corp.
Palm Bay, FL

The Reviewer Replies

This is a reply to Programma International's letter regarding my review of their 80-Graphix board (February and April *80 Microcomputing*, 1981).

Rereading my review, I did become aware of a possible misinterpretation about the use of the cassette port as Jeff Lasman says. All the wiring is done internally and use of the cassette for I/O is completely unhindered.

However, I stand by the rest of my review. My major objections are still quite valid. The point is that the 80-graphix board is a *character generator*, and as such does not give one better graphics than the Apple. To say that you have 384 by 192 resolution because you can program 64 characters of up to 72 pixels each is, to be blunt, as ridiculous as saying that you have the same resolution with normal TRS-80 graphics since CHR$(191) will turn on all of the Programma's 72 pixels/byte. The Apple has far superior graphics to the TRS-80 with or without the 80 Graphix board. For some uses, such as creating character sets, the 80 Graphix board is excellent, but for high res graphics, such as in high density graphics, it is not *at all* useful. I dearly love my TRS-80, but it is *not* a good graphics computer.

I fail to see Mr. Lasman's point about the use of Set and Reset. I said that they are effectively disabled, since the graphics characters are created and ordered in such a way that if you OR the contents of the byte with 80H and OR it with a pixel position, you set a new point. If you create a different order, you effectively destroy the utility of Set/Reset/Point. Mr. Lasman says, "sure you can use Set and Reset. Just design a character set identical to the one you already have" (freely paraphrased by myself). For this, I get to pay $150? I can do the same thing by remaining in normal TRS-80 graphics mode (and do it more cheaply!).

I take exception to Mr. Lasman's comment that I "berated" the demo programs. I did not. They are decent demo programs.

And I still say that the character generator is not well thought out in terms of user-interface. Why not use arrow keys and turn on and off the blocks in a simulated "full screen" editor? Entry would be much quicker and easier. Then save the data statements as a "merge"-able file (that really isn't all that hard to do).

Bruce Douglass
Vermillion, SD

Users Groups

80 Microcomputing is gathering information on TRS-80 users groups for publication sometime this fall. If your group has not sent us any material, or has updated information, we would like to hear from you soon.

Send your address, phone number, person to contact, dues and other pertinent data to:

Michael E. Nadeau
80 Microcomputing
Pine St.
Peterborough, NH 03458.

EDUCATION 80

by Earl R. Savage

"The original cassette mass storage system is unsatisfactory."

I apparently hold some heretical views on mass storage. For now, I'll limit the discussion to the Model I because of the overwhelming number in the schools.

There is one point on which all of us can agree. The original cassette mass storage system is unsatisfactory. It works quite well (after the very early production runs) and it is inexpensive, but *slow*.

Every load and save takes minutes. Searching for a program seems to take hours! And that's with program material. The situation with *data* is even worse. The result is a lot of wasted time waiting for the tape to run. In school, that is lost instructional time.

Of course there is a better way—in fact, there are several. You must choose the method most suitable for your requirements. Let's take a look at them and I'll give you my impression of each one.

Disk

Probably the first thing you think about when you hear the words "fast mass storage" is disk operation. Well, in my opinion you are both right and wrong. Certainly disk is fast, and for some types of operation there seems to be nothing better. However, it is the most expensive method available.

Not only is there the cost of the drive(s), but you must have a disk controller (in an expansion interface of some type). In addition, you must purchase a disk operating system (DOS). The DOS, in turn, requires that you have more memory to make up for that which it occupies.

I believe the cost/benefit ratio is unjustifiable in most TRS-80 school applications. This is true especially in today's extremely tight financial circumstance of almost every school in the country.

Another great concern is the suitability of disk for school use. Both disks and disk drives require careful handling and, for that reason, do not hold up well in general use by relatively untutored students (and teachers). Elementary school students can wreak havoc on disks. At best, they may be suitable in secondary school computer labs. There, too, special efforts should be taken to reduce the quantity of dust that is prevalent in heavily-traffic classrooms.

I place fairly restrictive limits on disks in schools. For most such applications, I prefer some of the more rugged, less expensive and quite efficient means of mass storage.

Speed-up Modification

First, of course, there is the speed-up modification to the computer circuit itself. The most reliable one for general use provides a 50 percent increase in speed. I have had this option on my Model I for almost two years and it has functioned with absolutely no problems.

Do not scoff at a mere 50 percent increase since it applies to program execution as well as to Saves and Loads. At 750 baud, using the regular cassette operation, you will cut the time of mass storage and retrieval by one-third.

Stringy-Floppy

The closest thing to disk operation is the Exatron Stringy Floppy (ES/F). It is much better from the standpoint of being significantly less expensive and more physically rugged. The ES/F closely approaches disk in speed and in file management. It consists of an operating system and a digital recorder-player.

The ES/F does not require additional memory because the operating system is in a built-in EPROM. A special miniature cassette (wafer) contains tape in a continuous loop and this greatly reduces program/data search time.

The operating speed of the ES/F is about 14 times that of the regular cassette. That figure is somewhat misleading because of even greater efficiencies in all types of file management. Further, I have experienced absolutely dependable operation with the speed-up option which makes it 21 times faster than cassette.

The Poor Man's Floppy

Another alternative means of mass storage is the IC-8 by JPC Products. Called the poor man's floppy, it is available in kit or assembled form and includes a short relocatable utility in addition to the device and cables.

The TC-8 is five times faster than cassette, using a regular cassette machine by transforming the signal stream. It, too, handles data with increased efficiency and is faster than the speed would indicate. The TC-8 functions well with the computer speed-up option (7.5 times faster than cassette) if your cassette machine is in good condition and you use good tape.

HISPED

The final improved mass storage method is a software approach. HISPED by Palomar Software is one such utility program. It offers a choice of eight Save and Load speeds up to about five times the speed of cassette.

To achieve the highest speeds available with HISPED, requires a minor circuit changes in your computer and the use of a quality cassette machine. Like the others, it provides greatly improved efficiency in data handling.

As you have seen, you do not have to choose between cassette and disk only. Careful consideration of your school users, applications and finances should produce the method best for you.

Lemon Extract

When you furnish to your students a listing to study, do you give them a list of the variables in that program? When you or they want to modify a program, can you produce a list of the variables (other than manually)? If you answered no to either of these questions, you are missing a valuable utility.

Lemon Extract (from Hanson House, 55 Hanson Place, Stratford, CT 06497) will extract the variables in a program in alpha order and by the first occurrence. It prints out the program name, its length in bytes, the total number of variables, and the as-of date. Then, the variables are printed in alpha order in five vertical columns with the first occurrence of a line number.

You should have Lemon Extract or some utility which will print the variables in a program. You will find that it is a real time saver and very useful to both you and your students.■

THE ALPHA I/O SYSTEM

A COMPLETE FAILURE?

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).

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.

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).

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.

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

PRINTER STAND

$49.95

PRINTER CABLE EXTENDER

Adds 4 ft. to your existing cable, printer extension connects between Exp. Int. and your present printer cable........$27.50

ALPHA Product Co.

✓210

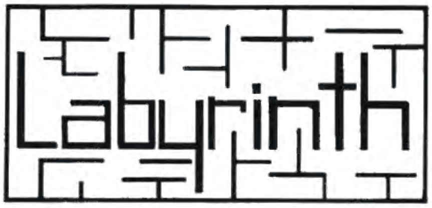
ADD $2.50 PER ORDER FOR SHIPPING AND HANDLING
ALL ORDERS SHIPPED FIRST CLASS MAIL.
WE ACCEPT VISA, MASTER CHARGE, CHECKS, M.O.
COD: ADD $2.00 EXTRA.
QUANTITY DISCOUNTS AVAILABLE.
N.Y. RESIDENTS ADD SALES TAX.

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

80 REVIEWS

Edited by Pamela Petrakos
and Michael Nadeau

"... after walking the hallways for hours, you rather expect to turn a corner and come face to face with yourself (playing Deathmaze)."



Labyrinth
Med Systems Software
Chapel Hill, NC
$12.95 cassette
$16.95 disk
16K, Level II or Model III

by Debra Marshall
***80* staff**

Well, here 'tis, Deathmaze fans—Frank Corr's (and friends') second excursion into the dark, dim corridors of three-dimensional graphics—kind of.

Once more you'll find yourself looking down a long hallway with doors opening on either side. You know that whichever way you go, it's going to mean trouble—usually, big trouble. Take your courage in hand, and do it anyway. Believe me, you may never get out.

There are more hallways, more boxes full of junk, more empty rooms, and a good selection of pits, as well. Labyrinth *looks* like Deathmaze, graphically, and the information sheet that comes with it is equally brief and bare of help or hints. But that's okay—finding commands that work is part of the mystery of this adventure. If you get stuck—really, truly engulfed—the Med System's people are very nice about giving hints. I assume that's because they don't want the responsibility of thousands of raving, game-playing maniacs on their souls.

Past the Minotaur

But let's face it—in a game like this, you should take weeks or longer to admit you're stuck. First you've got to get through the fog, and past the minotaur that I keep bumping into as soon as I step through the door. Then, there are several spots (or is it just one?) that you simply can't get out of. I've walked for hours in circles before realizing what I was doing. Like Deathmaze, this game has to be mapped.

Mapped, that is, if you can do it. The fog makes it difficult to see in, or get through, certain parts of the maze. And this is the only place I've ever visited where you can jump into a pit and come up somewhere on the very same level you jumped from. Mapping this maze is like putting together parts of a jigsaw puzzle. Expect to spend many, many hours doing it.

The mapping has to be done in sections, and very carefully. At the same time, prepare to suspend your normal perceptions. Up isn't necessarily up, it may be over, and down is usually a quick route to somewhere you'll wish you weren't. The labyrinth seems to overlap itself in at least one spot.

Bring Along a Picnic Basket

Be sure to bring along some extra torches and a picnic basket or two. If you don't, you're going to have to search for them, and they're well hidden, as are several nasty surprises that you'll just happen across on your way. You'll have your pockets full of junk by then; maybe some of it will come in handy.

If you ever get out, please let me know—I haven't. As another editor here fantasized, after walking the hallways for hours, you rather expect to turn a corner and come face to face with yourself (playing Deathmaze). This game just may send me to the Asylum. ■

T80-FS1 Flight Simulator
Sublogic Company
Champaign, IL
$25

by Jon Lindsay

Not being used to real-time simulations, the T80-FS1 Flight Simulator by Sublogic Company was something of an enigma. The ads promised me the chance to simulate an airplane flight. My equipment, however, did not support the program. That is, not until I acquired a TRS-80.

According to the accompanying manual, the T80-FS1 package is a reworked version of a program originally designed for the Apple II, improving the characteristics of the game/simulation. The manual, which also serves as "ground school" for the operation and conduct of the aircraft, stresses the performance characteristics of the program, pointing out the similarities between the program and an actual aircraft. The aircraft's characteristics are similar to the World War I Sopwith Camel as well as a Piper Super Cub. That is, the weight, length, service ceiling, horsepower and top speed of these aircraft are simulated. This brings to mind the question of whether this program is in fact a simulation or just a game. The answer should probably be left to the user to decide.

Program Operation

Using a cassette, the operator must first type SYSTEM in order to load the assembly language preloader. Then type FS1. This is the preloader. When the computer responds again with a *?, type /. This will load the actual program after several minutes. The program automatically executes and you are ready to fly.

Before you can think about playing the included war game (British Ace), first you must learn how to fly the aircraft. Despite having flown planes for several decades, flying this simulator was not as easy as it sounded.

From the Cockpit

The pilot finds the plane sitting on the approach end of runway 27 (that is, facing due west or 270 degrees), awaiting departure. The pilot has a view looking out through the forward cockpit. There are two views possible: One is out the front of the cockpit, and the other is from on top of the aircraft looking down.

The latter is the radar view which allows for much greater perception of the loca-

GAME BREAKTHROUGH:

THE STICK 80

- ☑ FEATURES THE FAMOUS **ATARI JOYSTICK.**
- ☑ WORKS WITH ANY LEVEL 2 AND DISK SYSTEMS.
- ☑ PLUGS DIRECTLY INTO KB. OR E/I (EXPANSION PORT).
- ☑ INCLUDES SIMPLE, DETAILED INSTRUCTIONS TO MAKE JOYSTICK VERSIONS OF MOST ACTION GAMES.
- ☑ COMPATIBLE WITH **ANY** OTHER TRS-80 ACCESSORIES.
- ☑ FULLY ASSEMBLED AND TESTED. READY TO PLUG IN AND ENJOY.
- ☑ FUN TO EXPERIMENT WITH IN **BASIC:** SIMPLY USE INP(Ø).
- ☑ 90 DAY PART AND LABOR WARRANTY.

UNCONDITIONAL **MONEY BACK GUARANTEE.** Try your STICK-80 for 14 days. If for any reason you are not delighted with it, return it for a prompt and courteous refund (including shipping and handling). PRICE INCLUDES ATARI JOYSTICK + ALPHA INTERFACE + DETAILED INSTRUCTIONS + DEMO PROGRAM. ONLY $39.95

BIG FIVE SOFTWARE SUPER FAST MACHINE LANGUAGE ACTION GAMES

If you and your TRS-80 have longed for a fast-paced arcade-type game that is truly a challenge, then **SUPER NOVA** is what you've been waiting for. In this two player machine-language game, large asteroids float ominously around the screen. Suddenly your ship appears and you must destroy the asteroids before they destroy you! (But watch out because big asteroids break apart into little ones.) The controls that your ship will respond to are thrust, rotate, hyperspace, and fire. All right! You've done it! You've cleared away all the asteroids! But what is that saucer with the laser doing? Quick! You must destroy him fast because that guy's accurate! ◆◆◆◆◆◆◆◆◆◆◆◆◆◆ **SUPER NOVA®**

The sound of the klaxon is calling you! Cruel and crafty invaders have been spotted in battle formation warping toward Earth at an incredible speed. Suddenly, your ship materializes just below the huge flock of invaders. Quickly and skillfully you shift right and left as you carefully fire your lasers at them. But watch out! A few are breaking out of the convoy and flying straight at you! As the whine of their engines gets louder, you place your finger on the fire button knowing all too well that this shot must connect—or your mission will be permanently over! With sound effects! ◆◆◆◆◆◆◆◆◆◆◆◆ **GALAXY INVASION®**

Your TRS-80 screen has been transformed into a maze-like playfield for this game. As your ship appears on the bottom of the screen, eight alien ramships appear on the top. All of them are traveling at flank speed directly at you! Quickly and boldly you move toward them and fire missiles to destroy them. But the more aliens you destroy, the faster the remaining ones become. If you get too good you must endure the wrath of the keeper of the mazefield: the menacing "Flagship". You must destroy him fast because, as you will find out, that guy's accurate! With sound effects! ◆◆◆◆◆◆◆◆◆◆◆◆◆ **ATTACK FORCE®**

With thousands of stars whizzing by you, your **SPACE DESTROYER** ship comes out of hyperspace directly under a convoy of aliens. Almost effortlessly, you skillfully destroy every last one. But before you can congratulate yourself, another set appears. These seem to be slightly more intelligent than the first set. Quickly you eliminate all of them, too. But your fuel supply is rapidly diminishing. You must still destroy two more sets before you can dock with your space station. All right! The space station is now on your scanners! Oh no! Intruders have overtaken the station! You must skillfully fire your neutron lasers to eliminate the intruders from the station before your engines run out of fuel and explode! With sound! ◆◆◆◆◆◆◆◆◆◆◆◆ **COSMIC FIGHTER®**

NEW

The second **Big Bang** has occurred and the galaxy is full of stray asteroids and meteors. As you look through your space port you see a belt of asteroids drifting across the screen blocking your path to the safety of the space station above. But be careful because meteor showers, exploding suns and invading aliens may strike your ship and send it hurtling back to ground level. How many times can you and your opponent maneuver through those obstacles before time runs out? With sound effects! ◆◆◆◆◆ **METEOR MISSION II®**

SPECIAL

INTRODUCTORY OFFER:

Order the STICK-80 with one or more games and choose one:

•FREE ALPHA GREEN SCREEN
OR
•FREE MAGIC ARTIST PROGRAM

★ IN ADDITION YOU MAY:
★ ORDER 2 GAMES AND TAKE 10% OFF the games
★ ORDER 3 GAMES AND TAKE 15% OFF the games

All games are written in machine language and supplied on cassette.

EACH GAME IS AVAILABLE IN 2 VERSIONS

A (SIDE 1: LEVEL II 16K FOR MODEL 1
SIDE 2: LEVEL II 16K FOR MODEL 3 $15.95

B (SIDE 1: DISK 32K FOR MODEL 1
SIDE 2: LEVEL II 16K FOR MODEL 1
THE DISK VERSION SAVES THE HIGH SCORES ON DISK $17.95

• ALL THESE NEW JOYSTICK GAMES WILL ALSO WORK USING THE KEYBOARD WITHOUT ANY MODIFICATIONS.

GOOD NEWS: If you already have a non-joystick version of these BIG FIVE SOFTWARE games, send the original tape(s) with your STICK-80 order + $3 per game. We will send you a new Joystick version of your game(s).

ALSO AVAILABLE **DOUBLE-STICK 80:** 2 ATARI JOYSTICKS + ALPHA DUAL INTERFACE + INSTRUCTIONS + DEMO PROGRAM: $59.95

✓551

ALPHA Product Co.

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

85-71 79 St. WOODHAVEN N.Y. 11421 ☎ phone: (212) 296 5916 tion of the plane. The problem with this latter view is that it is not very good for actually flying the plane. And the problem (at least initially) with the front view is that the limited graphics capability of the TRS-80 fails to give a realistic view of the attitude of the plane.

"Despite having flown planes for several decades, flying this simulator was not as easy as it sounded."

This is not to suggest that I thought the program failed in its objective to simulate flight. On the contrary, I was very impressed with the effort. It must be remembered that this aircraft is not meant to be a F-15 Eagle. Rather, it is a Sopwith Camel which did not have some of the more sophisticated equipment of the present day. But the fact is that the simulator does have more equipment than did the Camel. In the words of the manual, "If you can rationalize a tricycle-geared, good handling Sopwith Camel with a radar screen, you are ready to fly the FS1."

The flight controls consist of a number of keys on the TRS-80. The views are controlled by the up-arrow (front view) and the down-arrow (looking down on the plane). These are also controls for the throttle (left and right arrows), elevators (T and B or V keys), aileron/rudder (F, G and H keys) and landing gear (U and D keys). Brakes can be applied while on the ground through the period key. While the keys are scanned rapidly for commands, the response is not instant. You must hold a key down longer than an instant in order to make sure that the computer has read your request. Failing to do so can result in turn rates, for example, that can lead to a dead man's spiral.

Control of the moving aircraft on the ground is different from that in the air. On the ground, you steer the plane like a car. A left turn starts the plane (actually the environment) turning in that direction. But hitting the aileron/rudder G key (neutral) control will cause the plane to immediately proceed in a straight line. This does not apply when the plane is in flight.

In the air the kinematic effects of lift, gravity, drag and momentum are brought into play. The flat two-dimensional attitude of taxiing becomes three dimensional once in flight. As in actual flight, you start a turn and then neutralize it (with the G key). The plane's roll rate, however, is all that is neutralized. The plane itself is still in a turn until you apply opposite aileron/rudder to roll out of the turn. Leaving the aileron/rudder applied is likely to produce a barrel roll.

Instrumentation consists of the following gauges: airspeed indicator, altimeter, micro-altimeter (for precise altitude during landing), oil pressure, oil temperature, fuel, tachometer, compass, turn-rate indicator, vertical velocity, roll-rate indicator, elevator indicator, throttle, as well as three indicators for the game—score, bombs and ammo. The bombs are released by the X key and the machine gun ammo by the space bar. What is missing from the gauges that would be much appreciated is an artificial horizon. But then, one cannot have everything.

Taking the First Flight

We are sitting on the end of the runway, engine running and brakes set. Now is the time to fly! In seat-of-the-pants style, apply the throttle to about two-thirds open. As the prop overcomes the inertia of the plane, the plane slowly starts to accelerate down the runway. A flick of the B key will give one notch of up elevator. The airspeed climbs to 65 mph and at about half way down the strip, the plane slowly rotates.

The airport's altitude is 410 feet above sea level. As the plane lifts into the air, the micro-altimeter reflects the gaining altitude. The vertical speed (velocity) indicator (VSI) also reflects the climb by showing the rate of climb, depending on power and elevator setting. Looking forward you can see the end of the runway beginning to appear through the windshield. As it slides beneath the plane, hitting the down-arrow will give a radar view that will show the plane passing over the end of the runway at the same time.

The compass still shows 270 degrees. Since this is a Sopwith Camel, this heading indicator can be called a compass. It appears to perform more like a directional gyro, however. The latter is innately more stable during turns. Time to start a left turn. The F key applies left aileron/rudder. A glance at the turn-rate indicator shows an enlarging number with a minus sign in front of it. This indicates that a left turn is in progress. If turning to the right, the number is positive. Of course, just looking out the cockpit will show that you are in a left turn.

A standard rate turn is one that is executed at a rate such that a 360-degree turn will take two minutes to complete. This guage, the roll-rate indicator, measures the number of degrees per minute. A standard rate turn, therefore, is 180 degrees per minute. Since this is a very controllable rate, it is a preferred rate of turn. Less than that is somewhat shallow and will take a long time to complete. More than that increases the likelihood of a turn that quickly gets out of control. (Note that even though the aileron/rudder is in neutral, the turn is still being executed.) To return to straight and level flight, one must roll out of the turn by applying opposite aileron/rudder. . . just like a real plane.

With the turn properly established, the plane is allowed to continue until it approaches 90 degrees, or the opposite direction of the take-off. The plan is to make a landing back at the airport. As the plane turns, looking forward will show a series of moving lines which represent the one mile wide sectional lines as they appear before the windshield. These lines are part

"We are sitting on the end of the runway, engine running and brakes set. Now is the time to fly!"

of a large grid that makes up the world of your aviation adventure. Going beyond the grid is to venture into the vaporous ether of the unknown. Nothing is out there! That is not to say that you cannot go there. You just won't find anything.

Flying over the outer boundary of the grid on a heading of 90 degrees, for a minute or so, will give us a reasonable amount of time and distance to prepare for the approach to the airfield. Though we cannot always see it, we know where it is. The vision from the cockpit is changed by raising and lowering the landing gear. With the gear up, the grid is more readily seen but the runway detail is absent. Lowering the gear will allow enough detail to recognize the centerline of the runway during the approach. It is very important to a good landing that we can see the full runway in its greatest detail.

Again, executing a left turn at standard rate, the plane is allowed to come around to a heading of 270 degrees. This is where the runway lays. There is no guarantee at this point, however, that you are actually lined up with the runway. If you did not standardize your turns during take-off and approach, you will have to make adjustments during the final approach in order to correctly align the plane with the runway.

With the landing gear in down position, the view out the front, at this point, is remarkable. The paralax of the runway is readily apparent (but only if you are lined up). If your approach to the runway was perfect, it appears to be growing in size and perspective. With the power throttled back and slight aileron/rudder adjustments to finish aligning the plane with the runway, the elevator is gradually raised to allow for flaring of the plane as it touches down. Some braking will probably be necessary to bring the aircraft to a halt. Ideally, the plane should be sitting on the centerline of runway 27. To check that, look at the radar view. That gives you the final perspective on the landing.

This landing is more easily described than performed. And perhaps that is the challenge of this program. Repeated attempts netted only fair results at landing. And having made a good landing one time was no guarantee that it could or would be repeated another time. Because of the variation in approach technique, every landing is unique!

Looking at the World

Once you feel that you have mastered the flying aspect of the program sufficiently, you may want to explore the rest of the available world. The world, clearly described and mapped in the manual, consists of a grid of section lines. The actual world is comprised of 36 square miles of terrain, or six miles per side. Within this area is the British airbase (from which you depart), a civilian airport and an enemy airbase (with which you may conduct a war —hence the score indicator, ammo and bombs). Since this airplane is slow, it takes a bit of time to get anywhere. At the same time, fuel is being used.

Should you need to refuel, you must return to the British airbase from which you came, land and taxi to the hanger on the ramp and stop. Magically your Sopwith will be refueled with 38 gallons of petrol and you can continue warring or making touch-and-goes or lazy eights in the sky. If you are really anxious to get back into the sky, you may simply press the zero key. This relocates the plane back on the end of the runway, with the engines revving, ready for take-off. Reset relocates the plane and replenishes ammo and fuel. At the same time it turns off the war mode. It does not, however, reset the flight controls or the throttle. This can make for some interesting, unexpected take-offs from time to time.

"Because of the low resolution of the graphics, performing a maneuver like landing probably wouldn't aid you in landing a Cessna 150."

Game or Simulation

The program, to its credit, is both a game and a simulation. But the game is a real-time game. By hitting the W key, the war is on. There is no turning back. This can be a fun source of diversion for those interested in this kind of activity. And there is quite a challenge to it, hence the score, ammo and bomb indicators. I, however, enjoyed the challenge of routine flight maneuvering and the simple pleasure of cruising around the available world and exploring it as it presented itself. Landings and take-offs do require considerable amounts of skill in order to perfect them. There is a remarkable similarity of the subtleties of flight between the actual plane and the program.

Whether this simulator can actually result in transference of learned flight technique from simulator to actual aircraft is debatable. Certainly Sublogic makes no such claim. They are, however, eager to point out how similar the flight characteristics between program and plane are. I would certainly have to agree. Because of the low resolution of the graphics, performing a maneuver like landing probably wouldn't aid you in landing a Cessna 150.

Flying is as much an art as a science. The keyboard of the simulator would probably not suffice for a control yoke in that it offers very little feel. On the other hand, during the time of the simulation when you must rely on instruments for aircraft attitude, altitude and direction, I feel that the same mental acuity is necessary to accomplish the task here as under actual flying conditions.

Flying on instruments means that you must apply a rigid scanning procedure in order to avoid losing control. Fixating on an instrument for a moment may cause unsuspected surprises. Next thing you know the plane is in some very unusual attitude...in other words, out of control. Flight is not in one or two dimensions. It is a three-dimensional, real-time process. And it is this aspect that gives spice to the whole event. So the key to flying under Instrument Flight Rules (IFR) is to anticipate problems and correct them before they become large and uncontrollable. The T80-FS1 Flight Simulator allows you this capability.

Is It Any Good?

Because it is possible to place the aircraft in different spatial positions, each flight is both a new learning experience and challenge. The environment on the CRT appears to change with the aircraft's different attitudes, giving you a reasonable perspective of this orientation. The more you experiment with the program, the more skilled you become in flying the simulation. All in all, T80-FS1 is quite a delight to experience and one heck of a way to take a trip. Happy landings! ■

Everest Explorer
by William Godman and Don Knowlton
Acorn Software Products, Inc.
Washington, D.C.
$14.95 cassette
$20.95 disk

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

Have you ever wondered what it would be like to climb the world's tallest mountain? Can you imagine yourself climbing an iceflow, clinging to a near vertical ledge or working cautiously up a precipice while the thin wind whistles by? What is it like to stand on a piece of the earth that is higher into the stratosphere than most commercial airlines fly?

If you would like to answer these questions, don't bother with Everest Explorer, the new strategy game for the TRS-80 Level II. All this game is going to teach you is how to handle the problems of shifting supplies from one camp to another to insure everyone has enough food, water and a comfortable bed.

Expectations

Basically, I expected to have to make critical climbing decisions—how to get around or over a chasm, how to get up a rock face, that sort of thing. I did not expect graphics, which was just as well because there are none. I did expect an adventure where you make your move and the computer tells you the result of it and describes any new challenges you may face.

However, what I found was that I was completely drowned in the minutia of supplying the expedition while the computer had all the fun of deciding on actual climbing strategies.

Everest Explorer starts you in Katmandu, the traditional kicking-off point for Everst expeditions, with an operating budget randomly selected from a low of $80,000 to a high of $275,000. You spend this money on climbers, Sherpa guides (Tibetan people who are skilled climbers), tents, oxygen, food and fuel. The program has considerable flexibility, allowing you to go back and change selections until you are completely satisfied. It also will not allow you to overspend your budget.

Once you have organized your expedition the computer asks your last name and enters you as one of the climbers. It then totals things up, gives you a list of climbers and Sherpas and moves you to the base camp, a permanent installation at the base of the mountain.

Up until this point I enjoyed the program. The careful choices of climbers and supplies to face the challenge of the mountain was a good intellectual exercise, especially as I am not a mountain climber and have no real experience in organizing expeditions of this kind.

At the base camp, however, things began to go wrong.

The first challenge of the expedition, according to the instructions, is to find a route through the Khumbu Glacier to the site of camp two. To do this, you want to send out all available climbers and Sherpas. First, check the weather forecast, which is divided into three sections: weather conditions on the lower mountain, conditions on the upper mountain and the day's weather prediction. The instructions warn that the predictions are not 100 percent accurate.

The way you send climbers out is to select the fourth item on the menu—climber assignments—and call up each climber individually by a letter or (for the Sherpas) number.

Calling a climber up gives you a name, location and present condition. Pushing two letter keys assigns the person to work on the route to camp two. Enter returns you to the point where you can call another climber. I had 45 climbers and Sherpas. The process lost some of its excitement by the time I was finished.

No Climbing?

I thought that when this chore was completed the fun would begin. I pushed the key to indicate all climbers were assigned. The screen went blank for a minute, then it displayed "evening reports" and told me the path through the glacier was 25 percent complete.

"Wait a minute," I thought. "Don't I get to do any climbing?"

Five turns later I had established my route to camp two. I had gone just about crazy pushing command buttons by then. I had also discovered a data problem.

The program's menu gives four choices: Weather, camp status, climber status, climber assignment. I quickly learned that when doing the repetitive climber assignments I was wise to double check them by going through the climber status list afterwards. This gives a list of climbers by name showing only their condition (from excellent to critical) and climbing assignment. I had decided that I would not use climbers with only a good health status, the minimum acceptable status for exploration.

"At the base camp, things began to go wrong."

However, the problem was that when I found one or two climbers who were in very good condition on their assignment sheets they were only in good condition on the status list.

This was minor compared to the main problem, however. Once I established camp two, I began to move all my supplies up to it (no climbing supplies are needed at the base camp according to the instructions). At the same time I sent a small group of climbers up to camp three to establish it. Very soon I found myself moving supplies and climbers among three camps.

The climbing instructions became much more cumbersome. I had to assign each climber a destination, and decide what he should carry to that destination. I also had to make sure each camp had a sufficient number of tents, food and fuel to supply the expected climbers. This was complicated by the lack of a unified listing which would show how many climbers would be in camp the next night, what supplies would be waiting for them and what they would bring with them. In order to make up such a list so I could be sure of what I was doing, I would have had to go through three of the four data banks listed in menu and take written notes.

At this point I had already spent more than two hours playing this particular game. The instructions stated that if you want to save a game you can put it on a formatted disk. Unfortunately, something went wrong and the data was lost. It was just as well—I was getting a headache anyway.

That was the third and last attempt I made on Everest and the furthest I got. On the basis of my experience I would estimate it would take six or more hours to play a complete game.

If you enjoy dealing with the intricacies of logistics you will love Everest Explorer.

For those who have adventure in your heart—spend your money on a trip to a local hill and climb it. You'll be much closer to the Everest experience than with this program.■

Put Your Fingers On The Pulse Of Wall Street With Your Radio Shack® Computer

If you invest in the market, operate your own business or require updated information quickly to manage your affairs, Dow Jones Information Services provides the authoritative business and financial information you need. . .from Wall Street, Washington and around the world. Now Radio Shack owners can receive this valuable service through TRS-80® Videotex® software.

Get Information For Business And Investment Control. . .On Command.

Discover how quickly and easily your personal computer turns into a specialized, sophisticated library of information with access to the Dow Jones database. It's considered the world's most reliable and comprehensive source of business and financial data providing you with news up-to-the-minute as well as from the previous 90 days.

Now you can review thousands of items from The Wall Street Journal, Barron's and the worldwide Dow Jones News Service without having to maintain piles of newspaper clippings. Information about companies, stocks, labor, government, international market and monetary developments, and so much more, is yours on command.

Quotes From Four Exchanges

Now, in just seconds, you can obtain current prices and the volume traded for stock quotes (15 minutes delayed) on over 6,000 companies. . .plus 50 key financial indicators on 3,200 selected firms and 180 industries. You'll also have access to trading information on mutual funds, treasury notes and bonds.

Imagine being able to spot trends, research a company, obtain data that could affect your business, your investments, your investment decision-making.

But you don't have to imagine anything. . .Dow Jones Information Services provides it all right now.

Low Start-Up Cost. . .Free Password.

Stop by your nearest Radio Shack store and find out how easy and inexpensive it is to get Dow Jones investment data through TRS-80 Videotex. Videotex software for Radio Shack computers is only $29.95. What's more, you get a Dow Jones password absolutely free. . .along with one hour of free introductory time. And there are no start-up or minimum usage charges. In addition, software and usage costs will be tax deductible in many cases (consult your tax advisor for further information).

Customer Support. . . Just A Phone Call Away.

Dow Jones' professional customer service staff is ready to provide complete information about this important home information service now. . . and full customer support after you've made your purchase. Just call toll free. . .

800-257-5114

In New Jersey call: (609) 452-1511.

DOW JONES INFORMATION SERVICES

Providing Information Through Innovation

Dow Jones & Company, Inc. 425

Zork
Personal Software, Inc.
Sunnyvale, CA
$39.95 disk
32K Model I, Level II

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

We play adventure games and become willingly enthralled by the excitement, mystery and magic of their story. We become the hero and slay representatives of evil. We collect our just rewards: treasure and gold, perhaps a pretty princess. But most of all, we gain the lore of the land, and the knowledge that we have conquered it and can walk its paths without fear. We have won and now we look for new, unexplored adventures.

Zork is complicated and sophisticated. It is an underground kingdom, but its treasures lie above and below ground. One of its greatest treasures is its ability to react to fairly complex commands. You can do very specific things in Zork; you are not limited to a vague, general activity because the program recognizes only two-word commands. When you decide to attack a nastie, or dig a hole, you should have a definite plan of attack and weapon (which should be part of your command)—or you may find yourself fighting the forces of evil with your bare hands, or worse.

Zork is a joy to play. There are no graphics; the screen carries a description of your surroundings and the directions of travel that are available to you. Your number of turns, and points acquired, are also on the screen at all times. It is up to you to decide what to do, and how to do it. Climb trees and look around; check under rocks and piles of leaves; read walls and books. Explore, but keep in mind that directions change; just because you went north to get to the chasm doesn't mean you can return simply by going south. And, please, watch out for the pickpocket.

Reincarnation

Zork contains many areas, and consequently should be mapped. Because of its complexity, mapping will take a lot of time, and thoroughly exploring each room and site will take longer. A save procedure bookmarks your place in a game so you may avoid starting from the beginning each time. This can also be used by timid game players (or reviewers on deadline) before attempting a potentially dangerous move (you cavalier and stout-hearted daredevil) and manage to get yourself killed. You have the option of being reincarnated by various games-masters. You do lose points, however, for having proven yourself not very discriminating in your character's choice of activities.

A 15-page instruction book accompanies the disk. The book gives background, directions and explanations of the game, and also describes in great detail how to load and save the game on one and two-disk systems. The instructions are incredibly complete, they even include directions for initializing the TRSDOS disk you will need. Take it from a rank amateur; these instructions are clear and easy to follow.

Staggering Possibilities

Mysteries abound in Zork. I'm not sure it can be solved—there is so much in it and the possibilities seem staggering. Since the program lets you do pretty much what you want to, even if the consequences are much less than desirable, it leaves open marvelous opportunities for those of us who love to break into things to see how, and if, they work. I've spent many hours pondering why the earth shakes when you smash the mirror. I also have one very smashed jeweled mechanical bird for sale to the highest bidder.

Walk carefully along the paths, and enter no dark places without a light; there is always a Grue present in dingy spots which would love to eat you. The stars are high and the canyon is deep. Somewhere is a key to unlock the grate under the pile of leaves. Do not lose your mind in the loud room—the answer is there if you can only hear it.

Above all, be ready to ask questions and to giggle. The program has answers, and also a sense of humor.

Perhaps you can play against someone across the country on one of the modem systems—it is being done.

There is much to explore in Zork, and a lot to play with once you're there. The booklet describes Zork as Part 1; I truly hope that means we can expect a second part sometime soon. ■

Super Nova
Big Five Software Co.
Van Nuys, CA
$15

by Carl A. Kollar

Super Nova is a machine language program with some of the fastest graphics I've seen so far. Asteroids glide across the screen in all directions, and aliens shoot at your spaceship as they go by. When you hit a meteor it splits into fragments, and you've got to watch out for those too. When you get hit by a meteor or alien fire, the screen flashes, and all that's left of your ship is debris.

The game can be played with one or two players. If you're playing by yourself, you're given three ships; when they are destroyed, the game ends. If you're playing with two players, each is given a turn until his ship is destroyed. The play alternates between the two players until each has had three ships destroyed. The winner is then declared and his score is posted if it is among the top 10 scores thus far.

Meteor Field

An interesting feature of Super Nova is that when it starts, your screen simulates movement through a star field and then the title pages appears. While introducing the program, it asks whether you want instructions or would like to start the game. If you don't respond within about 45 seconds, it assumes you can't make up your mind and gives you the instructions anyway. At the end of the instructions it asks you to press a key to start the game. If you don't do this within about 45 seconds, it starts without you and plays by itself destroying asteroids and moving about as if someone was controlling it. Sometimes the score is pretty good too!

The game is thoroughly enjoyable and worth every dollar. Action is fast and the game is challenging. Especially when you get past a score of 10,000 and that nasty flagship appears and blows you to smithereens unless you're careful.

My only criticism has to do with the graphics for your ship. It is hard to tell which direction your ship is facing. The only way I've been able to tell which was the front was to fire a missile and see which end it comes from.

It seems to me that with all the work that went into this program, a little more effort could have been made to make the direction of the ship more recognizable. After playing this fascinating game and becoming familiar with it, I'd still spend the money if I had it to do over again. ■

ALPHA BYTE DOES IT AGAIN...

Call us for our never undersold price.

ALPHA BYTE STORES

(213) 883-8594

5115 Douglas Fir Road • Suite B • Calabasas, CA 91302

Invaders +
Larry Ashmun
Level IV Products
Livonia, MI
$19.95 cassette
$24.95 disk

Invasion From Outer Space
Chris Freund
The Software Exchange
Milford, NH
$14.95 cassette

Galaxy Invasion
Bill Hogue
Jeff Konyu
Big Five Software Co.
Van Nuys, CA
$14.95 cassette
$17.95 disk

by John T. Phillipp

Only someone who has been living on the dark side of the moon for the past two years could be unaware of the Invaders video arcade game. This mind-numbing experience consists of rows of aliens marching back and forth across the screen hurling missiles down at a defending base which is protected only by a few gas clouds and the fast reflexes of the player. The goal is to destroy all the aliens by shooting missiles from the base, as they come closer and closer and move faster and faster, accompanied by an increasing tempo of the sound effects.

There has been a proliferation of Invader-type games for the 80. This is a review of three of them.

Invaders + is a 2.9K machine language program which presents the player with 49 aliens of two types, arranged in seven rows of seven aliens each. An additional alien zips continuously across the top of the screen.

The animation of the aliens is good, as they seem to take little steps in their march across the screen, dropping missiles on the defending base. Five gas clouds protect the bases, at least until they are eroded away by the alien attack, or from below by the missiles fired the base. The base can be moved from side to side very quickly. Missiles are fired at the aliens by pressing the space bar, but cannot be fired while the base is moving. (Holding down the space bar causes repeated firing.)

Missiles from the base can destroy the aliens or the descending alien missiles. Sound effects are very good, quite similar to the arcade game. The score is incremented for each alien destroyed, each type has a different value. The number of shots taken, number of misses (there is a penalty for missed shots), and the previous high score are displayed after each game. The game continues until four bases have been destroyed but each time all the aliens are eliminated, the player starts with four new bases.

I have a few minor complaints. The base moves right and left so quickly that it is hard to control. It is not possible to fire at the aliens while the base is moving. There is no ongoing display of how many bases remain. And at the end of each game, the player has to sit through a rendition of the Star Wars theme before he can play again.

Invasion From Outer Space

Invasion From Outer Space is a 6K machine language program which presents the player with 105 aliens of only one type, arranged in seven rows of 15 aliens each. There is no animation of the individual aliens, they just move back and forth across the screen, dropping CHR$(191)s on the defending base, which is protected by three gas clouds.

The alien missiles have no effect on the gas; clouds, but they can be eroded by missiles from the base. The base can be moved rapidly from side to side under good control, firing CHR$(191) at the aliens by pressing the space bar while it moves. There is no auto-repeat firing; the space bar must be pressed for each shot.

Defending missiles cannot destroy descending alien missiles. Sound effects are appropriate to the action. The score is incremented for each alien destroyed, and there is no penalty for missed shots. The game continues until five bases have been destroyed. When a new screen of aliens appears, it is without new bases. An added addition is a mystery saucer which flies across the top of the screen. Hitting that the mystery saucer results in an extra base.

My complaints are minor ones. The previous high score is not displayed. The CHR$(191) used for missiles by both the aliens and the bases seem too big. And sometimes, when you press the space bar it doesn't fire a missile; presumably the CPU is updating the display, and not monitoring the keyboard.

Galaxy Invasion

Galaxy Invasion is an 9.2K machine language program which presents the player with 40 aliens of four different types, arranged in four rows of 10 aliens each. An additional six aliens in two groups of three provide an upper fifth row. The animation of the aliens is *superb*. Not only do they flap their wings as they march across the screen, but they attack the base by swooping down in singles and groups. They also drop missiles, represented by single graphics points, but their aim is poor.

There are no gas clouds for protection of the base. The base can be moved from side to side by pressing the space bar. Holding down the space bar causes repeated firing.

Aliens can be destroyed while on the march, or while flying. The sound effects are not like the arcade game at all. The score is incremented for each alien destroyed, each type has a different value which doubles if they are attacking!

The ten previous high scores with the scorer's initials are displayed. The game continues until three bases have been destroyed, and an additional base is awarded for each 10,000 points scored. There is provision for two players to compete alternately. For the single player, the game goes into a demonstration mode with the computer controlling the aliens and the base.

There are several complaints. The CPU is so busy with the display that there is often a delay between the space bar and the firing of missiles, which adds the frustrating element of luck to what should be a game of skill. The sound effects are the poorest of the three. The game is silent unless a missile is fired or an alien is hit. And before each game, the title is displayed and the number of players must be input, another unnecessary delay.

The Verdict

Invaders + is the closest to the arcade game in its animation and sound effects. It is fast-paced and challenging. The addition of levels of difficulty makes it interesting to beginners and experienced players.

Invaders From Outer Space is simpler but also fast-paced, and keeps the spirit of the arcade game.

Galaxy Invaders is the most sophisticated of the three. Its imaginative use of graphics provides a demonstration of the kind of animation the TRS-80 is capable of producing. Still, its poor sound effects and relatively slow pace make it less satisfying than the others. ■

Swamp Wars
Instant Software, Inc.
Peterborough, NH
Level II, 16K
$14.95 cassette

by Eric Maloney
***Kilobaud Microcomputing* staff**

Swamp War reenacts a hostile encounter between a tramp spacer called The Stellar Spaniel and the indigenous population of an unnamed planet in the Bragthos sector.

The local aborigines—known to the less informed as Slizards and Muck Monsters—are lazing around the lean-tos one afternoon when a fellow who is out washing his car happens to see the Spaniel land on one of the planet's nine islands. He naturally calls in a report to Mutual Aid, and pretty soon just about everybody has heard the news on their scanners.

As it turns out, the Spaniel is only stopping for repairs. But the townies have had previous difficulties with bounty hunters and mercenaries, and know that the rest of the solar system has about as much use for them as a bucket of cold farts on a rainy day. So they mobilize their SWAT squad to go out and deal with the situation.

But life ain't no chair of bowlies. The ship's captain, with the improbable name of Legion J. Muldoon, is a crafty SOB, and soon has four 'droids combing the islands for transporters left behind by a group of missionaries. He hopes to use the transporters to repair his ship, and book.

The Slizard SWAT team isn't too bright. In fact, there's a warrant on the agenda for the next town meeting to raise the entrance standards. They come charging up on the island hell-bent for bongoburgers, and Muldoon cleans the beach with their faces. He even picks off a few Muck Monsters, who shoot faster and with greater accuracy, before sending his 'droids to another island.

Naturally, the locals are upset. This sort of thing costs a bundle in ritual bereavement ceremonies. So they send out another squad, with instructions to stay low and kick ass.

These guys do a little better, backing one of the 'droids into a public restroom and turning him into swamp gas. But the 'driods bob and weave, transport from one part of the island to another, and generally make themselves difficult targets. The result is the same, and the Honor Guard deep-sixes 18 more shoe boxes.

These first two confrontations set a pattern that continues from island to island. The Slizards are beginning to feel like they're riding mopeds through the Lincoln Tunnel. The Muck Monsters start to pack their station wagons and make hotel reservations in Miami Beach.

But somewhere around island five, the action picks up. The Slizards kick out the jams and get serious, wasting two more 'droids. Muldoon is tired of pushing all those buttons, and the remaining 'driod is complaining about heartburn. He is about ready to toss a mud ball at the mother ship when a Muck Monster sneaks up from behind and grinds him into silicon dust.

That's it, of course, for Muldoon, who drinks six bottles of Scope and dies of minty breath.

The Slizards and Muck Monsters dismantle the ship and use the parts to repair their tractors. Muldoon's boots and the six empty Scope bottles are sent to the Museum of Modern History, where they are put in the basement and forgotten. The story makes the wire services, but the public loses interest when the Pope arrives for his first tour. ■

Haunted House
Tandy/Radio Shack
Fort Worth, TX
$9.95 cassette

by David Williamson

The second adventure program marketed by Radio Shack is called Haunted House, (the first is Pyramid 2000) and was developed for them by Device Oriented Games. It is short enough for a beginning adventure player, but needs the expertise of a professional; it is therefore not suitable for either.

The instructions are very brief—only one-third of a page with no real help provided to beginners.

The vocabulary is also limited, consisting of: N, S, E, W (for the directions North, South, East and West), Get, Drop, Read, Attack, Smash, Pour, Hit, Open, Climb, Jump, Look, Inventory and Kill (only N, S, E, W, Get and Drop are covered in the instructions). After using the enormous vocabulary of Hi-Res Adventures and the Scott Adams adventures, I was frustrated not only by the small vocabulary, but also by the omission of standard commands such as Go, Use and Throw.

On a positive note, the game uses a sense of humor in solutions and responses. For example, if you try to Get Cabinet, you will get the response: "Don't be ridiculous!" instead of the customary "You can't do that."

The object of the game is to get out of the Haunted House alive. The game is in two parts, in Part I you are on the first floor, dodging levitating knives, looking for a secret passage, and getting past animated suits of armour and fires. The second floor program is loaded when you get there. This is where I got stuck for a while.

On the second floor, you find a magic sword and three ghosts, one to the west, one to the south and one to the east. After destroying these three, you find a ghost to the west of the western ghost. You quickly discover he is immune to attack. The problem was I couldn't seem to get out of this particular room.

After trying almost everything, I felt desperate enough to call Texas, but I accicently discovered that by moving around the room in certain directions would trigger one of the entrances to open. It is debatable if others will be as lucky as I was; I think the assistance of an experienced adventure player would help.

Playing straight through, without any wasted commands, the program only takes six minutes and that includes loading time. It is short enough to suit the beginner who will go on to bigger and better things.

In conclusion, with the only plus being humor, and the minuses being limited instruction, vocabulary and game (most because of limited memory), I advise you to save your money for 16K or larger adventures. ■

Radio Shack Made the TRS-80® Color Computer Even Better!

At $399*, it's no wonder the 4K TRS-80 Color Computer is ideal for beginners. Now there's a TRS-80 Color Computer designed for more advanced applications, too.

TRS-80 Extended Color BASIC Computer. This high-performance computer includes a 16K ROM Extended BASIC with advanced graphics, eight brilliant colors, and sound for an unprecedented low price! You can draw fine lines, circles, rectangles, boxes and more with *easy-to-use one-line commands.* Four graphic modes with two color sets allow up to 49,152 programmable screen points (pixels). There's 225 separate tones for music or sound effects, too. All this on a 16K RAM machine (including video memory) loaded with the dynamic features a serious programmer demands. You get a 32-cpl x 16-line screen, multi-character variable names (two significant), editing, tracing, user-definable keys, 255-character string arrays, floating point 9-digit accuracy, and even machine language routines.

Priced at Only $599, the TRS-80 Extended Color BASIC Computer is useful, entertaining and educational. Yet using it can be as simple as plugging in one of Radio Shack's instant-loading Program Paks. The computer attaches to your TV, or our own $399 TRS-80 Color Video Receiver. For just $24.95, you can add a pair of joysticks which add flexibility to games and video displays. A built-in serial interface lets you attach a printer or a modem. A pair of tutorial, Extended Color BASIC instruction manuals are included, as well.

More Good News. Extended Color BASIC is also available as an upgrade kit ($99) for your 4K Color Computer (16K RAM required — $119). There's a modest installation charge for each kit.

New TRS-80 VIDEOTEX Software (with the modem shown below) offers quick, affordable access to various information and data services. The CompuServe® Information Service gets you local, national and international news, weather and sports from major newspapers, like The New York Times and The Washington Post, plus the Associated Press News Service; info on stocks and bonds; educational reference service; nationwide Electronic Mail and much more! The Dow Jones® Information Services provides stock exchange quotes — as recent as 15 minutes — plus feature selections from The Wall Street Journal and Barron's.

Only $29.95 Buys You VIDEOTEX Software including a free hour on both CompuServe and Dow Jones. Come see the new TRS-80 Color Computer, its programs and accessories, at your nearby Radio Shack today!

Radio Shack®

The biggest name in little computers™

A DIVISION OF TANDY CORPORATION

6100 STORES AND DEALERS, 140 COMPUTER CENTERS AND 135 SERVICE CENTERS NATIONWIDE

NEW

A Low-Cost, Direct-Connect Modem. A convenient alternative to an acoustic coupler. The TRS-80 Modem I lets you enter the world of microcomputer communication for only $149. Cable extra.

Specifications. Low-Power CMOS Circuitry: Full duplex 300 baud, 103 compatible ANS/ORIG. Sensitivity: −48 dB/−43 dB. Connectors: DB25/4-pin DIN; includes interface for Model I cassette port. FCC approved.

*Retail prices may vary at individual stores and dealers.
® CompuServe is a trademark of CompuServe, Inc.
® Dow Jones is a trademark of Dow Jones & Co., Inc.

Two Heads of the Coin
Adventure International
Longwood, FL
$19.95
32K Disk Basic, TRSDOS 2.2 or 2.3

by Dan Rollins

This program is a type of sophisticated game in which the player assumes the role of a character in a narrative novel. At various points in the dialogue you are prompted to ask questions or take actions. The input line may be as long as needed—an improvement over the two-word (verb/noun) input of the adventure series. The idea behind interactive fiction is intriguing and its author, Mr. LeFore, has put a lot of effort into Two Heads of the Coin. However, though I was initially very pleased with the program, I was ultimately disappointed.

The program requires 32K and at least one disk drive. You must boot up with TRSDOS 2.2 or 2.3. The program is initialized from DOS by entering IACHAIN which sequences through the Basic/Files/Memory Size/Run program prompts without operator intervention. You are prompted for your full name, gender and your favorite recreational relaxant. The opening narrative begins by setting the stage for the story.

Drama

Two Heads of the Coin is a Sherlock Holmes/Dr. Watson style drama set in Nineteenth-century London. The narrative is full of quaint British idioms. This added to the realism of the story and characters and inspired an occasional chuckle. The wife of a Mr. Conway has disappeared and you, playing the Holmes role, are expected to ask questions and gather enough information to finally solve the mystery. At times the Watson character (called Dr. Grimsby) will interject questions and comments of this own, especially when you ask a series of questions to which the program can find no response.

The program is written in Basic with a large random access file of responses on the disk. Each input string is searched for key words to which the program responds with one of several "I don't believe I follow you" comments, or the disk starts spinning and the reply is printed.

On the plus side, the responses were entertaining and intelligent. For example if your input line contains Her, She, Their or Them, the program will assume that the pronoun refers to the most recent character(s) you've mentioned. The lag time before the questions are answered is minimal—a credit to the programmer—as

a large number of key words must be checked. Also you can't get away with just typing some obvious key words. The program expects complete sentences and the game is much more enjoyable when you respond "in character."

Unfortunately, this program just did not live up to its advertised claims: First, "The dialogue of the characters, and even the plot, will depend on what you say." The other characters DO respond to your questions, but the plot—Holmes asking questions, Conway answering—cannot be altered. You can't leave your Baker Street flat. No action takes place aside from your asking of questions and Conway's entry and exit.

Second, "Most challenging of all, this program will tax your observational skills and above all your imagination." Being an adventure lover at heart, I expected Two Heads to provide days, even weeks of entertainment. Unfortunately, this game is so simple that it doesn't even include a Save Game option. Within less than an hour of questioning, I had solved the puzzle. In fact, I knew the answer before that, but considered it too obvious to ask! I am not attempting to display my intelligence here. Indeed, most adventure veterans will solve the puzzle in a fraction of that time. Even the complete novice will likely be disappointed.

I have even gone back and played the game again, looking for alternative ways to play it. I was still unsatisifed. The ending question to the game (the solution to the mysterious disappearance) can be asked at the very first input.

If you are looking for a challenge then I cannot recommend that you spend your money on this program. Buy, instead, two of the fine Adventure series (Strange Odessy is excellent!) or go to a movie! ■

TRS-Opera
Word Challenge
Richard Taylor
Acorn Software
Washington, DC
16K Level II, Models I & III
$9.95 each

by Darren DeVigili

Who among us would pass up an opportunity to show off the much maligned and underrated sound capabilities of the TRS-80? O.K.—so it only has one voice, so do I. What that single voice can do may surprise those who have not been lucky enough to have heard the monotone magic of TRS-Opera.

The program includes five songs, all classics. While my entire operatic knowledge will fit neatly between pages 51 and 52 of any magazine, three of the songs are quite familiar, and the other two passingly so.

As far as I can tell, only one of the songs is full length. This is most likely due to the memory restrictions involved in presenting a fair selection while keeping the program within the 16K margin. The song represented, in its entirety, is familiar and enjoyed by all: The William Tell Overture.

By using very fast alternating tones, amazingly full single voice sounds can be, and have been produced. What appears to be two notes is really one note quickly followed by another. This must be heard to be appreciated, as it accurately simulates multiple notes (chords). The tone quality is good, as far as tone goes for a TRS-80.

If you have any interest in computer music, this is for you. In my opinion, TRS-Opera is one of the most ambitious musical programs written for the '80.

Word Challenge

Word Challenge is an excellent example of entertainment using the well-integrated use of sound. The game is a cross between the conventional game of hangman and the television game show, Wheel of Fortune. I have seen many programs that use sound and graphic combinations to capture the interest of the user, and this one hangs with the best.

The game can be played by one or two players. In the one-player mode, the computer supplies the phrases to be guessed from its store of about 25. These phrases are good, but few. In the two-player mode, you have the option of creating your own phrases, but you can't save them to tape or disk. This is unfortunate because you could exchange data with a friend, or use the program for educational purposes.

There are minor problems in two delay loops; the screen clears before you have a chance to read it properly. These are quite minor and can be fixed, or safely ignored. Regardless, the program is a good one. The quality of the sound will amaze you.

Fidelity Bonus

Those of you who use a speaker or headset instead of the typical tiny amplifier get a fidelity bonus, as the sound quality in this program is just as good as that in TRS-Opera. Many of the sounds are akin to those sed in most arcade games, and are done neticulously.

When play starts, the phrase or word is musically spelled out as dashes (HI THERE/-- -----). You have a possible 50 points for each phrase. A wrong guess at a vowel d you lose 10 points and get a wisecrac any other incorrect guesses cost you two points. The program is designed to let you try four phrases in the one player mode, and up to four in the two-player mode.

Word Challenge is a good program, particularly for those of you with school-age children.

Galactic Empire
Douglas Carlston
Broderbund Software
Eugene, OR
$14.95 cassette
$19.95 disk

by Dan Cataldo

One of the best game programs currently available for the TRS-80 is Galactic Empire, written by Douglas Carlston of Eugene, OR. It is a 16K, Level II strategy game, which places you as the commander of the planet Galactica's forces. Your mission is to conquer the 20 worlds of the Central Galactic System within an allotted time period of 1,000 years. Your forces are fighters, transports, and scouts, and your method is—brute force.

The biggest obstacles facing you in your mission are distance and time. The Central Galactic System is approximately 60 light-years across and about half that in depth. The dimensions, and the locations of the various systems, vary randomly from game to game. It takes your fleet one year of game time to cover one light-year of distance. There is no faster-than-light travel in this vision of the future. However, travelling at high speed slows down your aging processes. Hibernation stops it completely. Thus, your normal life-span is extended to 1,000 years.

To succeed in your task, you must make skilled use of your fleet's resources and its officers. As the game begins, your fleet consists of 100 fighters, five scouts, and 100 unloaded transports. Each transport can hold 100,000 troops. You also have 1,000 megacredits in your treasury, which you can spend to construct new ships.

Your four officers are: Lt. Bayliss, in charge of taxation, ship construction, and enlistments; Lt. Starbuck, in charge of the scoutships; Navigator Kirman; and Dr. Henderson, in charge of hibernation. And, of course, there is Computer Central, which contains planetary directories, maps of the galaxy, and status reports.

This game has a very well designed display. A small view screen in the upper left corner of your monitor shows the planet which your fleet is currently orbiting, and its name. If your fleet is in transit between systems, this view screen will show a moving field of stars. When the new system is reached, a planet will flash into view and you will again be in orbit.

A CRT display is in the lower left corner of your screen. The CRT is used to relay communications from Computer Central and to list your officers and their duties. When an attack is underway, the CRT will show the number of fighters and transports lost on either side, and your probability of success at any given moment.

Stardate

In the upper right corner of the screen is a calendar showing the current stardate. Below it is a resource chart, showing the number of fighters, transports (loaded and unloaded), scouts and credits available. At the bottom right is a summary of the four main command options: A-Attack, E-Embark, C-Computer and O-Orders. There are also two other commands: R-Retreat, and Q-Quit.

The best way to see how these commands are utilized is to follow through a short sample of the program. As the game begins, you are orbiting Galactica. You press C to call up Computer Central. A display of Computer options appears on the CRT: A-Star Maps, B-Planetary Directory, C-Status. You press A and another menu of options appears: local map, range finder or galaxy map.

You call up the local map, and the CRT displays the systems close to Galactica. To determine the names of the systems you have two choices: First, you can move the blinking cursor at the top of the map with the arrow keys. When the cursor is superimposed on one of the systems, its name and coordinates will appear. Second, you can simply key in the irst letter of a system's name. If that system is on the map, it will blink repeatedly, and its name will appear as before.

Once you've identified the nearby system u can use the range finder to detern heir exact distance from Galac stems that look close together map may actually be far apart. e maps are, in effect, looking d he galaxy, the element of depth

Aft ve located four or five systen are close by, choose one as the dest on of the fleet. The others can be investigated by your scout ships. Call up your officers and issue appropriate commands to Lt. Starbuck. He will ask where to send the scouts, and to what system they should return. The scouts in this case would be told to return to the fleet's destination.

Lost In Space

The timing of these missions, and the arrival of your construction orders, is very critical. If a snip arrives at a target system and is not met by the fleet within five years, it will run out of fuel and be lost in space. If the fleet is in the system but you hibernate past the five-year period, the ships again will be lost.

Having dispatched your scouts, call upon Lt. Bayliss to tax Galactica. With the additional funds, call on Bayliss to place construction orders for more ships. The first year's orders are delivered immediately. The following years' orders must be directed to your expected destination. The number and type you build will depend on your needs, and on your experience in playing the game. Finally, Bayliss (who by this time is approaching exhaustion) is called upon to enlist troops to fill your empty transports.

As you race through space, you will see the years pass on the calendar, just as the stars march slowly by on the view screen. When you reach your destination the CRT will announce the arrival of any scout ships and construction orders that are waiting for you.

Next, you can then call up the Planetary Directory in the computer to determine the population and technological level of the system. The worlds of the Central Galactic System are divided into four technological classes: primitive, limited atomic, sophisticated (the rating of Galactica) and superior. The populations of the planets can vary from less than 100 million to more than a billion. The characteristics of the various systems will change from game to game. If the first system you visit has a greater population than Galactica or is sophisticated, it is good strategy not to attack. You should wait for your returning scouts and construction orders. You can then use the Planetary Directory to examine the systems visited by the scouts for an easier target. By using the Status Report option of Computer Central, you can determine if all your scouts and construction orders have arrived before moving on to the next system.

Conquering the Galaxy

And so the game continues: You move from system to system, conquering when you can. All your actions have to be carefully considered, if you expect to conquer the galaxy in a mere thousand years!

Although Galactic Empire is well designed and fun to play, there are some problems. This is a very long game; one session can easily last from four to six hours. You not only have to have the organizational skills of a conqueror, but also the patience of a saint.

Older versions of Galactic Empire (which are apparently still being sold by some software houses) have two major problems. The first is that you can't save a game in progress, an absolute must for a game of this length. The second problem concerns placing construction orders. In the older version, if you place orders for more than 50 years, the earlier years' orders will disappear, or you may get a BS error while the program is running.

Mr. Carlston has informed me that newer versions of Galactic Empire will indicate when you have reached your order limit. The 'save game' option has also been included, and travel between systems goes two and a half times faster than before. Those of you who have purchased the earlier version of Galactic Empire may send a tape to Broderbund Software for a free correction. It's nice to see a company that cares about its reputation and its customers.

One final problem concerns cheating—that is, you can make the program do things it isn't supposed to do. By skillful manuvering, you can have a planet which is supposed to be capable of producing only five ships a year produce 10 or even 15 ships a year. This is done by placing orders at system A, racing to nearby system B, then back again to A to place more orders. The additional orders should be built only after the first set is completed, but in fact that isn't the case. The additional orders can and do overlap—they are produced concurrently, not consecutively. Don't do this if you want to play the game the way it was meant to be played. This problem exists on all versions of Galactic Empire.

The good points far outweigh the bad: Galactic Empire is never dull. The continuous changes in layout and planetary characteristics present you with different problems each game. The attacks can be full of surprises: You can lose a high probability battle if you aren't watching carefully. You can also pull off a victory from what seems like a hopeless position if you have the guts not to retreat.

You can easily change the names of the planets in the Central Galactic System by editing one of the data lines, if you want the thrill of playing in one of your favorite sci-fi worlds.

I heartily recommend Galactic Empire to anyone seeking a complex, intelligent, and, of course, fun program.■

Winged Samurai
Discovery Games
St. Paul, MN
$19.95

by Art Little

In Winged Samurai, Discovery Games has assembled a program with some intriguing ideas.

The Situation: Spring, 1942. The Allied bombers still 28 miles from their target, are spotted by 16 Imperial A6M2 fighters (the dreaded ZERO).

You are a Japanese Squadron Commander defending a New Guinea naval base. In this game, you can choose one of 13 different fighters, each with its own flight characteristics, for your mission. You will be facing 15 types of Allied planes.

Packaging: The program comes in a dustproof box, illustrated with a color drawing of an air battle above the Bismarck Archipelago. The cassette tape within loads on a TRS-80 Model I. On the same cassette are copies of the program intended for the PET and Apple II, all written in machine code.

Documentation: The instructions are in three parts. First come the loading instructions which, though detailed, are slightly confusing to the novice user.

Second, is a sheet listing all of the possible commands and their meanings including an explicit description of the action commands with suggestions for their use.

Third is a 12-page booklet that gives the history of the battle of Rabaul and provides details on all the aircraft used in the Pacific Theater. It includes information on speed, maneuverability, firepower, sturdiness and rate of climb—all of which are programmed into the game.

The Game: After the title pages load, you are welcomed to the game and asked your name. You are shown a side view of the 30-mile air corridor leading to Rabaul (X-Coordinate) and the altitude in feet (Y-Coordinate). The positions of all attacking and defending aircraft are displayed. Below this are listed the number of bombers, escorts and fighters; their altitude and range.

The action proceeds as you type in commands to your squadron. There are 25 two-character commands. After you enter a command, the sideview is updated with information and you make another command decision.

Your goal is simple enough. Destroy all of the bombers before they reach your base. What makes the game interesting is that the bombers have no intention of cooperating. They proceed at full speed toward the target, raining cannon fire. If accompanied by an escort, the situation is even more difficult. You must lure the escorts away, flank them, then return and inflict damage to the bombers before the fighters return.

It is easy to learn... not easy to play well because with all the possible aircraft combinations, each game is very different. In Winged Samurai, as in real aerial combat, chance plays a major role. This is simulated by including a random element into all encounters. Thus, for no apparent reason, your fortunes may turn sour rapidly.

But the code of the Samurai demands that you renew your attack, for it is both an honor and a responsibility to be called Toryu—Killer of Dragons!■

STOCK MARKET SOFTWARE

By H & H Trading Company for Mod. I, II, & III TRS-80®s

STOCK TRACKER™ times trades on individual stocks and options. Uses a technical volume analysis. Completely objective. Gives buy, sell signals.

MARKET TRACKER™ times tops & bottoms on the Dow Industrials; a composite of six technical indicators. Some judgement required. Gives buy, sell signals.

These two programs are accurate.
Hundreds of satisfied users.

ASK FOR THE EVIDENCE!

FOR MORE INFORMATION OR TO ORDER, CONTACT:

H & H TRADING COMPANY
POST OFFICE BOX 23546 ✓223
PLEASANT HILL, CALIFORNIA 94523
Telephone 415/937-1030

VISA & MASTERCARD

®Regd. T.M. of Radio Shack ©

Dough'flo (dō'flō) **n.** [Colloq] a highly interactive business or home finance program, designed to instantly analyze **where the hell all that money went.**

Doughflo. It's a business program. It's a home finance program. Doughflo is (now hold your breath), "an expenditure oriented data base management system." (Whew.) But best of all, it's easy to use.

What the heck does Doughflo do? Lots.

- Allows instant analysis of business or home cash flow. (Flo?)
- Keeps tabs of expenses with up to 26 user defined categories.
- Allows separation of home and business expenses for tax, Social Security, deductions, etc.
- Ideal for preparing income and state taxes.
- Is able to search, sort, and produce hard copy of any number of items and checks, over any range of dates, categories, amounts, vendors, or descriptions.

Be your own business analyst! Virtually obsoletes a bookkeeper, because your computer does the work for you. Use Doughflo to...

- Instantly retrieve up to 450 records at a time **without** time consuming disk access.
- Easily generate detailed reports of all expenditures.
- Satisfy your boss, wife, (husband?), or yourself that you know how wisely (or foolishly) money is being spent.

Exceedingly simple to use (we tested it on simple people!), now there is no excuse not to know where the hell you lost all that money, whether it was last year or last month. Best of all, maybe now you'll be able to save some of that dough for the future! Requires 32K and one or more disk drives. Specify Model I or III. At only $39.95 postpaid, it's a steal sure to save you money!

✓124

TIME ...

METHUSELAH™ has time for you and your TRS-80®

The name Methuselah has always been associated with long life. The Methuselah computer clock board with its 24 hour clock and its perpetual calendar is indeed worthy of the title. Methuselah puts state of the art technology into action with the on board four year lithium battery back-up to keep it running even when your computer is off. That means no more fooling around with the software or hardware "clocks" that become Rip Van Winkles when the computer is reset or turned off.

Methuselah has many other timely features. Software patches keep the DOS and BASIC time and clock commands ticking. Only two screws mount Methuselah inside the expansion interface without soldering, clipping, cutting, or jumpering. There is even a spare I/O port and four different interrupts available for people who love to tinker.

SPECS: MM/DD/YY, HH:MM:SS and day of the week. Four year lithium battery back-up. Crystal controlled timing (adjustable 32.768 oscillator). Twenty four hour clock and perpetual calendar.

Send check or money order to:

NDM DESIGNS ✓135
6330 W. North Ave.
Wauwatosa, WI 53213
(414) 259-0120

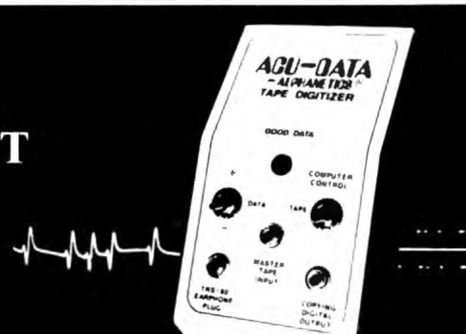
M-S1 .... $97.00
(add $4.50 P and H)
S-S1 .... $17.50
(add $2.50 P and H)
(If you wish to use Methuselah and the RS-232 board (26-1145) together, order this option which includes a new cover and connector.)
WI residents add 4% sales tax.

METHUSELAH is a trade mark of NDM Designs

® Reg. T.M. Tandy Corp.

alphanetics P.O. BOX 597 FORESTVILLE, CA 95436
24 hr. phone (707) 887-7237

"A PERFECT CLOAD EVERY TIME"*



ALPHANETICS TRS-80 TAPE DIGITIZER

* From a review in the September-October 1980 Elementary Electronics. Reprints available upon request.

At last there is a cure for TRS-80 tape cloading blues. For over three years, Alphanetics has been selling the TRS-80 Tape Digitizer, a proven hardware solution for your software problems. No longer need you juggle the recorder's volume control endlessly, trying for a perfect cload of a pre-recorded program. Just pop the tape into the cassette recorder, process the signal through our digitizer, and you're ready to RUN a perfect load!

Just check out the Tape Digitizer's features...

- Makes tape program loading virtually independent of volume control setting.
- Allows copying system and normal tapes without using computer.
- Makes a perfect digital copy of any tape, without using computer, removing hum, noise, and cures minor dropouts.
- Cassette switch allows manual control of cassette recorder, independent of computer control.
- "Good Data" indicator easily enables setting proper volume ... doubles as a tape monitor.
- A.C. powered — no batteries to replace.
- Housed in a sturdy, attractive metal case.
- Completely compatible with level I & II, also LOW speed Model III.

Feed your cassette to the Alphanetics Tape Digitizer and feed your computer the exact digital waveform the TRS-80 gave your tape. Get rid of your tape bugs today — $64.95 postpaid or return within 10 days for a full refund!

80 ACCOUNTANT

by Michael Tannenbaum C.P.A.

"I was surprised to see several vendors selling Model II software designed to assist programmers in developing custom programs."

From May 21 to May 24 the first Annual National 80 Microcomputing show was held in New York City. Although other computers were displayed, 98 percent of the show was devoted to Models, I, II, III, and the Pocket Computer.

There was a poor turnout at the show, no doubt from a lack of publicity. This was unfortunate. The concentrated number of exhibitors provided a feast for the 80 owner.

As I had expected, games and computer supplies dominated the show. However, I was surprised to see several vendors selling Model II software designed to assist programmers in developing custom programs.

Two interesting program packages were offered by Snapp Inc. and Johnson Associates Software. Snapp Inc. offered extensions to the Model II Basic interpreter. The extensions were offered in several different modules. The one I purchased, Snapp III, restored PEEK and POKE commands to Model II Basic and provided a fast built-in machine language sort. Snapp's other modules provide further built-in functions. Snapp Inc. is located at 8160 Corporate Park Drive, Cincinnati, OH 45242.

Johnson Associates offered two amazing products: a machine language ISAM file subroutine that can be integrated into any custom program and a display screen generator. These products should, when I learn how to use them, take the effort out of writing custom Basic programs.

Their demonstration of the screen generator was a real knock-out. Anyone who has written custom programs appreciates how hard it is to prepare good display screens with proper interactive edit features. The Johnson program whips up a beautiful display screen with amazingly few commands. Johnson Associates is located at P.O. Box 1402, Redding CA 96001.

Super Accounting System

One of the most interesting accounting systems at the show was offered by Taranto & Associates Inc. Taranto has refined an already good system, the Osborne System (see June and December 1980), from its original form into a super Model II record-keeping system. When I stopped at the Taranto booth, I met Irwin Taranto who gave me a brief demonstration of its features.

Irwin cautioned me the demonstration package was not the way he usually sold systems. Prospective buyers are requested to fill out a brief questionnaire and based on that, Taranto & Associates Inc. will customize a package for the purchaser.

This approach to selling accounting software makes good sense. No matter how good the software is, if there is a mismatch between the expectations of the purchaser and the capacity of the program, there is a good chance of failure. By having the potential customer document his requirements on a written form, Taranto does his customers an enormous service.

Through use of the questionnaire, Taranto can meet some of the special requirements of his customers. For example, the customer is requested to specify how many months of data will be retained and decide whether or not to suppress printing of account numbers on the financial reports.

A key question is the number of disk drives on the system. A complex integrated system might exceed the capacity of a single-drive Model II. With this knowledge Taranto can tailor his system to fit the available space. If there is inadequate room, he can warn his customer to acquire additional storage before implementing all features of the package.

Taranto complies with copyright requirements by requesting his customers to specify they have been licensed by Radio Shack to use TRSDOS. He protects himself by requesting the purchaser agree to a nonexclusive and nontransferable licensing agreement to use the software. The number of cheap Osborne systems available attest to his need for caution.

As a nice touch, systems where volume of transactions could be a limiting factor have the limitations specified in the questionnaire. For example, payroll has a maximum total capacity of 300 employees; inventory control has a capacity of 5000 items; and accounts receivable (open item), 2000 customers plus open invoices.

Since the Osborne accounting system is described in several manuals published by McGraw-Hill, most vendors of this system do not include much documentation. Taranto, on the other hand, publishes a manual with each application package designed to amplify and clarify the original text.

In the general ledger module, the file and report structures have been altered to accommodate budgeting and monthly comparative financial reports. Although the reports in the original package were quite good, the extended reports in the Taranto system are the best I have yet seen in a microcomputer system. However, they still do not include the sources and application of funds statements, but I was assured this product is on the drawing board.

The accounts receivable module has an invoicing module consisting of four screens to control data entry. Screen one allowed selection of the customer, specification of invoice type, date, shipping date, customer order number, FOB information, ship via data and G/L account code.

The invoicing module required the presence of a customer master file record before invoicing could proceed. Once the heading was specified, an invoice body screen was displayed. This screen allowed entry of up to 51 line items of invoice data. If the accounting system included an integrated inventory control system, entry of a period after a stock number initiated a search of the inventory master file. When located, the price was extracted from the file and displayed. This price could be accepted or rejected by the operator as desired.

Entry of a unit price could be either zero or a fraction of a cent. Calculation of the item amount was automatically performed by the computer. At this point, the operator could specify whether or not the item was taxable and indicate an inventory product code. The specification of taxability on an item by item basis permitted mixing taxable and nontaxable items on one invoice. The use of a product code facilitated a post-invoicing sales analysis. Since only a single sales account could be specified during the invoice heading preparation, sales of dissimilar products contained on the invoice would be combined. The post invoicing sales analysis separates sales data for accounting purposes.

To facilitate preparation of sales commission reports, a salesperson's code is assigned to each sales detail record during the invoicing procedure. This code is obtained from the customer's master file. With this data, it is possible for the system to calculate a different sales commission on each item. This is the first time I have seen this feature in a microcomputer system.

Once the body of the invoice has been completed, a summary screen is presented detailing the invoice subtotal, tax, shipping charges and final totals—both taxable and nontaxable. The operator can enter a tax rate, shipping charges or return to the detail entry section for review and correction. Termination of the invoice entry procedure is accomplished by specifying a "save this invoice" option on the final screen. Of course this screen permits a review of the entire invoice.

In general, the invoicing procedure and screens are clear and simple to follow. The only negative factor in the entire procedure is the lack of an abort facility if an improper customer number is entered and accepted. The only option available, should this type of error occur, is to enter dummy data and cancel the invoice on the final screen. This problem will be severe during the early stages of implementation and should be less troublesome as an operator becomes more skilled.

As is typical with all Osborne system data entry procedures, invoice data is entered, printed and converted to accounts receivable data in batch. The printing program is designed to accept a special preprinted invoice form but can be altered for special customer needs. The printing program accepts a special 50-character message that may be printed at the bottom of each invoice. The printing program does not kill the invoice file after printing. Therefore an invoice can be reprinted as required. To assist invoice file corrections, a transaction file maintenance program is provided.

In time the invoice file will become quite large. Purging the file is accomplished by the summary transaction program which compresses the invoice file and generates the data required for the inventory control and accounts receivable processing modules.

To ice the cake for the accounts receivable user, Taranto supplies a mailing label program which can generate up to nine labels across and nine labels per customer. Labels can be generated in customer master file order or in zip code order, whichever is required.

Accounts payable record keeping begins with preparation of a purchase order. The purchase order data entry procedure is similar to the invoicing system used for the accounts receivable module. The result is to create a purchase order which is sent to the vendor of the product and provides a clear description of the items, quantities and prices required.

After the merchandise is received, the detailed purchase order image is used to enter the vendor's invoice data into the accounts payable system. If this system is fully integrated into an inventory system, the purchase order and receipts quantities can be used to update the on-order and on-hand columns of the inventory master file. The system is integrated into the general ledger and summary accounting activity can be transferred at the end of the month.

The inventory control module appears to be the latest in the group of systems furnished. As was true for accounts payable, there is no indication as to how this module integrates into the other systems.

It does provide two main reports on inventory activity: an on-hand inventory report and a sales report. The inventory report is a "snapshot" of inventory status. It details on-order and on-hand status as well as cost, value, selling price, reorder point and reorder quantity. The sales report indicates current, year to date and prior year sales in dollars. If this is the only sales report, it does not appear to have more information than the accounts receivable reports. What is needed is a unit sales report showing a sales trend. Such a report would be quite valuable in calculating optimum reorder points and quantities.

Although the payroll system was not included in my review package, it appears Taranto has put together a complete accounting and inventory record-keeping system that can be installed on all Models I, II and III. If you are looking for a comprehensive and well-documented system, Taranto has an impressive product at a very reasonable price. Taranto can be contacted at 121 Paul Drive, San Rafael, CA 94903.

Dictionary Software

Once word processors became common, it was just a matter of time before someone would integrate a dictionary to provide an edit function. This is one of those concepts that sounds simple but is quite difficult to implement. According to my wife, the English teacher, everyone has a slightly different vocabulary; establishing a common base for editing is quite difficult. Cornucopia Software, P.O. Box 5028, Walnut Creek, CA 94596 has attempted to meet the challenge.

They offer an editor (Microproof) that can be integrated into Model I, II, or III Scripsit or used as a stand-alone editor. Containing 50,000 words, the Microproof system must use a segmented file structure to fit on Model I disks. In use on my Model II, the segmented data base was unnoticeable. The program worked flawlessly and quickly demonstrated the limitations of an editor.

Apparently the editor does not contain certain words such as bytes, Scripsit, TRS and other terms which usually dot my literary output. Also missing are common names like Osborne, Ohio and California. If these are in common usage, they can be added to the dictionary. Additions are limited by the amount of disk capacity available.

As reviewed on the Model II, the subject file had to be converted to an ASCII file. If the word processor in use is Electric Pencil, the file is already in the proper format. On the other hand, Scripsit uses a special file and contains its own internal directory. By using the merge file option, the necessary output file can be created. Unfortunately, this requires a second drive. Once the file is generated on drive one, the diskette should be placed in drive zero and reset. Selecting the Microproof program and following the information on the screen does the rest. At deadline time, an integrated version was released for the Model II. This version makes Microproof accessible from within Scripsit, eliminating the need for a second drive.

The proofing program operates quite efficiently. A 15-page document can be proofed in about two minutes. At the conclusion of the proofing process, a list of unreferenced words are displayed or printed on the line printer. By using the global search facilities of the word processor, corrections can be made quite quickly.

Accompanying the program is a detailed explanation of why some words will be listed as errors and misspelled words will be accepted by the editor.

If Microproof is integrated with Scripsit on a Model I, it can be called from the Scripsit program by executing an M command. Corrections can be made to the Scripsit text since it remains resident in core while the editor "does its thing."

I have already found that the use of the editor has greatly enhanced the quality of my letters and reports. This is a very useful product and should be obtained by anyone who uses a word processor.■

THE EXCLUSIVE ORACLE

"I live in an old, non-renovated apartment with ancient wiring. The light flickers every time the refrigerator goes on."

Q: My machine goes back to the Memory Size question, the keyboard gets "locked out", and sometimes the keyboard shuts off its own power supply by itself. Several times this has happened as I inadvertently passed my hand near the power supply box—too often for coincidence, so I thought maybe stray capacitance from my hand had something to do with it. But usually this has no effect; most often the reboots happen for no reason at all.

The first thing I suspected was power line spikes and hash. I live in an old, non-renovated apartment with ancient wiring. The light flickers every time the refrigerator goes on. On the recommendation of a person in the local computer club, I bought an Isolator, made by Electronic Specialists. That seemed to help slightly—the reboots were less frequent (only a couple times per night). I was less than satisfied with this result, so I traded the Isolator in for the Super Isolator by the same company. That seemed to do the trick—for about three days. Then, the same problem returned and the "dancing screen" on my monitor was worse. I figured that the device might work for sudden spikes but was no good on slower fluctuations, like those caused by appliances.

Looking at the schematic in the *Technical Reference Handbook*, it was clear that the 723 voltage regulator in the 12-volt supply inside the keyboard pretty well controlled things. It could even cause a power shutdown through a feedback loop by triggering Q6. That could explain the seldom, but puzzling, total power loss in the keyboard.

Anyhow, I replaced Z2 with another 723. It seemed to work for a week, no glitches. But now they're back again with or without the Super Isolator. Could it be that the new 723 has already been blown by my fluctuating line voltage? Or maybe I'm entirely off base.

Just in case you do tell me to replace the one or the other 723, could I safely socket them? Would the heat buildup destroy the socket?

Mendel Cooper
New York, New York

A: Your problem diagnosis and approach are excellent, leaving only a few remaining avenues:

1. The likelihood of power supply spikes killing your 723 is small. Short-term spikes of 150 percent or more are not impossible, but it isn't likely that these spikes would increase the 19-volt output of the TRS-80's transformer so high as to exceed the 723's 40-volt maximum input. The other 723 (Z1) has only a 14-volt input. The use of the Super Isolator nonetheless should help prevent voltage disturbances from upsetting the regulator enough to cause memory crashes.

2. If you still suspect that spikes in your area might be disturbing the 723 regulators enough to cause crashes, replace them with a Signetics type NE550, which is a faster, high-performance 723 equivalent.

3. Use an accurate voltmeter to check the calibration of R5, which controls the 5-volt output line, and R10, which handles the 12-volt supply. Most TRS units were shipped without these controls being lacquered, and they can easily be knocked out of adjustment when the case is opened.

4. If your keyboard unit is actually shutting down (the LED flashes off), then the 5-volt line is definitely suspect, which is controlled by Z1. Consider:

   a. A bad fuse wire or cold solder joint in the power supply transformer; later units had a case which could be opened and this wire replaced.

   b. A broken connection in the cable or on the power supply section of the keyboard unit.

   c. A solder ball or splash on the circuit board somewhere in the unit.

5. A broken or cracked keyboard connector cable which can cause lockup and crashing (but not power-down). Check this cable carefully.

6. You might have a balky Q2, Q3, or Q5. Since you ask if the 723 can be socketed because of heat (it normally can be), it sounds like there's more heat than necessary in the power supply area. The heat sink around Q2 should feel very hot, but not so hot that components in the vicinity are too hot to touch. Replace Q3, Q2 and CR1 if this seems to be the case. Likewise, if the heat sink on Q6 is hot, suspect a problem there. This heat sink should only get moderately warm.

In sum, it sounds like the remedies you have tried point to an overload on the output side of the 723 and its associated high-current transistors controlling the 5-volt supply. A short on the board, the intermittent failure of the pass transistors, or (last of all and most difficult), intermittent failure of an onboard chip's internal supply, could all be the cause of your woes.

Q: Running the program below I get what appears to be ROM data appended. That's no problem—I just delete lines not required. However, if random lines of ROM are available, how can I get all the ROM printed?

```
(The following is the end of the printed program)
280 PRINT"RESTS CHAR/LINE = POKE 32250,0"
290 PRINT"SET CHAR/LINE = POKE 32372,pD
SYSTEMRESTOREX
dl = !
I
M
FORAI%9PFOR = 8FOR9 = 15
0FOR-
e%8FOR%LFOR9 = &FOR = A
I
Q% = 9
0DATA
1752 MEMINPUT% + FORUAA
I
```

E. L. Armson
Aurora, Ontario

A: First of all, what you see are not lines of ROM. The Level II Basic language in ROM does not use Basic keywords, but rather unfamiliar-looking machine language. What you have is garbage being translated into Basic-like words by the List command.

You haven't described the conditions under which this list appears. If it appears immediately upon loading the program, perhaps some of the final Basic lines of the program loaded badly. This would cause garbage to be printed in a list. Or, if you have already run the program, the active POKEing around memory that this particular program does may be interfering with the data stored in RAM, causing the garbage during list.

By the way, if you want a quick look at the contents of the ROM as stored in memory, run this line: 10 FOR X = 0 TO 12288 : PRINT CHR$(PEEK(X)); : NEXT.

Q: I noticed some articles in *80 Microcomputing*, and I have a few questions. Do you know if the reverse video, high speed, and the upper/lowercase modifications are compatible with each other? If not, do you have or can you recommend a lowercase mod that is compatible? Also, in the reverse video mod you say that graphics may squirm. Is that when the video is reversed or whenever the mod is present?

Brian Harris
Westhampton Beach, New York

A: All the video, speed, and upper/lowercase modifications published in *80 Microcomputing* to date are compatible with each other. The lowercase mod published in the March 1980 issue is not, however, compatible with the standard Radio Shack or Electric Pencil modifications. If you write to the authors of that article and enclose a self-addressed, stamped envelope, they will forward the updated modification instructions to you.

The graphics in the reverse video modification squirm only in the reverse mode. In fact, they are improved in the normal mode. An inexpensive reverse video modification manufactured by Archbold Electronics includes a separate "non-squirm" board for the video monitor.

Q: Shortly after upgrading my TRS-80 to 16K, I punched out a copy of your RAM test program, Babyroot (May 1980 p. 76). I have now upgraded my unit to 48K. Do you have a means of expanding Babyroot for use in my current system?

Also, I would like to know if you are aware of any patch that will prevent the Radio Shack Editor/Assembler form disabling the Radio Shack lowercase modification driver software. No help has been forthcoming when I posed this question to the experts on the Fort Worth hotline.

C. W. DeLaughter
California, Maryland

A: Yes, the modification to Babyroot is simple. For 32K, change byte 502C from 80 to CO and change byte 50E6 from F0 to B0. In the Basic verson, change the 13th data value in line 50 from 128 to 192, and the 7th value in line 240 to 176. For 48K, change the bytes noted above to 00 and 70, respectively, and the values in the data lines from 0 to 112, respectively. Be prepared to wait 25 minutes per 32K cycle, and 35 minutes per 48K cycle.

The reason the lowercase driver is disabled when using EDTASM is because the program uses its own keyboard scan routine. It was written for use with either Level I or Level II, so makes no calls at all to ROM. If you want to add debounce, repeat, lowercase, etc., you will have to patch that routine into EDTASM's own keyboard driver block, located at 4301 hex.

For starters, below is a hex dump of a debounce/repeat/beep routine that patches into EDTASM version 1.1. You can add your lowercase driver to this, entering all of this code and the EDTASM changes using a relocated version of T-Bug or a similar monitor:

```
5D00  ·  21 21 40 01 01 38 16 00 0A 5F A3 20 A1 77 14 2C
5D10  ·  CB 01 79 D6 80 20 F1 00 06 07 2D 86 10 FC FE 00
5D20  ·  3E 00 C0 32 1A 40 C9 A6 28 10 3A 1A 40 3C 32 1A
5D30  ·  40 FE FF 20 D9 3D 32 1A 40 7B 73 C5 01 00 02 CD
5D40  ·  60 00 C1 0A A3 C8 C5 E5 F5 06 40 3A 3D 40 E6 FD
5D50  ·  67 F6 02 6F 7D D3 FF 7C D3 FF C5 06 40 10 FE C1
5D60  ·  10 F2 F1 E1 C1 C3 07 44
```

You will need to make the following changes to EDTASM itself:

Addresses	Old Values	New Values
4710 4A07	11 F9 5C	11 00 5E
4ADB 4B50 4D39		
4D80 5227 468A	21 F9 5C	21 00 5E
43EF	21 21 40	C3 00 5D

Brian Donlan of Lake Charles, Louisiana, has written a simple debounce-only modification for EDTASM:

"The keyboard driver for EDTASM begins at 43EFH and continues through 445FH. I replaced the routine here with the routine I found in Tandy's KBFIX starting at 7FDAH. There was one jump that had to be changed. After loading this routine over the one in EDTASM, I changed the instructions now residing at 4402H from F2 E2 7F to F2 F7 43.

"If you wish to punch in the changes with your monitor, here is the code already including the above instruction changes:"

```
43EF  ·                                                21
43F0  ·  36 40 01 01 38 16 00 0A 5F AE 73 A3 20 08 14 2C
4400  ·  CB 01 F2 F7 43 C9 5F C5 01 DC 05 CD 60 00 C1 0A
4410  ·  A3 C8 C3 FB 03
```
■

Desperate? Send your questions on Model I, Level II, TRS-80s to: Dennis Bathory Kitsz, Roxbury, Vermont 05669.


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
PHOTO COPYRIGHT · DON DIXON
*TRS-80 is registered Trademark of the Tandy Corp.

THE ASSEMBLY LINE

by William Barden, Jr.

"The May issue. . . contained eight questions for The Fourth Great Assembly Language Contest. This month we'll give the winning solution."

The May issue of this column contained eight questions for The Fourth Great Assembly Language Contest. These contests are held in this column periodically to test both beginners and hackers alike. This month we'll review the eight questions, discuss some of the subtle aspects, give the winning solution, and then (with a fanfare of rebooting disk drives and a flourish of Line Printer VIIs), announce the names of the two winners—one in the novice class and one in the hacker class.

Problem Number One

What is the tightest timing loop that can be written for counts of 1-65,536? An example of a slow loop is:

```
        (HL) = Count, 0 – 65,535
LOOP  DEC   HL
      LD    A,H
      OR    L
      JR    NZ,LOOP
```

Timing loops keep rearing their ugly heads in many assembly language programs, such as cassette driver bit timing, RS-232-C timing in software, computer-generated music and so forth.

To figure out the timing of any set of instructions or any loop, start with the number of T cycles for each instruction. These are given in the Radio Shack Editor/Assembler documentation, Zilog reference materials or other documents, usually for a 4-MHz clock. A T cycle is one clock period. For the TRS-80 Model I, a T cycle is .56375 microseconds. For the timing loop above we have:

```
LOOP  DEC  HL       6 T cycles
      LD   A,H      4
      OR   L        4
      JR   NZ,LOOP  7/12
                    ------
                    21/26 T cycles = 11.84/14.66
                    microseconds
```

The JR NZ,LOOP takes 12 T cycles if the jump is made and 7 T cycles if the jump is not made. If the jump is not made, the jump address does not have to be jammed into the program counter, which has already been updated by one to point to the next instruction in sequence; if the jump is made, the program counter has to be loaded with the jump address.

The timing loop takes 14.66 microseconds for every iteration except the last and 11.84 microseconds for the last, so for any count in HL we have: TIME = (CNT – 1)•14.66 + 11.84 microseconds.

As a count of zero is treated as 65,536, the maximum delay will be 960,755 microseconds, or almost a second.

What I was after here was something like this entry:

```
(HL) = timing   count
(BC) = – 1
LOOP ADD        HL,BC
     JP         C,LOOP
```

"With a fanfare of rebooting disk drives and a flourish of Line Printer VIIs, we announce the winners of The Fourth Great Assembly Language Contest."

This loop takes 11 plus 10 T states, or 21 T states. By the way, this loop terminates when the count in HL goes negative, due to the test on the carry!

Wayne Westmoreland of North Augusta, SC, however, taught this old dog of a columnist a new trick with this tight solution:

```
(BC) = count, 0 – 65,535
LD    D,H
LD    E,L
LDIR
```

Several people took this approach. Here, a block move moves data to itself, essentially doing NOPs. The number of bytes transferred is the byte count in BC. The timing is 21 T states for the LDIR on all but the last cycle, where it is 16. This results in a tight loop of 11.84/9.02 microseconds.

Problem Number Two

What do these three instructions do to the contents of HL?

```
SR1  CALL  SR2
SR2  ADD   HL,HL
     RET
```

This is a tricky question as we've made some assumptions about the stack. We've assumed that this code is a subroutine Called from another place, with the return address in the stack. The code executes like this: HL contains some value, which we'll call X. The Call instruction calls the code at SR2, which adds HL to itself, multiplying X by two. A Return is then made back to the instruction following the Call. As this instruction is at SR2, another Add of HL to itself is done, yielding 4•X, and a return is made back to the unspecified call. Almost all entries had the correct answer on this one.

Problem Number Three

What does the following code do?

```
LD    (HL),A
DEC   BC
LD    D,H
LD    E,L
INC   DE
LDIR
```

To see what's being done here, let's look at the LDIR, the key to this puzzle. LDIR is a block move that moves a block of data from a source area into a destination area for a specified number of bytes. Before the move is done, HL holds a pointer to the start of the source area, DE holds a pointer to the start of the destination area, and BC holds the byte count. The move is then done beginning to end. The first byte from the source area is transferred to the first byte of the destination area, HL and DE are incremented to point to the next byte, and BC is decremented. The LDIR loops on itself until BC has decremented down to zero.

Some strange things happen if the source area and destination area are overlapping. Suppose that the source area is 8000H, the destination area is 8001H, and the number of bytes is three, as shown in Fig. 1. 800H contains a zero, 8001H a one, 8002H a two, 8003H a three and 8004H a four. The first LDIR iteration puts a zero into 8001H and adjusts HL to 8001H, DE to 8002H and BC to two. The next move puts the contents of 8001H, now a zero, into 8002H, and adjusts HL to 8002H, DE to 8003H, and BC to one. The next move puts the contents of 8002H, now a zero, into 8003H, and adjusts HL to 8003H, DE to 8004H and BC to zero. You can see that the first byte has been repeated down through the entire area!

That's the secret of this code—it uses the replication effect to fill a block of memory with a given value. Suppose that A is 33H, HL is 8000H and BC is four. A is stored in 8000H by the LD (HL),A. BC is decremented from four to three. DE is loaded with HL, 8000H, and incremented by one to give 8001H. The LDIR will now replicate the first byte of the source in 8000H down through 8001H, 8002H and 8003H. We've used the LDIR as an efficient fill character routine. The routine uses A as the fill characters, HL as the start of the area to be filled, and BC as the number of bytes to be filled. Several people pointed out that BC must be two or greater. If not, the fill will work on the entire memory!

Problem Number Four

A value of zero to 15 decimal must be converted as shown below. What are the fewest number of instructions to do this?

0 to 30H	4 to 34H	8 to 38H	C to 42H
1 to 31H	5 to 35H	9 to 39H	D to 43H
2 to 32H	6 to 36H	A to 40H!	E to 44H
3 to 33H	7 to 37H	B to 41H	F to 45H

I thought some of you might misread this, and I put an exclamation mark after the 40H! Values of zero through nine are changed to their ASCII equivalents, while values of B through F are changed to their ASCII equivalents minus one. A value of A is changed to an at-sign (@). Admittedly, this routine is not extremely useful.

This is obviously a trick question, but as one reader said, "Thanks for this, it made me think!" It turns out that the conversion can be done in two instructions:

```
ADD   30H
DAA
```

The DAA instruction is the Decimal Adjust Accumulator. It can be used to do adds and subtracts on decimal, also called binary coded decimal or BCD data. BCD is a carry-over from the earlier days of computers when some machines used BCD format; in addition, there's a lot of instrumentation that uses BCD format for input and output.

In BCD, there is one digit per four bits, and two per byte. Valid BCD digits are zero through nine. The values of 1010 through 1111 are not allowed. The eight-bit value 10011000, for example, reads 98 in BCD and represents the decimal value of 98, rather then the binary value of 152. The Z-80 permits BCD adds and subtracts to be done by means of the DAA. An addition of 00101001 and 00110011 (BCD 29 and BCD 33), for example, would yield 01100010, the BCD result of 62. All that must be done to get the true BCD result is to follow an add or subtract with a DAA instruction:

"Some strange things happen if the source area and destination area are overlapping."

```
ADD   A,B    ;add two bcd digits
DAA          ;adjust from binary to BCD
```

The conversion is done in the DAA by using the carry flag, the half-carry flag, the two four-bit results and the add/subtract flag. A value of six on an add or a value of 0FAH (minus six) on a subtract is added to each four-bit result to adjust it to be a true BCD digit if necessary.

Of course, what I was really looking for when I first thought of the problem was a tricky way to convert from a binary value to its ASCII equivalent. I settled for the problem above when I couldn't find it.

About half of you got the correct answer. In looking over the replies, I saw the following code from David Lamkins of Marlboro, MA:

```
ADD   A,90H
DAA
ADC   A,40H
DAA
```

David said, "I don't claim originality, though the source is unknown." David's response doesn't solve the problem as posed but it is a gem! It solves the problem I wanted to pose. It does convert from a binary value to ASCII in four instructions! (Try it!) A special honorable mention, therefore, goes to David.

Problem Number Five

Find the integer portion of log base 2 of a value in the A register, with the result in any register. Examples:

1 produces a value of 0
2 produces a value of 1
3 produces a value of 1
4 produces a value of 2
5 produces a value of 2
15 produces a value of 3
25 produces a value of 4
37 produces a value of 5

The question posed in an alternative way is, "Write code to find the next lower power of two for a given number. Return the power of two represented."

We know from exhaustive study of binary, hexadecimal and bi-quinary in late evening hours that each bit position in a binary number represents a power of two. A value of two to the power of zero (one) is in bit position zero at the right, two to the power of one (two) is next, two to the power of two (four) is next, and so forth. To find the first power of two represented by a binary value, all we have to do is to look for the leftmost one-bit in the value. Sup-

	BEFORE LDIR	AFTER FIRST ITERATION	AFTER SECOND ITERATION	AFTER THIRD ITERATION
8000H	0	0	0	0
8001H	1	0	0	0
8002H	2	2	0	0
8003H	3	3	3	0
8004H	4	4	4	4
HL =	8000H	8001H	8002H	8003H
DE =	8001H	8002H	8003H	8004H
BC =	3	2	1	0

Fig. 1. LDIR with overlapping source/destination.

This is . . .

. . . Your last chance

to get 80 MICROCOMPUTING for only $18 a year.

August 1, 1981 the price of 80 Microcomputing is going up. The cover price will be $2.95 and the subscription rate will be $25 for one year. But you still have a chance to renew or subscribe at the old rates. . . . send in the card (if the cards are gone, a photocopy of the coupon will do) before October 1, 1981 and you can get 80 Microcomputing at the old rates.

You get more information for your TRS-80* from 80 Microcomputing than from any other single source. One of the numerous (we've been averaging thirty a month) programs from each issue put to use can easily pay for the subscription price. Even at $25.00 a year. But beat the price increase and continue to get the hundreds of pages of reviews, articles, programs, and applications for only $18.00.

Remember after October 1, 1981 no previous offers to 80 Microcomputing at a price less than $25.00 will be accepted. So send in the card today.

Wayne Green Inc., Peterborough NH 03458

☐ **Yes,**

I want another year of 80 Microcomputing at the low price of $18.00

Bill me ☐ **Bill my** ☐ **VISA** ☐ **AE** ☐ **MC** **Check enclosed** ☐

Name ______________________

Address ______________________

City ______________ **State** __________ **Zip** __________

Card # ______________ **Expire Date** ______________

Signature ______________ **Interbank #** ______________

Offer expires October 1, 1981

If you are renewing,
the number from your address label would be helpful ______________

Canadian 1 year only/US funds $20.00 Foreign 1 year only/US funds $28.00
80 Microcomputing • PO Box 981 • Farmingdale, NY 11737 **318B8**

pose we have the value of 75 decimal. This is represented by 01001011, or two to the power of six plus two to the power of three plus two to the power of one plus two to the power of zero. The leftmost one-bit is two to the power of six, or 64. The following routine, from Bill Smythe of Chicago, IL, is representative of the answers to return the leftmost power of two:

```
        LD      B,7
LOOP    RLCA
        RET     C
        DJNZ    LOOP
        RET
```

A bit position value in B is initialized to seven. A is shifted left by an RLCA. If the carry is set, B contains the first power of two with a one-bit. If the carry is not set, the bit position number is decremented by the DJNZ, and another pass is made. As Bill says, "Zero is returned in B if A equals zero, which is technically incorrect, but minus infinity is hard to represent in one byte!"

Problem Number Six

What does the following code do?

```
        JR      NEXT+1
NEXT    JR      NEXTP-1
NEXTP ...
```

To solve this one, we've got to see what the assembly listing produces for machine code. The JR instruction produces a two-byte instruction of 18H plus a one-byte displacement. The one-byte displacement is a signed eight-bit value that, when added to the contents of the program counter, will point to the location specified. As the program counter points to the next byte after the JR, the one-byte displacement value allows us to jump minus 126 bytes back from the JR or plus 129 bytes forward, for displacement values of minus 128 through plus 127, respectively.

The listing looks like this:

```
8000  18  01              JR  NEXT+1
8002  18  FF  NEXT        JR  NEXTP-1
8004          NEXTP ...
```

NEXT + 1 is at 8003, so the JR NEXT + 1 jumps to 8003, which is a 0FFH. Now the next question is, what instruction is a 0FFH? Close scrutiny of our Zilog reference card reveals that it is an RST, with a three-bit field of 111. This is an RST 38H, and that's what will be executed.

Problem Number Seven

A contains a signed value. What does it contain after:

```
NEG
CPL
```

This was an easy one. The NEG negates the value in A, changing it to the same magnitude, but opposite sign. Suppose, for example, we had 00000101, or plus five, in the A register. The NEG takes the two's complement by changing all the ones to zeroes, changing all the zeroes to ones, and adding one. The result after the NEG is 11111011, or minus five. If the original number is N, the NEG finds $-N$, the CPL performs a one's complement. A one's complement changes all the ones to zeroes and all the zeroes to ones. If N is the number before the CPL, the CPL finds $-(N+1)$. The two instructions together find $-((-N)+1)$ or $N-1$. Examples are:

Value	After NEG	After CPL
00000000 (0)	00000000 (0)	11111111 (−1)
00000001 (1)	11111111 (−1)	00000000 (0)
00000010 (2)	11111110 (−2)	00000001 (1)
01111111 (127)	10000001 (−127)	01111110 (−126)
1111111 (−1)	00000001 (1)	11111110 (−2)
11111110 (−2)	00000010 (2)	11111101 (−3)
10000000 (−128)	10000000 (−128)	01111111 (127)

You can see that there is one overflow condition for minus 128 in which the final result is erroneous.

Problem Number Eight

What is the purpose of TWDEE and TWDUM?

```
SR      CALL    TWDEE
        .
        .
        .
TWDEE   EX      (SP),HL
        PUSH    DE
        PUSH    BC
        PUSH    AF
        LD      DE,TWDUM
        PUSH    DE
        JP      (HL)
TWDUM   POP     AF
        POP     BC
        POP     DE
        POP     HL
        RET
```

Let's see what this code does. SR is a user subroutine. It has been called by other code that does not appear. The return address from the Call is on the stack. SR calls TWDEE, and as it does so, the return address after SR is put onto the stack. The first instruction of TWDEE exchanges the top of stack with the HL register. HL is assumed to contain valid data that must not be destroyed. In doing the EX instruction, the HL register has automatically been saved on the stack in place of the return address for the SR CALL.

The DE, BC and AF registers are now pushed onto the stack. At this point HL, DE, BC and AF have been saved on the stack, in that order. Now DE is loaded with the address of the TWDUM subroutine and pushed. The stack now holds the user return for the SR Call, HL, DE, BC, AF and TWDUM address. Next, a jump is made indirectly to HL. As HL was loaded with the return address in the EX instruction, this causes a return from TWDEE.

At some later point in the user subroutine, a RET is executed. The return causes the stack to be popped and the return address to be put into the program counter. The first item on the stack, however, is the TWDUM address, and the code at TWDUM is therefore executed. TWDUM restores AF, BC, DE and HL from the stack, and then executes a return. This return causes a return to the location following the Call to SR.

This whole sequence, therefore, is a clever way to automatically save all general purpose registers with one instruction, and to do an automatic restore and Return with the RET instruction at the end of a subroutine. The user subroutine employing this technique would have something like:

```
SR  CALL TWDEE    ;save registers
    (body of SR)
    RET           ;restore and Return to user
```

Since subroutines typically save registers at the beginning and restore them at the end, this technique saves the user from performing a lot of PUSHes and POPs with one Call and a single Return.

The Winners

And now for the winners. The bulk of mail I got was considerable, so I used the following technique to determine the winners: I looked for correct answers to questions two through eight first and then went back to compare code for question one. Those entries not specifying novice or hacker were assumed to be in the hacker category.

The winner in the novice class was Bryon Wallace of Refugio, TX. The winner in the hacker class was David S. Raley of Roosevelt, NY (honorable mention: Smythe, Yelvington, Westmoreland, Hutchins, Rollins, Nelson, Delfs, Black, Felsheim, Pirog, Cunningham and Miller). These two winners will receive copies of my latest book *Microcomputer Math* and copies of the Prentice Hall book *Microcomputer Interfacing* by Artwick.

That's it for this month. I hope you enjoyed the contest. I know I did, and picked up some good techniques from the entries! ■

80 APPLICATIONS

by Dennis Kitsz

"Midsummer is a time of relaxation. I would like to present a little entertainment first. . ."

Midsummer traditionally is a time of relaxation. In keeping with that, I would like to present a little entertainment first, and save the brain-cracking for later.

Fifteen years ago, a British mathematician by the name of John Conway envisioned an orderly universe populated by obedient but boring creatures. These creatures followed strict rules that governed their existence, rules that were coincidentally quite mathematically regular. He called his universe, together with its rules, "The Game of Life."

The Game of Life is not a board game nor an adventure—it is, rather, a *ludus*, a pastime, an entertainment, or miniature theatre. A mythical civilization of uniform beings is born, and its members give birth, live, and die according to a specific decree. The laws are simple:

1. All beings must live within the bounds of a finite universe.
2. That universe consists of a regular grid, in each position of which a being may live. In this way a being may have no more than six immediate neighbors.
3. When a being is surrounded by two or three immediate neighbors, it may live.
4. When a being is surrounded by more than three neighbors, it must die.
5. When a being is surrounded by less than two neighbors, it must die.
6. When a position in the universe is immediately surrounded by exactly three other beings, one is given birth at that position.

The universe for the Game of Life could be infinite, but it would then take an infinite amount of time for a computer to figure out the next generation of beings within it. Since I only have till next month to work this program out, I've scaled the universe down to the 64 × 16 characters on the TRS-80 screen (what a coincidence!).

There are many beautiful patterns possible with the Game of Life, with names like gliders, ponds, flashers, and spaceships. In the ideal Game of Life these will flicker and flash in exquisite patterns. (If you're interested in a description of these forms, as well as a more extensive 128 × 48 version of the game, turn to the June, 1980, issue of *80 Microcomputing*. Also, the theory and myriad versions of the game were first presented in *Scientific American*, and has been covered there and in *Byte* over the years.)

The Game of Life is a perfect mental exercise in programming, because the display should be fast and fluid. The number of calculations, however, can be extraordinary: for the TRS-80, the fastest 64 × 48 grid I have seen is Charley Heath's (sold by Instant Software), at about 180 generations per minute; a later revision of my own version in *80 Microcomputing* is the fastest 128 × 48 grid, at about 80 generations per minute. In these, the bit-addressing scheme used in the TRS-80's video graphics system makes these calculations slow; using entire bytes for each character makes the Game of Life run much faster.

The version presented here has an average speed of 740 generations per minute—over 12 per second! If you have a 50 percent speed-up modification in your TRS-80, you will see the generations flash by at over 1,100 per minute.

What's an Algorithm?

The heart of any program, the method used to solve a problem, is called an *algorithm*. Here's an example: given the problem, "What is the result of 355 divided by 113," how do you find a solution? Your human algorithm might be long division, quick successive approximation, a grab for the calculator, or the knowledge that this particular division is a pretty good approximation of pi.

The scheme by which you solved that problem might be called an algorithm. In a more realistic situation—since there are more numbers in the world to divide than 355 and 113—the problem would be stated as "What is the result of x divided by y." Your task then would be to come up with an overall solution, a generalized algorithm (which still might be to grab for the calculator!).

All the rules for the Game of Life are presented above. The most common algorithm developed from them is to evaluate the number of live cells immediately neighboring a given grid position, and alter that grid position accordingly.

A better solution was developed by Philip K. Hooper (The Codesmith, Northfield, VT) for the KIM-1. Dr. Hooper is a programmer specializing in 6502-based computers, and his methods are determined by the most elegant and efficient achievement of programming goals. His solution was like the round earth. . . it was obvious only after being discovered. Dr. Hooper's version of the Game of Life assumes (given any universe and Conway's rules) that after the first few generations have been produced, there will always be more empty cells than filled ones. Applying logic and math to the rules shows that this is true.

That implies that an individual cell need not be examined for its neighbors, but rather that the life or death of the *neighbors* may be determined by the state of the cell being looked at. With this in mind, it's clear that the program's speed will be increased over the first method because it will ignore any unfilled grid positions, stopping only to test and store information about the neighbors of live positions.

Creating the program turned out to be quite simple. Program Listing 1 will run under Level II Basic, or can be loaded from a disk system (disk users have been taking me to task for ignoring them in this column), and the program occupies only 154 bytes of memory. 1K of additional memory is used to store a working image of the screen. Program Listing 2 is the same program converted to a series of Basic POKE statements.

Invading Hordes

When loaded as a System tape, the program will automatically take control of the computer by patching into the keyboard scan and directing that loop to itself. It then returns the original address so the program may be exited at any time by pressing the reset key. Disk users may save it as a CMD file. The Basic version will automatically execute when the USR(0) statement is encountered.

Interrupts are disabled in order to increase the speed of the program. The screen will clear (a ROM call in line 310 of Listing 1), and a block of higher memory will be filled with −1's. These two steps initialize the program with a blank-screen

TRS-80™ MODEL I & III PROGRAMS
SUPER SOFTWARE AT SUPER PRICES

TRS-80™ A Trademark of Tandy Corp.

THE ULTIMATE DISK-OPERATING SYSTEM

LDOS™

A completely documented new generation of operating system for the TRS-80. Far superior to any on the market. It is a totally device independent system, capable of device linking, routing, setting, and filtering. LDOS will support 5" and 8" floppies, single/double density, single/double sided, and up to 80 tracks. Full customer service proivded with a toll free 800 number. Excellent documentation in a tab indexed manual (over 250 pages). Call or send for more information today.

AVAILABLE for MODEL I or III ... **ONLY $149.00**

LDOS™ is a product of Logical Systems Inc.

BUSINESS SYSTEMS & MAILING SYSTEMS

INVENTORY MASTER SYSTEM

With today's market, keeping on hand what your demand calls for is reason enough to let your TRS-80 accurately and effectively take care of your Inventory. Always know what to stock and when to stock it. This system has many features which were modeled after a main frame system of large capacity. More information available upon request.

MODEL I VERSION JUST **$159.00**
MODEL III VERSION JUST **$259.00**

MAIL/FILE - SERIES I

Mail/File - Series I is the system which originally set the standards for the industry. It's the original Name, Address & Phone Number Data Base Manager which has made Galactic customers happy for years. A flexible and capable program for any small business or person with mailing list needs. Call or send for more information today.

AVAILABLE FOR THE MODEL I ONLY

.... originally **$99.00**
HURRY* Special limited time offer ... **$59.00**

MAIL/FILE - SERIES II

NEW Series II Mail/File has all of the outstanding features of the original Series I Mail/File, PLUS many additional qualities which again will set standards for this type of system. "Word processor" type input editor, fast sorting, and fabulous editing capabilities are a few of its features. Name and ZIP code are under constant sort. Series II is compatable with Series I data files.

MODEL I SYSTEM will handle up to 600 Names
MODEL III SYSTEM will handle up to 1200 Names
Either SYSTEM AVAILABLE for **ONLY** . **$159.00**

ALSO AVAILABLE
... For the MODEL I or III

STOCK MARKET MONITOR

Cassette version **$89.00**
Disk version **$99.00**

ULTRA-TREK GAME

Can You Master This Trip Through the Galaxies? Climb aboard the Star-Ship Enterprise and head for a universe of unknown encounters, enemies and stellar battles. As commander, will you be able to keep the Enterprise and its crew en route and on patrol? Can you destroy the hostile base? ORDER TODAY.

MODEL I OR III VERSION ONLY **$19.00**

All of Galactic's products have a one year limited warranty with customer support included with each purchase. Warranty information is available upon request in writing.

MONEY ORDERS, CREDIT CARDS, & COD'S SHIPPED WITHIN 24 HOURS.

MasterCard VISA

galactic software ltd.

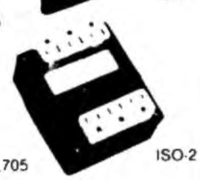
A Division of GS & WS, Inc.

11520 N. Port Washington Rd. • Mequon, Wisconsin 53092
Telephone: (414) 241-8030

**DISK DRIVE WOES?
PRINTER INTERACTION?
MEMORY LOSS?
ERRATIC OPERATION?**

Don't Blame The Software!

ISO-1



ISO-2

Power Line Spikes, Surges & Hash could be the culprit!

Pat. #4,259,705

Floppies, printers, memory & processor often interact! Our patented ISOLATORS eliminate equipment interaction AND curb damaging Power Line Spikes, Surges and Hash.

- **ISOLATOR (ISO-1) 3 filter isolated 3-prong sockets; integral Surge/Spike Suppression; 1875 W Maximum load, 1 KW load any socket $62.95**
- **ISOLATOR (ISO-2) 2 filter isolated 3-prong socket banks; (6 sockets total); integral Spike/Surge Suppression; 1875 W Max load, 1 KW either bank $62.95**
- **SUPER ISOLATOR (ISO-3), similar to ISO-1 except double filtering & Suppression $94.95**
- **ISOLATOR (ISO-4), similar to ISO-1 except unit has 6 individually filtered sockets $106.95**
- **SUPER ISOLATOR (ISO-11) similar to ISO-2 except double filtering & Suppression $94.95**
- **CIRCUIT BREAKER, any model (add-CB) .... Add $ 8.00**
- **CKT BRKR/SWITCH/PILOT (-CBS) .... Add $16.00**

AT YOUR DEALERS

**Master-Card, Visa, American Express
Order Toll Free 1-800-225-4876
(except AK, HI, PR & Canada)**

ESP Electronic Specialists, Inc. ✓58

171 South Main Street, Natick, Mass. 01760

Technical & Non-800: 1-617-655-1532

MANAGEMENT SYSTEMS SOFTWARE, INC.

Dr. Jerome S. Osteryoung, renowned business educator, and author of 5 business texts has written the following programs.

1. PROFORMA CASH-BUDGET PROGRAM

Allows the user to project the cash-balances for up to twelve periods in the future. Amount of loan, if needed, is computed as well as computing funds available for short-term investment. **(Price $125)**

2. LEASE-PURCHASE PROGRAM

Evaluates the lease vs. purchase decision incorporating all the latest tax laws including the investment tax credit and accelerated depreciation. This program gives the user all the information necessary to make this decision. **(Price $50)**

3. BUSINESS PROGRAM PACKAGE

13 Business programs (e.g., capital budgeting, cash-management, ratio analysis, debt management). These programs will be very useful to the business manager. **(Price $200)**

4. PROCUREMENT PROGRAM

Ascertains purchase amount when future price of commodity is varying. A must for all managers who have purchasing responsibilities. This program takes into consideration inventory levels, inventory capacity, and financial carrying cost in determining the optimal amount of an item to purchase when future prices are varying. **(Price $150)**

5. COLLEGE ENROLLMENT PROJECTION PROGRAM

Forecasts the enrollment for colleges using several different statistical techniques. User can specify the number of periods for which a forecast is desired. **(Price $100)**

Extensive Documentation With Each Program

Write or call for a brochure which describes the product in greater detail.

✓87 MANAGEMENT SYSTEMS SOFTWARE INC.

5200 Brittany Drive, #1006 St. Petersburg, Florida 33715
813/864-4347

✓See List of Advertisers on page 322

Program Listing 1. Assembly Listing of The Game of Life

```
               00100 ; *********************************************************
               00110 ;              SOURCE PROGRAM FOR QUICK LIFE
               00120 ;                DENNIS BATHORY KITSZ
               00130 ;               ROXBURY, VERMONT  05669
               00140 ;    BASED ON AN ALGORITHM DEVELOPED BY PHILIP K. HOOPER
               00150 ; *********************************************************
               00160 ;
4016           00170        ORG     4016H           ;KEYBOARD INTERCEPT
4016 00        00180        DEFB    000H            ;LOW BYTE OF LIFE START
4017 6F        00190        DEFB    06FH            ;HIGH BYTE OF LIFE START
6F00           00200        ORG     6F00H           ;LIFE PROGRAM STARTS HERE
6F00 211640    00210        LD      HL,4016H        ;KEYBOARD INTERCEPT
6F03 36E3      00220        LD      (HL),0E3H       ;RESTORE ORIGINAL VALUE
6F05 23        00230        INC     HL              ;GET INTERCEPT HIGH BYTE
6F06 3603      00240        LD      (HL),03H        ;RESTORE ORIGINAL VALUE
               00250 ;
               00260 ; *********************************************************
               00270 ; CLEAR SCREEN AND PREPARE LIFE BUFFER WITH -1'S
               00280 ; *********************************************************
               00290 ;
6F08 F3        00300        DI                      ;GET RID OF INTERRUPTIONS
6F09 CDC901    00310        CALL    01C9H           ;ROM CLEAR SCREEN ROUTINE
6F0C 210070    00320        LD      HL,7000H        ;BEGINNING OF LIFE BUFFER
6F0F 110170    00330        LD      DE,7001H        ;TRANSFER DESTINATION
6F12 01FF03    00340        LD      BC,3FFH         ;NUMBER OF BYTES TO CLEAR
6F15 36FF      00350        LD      (HL),0FFH       ;CLEAR AREA WITH ALL -1'S
6F17 EDB0      00360        LDIR                    ;BLOCK MOVE TO CLEAR
               00370 ;
               00380 ; *********************************************************
               00390 ; BEGIN INCREMENTING NEIGHBORS OF LIVE CELLS
               00400 ; *********************************************************
               00410 ;
6F19 21003C    00420 ONE    LD      HL,3C00H        ;GET START OF SCREEN
6F1C DD210070  00430        LD      IX,7000H        ;GET START OF LIFE BUFFER
6F20 010004    00440        LD      BC,400H         ;GET TOTAL BUFFER LOC'NS
6F23 7E        00450 TWO    LD      A,(HL)          ;GET VALUE AT SCREEN
6F24 FE20      00460        CP      20H             ;IS IT A SPACE?
6F26 CA416F    00470        JP      Z,THREE         ;IF A SPACE, MOVE ONWARD
FIELD OVERFLOW
6F29 DD34BF    00480        INC     (IX-41H)        ;NO SPACE - INC. NEIGHBOR
FIELD OVERFLOW
6F2C DD34C0    00490        INC     (IX-40H)        ;NO SPACE - INC. NEIGHBOR
FIELD OVERFLOW
6F2F DD34C1    00500        INC     (IX-3FH)        ;NO SPACE - INC. NEIGHBOR
FIELD OVERFLOW
6F32 DD34FF    00510        INC     (IX-1)          ;NO SPACE - INC. NEIGHBOR
6F35 DD3401    00520        INC     (IX+1)          ;NO SPACE - INC. NEIGHBOR
6F38 DD343F    00530        INC     (IX+3FH)        ;NO SPACE - INC. NEIGHBOR
6F3B DD3440    00540        INC     (IX+40H)        ;NO SPACE - INC. NEIGHBOR
6F3E DD3441    00550        INC     (IX+41H)        ;NO SPACE - INC. NEIGHBOR
6F41 23        00560 THREE  INC     HL              ;GET NEXT SCREEN POS'N
6F42 DD23      00570        INC     IX              ;GET NEXT LIFE BUFFER POS
6F44 0D        00580        DEC     C               ;DEC. BC REG. (FAST WAY)
6F45 C2236F    00590        JP      NZ,TWO          ;LOOP IF NOT DONE YET
6F48 05        00600        DEC     B               ;DEC. BC REG. (FAST WAY)
6F49 C2236F    00610        JP      NZ,TWO          ;LOOP IF NOT DONE YET
               00620 ;
               00630 ; *********************************************************
               00640 ; BEGIN EVALUATING WHETHER NEIGHBORS ARE IN CONWAY'S RULE
               00650 ; *********************************************************
               00660 ;
6F4C 21003C    00670        LD      HL,3C00H        ;GET FIRST SCREEN POS'N
6F4F 110070    00680        LD      DE,7000H        ;GET LIFE BUFFER START
6F52 0604      00690        LD      B,4             ;BC=400 (C IS ALREADY 0)
6F54 1A        00700 FOUR   LD      A,(DE)          ;GET VALUE IN LIFE BUFFER
6F55 3D        00710        DEC     A               ;DECREMENT FOR TESTING
6F56 CA646F    00720        JP      Z,SIX           ;IF 0 THEN NEIGHBORS=2
6F59 3D        00730        DEC     A               ;DECREMENT FOR TESTING
6F5A C2626F    00740        JP      NZ,FIVE         ;IF NOT 0 NEIGHBOR >= 3
6F5D 362E      00750        LD      (HL),2EH        ;PUT LIVE CELL IN PLACE
6F5F C3646F    00760        JP      SIX             ;GO OUT TO NEXT CHARACTER
6F62 3620      00770 FIVE   LD      (HL),20H        ;PUT DEAD CELL IN PLACE
6F64 3EFF      00780 SIX    LD      A,0FFH          ;GET -1 VALUE IN A REG.
6F66 12        00790        LD      (DE),A          ;PUT -1 INTO LIFE BUFFER
6F67 23        00800        INC     HL              ;GET NEXT SCREEN POS'N
6F68 13        00810        INC     DE              ;GET NEXT LIFE BUFFER POS
6F69 0D        00820        DEC     C               ;DEC BC (FAST WAY)
6F6A C2546F    00830        JP      NZ,FOUR         ;REPEAT 400H TIMES TOTAL
6F6D 05        00840        DEC     B               ;DEC BC (FAST WAY)
6F6E C2546F    00850        JP      NZ,FOUR         ;REPEAT FOR FULL SCREEN
               00860 ;
               00870 ; *********************************************************
               00880 ; CHECK FOR THE HUMAN WHO MAY BE PRESSING THE KEYS
               00890 ; *********************************************************
               00900 ;
6F71 3A1038    00910        LD      A,(3810H)       ;GET 0-7 KEYBOARD ROW
6F74 A7        00920        AND     A               ;CHECK IF KEY PRESSED
6F75 CA196F    00930        JP      Z,ONE           ;NEXT GEN. IF NO KEY
               00940 ;
               00950 ; *********************************************************
               00960 ; BEGIN CHARACTER GENERATION BASED ON SCREEN CONDITION
               00970 ; *********************************************************
               00980 ;
```

Program continues

universe and a clear area in which living and dead cells may be tested.

The heart of the program begins in line 420, where the HL register pair is given the beginning screen address (3C00H) and the IX register is given the cell-test address (7000H). The BC register pair is handed the total count of cells on the screen (400H, or 1024 screen locations). Each screen location is tested to determine if it contains a space (line 460). If a space is present, no further work is done, and both the cell-test area (IX) and the screen position (HL) are bumped ahead to the next location. The C register is decremented and tested for zero; only then is the B register decremented. When both are zero, the routine terminates.

If a character is found on the screen (when pointed to by HL), the six neighbors of this live cell are incremented. The IX register can be addressed indirectly, using an offset—that is, the contents of memory to which it points may be changed, as well as the contents of memory within a range of ±127 bytes of it. When a live cell is discovered, this program increments neighbors IX-41 (upper left), IX-40 (upper), IX-3F (upper right), IX-1 (left), IX+1 (right), IX+3F (lower left), IX+40 (lower), and IX+41 (lower right).

Translating the process into simple terms, when a living cell is found, each of its neighbors (or potential neighbors) is marked. Since neighboring cells will have some neighbors in common, you can see what the result will be. When the process is complete, the total number of a cell's neighbors will be stored in a block of memory.

The neighbor test begins at line 670. The HL register pair is set back to the beginning of the screen, and the DE register pair is pointed to the block of cell-test memory. The DE register pair can be used this time because it is faster than the IX register, and because no indirect offsets will be needed for this process. B is set to 4 (C is already at 00, so this saves a small bit of time) for the total cell count, and the testing begins.

Recall that the original cell-test field of 1,024 bytes was set to −1; this makes testing faster. DEC A in line 710 will produce a zero if the number of neighbors is two (initial −1 in the test field +2 neighbors = 1; when 1 is decremented, it becomes zero). Two neighbors means that a living cell remains living, with no new cell being born. No change would then be made, and the routine is exited (via line 780). There the neighbor field is set back to −1, and the next test locations are obtained for HL and DE.

DEC A in line 730 will produce a zero only if the number of neighbors is exactly three. Three neighbors means that a cell must be given birth. That is done at line 750. If no zero is the result of line 730, the number of neighbors must have been greater than three or less than one (a cell-test value greater than +2 or less than zero). A space is then displayed (line 770).

A simple keyboard check is made at line 910. If no key is pressed in the 0 to 7 row, the entire Life process is repeated for another generation (line 930, looping back to line 420). If any key in the 0 to 7 row is pressed, the program produces invading hordes of new characters from the top of the screen. Exactly how many new characters are produced, and where they come from, is a function of the keyboard character pressed, the current state of the screen, and the present value in the R register. I'll leave any further explanations of this process; it's not random, but it's not easily predicted.

Program continues

```
6F78 57         00990          LD     D,A           ;D = KEY VALUE IN A
6F79 ED5F       01000          LD     A,R           ;GET CURRENT REFRESH VAL.
6F7B 5F         01010          LD     E,A           ;SAVE R IN E REGISTER
6F7C C5         01020          PUSH   BC            ;SAVE LIFE LOOP VALUE
6F7D 47         01030          LD     B,A           ;SAVE R IN B REGISTER
6F7E 10FE       01040          DJNZ   $             ;DELAY LOOP R TIMES
6F80 C1         01050          POP    BC            ;RESTORE LIFE LOOP VALUE
6F81 ED5F       01060          LD     A,R           ;GET ANOTHER R VALUE
6F83 83         01070          ADD    A,E           ;A = R PLUS E
6F84 82         01080          ADD    A,D           ;A = R PLUS E PLUS D
6F85 5F         01090          LD     E,A           ;E = R PLUS E PLUS D
6F86 7A         01100          LD     A,D           ;A = D (ORIGINAL KEY VAL)
6F87 163C       01110          LD     D,3CH         ;D = 3C (SCREEN HI BYTE)
6F89 C5         01120          PUSH   BC            ;SAVE LIFE LOOP VALUE
6F8A 4F         01130          LD     C,A           ;C = D (ORIGINAL KEY VAL)
6F8B 47         01140          LD     B,A           ;B = D (ORIGINAL KEY VAL)
6F8C ED5F       01150 SEVEN    LD     A,R           ;A = R (PART OF RANDOM)
6F8E E602       01160          AND    02            ;TEST FOR SET BIT
6F90 C2976F     01170          JP     NZ,EIGHT      ;IF NO BIT SET
6F93 3E2E       01180          LD     A,2EH         ;CREATE CHARACTER IF BIT
6F95 12         01190          LD     (DE),A        ;PUT CHAR. IN 3C+R+C+D
6F96 13         01200          INC    DE            ;INCREMENT SCREEN POS'N
6F97 10F3       01210 EIGHT    DJNZ   SEVEN         ;DO THIS D + D TIMES
6F99 C1         01220          POP    BC            ;RESTORE LIFE LOOP VALUE
6F9A C3196F     01230          JP     ONE           ;BACK FOR NEXT GENERATION
                01240 ;
                01250 ; ********************************************************
6F00            01260          END    6F00H         ;BEGIN AT LIFE ENTRY
00004 TOTAL ERRORS

EIGHT   6F97 01210    01170
FIVE    6F62 00770    00740
FOUR    6F54 00700    00830 00850
ONE     6F19 00420    00930 01230
SEVEN   6F8C 01150    01210
SIX     6F64 00780    00720 00760
THREE   6F41 00560    00470
TWO     6F23 00450    00590 00610
```

Changes

The character produced in this Game of Life is a simple period. If you want to change the character, load the program and press Reset (Break/Reset in disk systems). Then POKE 28510,X and POKE 28564,X. X is the ASCII value of the character you want to use; stars (42), lowercase o's (111) and small blocks (129) make an interesting display. To re-enter the program, type System, and respond to the *? prompt with /28424.

I want to express my appreciation to Philip K. Hooper for the kind permission to explain his algorithm in this column.

Program Listing 2. The Basic version of the Game of Life

```
10 REM * QUICK LIFE PROGRAM MACHINE CODE ENTERED THROUGH BASIC
20 REM * AS REQUESTED BY READERS, A CHECKSUM IS CALCULATED TO
30 REM * VERIFY THAT THE DATA LINES HAVE BEEN TYPED CORRECTLY.
40 REM * BY D. B. KITSZ, BASED ON AN ALGORITHM BY P. K. HOOPER
50 X = 28424 : Y = 0 : Z = 0 : REM * SET UP MEM, CHECK, & POKE
60 READ Z : Y = Y + Z : REM * READ VALUE AND START CHECKSUM
70 IF Z = 999 THEN 100 : REM * 999 IS END-OF-DATA INFORMATION
80 POKE X,Z : X = X + 1 : REM * POKE VALUE & ADVANCE MEM LOC'N
90 GOTO 60 : REM * LOOP BACK, READ NEXT DATA, CHECKSUM, & POKE
100 IF Y = 16396 THEN 150 : REM * EXAMINE VALIDITY OF CHECKSUM
110 PRINT "BAD CHECKSUM.  CORRECT DATA AND RUN AGAIN." : END
150 POKE 16526,8 : POKE 16527,111 : REM * LEVEL II START ADDR.
160 'DEFUSR0 = 28424 : REM * DEFINE DISC BASIC START ADDRESS
170 M=USR(0) : REM * LEVEL ii ENTRY -- USE M=USR0(0) FOR DISC
200 DATA243,205,201,1,33,0,112,17,1,112,1,255,3,54,255,237,176
210 DATA33,0,60,221,33,0,112,1,0,4,126,254,32,202,65,111,221,52
220 DATA191,221,52,192,221,52,193,221,52,255,221,52,1,221,52,63
230 DATA221,52,64,221,52,65,35,221,35,13,194,35,111,5,194,35
240 DATA111,33,0,60,17,0,112,6,4,26,61,202,100,111,61,194,98
250 DATA111,54,46,195,100,111,54,32,62,255,18,35,19,13,194,84
260 DATA111,5,194,84,111,58,16,56,167,202,25,111,87,237,95,95
270 DATA197,71,16,254,193,237,95,131,130,95,122,22,60,197,79,71
280 DATA237,95,230,2,194,151,111,62,46,18,19,16,243,193,195,25
290 DATA111,999 : REM -------->  CHECKSUM SHOULD BE 16396
```

How Was Your Week?

My computer also knows that summer's lazy days are here again. A week before the summer solstice, it decided to take a vacation—one not even close to home.

It all began innocently enough when graphics characters turned up unexpectedly in my text files. A vigorous cleaning of the edge-card contacts, thought I, would do the trick. Out came a cloth and some polish, and soon the connectors looked better than new.

I reloaded the damaged text file and began typing away, calling for an occasional word count. Nothing seemed amiss until I noticed that the word count was getting *smaller*, instead of larger. That's a sure sign of unwanted memory changes fooling the text editor into believing it's at the end of the material when it really isn't. These monthly Applications columns aren't done as paper drafts, they're composed right on the screen. Not being a hunt-n-peck typist, I had the uneasy sensation that many hundreds of words were about to vanish forever.

Right I was. Three years and two months my TRS-80 had been left on, with a few hours relaxation while being transported from place to place. Three years and two months it had lived without a buffered cable, without a cassette-load modification, without any improvement from the Tandy engineering lab. Now, exhausted with exertions even IBM couldn't meet, it was demanding a few electronic comforts.

Reluctantly I searched for the buffered cable I had ordered from Radio Shack, but never installed. Since the Shack provided no instructions with the cable (they would install it, they said, but wouldn't tell me how), I had to rifle through my bag of electronic tricks to figure it out. If you haven't

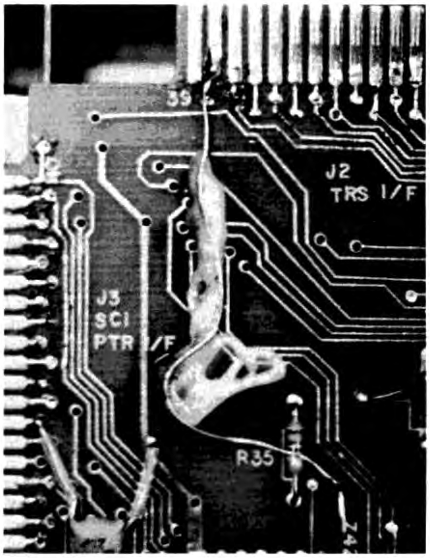


Photo 1. *Sending 5 volts to expansion interface pin #37.*

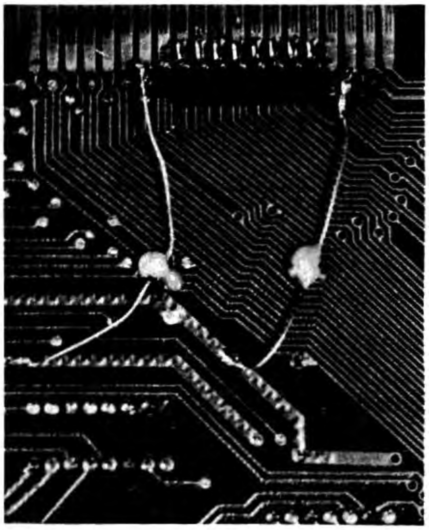


Photo 2. *Resistor packs soldered to data lines 0.7. Five volts and ground are shown routed to the packs.*

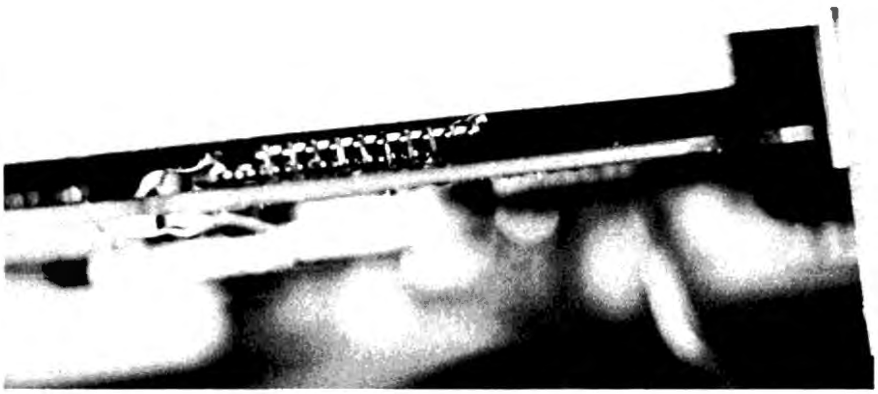


Photo 3. *There is little clearing between resistors on edge card and the expansion interface case.*

installed your buffered cable, here's how to do it . . .

Adding a Buffered Cable

● If you have a Level II machine, you will find that pins 37 and 39 of the edge-card connector (looking from the back, these are the top right-most two connections) are attached to each other. These may be connected either after manufacture (soldered with a piece of heavy wire), or may be original circuit board plating. A trace from pin 37 is probably cut with a deep gash. You must open the case to see these connections clearly.

If these two pins have not been connected, things are easier. Merely *cut* the trace leading from pin 37, which is the old 5-volt connection used on earlier Level I TRS-80's. During upgrade to Level II, some repair centers forgot to cut this trace and bridge it to pin 39. If pins 37 and 39 are already connected, separate them from each other.

● For the next step, open your Expansion Interface. Locate pins 37 and 39 on the expansion box edge connector. These will be on the bottom left of the *unopened* expansion box when viewed from the front, but just to be sure, the numbers have been silkscreened on the board. It is the side of the board on which the components are mounted. Cut loose connector 37 from connector 39, making a good gap between pin 37 and any traces around it.

Now find the large electrolytic capacitor near the edge connector. It is a tubular part about three-quarters of an inch long (usually encased in a blue or grey plastic sheath), and its polarity will be marked on one end. It may have a black arrow with a minus (-) sign pointing toward one end, simply a plus sign on one end, or both. Connect a wire from edge-card pin 37 to the *positive* (plus) end of this capacitor. The five volts feeding this capacitor will also supply power to the buffered cable.

● Now for the hard part. For this job you will need sixteen 470-ohm resistors. Use 1/4-watt or 1/8-watt resistors, which are small enough to fit into place. Half-watt resistors are too big: Radio Shack 231-1300 series resistors will be fine. Alternatively, you can use single-inline (SIP) resistor packs sold by Digi-Key in 470-ohm values.

Locate pins 18, 20, 22, 24, 26, 28, 30, and 32 on the expansion box edge connector. This is the solder side of the circuit board (opposite the side on which the components are mounted). You will be soldering two resistors to each of these pins, but you will have to do some careful planning before attempting it. First of all, there is very little headroom for the resistors when the case is reassembled. Secondly, if you don't solder to the very *back* of the connector pads (farthest from the edge of the board), there won't be room for the cable from the keyboard unit to slide in place.

To each of the eight edge-card pins above, solder one end of a 470-ohm resistor. Solder together all the loose ends of the eight 470-ohm resistors, and run a wire from that common point to the positive end of the nearby capacitor (the same one used to supply the buffered cable power). Next solder the same eight edge-connector pads to one end of each of the remaining eight 470-ohm resistors. Solder together all these loose ends, and run a wire from there to the negative end of the same nearby capacitor (see Fig. 1).

● The hardware work is complete. Power up the keyboard unit only, and verify that it works properly. Turn it off, and plug the "CPU" end of the buffered cable (not yet connected to the expansion interface) into the keyboard unit. Power up again and verify that everything is working.

Turn off the system again, and attach the buffered cable to the expansion box as well as the keyboard unit. Power up the keyboard unit, but not the expansion interface, and check that the system is okay. Finally, power up the expansion box, hold down the Break key, and press Reset. Test for your full system memory.

If any of the above steps don't work correctly, *turn the computer off immediately* and double-check your work. Since changes have been made which affect areas used by both the keyboard and expansion box power supplies, don't take

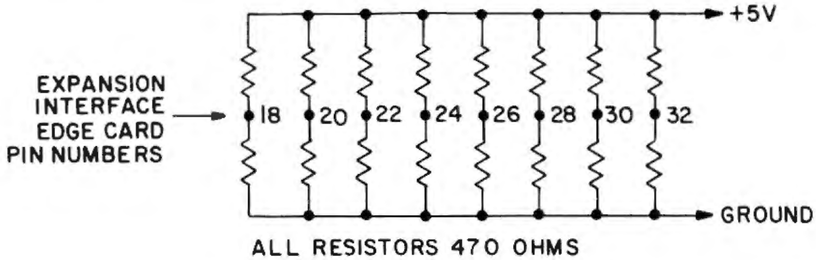
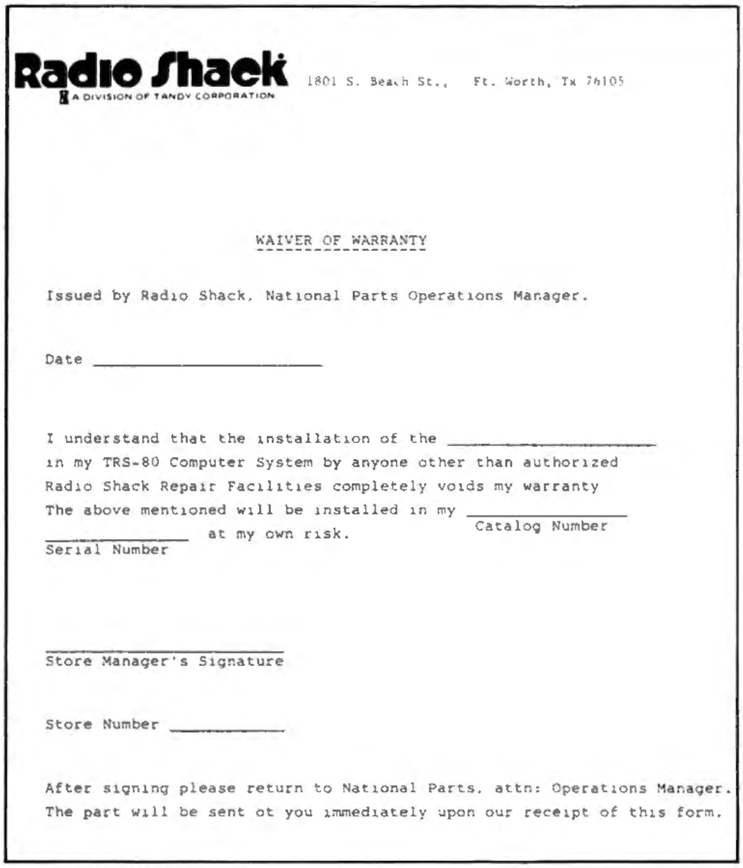


Fig. 1. Terminating the TRS-80 Data Lines



Radio Shack 1801 S. Beach St., Ft. Worth, Tx 76105
A DIVISION OF TANDY CORPORATION

WAIVER OF WARRANTY

Issued by Radio Shack, National Parts Operations Manager.

Date ____________________

I understand that the installation of the ____________________
in my TRS-80 Computer System by anyone other than authorized Radio Shack Repair Facilities completely voids my warranty
The above mentioned will be installed in my ________________ (Catalog Number)
____________ (Serial Number) at my own risk.

____
Store Manager's Signature

Store Number __________

After signing please return to National Parts, attn: Operations Manager.
The part will be sent ot you immediately upon our receipt of this form.

Fig. 2. Radio Shack Waiver

any chances.

Next month, in The EXclusive ORacle, I'll reveal how to install the "twisted pair" modification, Tandy's second attempt to make the expansion interface work reliably. Don't forget that the buffered cable, the twisted pair, and the cassette modifications are still available for free at Radio Shack. They will do the installation for you if you wish, or (especially if your unit has been modified by someone other than the Shack), you can fill out a "Waiver of Warranty" form to obtain the parts. You are entitled to only one set per serial number. If your Radio Shack dealer denies that such forms exist, show him Fig. 2.

Back to the Story

My problems, however, were far from over. In my enthusiasm to terminate the data lines (that's what the resistors are used for), I had made some of my peripherals sigh and throw up their electronic hands.

Termination is the electronic equivalent of leaving no loose ends. Imagine a rope about two or three feet long. When you hold one end tightly and swing your arm up and down, you can get the rope to form a regular up-and-down pattern, like a wave. Now increase the length of the rope to ten feet, and try to do the same thing. It gets considerably harder, and sometimes you just can't keep the pattern going continuously. Now increase the rope to 50 feet. You would need extraordinary strength and agility to keep the rope's entire length responding to your swings.

Now envision the opposite end of that same 50-foot rope tied to a wall. It takes little exertion to keep a regular pattern going, as long as the rope is supple and you haven't lost your strength after the previous exercises. That's the way it is with the computer's data lines. The longer they are, the more energy it takes to keep the signals moving cleanly. Gremlins such as capacitance between traces, resistance, dirt, and electromagnetic induction all influence those signals. They can be laggard, get noisy, overshoot their voltage destination, or pick up bits and pieces of other signals. This is called "signal degradation."

The termination resistors effectively tie the other end of the electronic rope, pulling the rope taut halfway between the positive voltage and ground from the power supply. (As I write this, my printer has just fried a resistor, filling the air with malodorousness. More to come...)

I had pulled the rope too taut, and signals from the peripheral equipment were unable to overcome its rigidness. The peripherals in question—the Exatron Stringy-Floppy and my own Memory Sidecar—contain no buffering between their internal memories and the outside world. Their memories have drive capabilities only about one-eighth that of the TRS-80's buffers, so I lightened the termination resistors to 470 ohms. I originally used 220 ohms, a pretty stiff demand on the memories in question, and also an undesirably heavy drain on the Expansion Interface's power supply. Quick math shows that two 220-ohm resistors in series is 440 ohms. Eight 440-ohm resistors in parallel is 55 ohms. At five volts, 55 ohms represents 1/11 of an amp—too much demand to add to the power supply already driving 32K of memory, an RS-232 board, and a buffered cable!

Using 470-ohm resistors lightened the demand to only 42 milliamps. The power supply was happy, and the peripheral devices relaxed. My system worked once again.

But Not Completely

Next morning it was time to sit down and prepare this column. I finished a few hundred more words, brought the cursor back to the start, switched on the printer, and started the printing process. No reaction from the printer. "Aha," thought I, "press the print-select button." I pressed it again. My computer was locked up, waiting for a response from the printer. I had, of course, neglected to save the text before printing it.

With a handy screwdriver, I shorted two pins on the Expansion Interface printer port edge card, pin 21 to pin 20. If you have an LPRINT hangup, you can short those pins as well. This sent an impostor acknowledge signal back to the computer while I held down the Break key. Text recovered, but printer dead.

All you who believe in optical isolators, hearken: They too can self-destruct. In this case, the optical isolator in my Centronics 102 printer (vintage 1975 or so) had broken down, allowing 110 volts to flow throughout the circuitry. Disassembling the printer strained an elbow (it weighs 143 pounds), finding the fried parts strained my brain. After two days and only two integrated circuits (anybody ever use a 7406?), the printer was running again. Relief.

I powered up the system, and rebooted my lone disk drive. Nothing. What? Everything was on. The disk select indicator was lit, at least as long as it was supposed to be. Even the stepper motor moved to track zero. But nothing else. No spinning disk. A quick call to *80 Microcomputing* got my deadline extended.

Now, this disk drive (a Tandon Magnetics 40-track number) has long sounded like a freight train, but I paid it no mind. I dug first into the Expansion Interface, turned my meter to its 10-volt scale, and stuck my probes into it—the black one to a ground, the red one to the "Motor On" signal output going to the drive cable. Reset. The voltage dipped to zero, then up again when the drive select indicator went out. That revealed that the correct signal was going to the drive, since the schematic showed that the "Motor On" signal is active low (zero volts).

I pulled the cable from the drive, and tested it at the far end. Sure enough, the signal was there too. With the cover off the disk drive and the cable reattached, I used the schematic to follow the motor signal through each integrated circuit in its path, and found the signal present throughout. All that remained was to examine the motor itself and the power leading to it.

I looked for evidence of damage. There it was. The motor's power circuit trace had burned clean off the board—and yet the fuse remained intact! I gently moved each motor by hand, discovering resistance in both the servo motor (which spins the disk) and the stepping motor (which seeks the track). Metal and plastic parts were rubbing against each other with absolutely no evidence of lubrication, not even light silicone film. Turning to the Tandon Magnetics service manual, I found this disturbing information: minimum parts order, $50. Drive repair, $95. Module repair, $80. Circuit board repair, $40. Realignment and test, $50 extra. A footnote stated "excludes damage." I winced. Tandon had convinced me to do the work myself.

The burned power supply trace was repaired first by bridging it with a heavy wire; I then tested for damaged parts. All seemed okay (diodes showed current flowing only in one direction, capacitors and transformer showed no shorts), and applying power and a computer re-boot indicated everything was normal—except for the freight-train sound.

The service manual had nothing to say about lubrication, so I had to decide whether to lubricate any parts of the drive. The considerations, which might be yours if faced with a similar electromechanical dilemma, were:

- Jargonists might say something like, "the adjacent parts were contacting constrictively and impacting negatively upon their respective operations, with untoward sonic emanations." That is, they rubbed and squeaked. Few other disk drives are quite as loud.
- There was no evidence of lubrication on any part of the drive.
- No added lubrication could be allowed to come into contact with the recording head, pressure pad, or any other parts which touch disks.
- Fine silicone lubricants are generally used for delicate electromechanical systems because such lubricants don't spread, bleed, run, or attract dust and grit.

I decided to lubricate with a Teflon-based product called "Break-Free CLP." One drop on the upper and lower stepper guide rails, one on the guide band, one on the pressure-pad guide cam, and one on the disk bearing. After allowing it to penetrate, I wiped the surfaces with a photo chamois and restored the power. All the noises were gone, and my disks load more quickly with none of the "re-seeking" I had been experiencing. Once more I felt relieved. The deadline might be met.

Curses

I was, naturally, optimistic. The next morning, disk operation acted like my '64 Valiant on a January morning here in Vermont.

No signals were emanating at all from the disk drive cable, and none could be found at the edge connector. Inexplicably, my Percom Data Separator had failed. That gem, which had increased my disk's reliability tenfold, had itself given up the task. As many of *80*'s readers have discovered, the Percom board is a true all-American value at only $30, and during the three (happy) months spent in my expansion box, it paid for itself in recovered disk data and mental collapse ten times over.

But now the Separator was not working. And if there are any black marks for Percom, it is that no schematic or service information is provided with the board. Not having the patience to draw a schematic from the circuit board itself, I sent it back. (A full report on Percom's service policies and quality in a later column.)

The disk controller chip was still in good shape, so I reinserted it in its socket. Once again, the disk system was in operation. . . for fifteen minutes, until the weakest link in the chain broke for the second time.

The cable to the disk units is driven by a type of integrated circuit which is capable of producing output only under certain conditions. It depends on a "pull-up" resistor external to itself; otherwise, there is no apparent output. Because the final transistor inside the IC is left unconnected, these are called "open collector" circuits, and are used where long or noisy lines must be powered. The circuit's pull-up resistors are found in disk drive zero, and are usually 150-ohm resistors in a single dual-inline (SIP) integrated-circuit style package.

These circuits are sometimes electronically fragile. In the keyboard unit, open-collector ICs are used for the keyboard connections, and in the expansion interface they are found driving the disk cable. I have seen more than a half-dozen keyboard failures and three disk failures due to breakdown in these circuits.

Evidence of breakdown or marginal operation is: continual re-seeking in the disk system; no motion by the stepping motor at re-boot; consistent writing errors; frequent formatting and/or copy failure; and banging of the disk drive assembly against either end of its path of travel. In the keyboard, the failure will show up as: incorrect or unexpected characters; repeating characters (where there's no software to do it); or multiple characters when only two keys are pressed. The only thing to do is replace the integrated circuit. That I did, and the system was (finally) running normally.

Oh yes. The frying resistor in my printer was the result of a paper clip falling into the circuit board. The damage was light but the smell was heavy. One more thing. The first of these disasters happened Monday; today is Thursday. And Fred's broken TRS-80 just arrived in the mail. . . ■

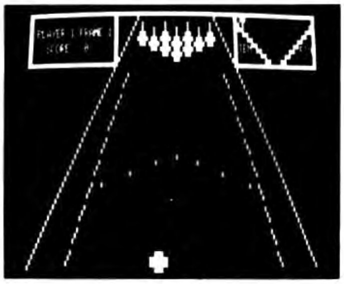
ACORN ARCADE!

FOR TRS-80*

PRESENTS ACTION GAMES BY JOHN ALLEN

NEW!

TENPINS

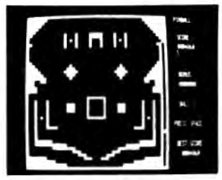


By John Allen

John Allen works his machine language magic again! With the realistic action and sounds made famous by his bestselling "PINBALL" and "BASKETBALL" arcade games, TENPINS now brings all the thrills of championship bowling to your TRS-80* model I or III.

From one to four players participate, and the program automatically senses the skill of each player. Beginners can simply position the ball and "roll" it while more skilled players can vary the force, roll a curve, and cause it to spin as it heads for the pins. All this—plus 3-D graphics and sound effects—adds up to a realistic and thoroughly challenging bowling game.

Scoring, pinsetting and ball return are all automatic, and you can view the entire scoresheet at any time. TENPINS even has factors to simulate the imperfections found in bowling lanes, adding even more realism—and sometimes frustration—to the game. It's almost as hard to roll a perfect 300 score in TENPINS as it is in real bowling, but it sure is fun trying! Available for only $14.95 for 16K tape version, $20.95 for 32K disk.



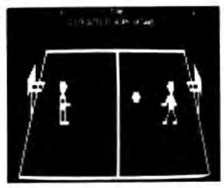
PINBALL

By John Allen

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—but be prepared for a lot of practice if you ever hope to master the fastest speed.

Can you beat your friends' scores? Will you avoid the dreaded "Bermuda Square?" Get PINBALL today and find out. Available for model I and III, $14.95 on tape or $20.95 on disk.



BASKETBALL

By John Allen

Fast machine language action game, with sound, from the author of the acclaimed "PINBALL"!

Try to keep up with the action and outscore your opponent in five minutes of one-on-one basketball. 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. Compete against a friend or your computer.

It's all there but the cheers—for model I and III on tape for only $14.95, disk version for $20.95.

Available now from Acorn dealers everywhere

AUTHORS & DEALERS WANTED

Acorn
Software Products, Inc. ✓34
(202) 544-4259
634 North Carolina Avenue, S.E., Washington, D.C. 20003

ALL ACORN ENTERTAINMENT SOFTWARE IS NOW AVAILABLE FOR TRS-80* MODELS I & III !

*TRS-80 is a trademark of Tandy Corp.

✓See List of Advertisers on page 322

80 NEWS

edited by Bert Latamore

". . . the companies involved are too busy designing,writing and selling games to worry about what portion of the total market they have. . ."

Micro Games Sell Like Hot Cakes, Suppliers Scramble to Keep Up

International Resource Development (IRD) Norwalk, CT, estimates the software games market at $80 to $90 million retail last year and it is growing rapidly The market's suppliers often cannot define its size in numbers, but describe it as "very attractive" or "unlimited."

This market is undoubtedly one of the fastest growing segments in the microcomputer field. In fact, it is changing so fast the companies involved are too busy designing, writing and selling games to worry about what portion of the total market they have or how big it will be in five years.

Apparently IRD is the only firm to have researched this market. They recently completed a study of the related electronic hand-held games market, which they predict will eventually be largely replaced by software games as microcomputers become more common in the home.

According to an estimate supplied by IRD President Kenneth G. Bosonworth, retail sales of microcomputer software last year totaled $700 million, of which microcomputer games contributed $80 to $90 million.

Put in those terms, games hardly look like a significant force in the software field. However, games reach a much wider audience than any other kind of program. Professional people play games on their lunch hours, their children play educational games at school. People with no use for an accounting program buy games, and people who would never think of using word processing also play games.

Dianne Asher, marketing director for Automated Simulations, Cupertino, CA, which specializes in fantasy/adventure software games, estimates that 90 percent of the TRS-80s in service see a game at least once or twice a month. While no one knows how many '80s have been sold, Asher expects at least 100,000 will be in operation by the end of the year.

She estimates the games program market may become as large as half the total software market.

Bob Bosen, head of Bosen Electronics, Provo, UT, estimates there are 150,000 to 200,000 TRS-80 owners now, and that at least 30 percent of these are buying a game every one to three months, no matter what they use their computers for most of the time.

Other producers have less of an idea of the size of the market they are supplying. But they do know that if they produce a decent game—one with some interest and no major programming flaws—they can sell it as fast as they can manufacture copies.

The games market gives plenty of other indications that it is large and growing fast, perhaps too fast for any one to count it accurately.

One of the more recent indications has

Big Five Wants to Change Your Micro into an Arcade

Most companies which make arcade games do not consider them their top products. Bill Hogue, president of Big Five Software, Van Nuys, CA, takes a different view. His firm produces *only* arcade games, and he says quality is just as important in them as in adventure games.

A good arcade game has a high scorer's table, can count scores far beyond what anyone can achieve in the game, has a flashy beginning that does unusual things and attracts attention, and offers arcade-type sound without freezing the game action.

Basically, he said, he tries for the feel of a true arcade, a thing which is hard to define but which he feels most microcomputer arcade games do not have.

Good action graphics are the primary consideration, he said. His own favorite game is Galaxy Invasion because of the graphics.

"It's almost as much fun to watch somebody else play it as it is to play it yourself," he said.

Hogue said his original games did not have sound because he didn't like the way sound routines froze the game graphics every time they went into action. However, he said, he has since developed new routines which don't effect the graphics at all.

He said he plans to go further in his next games, using voice synthesis so the game can actually talk back. He is investigating two alternative methods—a true voice synthesizer, which would allow a large vocabulary, or a more limited voice process using the computer's cassette port.

"Even with the cassette port I could come up with some real witty phrases," he said.

Interestingly, Hogue says he is not a good arcade game player himself, although he obviously likes them. He also enjoys adventure games, but he sees advantages to the arcade game for the buyer.

"Once you've solved an adventure, that's it," he said. "But an arcade game always offers the challenge of getting a better score."■

Bert Latamore
80 Microcomputing *staff*

been the entry of some traditional game companies into the software market. Avalon Hill Game Co., Baltimore, MD, a firm known for its high quality, complex strategy games, started putting out software games about a year ago, supporting three machines including the TRS-80.

Avalon Hill entered the software market believing that many of its existing customers own or will soon own microcomputers and will want to buy games for them. So far they have depended mostly on their own magazine, *The General*, for their publicity, and have marketed their software through the same toy stores that handle their board games. Recently, however, they have started to reach a new market, people who may never have played their board games but own computers.

"We believe the future is in computerization"

Jackson Dott, marketing coordinator for Avalon Hill's microcomputer games, said they are still a small producer in this field and their main effort remains in board-based games. However, they have been pleased with the market response to their games and this summer have brought out several new, more ambitious games. They are marketing fantasy role-playing software, sports games, business games and war games and are considering combination games in which the computer takes care of the paperwork but the player still uses a mapboard and playing pieces.

"We believe the future is in computerization," Dott said.

Other very large companies agree with him. Mattel and Atari have both recently upgraded their video game modules for microcomputers, thereby entering the market from the other end.

The other large game companies are not yet talking about their plans or lack of them. However, Dana Lombardi, editor of Game Merchandising Magazine, Clifton, VA, said several established game companies are working on microcomputer games. The market is big enough to attract them, and they expect microcomputers to be the new home utility of the 1980's.

The market also demonstrates its strength in the kinds of companies it supports. Not only are there firms which specialize in software games, many are specialists in one particular kind of game or support only one or two machines. Big Five, for instance, is a prosperous, small firm which produces only TRS-80 arcade games.

Even companies which make other kinds of software find games to be an important part of their total product line. William Denman, president of Med Systems, Chaple Hill, NC, said, "We make most of our money on games. I consider that to be a really growing market."

Med Systems makes all kinds of games, although perhaps their best known are Labyrinth and Deathmaze. Denman said they have seen a large expansion in the market in the last few months. All kinds of games are selling well, although high speed graphics adventure games sell best. Besides the TRS-80 their games support the Apple and they are considering the Atari.

Damon Crumb, assistant to the president at Instant Software, Peterborough, NH, said his firm is the largest manufacturer of software for microcomputers in all areas including games, with one of the largest inventory lines available.

He said the ideal microcomputer game would be a fantasy game for Tandy's Color Computer with high resolution graphics, interactive capabilities and an adventure format. That, he said, would be an automatic hit for at least six months.

"Dana Lombardi. . . said established game companies are working on microcomputer games."

"We're all working on it," he said, referring to the various game companies. "It's a race to see who gets it out first."■

Bert Latamore
80 Microcomputing *staff*

Designer Took Five Years To Create Complex Game

Space Raiders may be the ultimate Star Trek-type game and, possibly, the most complex microcomputer game ever attempted. It combines elements of flight simulation, including celestial navigation, tactical level war gaming and arcade gaming in a single package. All of this is packed into a 16K program for the Model I.

Producing a program of that complexity takes time, and Bob Bosen took it. Constructing the game has been a five-year hobby, and while it did not make him become a programmer, Bosen said it did greatly influence the kind of programmer he became.

"I've had microcomputers for quite a few years," he said. "The first game I ever played was a Star Trek game, but I wasn't very happy with it."

Bosen was one of those kids—he was still in school then—who has the need to change things he doesn't like. He decided to improve on that Star Trek game.

"I started studying celestial navigation and spherical trigonometry to learn what the stars should look like as you fly past them," he said. After a great deal of research he developed a basic algorithm to duplicate the effects of traveling through the stars on the computer screen, but he found Basic just didn't give him enough speed.

He tried the effect in compiled Fortran, but it was still too slow. Finally he found a way to speed it up enough to be successful.

The game includes a full floating-point math system and a full trigonometric navigation package that allows the player to navigate among the stars and change the starfield patterns as he moves. It also includes an animation package and a combat simulation module which allows you to shoot down Klingon patrol ships.

Bosen runs his own firm (Bosen Electronics) in his spare time and works full time as vice president of research and development for Computers International, Provo, UT.

So far Space Raiders is his only game, although he has ideas for others. It is also his favorite game. Writing it was a large challenge, but he said his real reason for doing it was more personal.

"We all like to live out a fantasy we can't do in real life," he said. "There's a little bit of Dr. Frankenstein, a little bit of Neil Armstrong in all of us."■

Bert Latamore
80 Microcomputing *staff*

Micros Invading Sports Arena

Sports is one of the few areas that has been slow taking advantage of the microcomputer's capabilities. A major reason for this is that software for sports applications has been unavailable until recently.

At least two companies are now pioneering sports-based software and hardware development. Sports Stats U.S., Cupertino, CA, is currently marketing its SS100 Basketball Scoring System; and the Trinity Group, Newtown Square, PA, offers Procap, a program to predict football scores.

SS100

The SS100 system uses the TRS-80 Model III and consists of the necessary software and a specially-developed keyboard. It uses the Line Printer IV and Scripsit. The program takes up 6,000 bytes and is written in assembly language.

According to Jim McCabe, a Sports Stats spokesman, the system was developed over a two and one-half-year period, and was done "more as a hobby" than a business venture. At this writing, Sports Stats has sold three systems, all in the NBA. The Milwaukee Bucks are the only team to have used the SS100 during the past season, though the Phoenix Suns and the Golden State Warriors have both installed the system in their arenas.

". . . coaches now get the first-half statistics at the beginning of half time. . ."

Bucks Publicity Director Bill King is very pleased with the SS100's performance. "The software is limitless. . . and I'm very happy with the hardware," he said.

An early version of the system used a Model I, which King claims had problems with microwave interference. However, the added shielding of the Model III has solved that problem.

McCabe noted the same problem with the Model I but said, "The Model III has been extremely reliable for us."

King said the only maintenance done so far has been a realignment of the disk drive.

The SS100 keeps score, tallies individual and team statistics and provides a compilation in an instant. The keyboard is set up so the user merely punches in the number of field goals, fouls, etc. as they occur. This eliminates the need for pencils and paper. King said the coaches now get the first-half statistics at the beginning of halftime rather than at the end.

This statistic-keeping efficiency does not give the Bucks a strategic advantage over other teams, according to King. In fact, the Bucks provide visiting teams with their data. King said the system's value is as a coaching tool and not as a game-winner.

The few things the system does not do

Frank Corr—Making the Ultimate Maze

"I don't want anyone to solve in eight hours what took me three months to write," Deathmaze 5000 author Frank Corr admits. Computer game aficionados would have to agree he got his wish.

Corr, an 18-year-old freshman at MIT, didn't start out writing adventure games. His first commercial game program, Rat's Revenge, was simply a maze in which the player had to find the cheese. This Basic program provided the graphics for Deathmaze.

When Rat's Revenge was finished, Corr got an offer from Mike Denman of Med Systems Software, Chapel Hill, NC, to market the program. Corr said he hadn't planned to sell the game—he just wanted to play it. But he told Denman, "Fine, you can market it, but I want to learn machine language first."

Corr wrote an English research paper on machine language, and Denman sold Rat's Revenge. Then, last June, when Corr was looking for a job and couldn't find one, he got the idea to use the graphics from Rat's Revenge in an adventure game.

This was a challenge to Corr because he "never played an adventure game until I was halfway through Deathmaze." Deathmaze was written in machine language. Corr said the most difficult parts were the maze drawing routines.

It is the graphics that separate Corr's games from the competition. The maze graphics were an inspiration to Corr to make Deathmaze as difficult as it is. Yet Corr said he gets a lot of fan mail; apparently Deathmaze entusiasts appreciate the challenge of the maze. As one fan put it, "Words are boring." Corr plans to provide more clues (the main complaint he receives) in upcoming games.

Labyrinth obviously grew out of Deathmaze, but Corr gave Denman much of the credit for writing the program. Corr is proud, though, of his contributions to Labyrinth, and, as one could have guessed, those contributions intensify the frustration level of the game. Corr called his teleportation squares "a cute little stunt," which transport the player to another part of the maze with no indication of it being done. He is also fond of the man in the black cloak.

Asylum was Corr's latest project, though Denman also worked on it. "Everything is twice as good as Deathmaze," Corr said.

He attributes much of the improvement to a routine he discovered in January that allows graphics to be stored as data. This game accepts simple sentences, provides hints and has 135 bytes more of maze, yet the program is 22 bytes shorter than Deathmaze, Corr said.

Anyone looking for a tougher challenge, though, will be disappointed. "Asylum is by far an easier game," Corr said.

What else is in store? Corr said he plans at least one more adventure game for the TRS-80. The graphics will be vastly improved, with octagonal rooms, and use a space station or similar setting, Corr said.

After that, Corr wants to write real-time arcade-style games based on the Atari micro, whose graphics capabilities Corr prefers. He intends to stay with Med Systems, however.

Corr offered some interesting observations on the future of microcomputers. He sees technology improving greatly but expects prices to remain the same (not going down as some people predict). He also expects nearly every household to have at least one or two micros to perform a variety of functions. But perhaps the most revealing remark of all was, "Most people will be home playing games and not watching TV, which is good!" ■

by Mike Nadeau
80 Microcomputing *staff*

(home versus road and bench versus starters statistics) are due to a lack of software, which King hopes to have soon. When the system is not being used for statistics, King uses it for writing press releases, or compiling the season ticketholder mailing list.

McCabe believes that by next season about half the NBA teams will have the SS100. Considering the time saved, the $6,000 to $7,000 price tag for the SS100 seems like a bargain to any NBA team.

Sports Stats has several other projects in the works. Football and baseball statistics-keeping systems are in the development stage, according to McCabe. The San Francisco Giants and the San Francisco 49ers are cooperating with Sports Stats on these projects.

> *"(The demand has) been fantastic. . . We hope to have 150 to 200 systems installed within three years."*

McCabe said they also hope to have a system for use in colleges soon that would come with several keyboards and different software for a variety of sports. With a change of a disk and a plug-in keyboard, this TRS-80 based system could change from a baseball scoring system to one for track.

McCabe is impressed with the demand for sports-based systems. "It's been fantastic," he said. "We hope to have 150 to 200 systems installed within three years."

Procap

The Trinity Group calls its Procap program "a unique marriage of statistics and probability theory with the ever-increasing power and capabilities of personal computers."

The TRS-80-based program computes a football team's offensive rating by comparing its score with the relative defensive strengths of opponents played to date. It computes a team's defensive rating by comparing the team's score-limiting performance against the relative offensive strengths of opponents played to date.

A team's overall power rating is calculated by subtracting the defensive rating from the offensive rating. Average teams will have a power rating close to zero—the higher the rating, the stronger the team. Matching teams on any given week involves four separate computation methods. These computations give two possible point totals for each team which are arranged as best and worst case scores for the favorite. An average case is listed as the probable score.

Terry Wright, a Trinity Group spokesman, claims an 85 percent success rate for the program. He said the program's author, Joe Mascio, won $2,500 in bets using Procap last year. Wright did offer one disclaimer, though. "No matter how good the (statistics/probability) program is, it can't predict upsets."

Though the program can obviously be used for gambling purposes, that was not the main intent of the developers. Wright said, "We want people to have some fun making predictions."

Wright believes the reason the software companies are just now bringing out sports-based programs is the trend towards a merging of the microcomputer as a personal and professional tool. "The hobbyist probably viewed the computer as a pastime. When the use of the product broadened, interests overlapped," he said.

With the overlapping of interests and users looking for more varied applications of the micro, sports programs began to emerge, Wright said. One can imagine a football fan/computer hobbyist watching a Sunday game. His eyes wander from the action to his TRS-80. Suddenly, it dawns on him that there may be more to the '80 than adventure games and home-budgeting. He ignores the rest of the game to work on a football statistics program. This probably isn't how Procap was written, but it reflects the image Wright suggests.

The Trinity Group has faith in the growth of the sports fan market for software. Wright estimates that 3.5 million homes will have microcomputers by 1984. He said his company hopes to get two to three percent of the market share this year with Procap—roughly 5,000 to 10,000 users.

"We want to capitalize on the fantastic growth of the personal computer market," he said.

Wright wouldn't comment on plans for other sports software from the Trinity Group except to say they are in the early stages of a golf program. He said other projects were "too preliminary" to discuss.

Track-Scoring Program

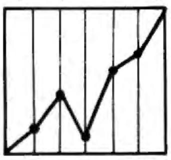
Another interesting sports program is not for sale; in fact, it has been used only once, but with great success. Calvin Perry, head track coach at Grambling University, Grambling, LA, borrowed a TRS-80 from the school's computer center, had a friend write the program he needed and scored

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 $100
MODEL II $140

+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 $250
MODEL II $300

+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 $250
MODEL II $300

+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 $250
MODEL II $300

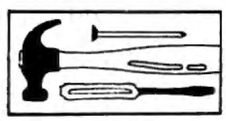
+GENERAL LEDGER

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 $250
MODEL II $300

+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 $250
MODEL II $300

SPECIAL! All of the above: MODEL I/III $1250 MODEL II $1500

+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
1-800-528-6050 EXT. 1562
(In Arizona, 1-800-352-0458 Ext. 1562)

+PLUS

PLUS COMPUTER TECHNOLOGY, INC. ✓131
POST OFFICE BOX 1152
ANGLETON, TEXAS 77515 (713) 849-1108

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 322

the South Western Athletic Conference Indoor Track Meet in record time.

The program, though it doesn't actually tabulate scores, "saves an enormous amount of time—about 10 minutes for each event," Perry said.

At the end of each race, an operator just keys in the individual times. Each athlete (there were 260 at the meet) is a number to the computer, and the '80 tracks each one throughout the meet according to heats and place.

When a coach wants to know the status of a particular athlete, that athlete's number is keyed in and his activities to date are listed, Perry said.

This system "knocked out three or four people copying information," Perry said.

Each coach got a printout of the race results immediately after its conclusion, and the day's final results were given to the coaches and the media soon after the end of the meet. That kind of speed is almost unheard of in a meet that large.

The program's author, Donald Lee of Tulsa, OK, could not be reached for comment, but Perry had much praise for his work. "The guy is really a wiz," he said. ■

by Michael Nadeau
80 Microcomputing *staff*

Small Ohio Paper Goes Electronic

A small afternoon newspaper in Ohio may become the world's first publisher of a local electronic newspaper this fall.

The Tiffin, OH, *Advertising-Tribune* is going ahead with plans to put out an electronic news service for its subscribers using a TRS-80 Model II and Tandy's Communications Multiplexer as the host.

The paper's publisher, Kaj Spencer, said the electronic service will supplement the 11,500 circulation evening daily, giving people news that needs to be updated more often than once a day or things that the small paper doesn't always have room to publish, such as stock market reports.

"We believe some form of electronic newspaper will emerge as an important medium in the near future," he said. "We want to be involved and have a hand in developing the future landscape."

Spencer said the concept is so new no one really knows yet what they will put on it. He said they had not received the equipment from Radio Shack at the time of this writing and they would have to play with it some before they decided just what form the electronic paper would take.

Spencer said, however, that he envisioned it as a specialized service designed for the needs of some segments of the *Advertiser-Tribune*'s readership. It would carry news updates, sports scores and similar items. The service is designed to supplement the printed paper. He does not see it as replacing the newspaper.

The service would be interactive, and Spencer said they plan to publish a daily menu in the printed paper.

The *Advertising-Tribune* is a member of the Buckner News Alliance, a group of six small papers. While Spencer feels the electronic news experiment involves only his paper and he could not speak for the rest of the group, he did not rule out the possibility that other papers in the group would put out electronic editions if the Ohio experiment proves successful.

Spencer said they had not decided on charges for the new service, but he thinks they will offer unlimited access for a monthly fee. They may have a second tier of more expensive information based on what it will cost them to buy that information.

Organizing a project of this sort is complex, and they expect to spend about three months arranging things, working out technical problems and refining the process before "going to press." However, Spencer said, he hopes to have the service out in the fall.

Spencer said they picked the Radio Shack machines for practical reasons. He had been watching developments in electronic data base services for some time, but most videotex systems are either too large or too expensive for the *Advertiser-Tribune's* limited resources and needs. The Tandy equipment, including the multiplexer originally developed for the experimental "Project Green Thumb", seems to him to offer a more practical alternative for a small newspaper.

The electronic paper would be sent out over telephone lines to users with microcomputers or Radio Shack Videotex terminals. Radio Shack sells software allowing all its microcomputers to access the system and a program to allow "dumb terminals" to access it. ■

by Bert Latamore
80 Microcomputing *staff*

Model IIs Talking to IBMs Thanks to New Programs

Tandy/Radio Shack, Fort Worth, TX, is trying to beat IBM to the punch by making the TRS-80 Model II compatible with IBM equipment before the industry giant brings out its own micros.

This move has been backed with the announcement of the development of two special software packages, the Reformatter, which allows transfer of files between the Model II and IBM-format disks; and the Bisynchronous Communications package, which allows the Model II to act as an IBM-compatible terminal for both on-line and batch processing.

These moves come in the face of persistant rumors that IBM will soon be producing its own microcomputers. IBM refuses to comment publically on these rumors, which assert that the computer giant has one or, possibly, two micros in production in a plant in its General Systems Division's Boca Raton, FL, plant. It has announced, however, that it is producing five-inch floppy disks which would not be used in its mainframe machines. This announcement only adds fire to these speculations.

Speaking of new Radio Shack programs, Tandy President John Roach said, "These new software packages give significant new utility to the TRS-80 Model II computer by entering it into the IBM world. The Model II has rapidly become the leading unit-sales 'very-small-business computer', and each new capability gives it an even broader base."

The Reformatter allows the Model II to read from, write to and work with IBM-compatible memory disks. According to a Tandy spokesman, programs and data can now be prepared on the Model II for use on IBM equipment, and most programs and data can now be prepared on IBM equipment for use on the Model II and most programs and data prepared for IBM computers can be used by the Model

✓See List of Advertisers on page 322

II. This, he said, essentially makes the Model II the equivalent of an IBM work station at a much lower cost, while, of course, still handling all normal Model II functions.

Reformatter automatically reorganizes data from the TRSDOS format to IBM 3741 single-density format along with translating character code from ASCII to EBCDIC. Disks are automatically initialized to the IBM format with all necessary fields, eliminating the need for specially pre-formatted disks.

Nine specific functions are provided to create compatibility between the Radio Shack and IBM disks:

- *Initialize* structures a disk in IBM 3740 format.
- *Transfer* permits direct data transfers between TRSDOS and IBM formats or between TRSDOS and an active data set.
- *Display* presents the contents of existing records of an active IBM data set.
- *Directory* displays the directory contents of either an IBM or TRSDOS disk.
- *Edit* modifies the contents of existing records in the IBM format or an active data set.
- *Dump* displays the contents of specific sectors of an IBM disk.
- *Identify* displays and alters the volume of an IBM disk.
- *Redefine* modifies parameters of an existing data set on an IBM disk.
- *Delete* renames an active data set and resets it to inactive status.

Reformatter requires a Model II with 64K memory and two or more disk drives. It includes a non-exclusive paid-up license for use on multiple CPUs, to allow the purchaser to copy the program as required for his or her own personal or business use.

The two Bisynchronous Communications packages, expected to be available in the third quarter of 1981, allow binary synchronous communications in full compliance with IBM standards. They include both the software necessary to implement communications protocols, code conversions and a hardware conversion to modify the Model II A serial port to meet IBM standards.

The on-line standard Binary Synchronous Communications packages, expected to be available in the third quarter of 1981, allow binary synchronous communications in full compliance with IBM standards. They include both the software necessary to implement communications protocols, code conversions and a hardware conversion to modify the Model II A serial port to meet IBM standards.

The on-line standard Binary Synchronous Communications 3270 program allows the Model II to communicate with IBM Systems 360/370 and 30-series central processing units or any other machines equipped with remote BSC-3270 communications capabilities.

With it, the Model II can act as a 3270 display station, giving screen formatting, polling responses, data-link control, time-out control and cyclic redundancy check. It makes the Model II fully interactive with IBM remote programs including TSO, CICS, VM/CMS, IMS, etc., a Tandy spokesman said.

The batch standard Binary Synchronous Communications 3780 program allows the Model II to work as an IBM-compatible Remote Job Entry (RJE) terminal. It allows users to select from IBM 2770, 2780, 3780 or 3741 terminal protocols for communications with IBM 360/370 or 30-series host systems, with DEC PDP-11 and VAX-11 host systems or with other machines equipped for binary synchronous communications.

Both packages require a Model II with 64K memory. Both are sold with non-exclusive paid-up licenses to allow the purchaser to reproduce as many copies as he/she needs for personal or business use.

Synchronous communications through the computer's A serial port operate at a nominal 9600 baud data rate, or at data rates up to 19,200, depending on the length and type of communications connection used. The programs operate on either half- or full-duplex communications facilities.

"It is our goal," Roach said, "to continue to make computers and communications more affordable for users. With our full services—leasing, maintenance and training—coupled with these new software packages, big computer users can now take advantage of the savings offered by microcomputers."■

by Bert Latamore
80 Microcomputer *staff*

Program Tabulates Polls, Adds Up Senate Victory

A psychologist—computer hobbyist has designed a poll tabulation program sparking an incorporated software business. Mark Stolzberg, whose program contributed to the US Senate victory of a little-known county supervisor, will soon market his software for broad use in marketing, research and the media.

A clinical psychologist and statistician, he designed his program with help from Dr. Hubert Howe, a professional programmer and music professor. It can handle up to 30 questions and tabulates data such as income, sex, religion and age.

"I think it will help at a grass roots level," Stolzberg said of the poll's political significance. A poll done by Stolzberg for Alfonse Damato convinced Damato he could successfully run for Senate. Before the sample, "he would sometimes call a news conference and nobody would show up," Stolzberg says. The poll helped Damato define his constituency.

Stolzberg used volunteers for the poll-taking, plus several consultants. Stolzberg Research Inc., formerly M, M, & S Software, is run out of his home in Stonybrook, NY, where he is often assisted by Howe, who operates Howe Software.

Designed it Himself

Stolzberg designed the poll himself, and Howe did the programming. Stolzberg thinks it could be modified for a number of uses, such as consumer/market research, education and advertising.

For instance, the software could easily compile test scores, and Stolzberg may use it for an evaluation of a math program in the Boston schools. It would assess differing student skills according to items answered—an "items analysis."

"Before the sample 'he would sometimes call a news conference and nobody would show up . . .' "

There's a possibility the software could tabulate data from psychology experiments, but only in the preliminary stages. The program is not sophisticated enough to handle the multi-variant analysis of most experiments, Stolzberg says.

He and Dr. Alan Cullen are now using the program to tabulate a review of new eye charts by the New York State Optometric Association. He put the association's ten criteria in place of the demographic variables, with each chart rated pass/fail.

Stolzberg is no stranger to statistical analysis. When he found Radio Shack's statistical package (RS 4000) inadequate,

he designed his own, which Creative Computing promptly bought for distribution. It handles three T-tests, co-variants, correlation, linear correlation, mean variants and standardized deviation. Stolzberg sold about 75 copies himself, mostly to universities and hospitals.

With Howe's help he has also put together an inventory program for a manufacturer that allows the user to call up a list of available products, products on order, total cost and total labor. A billing program for a chemical distribution company is currently being tested; it also carries accounts receivable, shipping, sales reports and marketing.

Started with Calculator

Stolzberg's computer hobby sprung from an interest in calculator programming.

"I'm a self-taught nut on the calculator," he says.

Then Howe introduced him to the TRS-80. Stolzberg needed a cheap way to store data for a dissertation, and Howe suggested the TRS-80. Now Stolzberg owns a Model I, a Model III and a Line Printer VI.

His polling program can be used on either the Model I, II or III, and the Maine Radio Information Network has been using it successfully on the Model II. Maine Radio Purchased it for their quarterly polls. Projects have included general issue polls and a nuclear referendum survey. Business Manager Malcom Leary says it's made their tabulations quicker, easier and more accurate.

Howe thinks Stolzberg's program is unique because he wrote it with the educational and professional researcher in mind. For instance, it can handle up to nine subcategories for a single question and can be programmed to categorize open-ended questions. Yet it is also accessible to the novice.

"There's good documentation," Howe says, so the beginner could use the program, but "I don't think the average person would know how to interpret the statistics," he added. For this reason the program is suited to the professional.

Though the software has yet to be marketed, Stolzberg has received several inquiries and plans to start advertising in the fall. Meanwhile his consulting work and other software projects keep him occupied.

He's also pursuing his doctorate in educational psychology at Hofstra University. This leaves about 20 hours a week for the software company.■

by Betty Thayer
80 Microcomputing *staff*

Writer Turns to Model I, Has Love-Hate Relationship

When prize-winning novelist Ernest Hebert started his new book, *A Little More Than Kin*, he decided to trade in his typewriter for a TRS-80 Model I and Scripsit. A year later the novel is finished, but, while the word processor speeded up the writing, it introduced problems of its own.

In the end, Hebert sold his machinery, replacing it with a much more limited IBM memory typewriter. Today, facing the prospect of starting a new novel, he isn't sure whether he wants to stay with the typewriter or buy a new microcomputer.

Hebert's word processing saga began when the 15,000 circulation daily newspaper he works for, *The Keene* (NH) *Sentinel*, went to a word processing system. At the time he was an editor, and he worked on the system eight hours a day.

"The possibilities of word processing are so incredible," he said. "You can't go back and be satisfied with what you were doing before."

At home, Hebert found he could not compose on a typewriter after using word processing. Fortunately, at about this time he received a grant from the National Foundation for the Arts. His first inclination was to use it to let him write full time, but the *Sentinel* would not give him a leave of absence, and he didn't have enough money to quit. So, he said, he "blew the money on equipment."

Besides refurnishing his study, he bought the Model I with two disk drives, 32K memory and an inexpensive printer. Scripsit, he discovered, had more editing features than the system at his newspaper offered.

"I had a wonderful machine," he said. "It just released me."

Hebert said word processing is "made for someone like me who is constantly rewriting, constantly revising, who is never satisfied." His writing method is to type a page, edit it, then retype. "I might do that anywhere from between 5 to 15 times before I got the thing where I wanted," he said. "Then I would go on to the next page."

The word processor eliminated the need for retyping. As a result, while *Dogs of March*, Hebert's last novel, took him four years to write, *A Little More Than Kin* took him one.

Unfortunately, the story has an unhappy side. First the printer broke down. The local Radio Shack manager loaned him another while his was sent out for repair. Then he started having disk problems.

"I was always having little things go wrong with this machine—things I couldn't fix," he said. "I was always calling Doug (Scribner, the local RS store manager) to come and bail me out, which he would do all the time and never charge me."

Being so totally and constantly dependent made Hebert increasingly nervous. On the other hand, he had no interest in making the machine into a hobby just so he could learn to repair it. While Hebert is fascinated with technology, the TRS-80 was solely a word processor. He never even bought any other software for it, much less learned to program.

"The possibilities of word processing are so incredible, you can't go back and be satisfied with what you were doing. . ."

"The anxiety was increasing," he said, "and I said to myself, 'what if this guy dies or gets sick of me? I'll be out of luck. . .' So while I could still get some money for it I sold it—for the book price, which means I took about a $1,300 bath."

The IBM, a surprisingly thin-looking black machine, was the replacement. It is, he says, only half as good as the microcomputer, with no screen and much more limited editing abilities. And it prints much more slowly.

"The value of this machine is I can put it on the IBM service contract for only about $250 a year and they come and fix it or replace parts if something goes wrong with it," he said.

Hebert hasn't given up on micros yet, however. He said he is considering another try at word processing, with either a TRS-80 Model III or some competitive system. He would like to have both the typewriter and the word processor, but he's not really anxious to go back to the typewriter.■

by Bert Latamore
80 Microcomputing staff

Seminars Cover Color Computer Chip

Motorola's Technical Training Group, headquartered in Phoenix, AZ, is offering a series of two and four-day seminars on their 6809 microprocessor chip (see Calendar). The 6809 chip is the brother of the powerful 6809E MPU which lies at the heart of Tandy's popular new Color Computer.

Though 50,000 Color Computers are in user's hands, there is a paucity of detailed information about the 6809E chip's instruction set, internal register structure and wide variety of addressing modes. The situation has resulted in a shortage of quality assembly language software for Tandy's new machine.

The Motorola courses, while not aimed directly at Color Computer owners, are designed to get those conversant with assembly language programming and microprocessor fundamentals familiar with the 6809E.

Two versions of the course are offered: a two day "update," for those who are acquainted with Motorola's 6800 family of chips and a more detailed four day treatment that covers everything from number systems to 6809 linking loaders and relocatable macro assemblers.

The four day course (MTT-11) is a fast-paced tour-de-force that includes two lab sessions within its 32-hour instruction period. Students not familiar with the jargon and habits of microprocessors will have trouble keeping up. However, those with a working knowledge of other microprocessors (such as the Z-80) will be provided with the information they need to write assembly programs that exploit the power of Motorola's 6809E.

That power is derived from the wide variety of addressing modes available in the 6809E (nine) and an instruction set that allows all registers to be pushed to the stack with one command (PSHU/PSHS), A and B accumulators combined for 16-bit operation (LDD) and the pointer register loaded with an effective address in the indexed addressing mode (LEAX, Y,S,U).

In an interview with *80 Microcomputing*, course instructor Ray Doskocil outlined the design philosophy Motorola adopted for the 6809: "Everything that connects to this MPU looks like memory as far as the chip is concerned. Thus, we are dealing with memory-mapped I/O. The advantage of this is that it is easy to hack chip functions up and make the chip perform."

Doskocil also indicated that the 6809 was the end of the line for eight-bit technology as far as Motorola was concerned. When queried about the chip's lack of popularity in the personal computer market, Doskocil cited Motorola's lack of aggressiveness in this segment of the marketplace as the reason.

Though not specifically about the E version of the 6809, MTT-11 covers all aspects of 6809E operation and outlines the differences between the 6809 and its E version. These differences are few, the most obvious being the lack of internal clock circuitry in the 6809E. While the 6809 requires only an external crystal for its system clock, the E version requires external clock circuitry.

Other differences include three control pins unique to the 6809E. The TSC (Tri-State Control) pin allows the address lines and data bus to be put in a high impedance state. The LIC (Last Instruction Cycle) pin is used for diagnostic purposes and signals that the next processor cycle will be an op code fetch. The Busy pin provides a common bus lockout signal for the MPU.

Course content is strictly Motorola, and when hardware add-ons like RAMs, ROMs and PIAs (Peripheral Interface Adapters) are discussed, all design examples feature Motorola chips. From a System design point of view this approach is logical, but Color Computer hardware hackers will undoubtedly shop around when modification time comes.

While not suitable for the rank beginner, Motorola's 6809 course will provide experienced assembly language software authors with everything they'll need to know in order to write code that will make the Color Computer perform.

Course costs range from $250 to $430 and discounts are available based on course location and number of students involved. Additional information may be obtained from: Mr. Ron Bishop, Motorola Technical Training Headquarters, P.O. Box 2953, Phoenix, AZ 85062. Tel: 602-962-2345.

Course	Dates	Location
MTT-11	July 21-24	Phoenix, AZ
Four day course	Sept 29-Oct 2	Phoenix, AZ
MTT-3	July 7-8	Los Angeles, CA
Two day course	July 9-10	San Diego, CA
	Aug 18-19	San Jose, CA
	Aug 20-21	Portland, OR
	Sept 29-30	River Edge, NJ
	Oct 1-2	Rochester, NY
	Oct 20-21	Houston, TX
	Oct 22-23	Dallas, TX
	Nov 10-11	San Jose, CA
	Nov 12-13	Denver, CO

by Chris Brown
80 Microcomputing *staff*

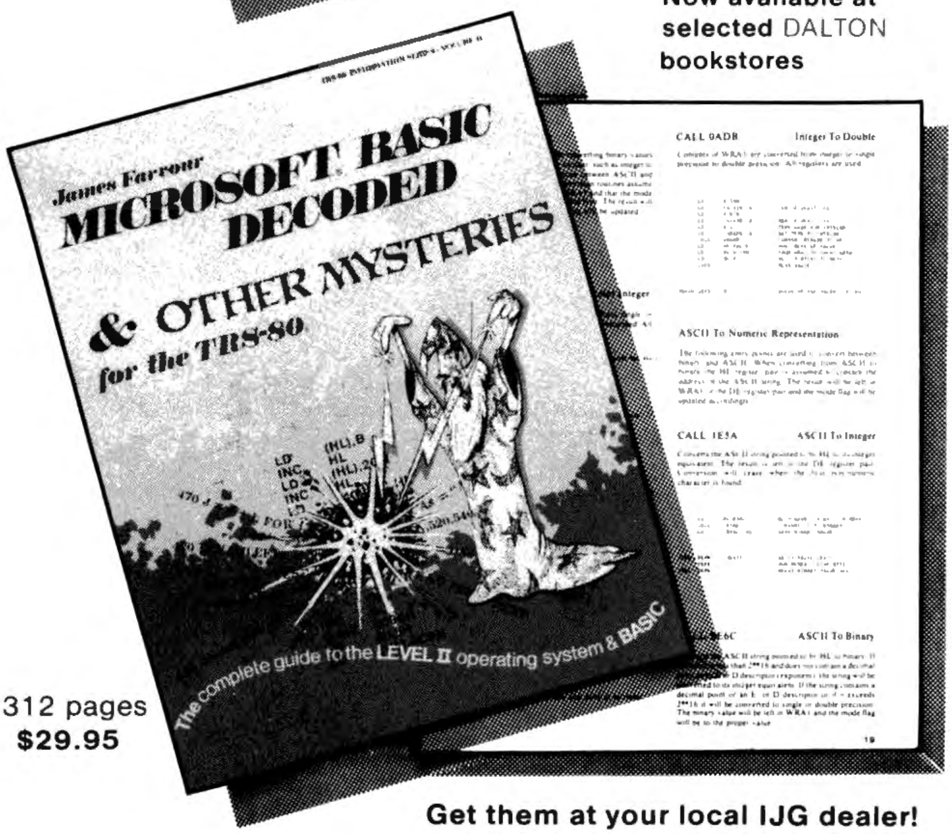
See List of Advertisers on page 322

'.. this is not only a worthwhile book but a great book. My advice is to get it and USE it!' - **William Barden Jr.**

OTHER MYSTERIES

132 pages
$22.50

Now available at selected DALTON **bookstores**



312 pages
$29.95

Get them at your local IJG dealer!

'It has twelve thousand one-liners in it, and every one is great!' - **Dennis Kitsz.**

Phone orders (714) 946-5805

TRS-80 Disk and Other Mysteries is the definitive **fixit** book for disk users. Writen by Harvard Pennington it has more than 130 pages of easy to read, entertaining and immensely useful information - find out how to recover disk files, the layout of information on the disk, memory maps, problem solutions . . . the list goes on!

Many readers have saved days of work by recovering files that had been unreadable. Now in its fourth updated printing, **TRS-80 Disk** and Other Mysteries has been getting rave reviews in several magazines. Yours for only **$22.50** (plus $1.50 shipping, CA residents please add $1.35 sales tax).

Microsoft BASIC Decoded and Other Mysteries is the complete guide to your Level II ROMs. With over seven thousand lines of comments and 6 additional chapters packed with information, this is easily the biggest, and best, book about the Level II ROMs available.

Exploit the full power of Microsoft BASIC, with the aid of more than 300 pages of tested examples, understandable explanations and detailed comments. Now available in a revised second printing, only **$29.95** (plus $2.00 shipping, CA residents add $1.80 sales tax).

IJG Computer Services, ✓37
1260 West Foothill Blvd.,
Upland, CA 91786

Please send me a copy of **TRS-80 Disk** and Other Mysteries for $22.50 plus $1.50 shipping. ☐

Please send me a copy of **Microsoft BASIC Decoded** and Other Mysteries for $29.95 plus $2.00 shipping. ☐

Please send me a copy of **The Custom TRS-80** and Other Mysteries for $29.95 plus $2.00 shipping. ☐

OK, send me all three **IJG** books for $82.40 plus $5.00 shipping. ☐

Overseas add $8.00 per book airmail.

Name ....

Address ....

City ....

State .... Zip ....

Charge my:

☐ MasterCard Interbank Code ....

☐ Visa Expiration Date ....

Card No. ....

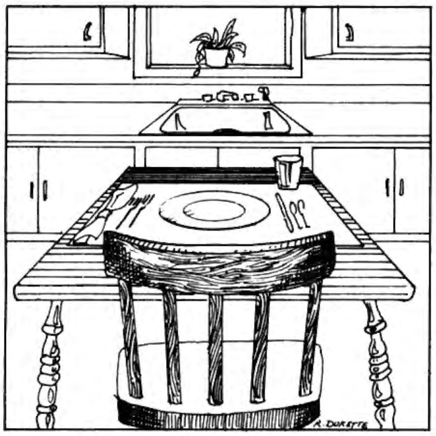
☐ Check enclosed for ....

☐ Ship COD ($2.00 extra)

Signature ....

News From KITCHEN TABLE SOFTWARE, INC.

by David Busch



That team of unknown geniuses at Kitchen Table Software, Inc., makers of DROSSDOS 1.1, at last report were flushed with success. Even though only 34 disks of the DOS have been sold, Kitchen Table has distributed more than 120,000 copies of the mandatory correction ZAPs at $4.95 each. Following this new influx of capital, the company has redoubled its efforts at writing and marketing innovative programs for the TRS-80. As the only reviewer willing to keep DROSSDOS 1.1 long enough to check out all its functions, I was selected by Kitchen Table to look at advance copies of these soon-to-be released gems. Here are some capsule commentaries on each.

Ready for 120,000 Baud?

Are you ready for 120,000 baud? This machine language program, Superfastlode, allows cassette-based systems to load Basic or System programs using the fast forward setting on the Radio Shack CTR-41 or CTR-80 recorders.

The program is supplied on cassette. Using the System command, the user first loads a short bootstrap program, which automatically slows down the TRS-80 cassette input speed to 10 baud. Kitchen Table says that this clever step ensures a good load of the speedup program which follows. The main program loads from tape in just under 17 minutes. Superfastlode resides in the top two bytes of memory, and automatically protects itself. To activate the program, the user merely hits /Enter, and CLOADs other programs in the normal manner, but using fast forward instead of play. Superfastlode has a built-in routine which overloads and burns out Radio Shack's XRX cassette loading modification, so there are no hardware modifications to worry about.

In benchmark testing, we were able to load a 16K program in under a tenth of a second. A friend brought over a program he had written containing more than 700 data lines which normally takes up 36K of memory. Superfastlode was able to suck up this prodigious code in less than a quarter of a second! A quick check revealed that this system is more than 99 percent reliable, an enviable record. For you mathematical types, that translates into only 360 bad bytes in a 36K program.

The only drawback I have been able to pin down is the lack of a correspondingly fast CSAVE function with Superfastlode. In fact, once the program has been run, it is impossible to CSAVE at all. Kitchen Table helpfully notes in the documentation that it will be necessary to purchase a second TRS-80 for program dumps if the user wishes to get the maximum benefit from Superfastlode.

Out of Sorts

This machine language bubble sort may be incorporated into any Level II or Disk Basic program and accessed through the USR routine. Numeric values or strings to be sorted should be first loaded by the user's program into an appropriate array. When the USR call is invoked, Out of Sorts quickly checks the array to make sure that the list is not already in the proper order. If so, the program punishes the operator by locking up the keyboard. Otherwise, a painstaking bubble sort is commenced.

For the neophyte, the bubble sort is the most popular, and therefore the least efficient, of all sorting routines. Even though Out of Sorts is a machine language program, it still required almost 11 minutes to sort a list of ten names. We felt compelled to ask Kitchen Table just how this program was developed. A company spokesperson explained that the programmer responsible for Out of Sorts had only finished reading the Level I manual the week before the program was written. Originally coded in Basic, Out of Sorts was later compiled because Kitchen Table felt that they could charge more for a 14K machine language program.

I can't recommend this program (though at $4.95 it is a bargain), but it may be of some interest to those connoisseurs of the various permutations of bubble sorts.

Fast Food Restaurant Accounting Package

Kitchen Table reports that this series of programs was prepared in response to literally hundreds of requests from micro users for a good, complete general ledger/accounts-payable/accounts-receivable package suitable for use by fast food franchises. This version is one of the most complete I've seen.

Ffrap has many special features which lend themselves to fast food operations. For example, the modules are remarkably friendly and commendably interactive, which makes them especially suitable for use by the typical 17-year-old fast food store assistant manager. The general ledger program guides the user through the required input at the end of each day. Special categories have been set up, with entries for overcooked burgers, over-rings and cashbox shortages. Accounts payable features allow the franchise owner to keep track of costs by category, so that all-beef-patties can be kept separate from the special-sauce-onion-cheese-pickles-ketchup-on-a-sesame-seed-bun expenses.

The package issues many reports. One tracks usage of straws on a month-by-month, year-to-date and comparable-period-last-year basis. Trial balances may be generated hourly, and there is a provision for using any TRS-80 as a stand-alone, point-of-service sales register. In this mode, the CRT is placed behind the Expansion Interface, and the cavity provided for the RS-232 board used as a cashbox.

Chess Handicapper

This is a fiendishly complex program, which aids in predicting the winner of any given chess match. The user inputs data about opponents of an upcoming match, such as results of last encounter, favorite gambits, most frequent opening moves, strength of end game, etc. The computer uses this data with a copyrighted algorithm to generate a random integer larger than zero, but less than three. The resulting figure will indicate whether player number one or player number two is destined to win.

This program also has a routine that will accept the status of a chessboard near the end of any given game and predict the winner with 50 percent accuracy no later than two moves before the losing player resigns. This program impressed me, a non-chess player. I used it prior to several matches I recently had with Sargon II. I lost every single game, and Chess Handicapper predicted this correctly more than 80 percent of the time! When used in the end-game mode, for example, I had two pawns and a king to Sargon II's King, Queen, two bishops, two knights, two rooks and six pawns. My downfall was forecast even *before* I was aware of the impending doom.

Program Packer

This machine language program selects the most interesting portion of any Basic program and compresses it in three or four multi-statement lines. The rest of the program is deleted entirely. It will reduce a 16K program down to less than 500 bytes. It's useful for those of you who invariable write long, useless programs. You can save disk space, but still have something to show for your programming efforts.

Flochart

This program takes any Basic program and outputs a concise flowchart to a printer or graphics plotter. It's very useful for college computer science courses, as well as for spotting bugs in an unusable program. Blocks of code that are never called are highlighted. Variables that are used for different purposes by different modules are listed in alphabetical order. Standard flowchart notation is used.

Paper, Rock, Scissors Adventure

This popular old-time game has a new twist. It's a fast, machine-language program with incredible graphics and sound which held my attention for nearly two weeks. In fact, I was nearly fired from my job when I didn't report to work for five days in a row.

I can't reveal the details of this game without spoiling the fun, but I can tell you that the adventure involves traveling through a mystic land in the guise of either a sheet of paper, a rock or magical scissors. You may encounter other paper, rocks or scissors along the way, each with unique attributes that give them invincible powers over the other players.

The object is to accumulate points by killing as many other players as possible. This game should be rated R for violence alone because of the possible demises which include smothering, crushing or hacking to pieces. Good, clean, adult fun.

Dice Simulator

One of the nice things about having a microcomputer is being able to computerize time-consuming tasks that once could be done manually. Dice Simulator is a clever application of this concept. This program realistically simulates the actual rolling of electronic dice. Without getting too technical, this is accomplished by having the computer select two random numbers with values between one and six. The total of these two number pairs is considered to be the total of the two dice.

> Paper, Rock, Scissors "should be rated R for violence alone . . ."

Dice Simulator has to be one of the most useful Basic programs I have ever run across. I sat down and compiled a list of literally hundreds of day to day applications which can be speeded up by computerized dice rolling. For example, do you find that a good game of Monopoly can squander several hours of your time? Dice Simulator can trim 10 percent of the time ordinarily needed to roll the dice.

One nice touch was a printout feature that allows the users to take advantage of the computer's unimpeachable randomness even if a game is going to be played outside the computer room. Dice Simulator will, on command, roll any number of die pairs and output the resulting totals to a printer. This printout can be stored and taken along to the game site. When each player's turn to roll comes up, he or she can simply check the listing for the next number supplied by the computer and use that instead.

Program Evaluator

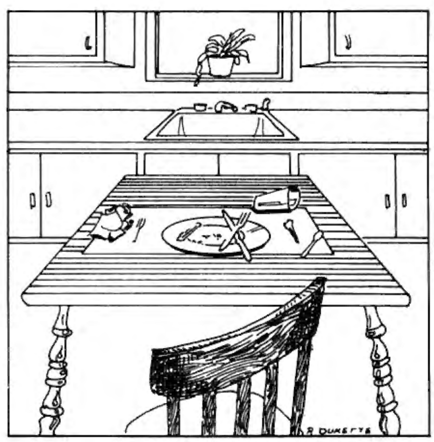
This program saves the user time by letting him or her know instantly whether or not a newly purchased program is any good. Instead of having to run the program serveral times and judge individually, Program Evaluator can do this chore automatically.

Unfortunately, this version works only with Disk Basic programs. The code to be evaluated is first loaded into the TRS-80 and then saved to disk in noncompressed (ASCII) format. Then Evaluator is loaded, run, and the file name of the test program entered. Each line is examined, and one of the following messages delivered:

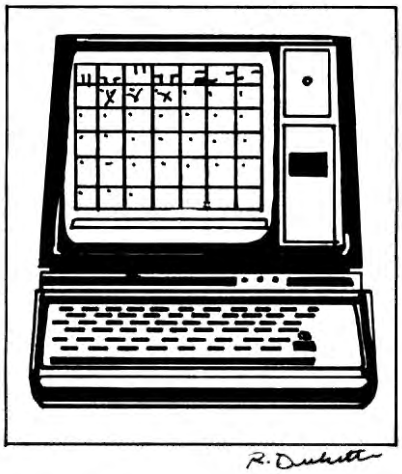
- Excellent program. You really got your money's worth. (This message also displayed if program bears Kitchen Table copyright notice.)
- Worthwhile program and should be very usable.
- Several bugs spotted. You may be able to fix them yourself.
- You got gypped. This program does absolutely nothing.
- This program is worse than useless—it may damage neighboring disk sectors.
- Please contact Kitchen Table Software. If you paid money for this program, we have a sorting routine you'll be thrilled with.

Far Eastern Hardware

Please don't write to me asking when these software items will become available. I only review programs, I don't market the stuff. Keep reading *80 Microcomputing* for constant updates on the doings of the Kitchen Table Software crew. I'm eagerly awaiting receipt of their latest developments, Word Winder (a word processing program), Space Shuttle Simulator and within a few months, their most ambitious hardware offering, the TLS-8E. I don't have any details yet, but I have been told that the TLS-8E is a 100 percent TRS-80 Model I compatible microcomputer, designed and built in Sri Lanka. ■



80 CALENDAR



August

Aug. 28–29 the **International Microcomputer Fine Arts Festival** will combine artists using or interested in using microcomputers with programmers and other technical people who have done work applicable to the needs of artists at the Teela-Wooket Camp, Roxbury, VT.

Worcester Polytechnical Institute will run **a series of two-day seminars for dataprocessing professionals through March 1982** at their Worcester, MA, campus and other Boston area locations. Interested parties can contact WPI for information.

September

Wentworth Institute of Technology, Boston, MA, will sponsor **a course designed to lead to a BA in Computer Science on Saturdays starting in September.** Students with at least 60 credit hours of higher education including prerequisites will be able to earn the degree which qualifies them as computer programmers.

October

Oct. 2 and 3 A two-day conference on **"Classroom Applications of Computers"** will be conducted at Independence High School, San Jose, CA, by Computer-Using Educators and Santa Clara Valley Mathematics Assn. The conference will cover mathematics, science, business, music, special education, language arts and administrative applications of computers at levels ranging from pre-school through college. Information is available from Computer-Using Educators, Independence High School, 1776 Educational Park Dr., San Jose, CA 95133.

Oct. 24 and 25 The second annual **New Jersey Microcomputer Show and Fleamarket** is scheduled at the Holiday Inn North Convention Center, Newark, NJ. The show will feature 75 commercial exhibitors and more than 100 outdoor fleamarket vendors and user-group meetings for TRS-80s. Admission for both the show and fleamarket is $5; for the fleamarket only, $3. It is sponsored by Kengore Corp., 3001 Rte. 27, Franklin Park, NJ 08823.

Oct. 26–Nov. 4 Virginia Polytechnic Institute and State University will conduct three workshops in October and November. **Digital Electronics for Automation and Instrumentation** will be Oct. 26–28; **Microcomputer Design Interfacing and Programming using Z-80/8085/8080** will be Oct. 29–31; and **Scientific Instrument Automation, Interfacing and Programming using the TRS-80 microcomputer** will be Nov. 2–4. Information is available from Dr. Lindy Leffel, CEC, Virginia Tech, Blacksburg, VA 24061.

✓ See List of Advertisers on page 322

NEW PRODUCTS

Edited by Janet Fiderio

Volume I Issued

Volume I of the *Encyclopedia for the TRS-80*, a library of useful information for TRS-80 users, has been issued by Wayne Green Books.

Volume I includes articles on graphics, producing custom invoices and methods of developing timing loops and a simulation game which makes you president of the US.

The Volume costs $19.95 for hardcover and $10.95 softbound, or you can subscribe to the 10 volume set for $164 hardcover or $83 softbound, from Wayne Green Books, Attn: Pauline Johnstone, Rte. 101 and Elm St., Peterborough, NH 03458.

Reader Service ✓ 325

Inventory Your Music

Record and Tape Collection Inventory catalogues, manipulates and searches your music collection, doing alphabetical sorts and alphabetical or numerical searches.

The program is for a 16K Level II Model I or III with cassette tape storage. It costs $15; the more sophisticated version for 32 or 48K disk machines costs $25, from Ozark Mountain Software, PO Box 2945, Fayetteville, AR 72701.

Reader Service ✓ 345

Alarm Signals Programming Errors

The Audio Alarm Device for the TRS-80 Model III plugs into the cassette port and gives a two-second beep when it detects errors in entering programs or data.

It comes with a cable and cassette port plug and is powered by a nine-volt battery. It requires one additional line of coding in your Basic program.

The cost is $36.70 (Ohio residents add $2.10 tax) from RSI, 231 Green St., Dayton, OH 45402.

Reader Service ✓ 170

Program Analyzes National Football League Matchups

Procap analyzes National Football League contests to predict the outcome of weekly games, probable scores and point spreads, evaluate team offensive and defensive strengths, review division standings, and team and division power ratings and scores of previous games.

Procap is available from Trinity Group, 829 Malin Road, Newtown Square, PA 19073 for $49.95.

Reader Service ✓ 162

Low Cost Modem

Radio Shack is now marketing a low cost modem to connect the TRS-80 and similar computers directly to standard single-line telephones.

The TRS-80 Modem I gives computer users an inexpensive way of enjoying the many over-the-phone services now available. The modem is FCC approved and Bell 103 answer/originate compatible. Connection to standard RS-232 connectors may require a connecting cable. Model I systems using the cassette port require a cable and cassette software.

The TRS-80 Modem I is available for $149 at Radio Shack stores, computer centers and participating dealers.

Reader Service ✓ 179

The TRS-80 Modem I

Micro Software Directory

The International Microcomputer Software Directory is a reference source of microcomputer software for all applications and systems.

The Directory, which has 5000 programs and 15,000 entries, has three main sections: system classification, for the user limited to a particular system; subject classification, for the user who can use any system or who has not yet purchased a system; and software house classification, for the user wishing to buy from a particular software house.

System-specific directories for the TRS-80 and other micros are available for $14.95. The complete directory is available for $28.95. For further information contact Imprint Software, US, 420 South Howes, Ft. Collins, CO 80521.

Reader Service ✓ 176

New Business Analysis Programs

Capital Investment analysis is a program which makes capital spending decisions such as: automatic depreciation, calculation of IRR and NPV, optional inflation adjustments, and the ability to handle multiple assets in one analysis.

The Internal Rate of Return/Present Value Calculator program calculates typical profitability measures from a cash flow and the IRR user-specified reinvestment rate, and plots net present value vs. discount rate.

The Depreciation Calculator program offers eight straight-line and accelerated methods. Options allow switching from declining balance to straight line at the optimal point.

A Level II Model III with 16 or 32K is required; a printer is optional. The cost for the Capital Investment program is $39.95; the second two programs combined are $19.95, or $49.95 for all three, from Jenson Associates, 59 Oak Drive, Ancient Oak, Macungie, PA 18062.

Reader Service ✓ 180

Assembly Developer Loads High

Intasm is an assembly language development system which loads into high memory and accepts assembly language source instructions entered as Basic-format lines, so both the interpreter and source are in memory at the same time.

This is very convenient for development and debugging, and Intasm accepts Z-80 register and flag display, display of next instruction, break-pointing and single-stepping.

Intasm costs $20 on machine language cassette, for 16K, 32K and 48K Model I systems, from Singular Systems, 810 Stratford, Sidney, OH 45365.

Reader Service ✓ 332

Utility Automates Land Sales

The settlement package for Models I and II for real estate oriented law firms and title companies does amortization schedules, maintains information for 300 settlements, performs all necessary computations and pro rations, outputs RESPA statements, does disbursements and all other functions needed to automate a real estate closing department.

It eliminates forgotten steps or charges by taking the user through the process in an orderly manner.

It is available for $3,000 from Scorpion Systems, 120 N. Court St., Federick, MD 21701.

Reader Service ✓ 331

A Project Management Support System

Target, marketed by the System Research Services organization (SRS), is a complete project management support system.

Based on the critical path method, Target can schedule and allocate real or hypothetical resources, as well as develop starting/ending dates for interdependent project activities, identify the critical path, compute time slack and tabulate resource requirements by type and cost over the life of the project.

Target is available from System Research Services, 1800 Old Meadow Road, Suite 1413, Mclean, VA 22102.

Reader Service ✓ 175

The Lynx Modem

Modem Compatible with Both Model I and III

The new Lynx telephone modem is compatible with both the Model I and Model III and replaces the original Lynx which was usable with the Model I only.

The Lynx has auto-dial and auto-answer, originate/answer, programmable word length, parity, active Clear and Break keys and control, programmable I/O ports and either keyboard dialing or stored number selection operation.

Lynx is $299.95 from ESI Lynx, 123 Locust St., Lancaster, PA 17602.

Reader Service ✓ 173

QuicKeys Label Keyboard

QuicKeys are clear, pressure-sensitive labels which mount on the face of key caps and include more than 425 commonly used words, commands and symbols, including those used by Level II and Disk Basic, popular word processing and communications programs commands.

Different command categories are color-coded so more than one label can be put on a key.

QuicKeys cost $9.95, from Pairr Products, PO Box 817, Montrose, CA 91020.

Reader Service ✓ 336

Computer Telephone Directory

The Online Computer Telephone Directory (OLCTD) is a quarterly newsletter reporting on microcomputer electronic message systems.

The newsletter contains information on accessing bulletin boards such as ABBS 4.0, CBBS, NetWorks, Peoples Message System, Connection-80, FORUM 80, NetWorks and remote Northstar. OLCTD also contains a directory of free-access system telephone numbers.

Subscription rates are $9.95 per year, $15.95 for two years, from On-Line Computer Telephone Directory, P.O. Box 10005, Kansas City, MO 64111.

Reader Service ✓ 178

TRS-80 Versions of Chemistry Programs

J & S Software has released TRS-80 versions of 15 high school and junior college chemistry programs originally written for the Apple computer.

The programs are available on cassette for the Model I Level II and Model III microcomputers for $19.50 for one, $75 for a set of six and $150 for all 15 from J & S Software, 140 Reid Ave., Port Washington, NY 11050.

Reader Service ✓ 171

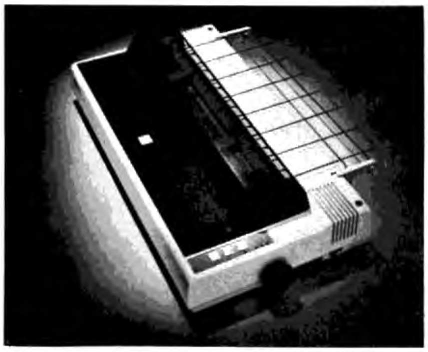
MX-100 Dot Matrix Printer

The MX-100 is a 136-column dot matrix printer with matrices from 9 x 9 to 18 x 18 providing letter-quality print in 12 different character weights and sizes.

It accepts 15.5 inch wide paper and can print 233 columns of information in condensed print mode. It prints bidirectionally at 80 CPS. It comes with both friction paper feed and adjustable tractors standard, and uses the Micro-Nine disposable print head and Graftrax high resolution graphics capability.

It costs $995 from Epson America Inc., 23844 Hawthorne Blvd., Torrance, CA 90505.

Reader Service ✓ 166



MX-100 Dot Matrix Printer

Terminall

Convert Micro into Communications Terminal

Terminall is an integrated hardware and software system which converts a Model I or III TRS-80 into a communications terminal.

It includes all necessary computer interfacing, audio demodulating, AFSK tone generation and transmitter keying hardware in one cabinet, and will transmit Morse or RTTY and Baudot codes.

Terminall is available for $499 from Macrotronics Inc., 1125 N. Golden State Blvd., Turlock, CA 95380.

Reader Service ✓ 162

Educational Software Exchange

Softswap, an educational software exchange project, offers approximately 240 public domain instructional programs collected for the TRS-80 and other microcomputers.

Most of the Softswap programs are short, stand-alone instructional units. Each disk contains from five to 28 programs for various subjects and grade levels, all for one microcomputer system.

If you are interested in donating a program to Softswap or would like to purchase one of the program's disks or tapes, contact: Softswap, San Mateo County Office of Education, 333 Main St., Redwood City, CA 94063.

Reader Service ✓ 177

Fantasy Role-Playing Game

Empire of the Overmind is a fantasy role-playing game with a literary twist. The object is to defeat the evil Overmind and restore the kingdom to its rightful rulers.

To accomplish this, you must first study "The Rhyme of Overmind," which tells the story of the conquest of the kingdom of the Overmind and contains vital information to help you in your quest. You then venture forth into a world described by the manufacturer as a blend of Tolkien, Babylonian mysticism and science fiction.

Empire of the Overmind is available for $30 on cassette and $35 on disk from Avalon Hill Game Co., 4517 Harford Road, Baltimore, MD 21214.

Reader Service ✓ 164

Lease Accounting System Released

Leasing is a lease accounting system for equipment leasing firms. It maintains files on lessees, leases, assigness and vendors; generates bills; processes payments; and prepares vouchers for authorizing payments to vendors and assignees.

It also provides audit trails and can assign late charges.

It runs on a Z-80 system using CPM. It costs $3,500 from COM Business Computer Systems Inc., 551 E. Genesee St., Fayetteville, NY 13066.

Reader Service ✓ 340

Marck's Lists School Software

Marck's spring catalog lists tested educational programs and related products for the TRS-80, Apple II, Atari and PET computers.

Programs cover all age levels and include traditional and innovative subjects.

Cost is $4.95 ($5.32 for CT residents) from Marck, 290 Linden Ave., Branford, CT 06405.

Reader Service ✓ 167

Pascal Disks Offered

The Pascal/Z Users Group is offering 10 disks of public domain software applicable to Z-80 micros using Pascal/Z.

The programs include tutorials, utilities and practical applications.

They cost $10 each from the Pascal/Z Users Group, 7962 Center Parkway, Sacramento, CA 95823.

Reader Service ✓ 344

The Omni Flip/Floppy Disk

Flip/Floppy, A Reversible Disk

The Omni Flip/Floppy is a dual-sided reversible 5¼-inch disk.

The manufacturer certifies that each disk is error-free at more than twice the error threshold of many disk drives and is rated at more than 12 million passes without disk-related errors or significant wear.

The cost: five disks for $21 and 10 for $40 from Omni Resources, 4 Oak Pond Ave., Millbury, MA 01527.

Reader Service ✓ 350

Financial Reporting Package

T/Maker II is a financial reporting package combining word and numerical data processing on one disk.

Compatible with all CP/M microcomputers, it includes automatic functions such as transcendentals and logrithms and is designed for professional offices and small businesses.

It costs $275, from Lifeboat Assoc., 1651 Third Ave., New York, NY 10028.

Reader Service ✓ 183

Gosub Friction Feed Kit

The Gosub Friction Feed kit for the Epsom MX80/70 printers allows you to use single sheet or roll paper, your own letter-

Most people just sell disks. I sell you a complete system, and then I help you make it work.

It's called support, and it's a rare commodity in the microcomputer world.

It's also one big reason why they call my programs "the standard of the industry."

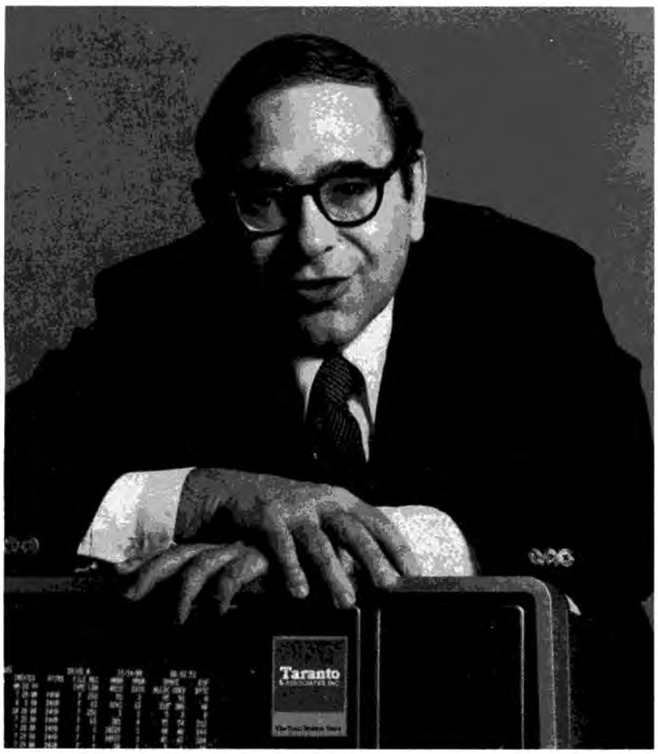
I'm Irwin Taranto, the one who changed the TRS-80* into a serious business computer. When you buy my TRS-80 systems (or, for that matter, one of my own computers that says "Taranto" on it), you buy me.

You buy my experience in making TRS-80 systems work in thousands of businesses around the world.

You buy the corrections, modifications and upgrades I constantly make on my TRS-80 systems.

And you buy my telephone number. You see, most of those thousand businesses needed a little help getting their systems up and going, and they called. We answered all their questions, and talked them through their problems. Every time the questions got really tough or really unusual, I'd answer them myself, on the phone, right then and there. I still do.

That pays off in two ways. It makes sure you get your systems working. It also alerts me to any little operating inefficiencies I might have designed into my systems. If there are any general business programs anywhere in the world, of any kind, that are checked out any better than my TRS-80 systems, I'd like to know about them.



I turned the TRS-80 into a serious computer.

The Model I, II and III business systems.

So far, I have six systems for the Model I, at $99 each:

Accounts Payable	General Ledger
Accounts Receivable	Payroll
Invoicing	Inventory Control

For the Cash Journal option on the General Ledger, add $50.

I also have six systems for the Model II:

General Ledger/Cash Journal	$ 299
Accounts Payable/Purchase Order	349
Open Accounts Receivable/Invoicing	349
Additional for Sales Analysis	100
Balance Forward Accounts Receivable	399
Payroll, without Job Costing	299
Additional for Job Costing	100
Inventory Control	399

For the Model III, we offer expanded versions of the six Model I systems, at $199 each.

Just call the number below and I'll send you any or all of the Model I or Model III systems by return mail. If you call about the Model II, I send you a questionnaire before I'll send you any systems. That lets me individualize the programs to your specific applications.

Why I call them "systems," not "programs."

There's a one-word answer: interaction. Each of the three sets of programs links to the General Ledger, and wherever it's useful, they cross-link to each other. For instance, "Sales Analysis" figures in a salesman's commission rate, so it links to "Payroll." Since it computes profitability within product categories, it links to "Invoicing."

That's what a system is. And that's one big difference between the Taranto TRS-80 business systems and somebody else's collection of business program disks.

If you like, I'll sell you the hardware, too.

I offer the TRS-80, Model II, along with selected peripherals. If you buy the computer from me, you get some extra advantages — hardware that's absolutely tailored to the programs, plus even more hand-holding from Taranto & Associates. The equipment won't cost you any more.

I can sell you a truly serious, completely supported, thoroughly proven business computer system for as little as $8000, hardware and software both.

There's nothing else like it in the market. Believe me, it's a far cry from that collection of program disks they're selling down the street.

✓45

Taranto
& ASSOCIATES, INC.

The Total System Store.™

121 Paul Drive, San Rafael CA 94903.
Outside California, toll free (800) 227-2868. In California, (415) 472-2670.
Authorized dealers throughout America.

*A trademark of the Tandy Corporation

✓See List of Advertisers on page 322

head, personal checks or non-tractor feeding labels.

The kit comes with a paper-roll rack for 8 1/2-inch teletype rolls. It utilizes "pressure feed," an opposing roller system, to pull the paper through.

The price is $49.95 from Gosub International Hardware Division, P.O. Box 275, Witchita, KS 67201.

Reader Service ✓ 181

Educational Software Library

Educational Software Library Inc. (ESL) is a nonprofit organization which provides a mail order library of educational programs for the Models I and III.

ESL publishes a bimonthly newsletter describing the expanding selection of software available, which ranges from preschool to high school levels.

A one-year membership for $20 entitles you to the newsletter, two educational tapes, the opportunity to trial-test programs and purchase them at discount prices. For more information write to: Educational Software Library 262 Park Lane, King of Prussia, PA 19406.

Reader Service ✓ 174

Plug 'N Power Controller

The TRS-80 Plug 'n Power Controller will allow your Model I, III or Color Computer to turn your lights and electrical appliances on and off, either in response to immediate commands typed into the computer or at preset times.

The Controller comes with its software for $39.95 from Tandy/Radio Shack.

Reader Service ✓ 161

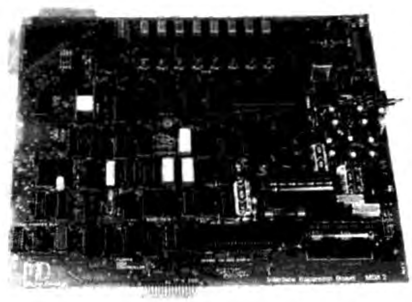
Centronics-type Printer Port

Cprint is a Centronics-type printer port for the TRS-80 Color Computer which connects to all parallel Radio Shack, Centronics, Epson, etc., printers.

It automatically reroutes all LList and Print#2 output; it has a screen-print function which can be started at any time; line width and page width can be set; the LPVII graphics can be used and blank lines are inserted between pages.

Cprint costs $49.95 from Micro-Labs Inc., 902 Pinecrest, Richardson, TX 75080.

Reader Service ✓ 160



The MDX-2

Expansion Interface Announced

The MDX series expansion interface boards for the Model I upgrade your computer to 32K memory. They also provide for a 2K or 4K EPROM, a direct-connect 300 baud phone modem, serial port, parallel port, dual-cassette option and hardware for a real-time clock. The MDX-2 also has a floppy disk controller.

The MDX-1 is $64.95 and the MDX-2 $74.95, from Micro-Design, PO Box 748, Mancanca, TX 78652.

Reader Service ✓ 335

Design Your Forms

Video Manager lets you design forms on your monitor using any combination of 80 and 40 character formats with both vertical or horizontal printing. You may then reproduce it exactly on your printer.

It also allows creation of graphic designs. It is written in assembly language and resides in high memory, so it is accessable from TRSDOS.

Available for the TRS-80 Model II, it costs $29.95 plus $2 postage and handling from Carl Larsen, Softwarehouse International, 5070 N. Sixth St., Suite 1038, Fresno, CA 93710.

Reader Service ✓ 329

Business Data Base Program

Business Data Base is a TRS-80 Model I and III package combining a worksheet-type program and a general purpose data base and storage system.

It is easily programmed for inventory control, payroll, accounts receivable, accounts payable, and general ledger applications.

It costs $89.95, from Charles Mann & Assoc., Micro Software Division, 55722 Santa Fe Trail, Yucca Valley, CA 92284.

Reader Service ✓ 348

Hear What You Type

Type-'N-Talk synthesizes English speech from typed input in much the same way as a typewriter for the blind works.

It requires an RS-232-C interface and handles an unlimited combination of English words and phrases. Its 750-character buffer will hold words until prompted or will echo words as they are typed.

It costs $345 from Votrax, 500 Stephenson Highway, Troy, MI 48084.

Reader Service ✓ 343

Cheaptalk Vocalizes TRS-80s

Cheaptalk gives a Level II TRS-80 with at least 16K RAM the ability to talk through any small audio amplifier connected to the cassette output plug. It also allows you to digitalize your own voice for reproduction.

It costs $19.95 on cassette from Alan Saville, PO Box 5190, San Diego, CA 92105.

Reader Service ✓ 342

Program Creates Data Tables

Electric Spreadsheet is a utility for the Model I or III which creates tables of data or spreadsheets on the computer screen, using a fast cursor to enter data or labels at the desired locations.

The program does calculations and presents the results in professional-looking tables, and can print these and save them to disk or tape.

Electric Spreadsheet is available on tape for the 16K Model I or III for $34.95 or for the 32K and 48K machines for $64.95 and on disk for the 48K Model III for $67.95, from Dan G. Haney & Assoc. Inc., PO Box 687, San Mateo, CA 94401.

Reader Service ✓ 330

A short history of games and game development.

A Walk on The Monster Side

by Bert Latamore
80 Microcomputing *staff*

Games have been around at least as long as Homo sapiens. No one knows when dice were invented; chess, checkers and parchisi go back into the dim times before written records. A couple of years ago S and R Games developed a game based on a board dug up in a Sumerian grave site by a group of British archeologists. The board is estimated to be 4,500 years old.

Today games are going through a distinct evolution under the influence of two separate forces. Near the end of the 19th century a group of free thinkers in England, including some of the most creative minds of the times, began experimenting with complex game forms. Alternative chess rules became popular, and rules were developed allowing simulations of historic battles using military miniatures. Incidently, one alternative chess game found its way into literature and can be read today at the beginning of Lewis Carroll's classic *Through the Looking Glass.*

This movement came of age in the 1960s and '70s with the appearance of games which allow you to fight whole battles or entire wars either with miniature figures or with cardboard counters on game boards which show all terrain features. At the same time, the concept of the fantasy-role-playing game, in which the player takes the part of a character and fights in hand-to-hand combat, explores mazes and interacts with other characters in fantasy-based situations appeared.

The latest major influence on games has been the computer. Games have been put on computers since Eniac, and, as computers become more common, more games are made and more people play them. The two movements came together early, probably because programmers were also players of the new kinds of game. The first computer games were war games, and fantasy-role-playing found its way onto the mainframe very early. "Adventure" is probably on every university and college mainframe system in the nation; people are playing it cross-country on electronic mail systems. Most microcomputer game designers are heavily influenced by it.

> "The most important concern of manufacturers is to produce a game which challenges the player, takes advantage of the computer's graphics capability and provides an escape from the real world."

At first, microcomputer games were fairly unsophisticated. Hangman and other simple ideas predominated, and the major concern was to write a program that worked as planned without bugs. Many game writers lacked knowledge of game design, and as a result many early microcomputer games were less than fascinating to play.

Things have changed, however. As the market grows, programmers learn more about games and more game players learn to program. Today bug-free operation is often guaranteed by the game supplier, and the emphasis is on developing more sophisticated games. Professional game designers are working with programmers, and their products are attracting serious game players.

The most important concern of manufacturers is to produce a game which challenges the player, takes advantage of the computer's graphics capability and provides an escape from the real world. They measure their success as much against board and arcade games as against their competition in the microcomputer market.

Defining what makes a good game from an esthetic point of view is hard. One major issue that has emerged, however, is better graphics in adventure games.

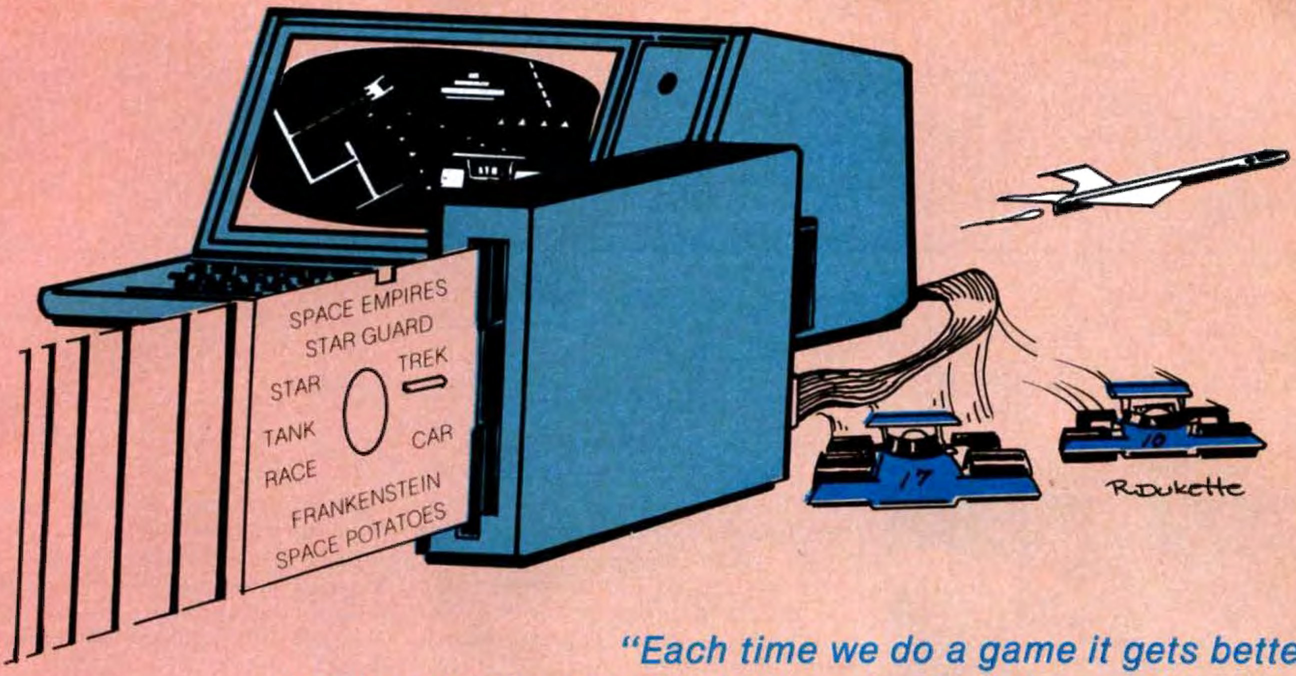
"When I get enough memory I want to do much better graphics," Dianne Asher, marketing director at Automated Simulations, Mountain View, CA, said. "There should be a way to get joystick operation into adventure games, smooth, consistent movement like aircraft simulators give."

Game Development

Game development for a company like Automated Simulations or the Avalon Hill Game Company, Baltimore, MD, is nearly identical to the process of developing a board game. These firms do most of their work in-house, seldom buying ideas or programs from the outside.

A game starts with an idea from a game designer. He works out the game mechanics so it will do what he wants it to do on paper. Then he takes it to the programmers.

Once the programmers are satisfied with their work, the game is extensively play tested. This is generally done at least partly out of house by serious game players who have contacts with the company. They make suggestions on how to improve the game, both from a design and a technical point of view. In the process, the company



"Each time we do a game it gets better. We handle the graphics better, we get things to fit better."

hopes they catch any bugs or other problems. Since they have no prior knowledge of the game, the play testers are a good check on whether the documentation is adequate.

Game Companies

Avalon Hill is unusual because it was a strategy board game company first and has only started into microcomputer games in the last year. So far its evolution has been the opposite of that of the typical microcomputer game company—they already had a sophisticated game outlook and the mechanism for developing a good game.

Their first games were designed to fit into 16K machines, according to Jackson Dott, marketing coordinator for Avalon Hill. As a result, he said, "Our first five games were mostly text. In other words, they were not graphically oriented. The reader feedback has been. 'The game is great but I want graphics.' "

At the same time, Dott said, they found many serious game players were expanding their machines to maximum memory. Therefore, their new releases, such as *Empire of the Overmind*, use much larger memories and provide much more in graphics and complexity.

Avalon Hill produces war, business, sports and fantasy games, and they are following this same pattern with their microcomputer games.

They originally decided to avoid computerizing successful Avalon Hill board games, and their first five releases were totally new designs. They have since adopted a more mixed strategy. Their releases have included new games, particularly new fantasy games, and computerized versions of proven Avalon Hill products.

Their first business game, an area which is almost totally neglected by everyone else, was a computerized version of their hotel chain game, *Acquire*. Dott said they hope to follow it with a computerized version of *Rail Baron*, Avalon Hill's most popular business game and a strong competitor with *Monopoly* for most popular business game on the market.

They are also working on computerizing some of the modules of one of their more popular tactical battle games.

However, their most ambitious computer project so far is to computerize *Diplomacy*, the classic WWI game. In it each player takes a major European power starting with the year 1900. Each wants to grow to dominate Europe. The game is unique because no one country can win on its own, each must make alliances to gain specific ends. Yet ultimately, each is playing on its own, no alliance lasts through the game. The successful player must know just when to stab his companion in the back.

Computer *Diplomacy* has not yet appeared on the market, but Avalon Hill does have the game running on a TRS-80.

Ed Friedland, head of Krell Software, Stony Brook, NY, is looking for modern classics to put on the computer.

"I try to concoct a game like Go. A very young child can learn the rules and play it but on the highest level it demands great expertise," he said. "I also want something that instructs the player in some way."

Friedland said his method of developing a game is very simple.

"I sort of sit around and ask what I would like to play," he said. "If nothing like it is out there I sit down with my programming staff and cook it up."

Med Systems, Chaple Hill, NC, produces some of the most elaborate maze games. This may be a direct result of part of the firm's philosophy. William Denman, Med System president, said they want a game "someone can play for weeks, with fast graphics and reasonably different from other games."

Anyone who has tried to find his way through Labyrinth knows what Denman means.

"Each time we do a game it gets better," he said. "We handle the graphics better, we get things to fit better. Asylum, our newest release, is a 1,200-location maze. A programmer would find it elegant, a game player will find it devastatingly interesting."

He said he is still writing most of his games for 16K memory because a lot of his customers, particularly overseas, are owners of 16K machines.

Curiously, while his company does not handle arcade games, he admits the arcades are his favorites, and he admires Big Five Software's products.

Another Approach

While most game producers develop their ideas in-house, Instant Software, Inc. (ISI), Peterborough, NH, does no initial development on any programs.

Of about 135 total programs on ISI's list, about 65 are games.

ISI is known for its flight simulation programs. They are about to release a new space shuttle program that lets the player fly and land the shuttle.

They are also involved in education games and have a hangman-type spelling game which speaks to the player instead of using text written on the screen.

One area ISI is not interested in is combination board and computer games. Avalon Hill has been experimenting with this concept in which the microcomputer keeps track of the paperwork of a game but your still move pieces around a physical board. Damon Crumb, assistant to the president, said he does not think that kind of hybrid will be successful.

"You are either visually oriented or textually oriented," he said. "There is little crossover among players."■

Klingons in space.

Startrek 4.0

by Jake Commander

No games issue would be complete without some sort of star trek. This is an addictive version, providing over an hour's playing time to zap those meanies (the Klingons) out of the galaxy.

This version is not programmed for sound, but a small transistor radio next to the keyboard adds to the effect. The program fits into 16K with no problem and works as printed on the Model I. Model III users should change the Out commands in line 5010 from: OUT255,8:OUT255,0 to OUT236,4:OUT236,0.

One extra feature missing from most star trek games is a real-time operating environment while you are at the helm of the Enterprise. This means that valuable stardates will tick away even as you peer at the screen. Also, as commander of the ship, the risk increases the longer you take to react to situations. Failure to take some sort of action will result in discovery of your ship's location by the enemy. As soon as you're spotted, they'll be in your quadrant, phasers ready. Don't just sit in space or you'll be a sitting duck.

Another interesting feature is the menu operation. Instead of a menu being displayed in typical list-like fashion, each command in the menu is called up by its command number. The command itself won't be executed until you hit (enter). As an example, say you hit the number two on the keyboard; this will display the command "Short-Range Visual Scan" at the bottom right hand corner of the display. If you then hit the number four, you will see "Fire Phasers" displayed. Neither command will be obeyed until it is entered, which allows you to change your mind, scan commands, and so on.

One tip to the neophyte starship commander: The more energy you place on your shields in defense of the ship, the greater your fuel consumption as you move around in space. This tempts you to leave your shields low in order to conserve fuel. The optimum shields-versus-fuel ratio will be determined by experience. You should expect, however, to be blown out of the sky the first couple of runs. The whole scenario appears very real and in fact, when I first wrote the program, I suffered quite a lapse into fantasy.

When I was close to completing the program and had thoroughly debugged it I thought about the real-time option and added that. Everything checked out. Finally I decided I didn't like the coordinate system, whereby a course of vector one headed the ship eastward; I preferred that the ship head northward for that vector. I changed the array that performed this calculation and tested it. I ran the program and tested the ship for correct movement. I found myself in the position of a test pilot running a new steering mechanism through its paces!

The ship moved to the four points of the compass with no difficulty. However, vector two moved the ship the wrong way. I realized I'd made a mistake with the vector array and had to correct the bug. I stared at the screen pondering what I needed to do. I decided a mere juxtaposition of two digits in the program would probably cure the fault. Without warning, out of nowhere, three Klingons appeared. What was I to do? Nothing was going to convince me to hit Break and shirk my responsibility as commander of the ship. I energized shields, and maneuvered into striking distance of the first Klingon. From there on I limped through the galaxy with a faulty steering mechanism and saved humanity from an awful fate.

You should have no problem with the thoroughly debugged ship provided.■

Program Listing

```
10 CLEAR150:DEFINTA,D-H,K-T:DIMS(8,8),G(8,8),Z(8,8),C(9,2),K(3,3
),D$(8),B$(30),C$(3)
35 Z$=STRING$(24," "):Y$="  ....................    ":X$=STRING$(1
4," "):W$=CHR$(30):V$="ENTERPRISE":U$="FEDERATION":CO$="## ,##"
120 CLS:PRINTTAB(25)"STAR TREK 4.0
"TAB(23)"BY JAKE COMMANDER

":INPUT"DO YOU WANT INSTRUCTIONS ";A$:IFA$="YES"GOSUB3460
160 FORX=0TO3:READC$(X):NEXT:FORY=1TO2:FORX=1TO9:READC(X,Y):NEXT
X,Y:FORX=0TO5:READA$(X):NEXT:FORX=1TO8:READD$(X):NEXT:FORX=1TO30
:READB$(X):NEXT
180 GOSUB5030:GOSUB5200:T=(RND(20)+19)*100+RND(100):T0=T+1:TT=32
:GOSUB5170:L=RND(8):M=RND(8):U=RND(8):V=RND(8):PRINT@640,"AWAIT
FURTHER ORDERS."
260 DATADOCKED,"GREEN ",YELLOW,"RED   ",,1,1,1,,-1,-1,-1,,-1,-1,
,1,1,1,,-1,-1,"   ",+++,"   ",<+>," . ",>!<,WARP ENGINES,S. R. S
CANNER,L. R. SENSORS,PHASER CONTROL,PHOTON TUBES,SHIELD CONTROL,
DAMAGE CONTROL,COMPUTER,,SET COURSE,SHORT-RANGE,VISUAL SCAN
270 DATALONG-RANGE,SENSOR SCAN,,FIRE PHASERS,FIRE PHOTON,TORPEDO
,,SHIELD CONTROL,DAMAGE CONTROL,REPORT,CALL ON-BOARD,COMPUTER,,F
INISHED,COMPUTER,GUIDANCE,STATUS,REPORT,PHOTON TORP-,EDO DATA,CO
URSE,CALCULATOR,GUIDED-PHOTON,TORPEDO,CUMULATIVE GA-,LACTIC RECO
RD
290 FORJ=1TO8:FORI=1TO8:R=RND(100):IFR>98K=3ELSEIFR>95K=2ELSEIFR
```

Program continues

Program continued

```
>80K=1ELSEK=0
370 KT=KT+K:R=RND(100):IFR<97THENB=0ELSEB=1:BT=BT+1
380 S=RND(8):G(I,J)=K*100+B*10+S:Z(I,J)=0:NEXTI,J
420 K7=KT:IFBT=0ORKT=0THEN290ELSEPRINT@640,"YOU MUST DESTROY";KT
;"KLINGONS WITHIN";TT;"STARDATES,
THERE ARE";BT;U$;" STARBASES IN THIS GALAXY.
450 CL=0:K=0:B=0:S=0:S(U,V)=3:IFL<1ORL>8ORM<1ORM>8THEN760
480 X=G(L,M):K=INT(X/100):X=X-K*100:B=INT(X/10):S=X-B*10
490 IFK=0THEN700ELSEIFG>200THEN660ELSEIFZ9=1ORZ7=1THEN550
500 PRINT"YOU HAVE ENTERED A COMBAT AREA WITH YOUR SHIELDS LOW."
;W$;"
THIS TIME THEY DIDN'T SHOOT FIRST. YOU HAVE BEEN WARNED....":Z9=
1:GOTO660
550 PRINT"COMBAT AREA  .. CONDITION RED - SHIELDS ARE DANGEROUSL
Y LOW.
660 GOSUB4900
700 IFB<>0FORI=1TOB:GOSUB3350:S(X,Y)=5:NEXT
710 IFS<>0FORI=1TOS:GOSUB3350:S(X,Y)=4:NEXT
720 ST=4:GOSUB5120
760 PRINT@204,USINGCO$;L;M;:PRINT@332,USINGCO$;U;V;:PRINT@640,;:
IFCL=1PRINT"SENSORS DETECT KLINGONS CLOAKING - PHASERS NECESSARY
";W$
770 IFD(2)<0O=0:A=2:GOTO850ELSEO=1:Z=81:FORY=1TO8:FORX=1TO8:PRIN
T@X*3+Z,A$(S(X,Y));:NEXTX:Z=Z+64:NEXTY
780 IFK<>0IFZ7=1IFRND(2)>1GOSUB2230ELSEELSEZ7=1
830 A=0:GOSUB860:PRINT@580,X$+CHR$(29)+"COMMAND?";:M2=0
835 H=0
840 H=H+1:IFH=350THEN870ELSEA$=INKEY$:IFA$=""THEN840ELSEH=0:IFAS
C(A$)=13THENPRINT@640,CHR$(31)ELSEGOSUB860:A=VAL(A$):PRINT@590,A
;:IFA=0ORA>8THEN840ELSEPRINT@494,B$(A*2-1);:PRINT@558,B$(A*2);:G
OTO840
845 IFA<3ORD(A)>=0PRINT@576,"###################";:PRINT@640,;:ONA
+1GOTO830,960,760,1460,1560,1740,2030,2090,2770,830
850 PRINTD$(A);" - INOPERATIVE";W$:IFA=4ANDCL=1PRINT"ILLOGICAL T
O REMAIN HERE, KLINGONS CLOAKING":GOTO830ELSE830
860 PRINT@591," ";:PRINT@494,X$;:PRINT@558,X$;:RETURN
870 GOSUB5170:PRINT@640,;:GOSUB1100:IFK<>0IFRND(2)>1GOSUB2230:GO
TO835ELSE835ELSEIFRND(10)<9THEN835ELSEGOSUB5170:IFL<1ORL>8ORM<1O
RM>8THEN835ELSEFORN=1TO8:FORQ=1TO8:X=INT(G(N,Q)/100):IFX=0NEXTQ,
N
890 G(N,Q)=G(N,Q)-X*100:G(L,M)=G(L,M)+X*100:K=X:PRINT"KLINGONS H
AVE JUST ENTERED THIS QUADRANT";W$:GOSUB4900:GOSUB5120:GOTO760
960 PRINT@640,;:INPUT"COURSE(1-9) ";C:IFC=0THEN830ELSEIFC<1ORC>9
THEN960ELSEPRINT@569,USING"#.#";C;:IFC=9C=1
980 PRINT@640,;:INPUT"WARP FACTOR (0-8)";W:IFW=0THEN830ELSEIFW<0
ORW>8THEN980
1020 IFD(1)<0ANDW>.2PRINTD$(1);" DAMAGED, MAX. SPEED IS WARP 0.2
":W=.25ELSEIFW<.9W=W*1.25
1040 GOSUB1100:GOTO1200
1100 IFST=0RETURNELSER=0:FORI=1TO8:IFD(I)>=0NEXTELSED(I)=D(I)+1:
IFD(I)=0R=I:I=8:NEXTELSENEXT
1110 IFD(7)<0RETURNELSEPRINT@640,;:IFR<>0PRINTD$(R);" NOW FUNCTI
ONING CORRECTLY";:GOTO1190
1130 IFRND(10)<9RETURNELSER=RND(8):ONRND(2)GOTO1150,1180
1150 D(R)=D(R)-RND(5):IFD(R)<0PRINTD$(7);" REPORT - ";D$(R);" OU
T";:GOTO1190ELSERETURN
1180 D(R)=D(R)+RND(5):PRINTD$(7);" REPORT - ";D$(R);" - REINFORC
ED";
1190 PRINTW$;:GOSUB5000:GOTO5150
1200 GOSUB2230:N=W*8:F=E/(5*(G/100+1)):IFN<=FTHENF=0ELSEN=F:F=1
1210 X=U:Y=V:S(X,Y)=0
1230 GOSUB1320:FORH=1TON:GOSUB1330:ONZGOTO1270,1275,1340
1270 NEXT:IFF=0THEN1310ELSEH=H-1:GOTO1290
1275 IFM2=1PRINT"SERIOUS DAMAGE HAS OCCURRED - CHECK ";D$(6):FOR
I=1TO8:D(I)=D(I)-RND(5)+1:NEXT
1280 PRINT"NAVIGATIONAL ERROR - ";:GOTO1300
1290 GOSUB5120:PRINT"ENERGY DEPLETED - ";
1300 X=X-X1:Y=Y-Y1:N=H-1:PRINTUSINGD$(1)+" SHUT DOWN AT SECTOR"+
CO$+W$;X;Y
1310 U=INT(X+.5):V=INT(Y+.5):S(U,V)=3:GOSUB5095:IFW<1THEN760ELSE
GOSUB5170:GOTO760
1320 D=C:X1=C(D,1)+(C(D+1,1)-C(D,1))*(C-D):Y1=C(D,2)+(C(D+1,2)-C
(D,2))*(C-D):RETURN
1330 X=X+X1:Y=Y+Y1:C=INT(X+.5):D=INT(Y+.5):IFC<1ORC>8ORD<1ORD>8Z
=3:RETURNELSEIFS(C,D)<>0Z=2:RETURNELSEZ=1:RETURN
1340 FORJ=1TO8:FORI=1TO8:S(I,J)=0:NEXTI,J:U=L*8+U+X1*N:V=M*8+V+Y
1*N:L=U/8:M=V/8:U=INT(U-L*8+.5):V=INT(V-M*8+.5):IFU=0L=L-1:U=8
1350 IFV=0M=M-1:V=8
1360 GOSUB5170:GOSUB5095:GOTO450
1460 O=0:A=84:PRINT@A,Y$;:FORJ=M-1TOM+1:FORZ=1TO3:N(Z)=0:NEXT
1480 FORI=L-1TOL+1:IFI<1ORI>8ORJ<1ORJ>8THEN1510
1490 N(I-L+2)=G(I,J):IFD(8)<0THEN1510
1500 Z(I,J)=G(I,J)
```

Program continues

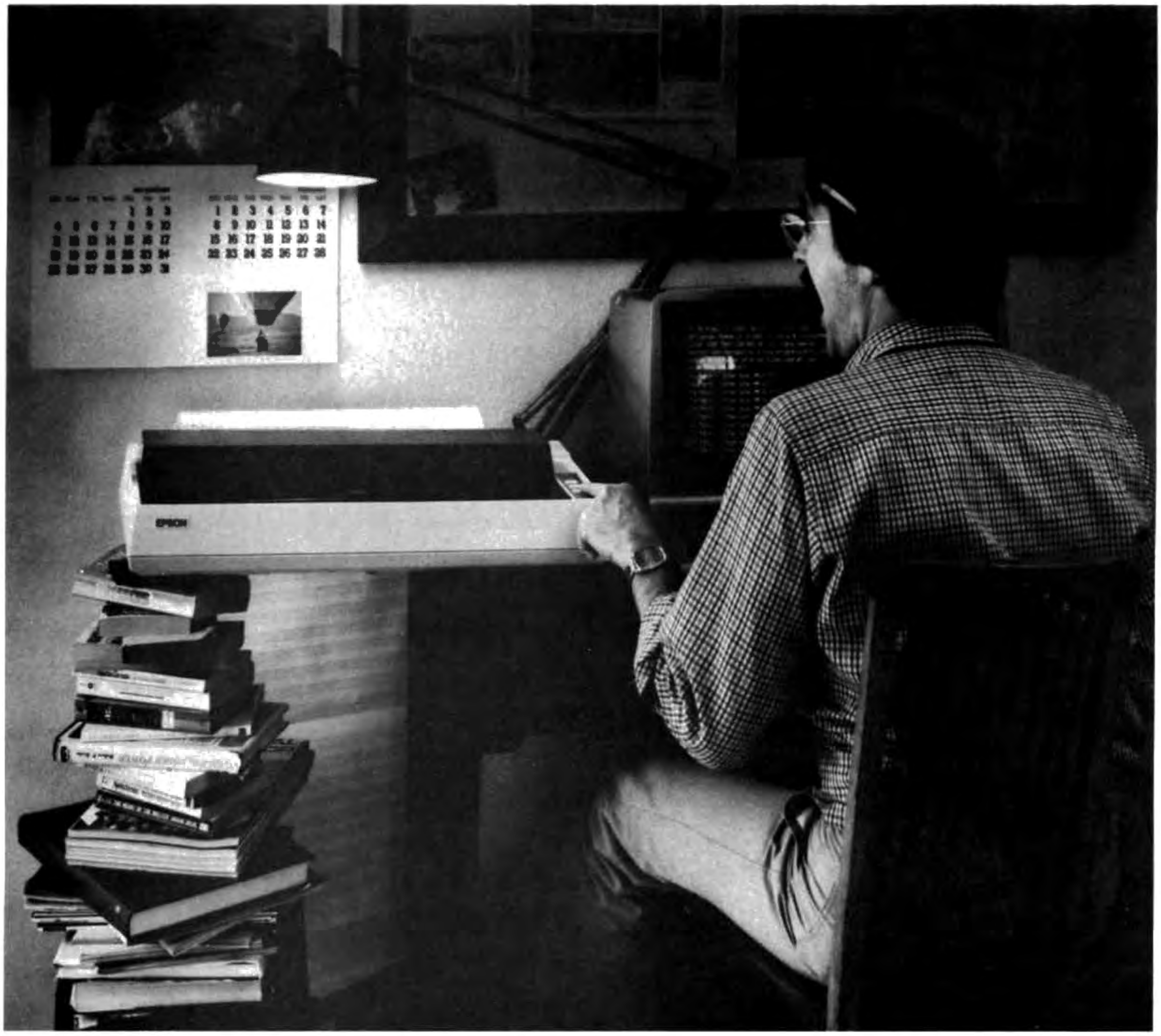
✓See List of Advertisers on page 322

Program continued

```
1510 NEXT
1520 A=A+64:PRINT@A,USING"  : ### : ### : ### :    ";N(1);N(2);N(
3);:A=A+64:PRINT@A,Y$;:NEXT:A=A+64:PRINT@A,Z$;:GOTO830
1560 IFST=ØPRINTC$(Ø);" - ";:GOTO850ELSEIFK=ØGOSUB5250:GOTO830
1590 GOSUB5150:Z7=1:IFD(8)<ØPRINTD$(8);" FAILURE. MANUALLY ";
1610 PRINT"LOCKED ON. ENERGY=";E;"... ";
1620 INPUT"UNITS TO FIRE";F:IFF<=ØTHEN830ELSEIFF>ETHEN1590
1640 E=E-F:GOSUB5100:IFD(8)<ØF=RND(F)
1650 R=640:FORI=1TO3:IFK(I,3)<ØTHEN1710
1670 H=F/K:J=.98:GOSUB4800:K(I,3)=K(I,3)-H:PRINT" KLINGON - "+W$
;:IFK(I,3)<ØX=K(I,1):Y=K(I,2):GOSUB5195ELSEPRINTUSING"AT SECTOR"
+CO$+" (### LEFT)";K(I,1);K(I,2);K(I,3):GOSUB5000
1710 NEXT:GOSUB2230:GOTO830
1740 IFP>ØTHEN1760ELSEPRINT"ALL PHOTON TORPEDOES EXPENDED.":GOTO
830
1760 IFCL<>ØGOSUB5260:GOTO830
1780 PRINT@640,B$(10);:INPUT" COURSE (1-9)";C:IFC=ØTHEN830ELSEIF
C<1ORC>9THEN1780ELSEIFC=9C=1
1800 Z7=1:GOSUB5180:GOSUB1320:X=U-X1:Y=V-Y1:GOTO1830
1810 PRINT@Q,"+";:GOTO1830
1820 PRINT@Q," ";
1830 GOSUB1330:IFZ=3THEN2020ELSEIFY=INT(Y)ANDO=1THENGOSUB5230:PR
INT@Q,".";:ONS(C,D)+1GOTO1820,1900,1820,1810,1970,1990ELSEONS(C,
D)+1GOTO1830,1900,1830,1830,1970,1990
1900 FORI=1TO3:IFC=K(I,1)ANDD=K(I,2)K(I,3)=-1:X=C:Y=D:R=640:GOSU
B5195
1920 NEXT:GOTO2025
1970 PRINT@640,USINGB$(10)+" CAPTURED BY STAR'S GRAVITY AT SECTO
R "+CO$;X;Y:GOTO2025
1990 GOSUB5210:PRINT@640,"***  STAR BASE DESTROYED  *** ... CONG
RATULATIONS!!":B=B-1:BT=BT-1:M2=1:X=C:Y=D:GOSUB5190:PRINT@704,"R
ESULTING EXPLOSION HAS BLOWN YOU OFF COURSE":C=RND(8):W=RND(8)/2
:GOTO1200
2020 PRINT@660," - ";B$(10);" MISSED."
2025 GOSUB2230:GOTO760
2030 IFST=ØPRINTC$(Ø);" - ";:GOTO850
2040 PRINT@640,;:INPUT"TOTAL UNITS ON SHIELDS";X:IFX<ØORX>E+GTHE
N2040ELSEE=E+G-X:G=X:GOSUB5100:GOSUB5110:PRINT@704,"ENERGY CONSU
MPTION IS NOW";5*(G/100+1);" UNITS PER SECTOR";W$;:GOTO830
2090 R=Ø:PRINT@576," NEXT...HIT '['  ";
2100 A$=INKEY$:IFA$=""THEN2100ELSEIFASC(A$)=13THEN830ELSEIFA$<>"
["THEN2100ELSER=R+1:IFR>8THEN830ELSEGOSUB5160:GOTO2100
2230 IFK=ØRETURNELSEIFST=ØZ7=Ø:PRINT@768,"STAR-BASE SHIELDS PROT
ECT THE ";V$;:RETURNELSER=640:FORI=1TO3:H=K(I,3):IFH<1NEXTELSEJ=
1:GOSUB4800:PRINTUSINGV$+" FROM SECTOR"+CO$+W$;K(I,1);K(I,2):G=G
-H:GOSUB5010:NEXT
2310 GOTO5110
2330 PRINT"THE "V$;" IS DEAD IN SPACE. YOU ARE AT THE KLINGONS'
MERCY":FORZ=1TO6:GOSUB5000:NEXT
2350 IFK=ØTHEN2390ELSEGOSUB2230:GOTO2350
2360 PRINT"IT IS STARDATE ";T;W$:GOTO2390
2380 PRINT@768,"THE "+V$+" WAS DESTROYED, THE ";U$;" WILL BE CON
QUERED
2390 PRINT"THERE ARE STILL";KT;"KLINGON BATTLE-CRUISERS SURVIVIN
G.":END
2410 PRINT"THE LAST KLINGON BATTLE-CRUISER IN THE GALAXY IS DEST
ROYED !!
THE ";U$;" IS SAVED !! YOUR EFFICIENCY RATING =";K7/(T-TØ)*1000:
END
2770 CD=Ø
2780 A=Ø:GOSUB860:PRINT@580,X$+CHR$(29)+"OPTION?";
2790 A$=INKEY$:IFA$=""THEN2790ELSEIFASC(A$)=13PRINT@576,:ONA+1GO
TO830,3320,2960,3060,3080,3300,2890ELSEGOSUB860:A=VAL(A$):PRINT@
590,A;:IFA>6A=Ø:GOTO2790ELSEF=A*2+17:PRINT@494,B$(F);:PRINT@558,
B$(F+1);:GOTO2790
2890 O=Ø:A=84:FORJ=1TO8:PRINT@A,;:FORI=1TO8:PRINTUSING"###";Z(I,
J);:NEXT:A=A+64:NEXT:GOTO2780
2960 O=Ø:FORA=84TO532STEP64:PRINT@A,Y$;:NEXT:PRINT@84,"      STAT
US REPORT:";:PRINT@212,"  KLINGONS  LEFT=";KT;:PRINT@276,"  STAR
DATES LEFT=";TØ+TT-T;:PRINT@340,"  STARBASES LEFT=";BT;:GOTO2780

3060 IFK=ØGOSUB5250:GOTO2780ELSECC=U:A=V:FORF=1TO3:IFK(F,3)<ØTHE
N3290ELSEW=K(F,1):X=K(F,2):GOTO3110
3080 INPUT"START AND END CO-ORDINATES ARE (X,Y,X,Y)";CC,A,W,X
3110 PRINT@704,;:X=X-A:A=CC-W:IFX<ØTHEN3200ELSEIFA<ØTHEN3230ELSE
IFX>ØTHEN3140ELSEIFA=ØTHEN3210
3140 CC=5
3150 IFABS(A)<=ABS(X)THEN3180ELSECC=CC+(((ABS(A)-ABS(X))+ABS(A))
/ABS(A)):GOTO3280
3180 CC=CC+ABS(A)/ABS(X):GOTO3280
3200 IFA>ØCC=7:GOTO3240ELSEIFX=ØTHEN3230
3210 CC=1:GOTO3150
```

Program continues



The MX-100. Not just better. Bigger.

Epson.

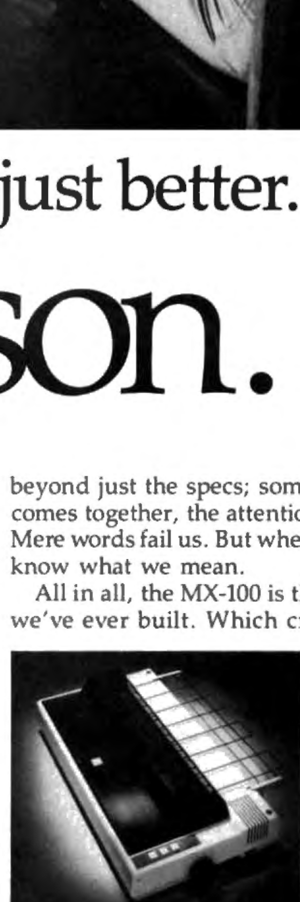
Our MX-80 was a pretty tough act to follow. I mean, how do you top the best-selling printer in the world?

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.

23844 Hawthorne Boulevard • Torrance, CA 90505 • (213) 378-2220 ✓404

See the whole incredible Epson MX Series of printers at your Authorized Epson Dealer.

✓See List of Advertisers on page 322

Program continued

```
3230 CC=3
3240 IFABS(A)>=ABS(X)CC=CC+ABS(X)/ABS(A)ELSECC=CC+(((ABS(X)-ABS(
A))+ABS(X))/ABS(X))
3280 PRINT@704,"COURSE =";CC;:CD=INT(SQR(X[2+A[2)):PRINT", DISTA
NCE =";CD:GOTO2780
3290 NEXT:GOTO2780
3300 IFD(5)<ØTHEN850ELSEIFP<1ORCL=1THEN1740ELSEIFCD<>ØTHENC=CC:G
OTO1800ELSE3330
3320 IFCD<>ØC=CC:W=CD:GOTO1020
3330 PRINT"NO DATA AVAILABLE FOR GUIDANCE SYSTEM":GOTO2780
3350 X=RND(8):Y=RND(8):IFS(X,Y)<>ØTHEN3350ELSERETURN
3460 PRINT"
YOU ARE THE COMMANDER OF THE STARSHIP 'ENTERPRISE'- YOUR MISSION
TO DESTROY KLINGON BATTLE-CRUISERS THREATENING THE FEDERATION -

        'TO BOLDLY GO WHERE NO MAN HAS GONE BEFORE'
";
3465 PRINT"
YOU MUST DO THIS WITHIN 32 STARDATES.";
3470 PRINT"

THE GALAXY IS REPRESENTED AS A 64 BY 64 MATRIX MADE UP FROM 8
8 BY 8 QUADRANTS - EACH CONTAINING 8  8 BY 8 SECTORS.

THE FOLLOWING OBJECTS ARE GRAPHICALLY REPRESENTED";
3475 PRINT"
<+> = THE ENTERPRISE.
+++ = KLINGON BATTLE CRUISER.
>I< = STARBASE.
 .  = STAR.
3480 GOSUB4790
3485 PRINT"
STARS ( . ) ... THESE ARE OBSTACLES. KLINGONS CAN HIDE BEHIND
THEM. IF YOU BUMP INTO THEM, YOU MAY DAMAGE YOUR SHIP.";
3495 PRINT"

STARBASES ( >I< ) ... REFUGE AND REFUELLING POINTS. TO DOCK WITH
A BASE, YOU MERELY NAVIGATE TO EITHER SIDE OF IT. REFUELLING IS
DONE AUTOMATICALLY - YOU MAY USE A BASE ANY NUMBER OF TIMES.";
3505 PRINT"

KLINGONS ( +++ ) ... THE ENEMY. THEY OFTEN SHOOT FIRST AND ASK
QUESTIONS NEXT. THEY ARE ARMED WITH PHASERS AND DEFLECTOR";
3515 PRINT"
SHIELDS, BUT SOME ARE ARMED WITH CLOAKING DEVICES WHICH MAKE
THEM INVISIBLE AND IMMUNE TO PHOTON TORPEDO ATTACK.";
3520 PRINT"

THE ENTERPRISE ( <+> ) ... THAT'S YOU. A MODERN POWERFUL WARP-
DRIVE SHIP, WITH ALL THE LATEST DEVICES, SUCH AS; SCANNERS,
PHASERS, PHOTON TORPEDOES, DEFLECTOR SHIELDS, AND AN ON-BOARD
COMPUTER. THESE ARE DESCRIBED NEXT.";
3525 GOSUB4790
3530 PRINT"
COMMAND 1 .... WARP ENGINE CONTROL. IT IS IMPORTANT TO KNOW THE
CIRCULAR VECTOR ARRANGEMENT AS SHOWN. INTEGER AND REAL VALUES
MAY BE USED. E.G. 1.5 IS HALFWAY BETWEEN 1 AND 2.";
3540 PRINT"

1 WARP FACTOR         8    1    2       VECTOR OF 9 IS UNDEFINED
MOVES EXACTLY           .  .  .         BUT VALUES MAY APPROACH 9
1 QUADRANT,         7. .  ...  . .3
.1 WARP FACTOR          .  .  .
MOVES 1 SECTOR        6    5    4";
3545 PRINT"

TO MOVE FROM QUADRANT 5,5 TO 6,5 YOU WOULD USE COURSE 3,
WARP FACTOR 1. USE WARPS LESS THAN 1 TO MOVE SECTOR TO SECTOR.";

3550 PRINT"

COMMAND 2 .. SHORT RANGE SENSOR SCAN. DISPLAYS ON THE ENTERPRISE
SCREEN THE IMMEDIATE QUADRANT, SHOWING POSITIONS OF ANY OBJECTS
IN THAT GALACTIC QUADRANT.";
3555 GOSUB4790
3560 PRINT"
COMMAND 3 .. LONG RANGE SENSOR SCAN. SHOWS CONDITIONS IN SPACE
FOR 1 QUADRANT EACH SIDE OF THE ONE YOU ARE IN, WHICH IS DISPLAY
ED IN THE CENTRE. THE SCAN IS IN THE FORM XXX , WHERE THE UNITS
DIGIT IS THE NUMBER OF STARS, THE TENS DIGIT IS";
3570 PRINT" THE NUMBER OF
STARBASES, THE HUNDREDS DIGIT IS THE NUMBER OF KLINGONS.";
3575 PRINT"
```

Program continues

Program continued

```
COMMAND 4 .. PHASER CONTROL. THIS ALLOWS YOU TO AIM PURE ENERGY
AT ANY KLINGONS IN THE VICINITY. EACH PHASER BURST WILL DEPLETE
HIS SHIELDS AND REDUCE HIS PHASER CAPABILITY - AND YOUR ENERGY."

3580 PRINT"
COMMAND 5 .. PHOTON TORPEDO CONTROL. ENABLES YOU TO COMPLETELY
DESTROY A KLINGON WITH A SINGLE HIT. TORPEDO COURSE IS THE SAME
VECTOR ARRANGEMENT AS ALREADY DESCRIBED.

":GOSUB4790
3585 PRINT"
COMMAND 6 .. SHIELD CONTROL. ALLOWS PROTECTION OF THE ENTERPRISE
BY PLACING SHIP'S ENERGY ONTO THE DEFLECTORS - INCREASES FUEL
CONSUMPTION THOUGH.";
3595 PRINT"
COMMAND 7 .. DAMAGE CONTROL REPORT. SHOWS STATE OF REPAIR OF ALL
DEVICES. A STATE OF LESS THAN ZERO MEANS THAT DEVICE IS
TEMPORARILY DAMAGED.";
3605 PRINT"

COMMAND 8 .. ON-BOARD COMPUTER - CONTAINS FOLLOWING OPTIONS:
  OPTION 0. FINISHED WITH COMPUTER, RETURN CONTROL TO CONSOLE.
  OPTION 1. COMPUTER GUIDES YOU ON THE COURSE FROM OPTION 4
  OPTION 2. STATUS REPORT - SHOWS HOW FAR THE MISSION HAS GONE
3610 PRINT"  OPTION 3. GIVES BEARING OF ANY KLINGONS IN THE QUAD
RANT
  OPTION 4. SHOWS VECTOR NEEDED TO TRAVEL BETWEEN INPUT CO-ORDS.
  OPTION 5. GUIDED PHOTON TORPEDO ON COURSE FROM OPTION 3.";
3615 PRINT"
  OPTION 6. CUMULATIVE RECORD OF ALL LONG RANGE SENSOR SCANS.
3620 GOSUB4790:RETURN
4790 PRINT@999,"PRESS ENTER TO CONTINUE:";
4795 A$=INKEY$:IFA$<>""THENRETURNELSE4795
4800 H=(H*2/SQR((K(I,1)-U)[2+(K(I,2)-V)[2))*(J*RND(0)+.2):R=R+64
:PRINT@R,USINGW$+"### UNIT HIT ON ";H;:RETURN
4900 IFK<>3IFRND(10)>7CL=1
4910 FORI=1TO3:K(I,3)=-1:NEXT:FORI=1TOK:GOSUB3350:S(X,Y)=CL+1:K(
I,1)=X:K(I,2)=Y:K(I,3)=200:NEXT:RETURN
5000 FORZZ=1E3TO1.8E3:NEXT:RETURN
5010 FORZZ=0TOH*.25:OUT255,8:ZZ=ZZ+.1:OUT255,0:ZZ=ZZ-.1:NEXT:RET
URN
5030 CLS:FORA=1TO28STEP27:Z=3:FORX=40TO87:SET(X,A):Z=Z+1:IFZ=6TH
ENZ=0:SET(X,A-1):SET(X,A+1):NEXTELSENEXTX,A
5040 FORA=38TO89STEP51:Z=2:FORX=1TO28:SET(A,X):Z=Z+1:IFZ<>3NEXTE
LSEZ=0:SET(A+1,X):SET(A-1,X):NEXTX,A
5045 X=1:Z=25:GOSUB5050:X=88:Z=19:GOSUB5050:FORX=88TO126:SET(X,2
8):NEXT:GOTO5090
5050 FORX=XTOX+38:FORY=1TOZSTEP6:SET(X,Y):NEXTY,X:RETURN
5090 PRINT@66,"STARDATE";:PRINT@194,"QUADRANT";:PRINT@245,"ENERG
Y";:PRINT@322,"SECTOR";:PRINT@373,"SHIELDS";:PRINT@450,"PHOTON T
.";:GOTO5150
5095 E=E-((N*5)*(G/100+1))
5100 PRINT@238,USING"####";E;:GOSUB5120
5110 PRINT@366,USING"####";G;:GOSUB5150:IFG<0THEN2380ELSERETURN
5120 FORZ=U-1TOU+1:IFZ<1ORZ>8NEXTELSEIFS(Z,V)=5THENZ=U+1:GOSUB52
40:NEXT:GOTO5130ELSENEXT
5122 IFK=0THEN5126
5124 IFST<>3ST=3:GOTO5140ELSE5130
5126 IFE>300ST=1:GOTO5130ELSEIFE>5*(G/100+1)IFST<>2ST=2:GOTO5140
ELSE5130
5128 PRINT@640,;:IFG=0THEN2330ELSEPRINT"YOU HAVE";E;"UNITS OF EN
ERGY. SHIELDS ARE TAKING";G;"UNITS":GOSUB5000:IFD(6)<0PRINT"BUT,
 ";D$(6);" IS OUT.SITUATION IRRECOVERABLE.":GOTO2330ELSE5124
5130 PRINT@110,C$(ST);:GOTO5150
5140 FORFL=1TO4:PRINT@110,C$(ST);:FORZZ=1TO50:NEXT:PRINT@110,"
   ";:FORZZ=1TO50:NEXTZZ,FL:GOTO5130
5150 PRINT@640,;:RETURN
5160 PRINT@576,X$;:PRINT@576,D$(R);:PRINT@590,D(R);:GOTO5150
5170 T=T+1:PRINT@77,T;:GOSUB5150:IFT>T0+TTTHEN2360ELSERETURN
5180 IFST=0P=10ELSEP=P-1
5185 PRINT@462,P;:GOTO5150
5190 GOSUB5210:S(X,Y)=0:G(L,M)=K*100+B*10+S:RETURN
5195 K=K-1:GOSUB5190:GOSUB5120:PRINT@R+28,USING"KLINGON AT"+CO$+
" DESTROYED"+W$;X;Y:KT=KT-1:IFKT=0THEN2410ELSEGOSUB5000:IFK=0CL=
0:RETURNELSERETURN
5200 ST=0:FORZ=1TO8:D(Z)=0:NEXT:E=5000:PRINT@237,E;:GOSUB5180:G=
0:GOTO5110
5210 IFO<>1RETURNELSEGOSUB5230:FORZ=1TO30:PRINT@Q+RND(3)-2,CHR$(
128+RND(62));:NEXT
5220 PRINT@Q-1,"   ";:GOTO5150
5230 Q=18.5+X*3+64*Y:RETURN
5240 IFST=0RETURNELSEPRINT@640,"SHIELDS DROPPED FOR DOCKING PURP
OSES";W$:GOSUB5200:GOTO5140
5250 PRINT"SENSORS DETECT NO KLINGONS IN THIS QUADRANT":RETURN
5260 PRINTD$(5);" WILL BE OF NO USE - ";B$(8);W$;:RETURN
```

Three word games in one.

Frankenstein

Robert A. Nicholas
2-B Lennox Heights
Lenox, MA 01240

Playing word games on the computer is fun. Using the machine to select a random word from a large word pool eliminates the need for another person to pick words for the player. There is one hassle involved, however, particularly when you use tape rather than disk—it takes a long time to load a game. When you get tired of one game and want to move on to another, you have to wait a few minutes while the new one is loaded.

Frankenstein eliminates that problem by providing three word games in one program:

- •Frankenstein-hangman.
- •Jumbles—unscrambling words.
- •Guess-a-word—a tougher word game!

Be sure to connect the auxiliary jack (the large gray plug leading to your tape recorder) to an amplifier or speaker so you can enjoy the sound effects while playing. A little animation has been added to Frankenstein to heighten your playing enjoyment.

The program requires Level II Basic with at least 16K and Disk Basic with 32K. There is a machine language sound routine, but you are not to reserve memory size for it—the program does that by itself! In fact, if you have set a specific memory size, you better reboot your system before loading Frankenstein or you may drive him crazy. (This is done in lines 30 to 70.)

A brief set of playing instructions is contained within the program itself.

Frankenstein

The object is to guess the computer's chosen word one letter at a time before the monster comes alive. Simply press the key for the letter you wish to try next—do not hit Enter. The computer will only accept uppercase letters, the asterisk and a slash.

If your guess is contained in the mystery word, the computer will put it in its correct position(s) and remove it from the alphabet list appearing at the bottom of the screen. If it isn't used in the word, the monster acquires another body part (recently removed from some grisly graveyard). If you enter a letter you've already guessed, the computer will inform you of your error and let you try again.

You may decide you know what the word is and would like to guess it right away. Press the slash key instead of a letter and you will be offered the opportunity to guess the unknown word. If you are correct you win but if you're wrong the monster gets another body part.

Frankenstein makes losing fun. Electric sparks zap his head from the two power generators on either side of him, he comes to life and waves his arms up and down in joy. He won't reach out and crush you, he just asks if you want to play again.

If you decide at any time that you want to give up and return to the menu, enter the asterisk as your response.

Jumbles

Examine the jumbled word the computer gives you and try to figure out what it should be. Type your corrected version and see if you were right: If you are, you get an ascending whoop and your score goes up. If not, a descending sound is played and your score plummets. If you decide to quit and return to the menu to try another game, press the asterisk as your response.

Guess-a-Word

This one isn't easy. The computer will show you how many letters are in the mystery word; your task is to discover what the word is. You can enter a guess word containing fewer than or as many letters as are in the word itself. If your guess is too long, the computer will ignore the extra letters.

After each guess, those letters will be eliminated from the alphabet list at the bottom of your screen. If any of your letters were in the word but not in their correct positions, they will be indicated beside the phrase "Correct but wrong place." Any correct letters in your entry that were correctly positioned are placed in the blanks on the screen.

This will be changed every time you make a guess, so be sure to keep a record of any information you receive.

For example, suppose the mystery word is "beast". It will be shown to you on the screen as five blanks (-----). If your first try was AEIOU, the computer will analyze your entry, and give you the following response:

```
-E---
CORRECT BUT WRONG PLACE-A
```

You know that E is the second letter of the word, A is contained in the word but not in the first position (which is where it was in your guess) and that I, O and U are not contained in the word.

Either A or E could be in the word more than once. You must enter those letters more than once in your guess words to discover this.

Now suppose your second guess is RSTMN. The screen will read:

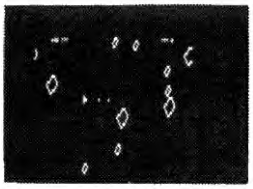
```
-----
CORRECT BUT WRONG PLACE—ST
```

What happened to the E and A we already

Games from BIG FIVE will turn your computer into a

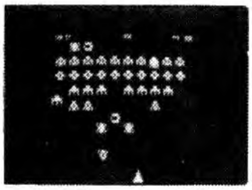
TRS-80 HOME ARCADE

SUPER NOVA©



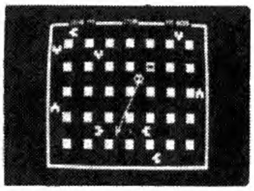
If you and your TRS-80 have longed for a fast-paced arcade-type game that is truly a challenge, then **SUPER NOVA** is what you've been waiting for. In this two player machine-language game, large asteroids float ominously around the screen. Suddenly your ship appears and you must destroy the asteroids before they destroy you! (But watch out because big asteroids break apart into little ones.) The controls that your ship will respond to are thrust, rotate, hyperspace, and fire. All right! You've done it! You've cleared away all the asteroids! But what is that saucer with the laser doing? Quick! You must destroy him fast because that guy's accurate! *As reviewed in May 1981* Byte *magazine.*

GALAXY INVASION©



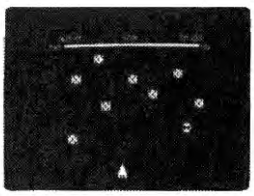
The sound of the klaxon is calling you! Cruel and crafty invaders have been spotted in battle formation warping toward Earth at an incredible speed. Suddenly, your ship materializes just below the huge flock of invaders. Quickly and skillfully you shift right and left as you carefully fire your lasers at them. But watch out! A few are breaking out of the convoy and flying straight at you! As the whine of their engines gets louder, you place your finger on the fire button knowing all too well that this shot must connect—or your mission will be permanently over! With sound effects!

ATTACK FORCE©



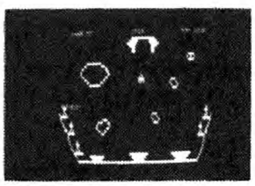
Your TRS-80 screen has been transformed into a maze-like playfield for this game. As your ship appears on the bottom of the screen, eight alien ramships appear on the top. All of them are traveling at flank speed directly at you! Quickly and boldly you move toward them and fire missiles to destroy them. But the more aliens you destroy, the faster the remaining ones become. If you get too good you must endure the wrath of the keeper of the mazefield: the menacing "Flagship". You must destroy him fast because, as you will find out, that guy's accurate! With sound effects!

COSMIC FIGHTER©



With thousands of stars whizzing by you, your **SPACE DESTROYER** ship comes out of hyperspace directly under a convoy of aliens. Almost effortlessly, you skillfully destroy every last one. But before you can congratulate yourself, another set appears. These seem to be slightly more intelligent than the first set. Quickly you eliminate all of them, too. But your fuel supply is rapidly diminishing. You must still destroy two more sets before you can dock with your space station. All right! The space station is now on your scanners! Oh no! Intruders have overtaken the station! You must skillfully fire your neutron lasers to eliminate the intruders from the station before your engines run out of fuel and explode! With sound!

METEOR MISSION II©



As you look down on your space viewer you can see the stranded astronauts that are crying out for you to rescue them. But first you must maneuver your shuttle down through the asteroids & meteors before you can reach them. Great! You've got one! But now can you get back to the space station to save your fellow shipmate or will you crash and kill both of you? You can fire your lasers to destroy the asteroids, but watch out, because there could be an alien FLAGSHIP lurking behind! Includes sound effects!

NEW JOYSTICKS!

For $39.95 it's now possible to have the famous ATARI joystick interfaced with your Model 1. All of our tapes are now completely compatible with the joystick. Packaged with complete instructions, you can even use it with your own programs! If your old tapes do not say "Joystick Version" on them and you wish to exchange them for new Joystick versions, enclose $2 and your old tape. (Call or write for info on Mod 3 joysticks.)

BIG FIVE SOFTWARE

P.O. Box 9078-185 · Van Nuys, CA 91409 · (213) 782-6861 ✓357

Prices per game: Level 2, 16K Cassette Mod 1/Mod 3—$15.95
Level 2, 32K Diskette Mod 1/Mod 3—$19.95
10% discount for 2 items, 15% for 3 or more (excludes upgrades).
Please add $1.50 per order for postage & handling, Calif. residents add 6% sales tax.
Outside USA please add $3.00 per order for postage & handling.
We accept checks, money orders, and MC/Visa orders ($2.00 extra for COD).

All games ©1980 by Bill Hogue & Jeff Konyu.
Programs are written in machine language for high quality graphics.
Disk versions are self-booting and compatible with Mod 1 and Mod 3 disk systems.
High scores are automatically saved after each game on disk versions.
TRS-80 is a trademark of Tandy Corp.
Dealers: All games now available in full color packaging, please inquire.

"Frankenstein makes losing fun. Electric sparks zap his head... he comes to life and waves his arms..."

knew about? Because we didn't guess them this time, the computer doesn't show them. That's why you have to keep a record of what's going on.

Since we already know that E is in the second place and A, S and T are in there somewhere, it shouldn't take too many more guesses to find that the answer is beast.

If you decide to give up on this word but want to continue playing Guess-a-Word, enter / as your response. If you want to throw in the towel completely and go back to the menu, enter an asterisk.

Typing the Program

None of the remark statements are referred to by the program. If you wish to eliminate them, feel free to do so, making sure the rest of the lines are numbered correctly.

Blank spaces have been eliminated to allow more memory space for the word list. If you want to create your own list of words,

Program Listing

```
10 REM          F R A N K E N S T E I N
20 REM            ROBERT F. NICHOLAS
30 POKE16561,150:IFPEEK(16562)>191THENPOKE16562,191
40 CLEAR2000:DEFINTA-Z:IFPEEK(16562)<191THENML=32664:POKE16526,1
52:POKE16527,127:GOTO60
50 ML=-16487:IFPEEK(14305)<255THENDEFUSR0=ML:CMD"T"ELSEPOKE16526
,152:POKE16527,191
60 FORX=MLTOML+23:READN:POKEX,N:NEXT:ML=ML+1:POKE16553,255:RANDO
M:NW=300
70 DATA 14,0,62,8,211,255,65,16,254,62,9,211,255,65,16,254,13,32
,238,62,8,211,255,201
80 CU$=CHR$(143):CB$=CHR$(32):ZP$=STRING$(12,94)+STRING$(11,93)
90 CLS:PRINT@452,CHR$(23);"I'M THINKING UP SOME WORDS!";:PRINT@0
,"CONNECT SMALL AMPLIFIER/SPEAKER  TO AUXILLIARY PLUG FOR SOUND.
";
100 DIMH$(NW),H(NW),A$(90),A(90),W$(90),W(90),U$(50),U(50)
110 REM READ WORD LIST
120 FORN=1TONW:READH$(N):PRINT@660,H$(N);CHR$(30);:NEXTN:CLS
130 INPUT"WOULD YOU LIKE DIRECTIONS";Y$:Y$=LEFT$(Y$,1):CLS:IFY$=
"N"THEN310
140 PRINT"      YOU WILL BE ABLE TO PLAY THREE WORD GAMES WITH T
HIS
150 PRINT"PROGRAM:  FRANKENSTEIN, JUMBLES AND GUESS-A-WORD!
160 PRINT"      THE OBJECT OF FRANKENSTEIN IS TO GUESS THE COMPU
TER'S WORD";
170 PRINT"ONE LETTER AT A TIME.  BUT YOU BETTER GUESS IT BEFORE
THE
180 PRINT"MONSTER COMES ALIVE!  TO GUESS THE FULL WORD, PRESS '/
' AS YOUR
190 PRINT"LETTER GUESS."
200 PRINT"      IN JUMBLES, YOU WILL BE ASKED TO UNSCRAMBLE THE
WORDS.
210 PRINT"SIMPLY ENTER IN YOUR ANSWER AND THEN PRESS <ENTER>."
220 PRINT"      GUESS-A-WORD IS A LITTLE TRICKIER.  IF YOU HAVE
A FIVE
230 PRINT"LETTER WORD, GUESS A FIVE LETTER ANSWER.  THE COMPUTER
 WILL TELL";
240 PRINT"YOU HOW MANY OF YOUR LETTERS WERE USED IN THE WORD AND
 WILL"
250 PRINT"PRINT THOSE THAT WERE IN THE RIGHT PLACE.  PRESS '/' T
O GIVE
260 PRINT"UP ON A WORD.
270 PRINT:INPUT"     HIT ENTER WHEN READY TO CONTINUE.";Y$:CLS:G
OTO310
280 REM MENU
290 CLS:PRINT@0,CHR$(23);"THE WORD WAS ";H$(H);"!":FORTI=1TO500:
NEXTTI
300 PRINT"********************************";
310 PRINT@128,CHR$(23);"MENU OF WORD PUZZLES:
320 PRINT@260," 1 - FRANKENSTEIN
```

Program continues

"See if you were right: if you are, you get an ascending whoop and your score goes up."

Program continued

```
330 PRINT@324,"  2 - JUMBLES
340 PRINT@388,"  3 - GUESS-A-WORD
350 PRINT:PRINT:PRINT:PRINT"('*' TO RETURN TO THIS MENU)
360 PRINT:INPUT" ENTER THE # OF YOUR CHOICE";CH:IFCH<1ORCH>3THEN
310
370 CLS:R=0:W=0:ONCHGOTO390,1070,1310
380 REM========== GAME OF FRANKENSTEIN =====================
390 GOSUB410:GOTO490
400 REM FIND A WORD NOT USED YET
410 PRINTCHR$(23);:FORX=1TONW:IFH(X)=0THEN430ELSENEXTX
420 FORX=1TONW:H(X)=0:NEXTX:H=RND(NW):GOTO440
430 FORT=1TO5:H=RND(NW):IFH(H)=0THEN440ELSENEXTT:H=X
440 H$=H$(H):H(H)=1
450 REM BREAK WORD DOWN INTO LETTERS
460 L=LEN(H$)
470 FORX=1TOL:W$(X)=MID$(H$,X,1):W(X)=0:NEXTX:RETURN
480 REM PRINT BOARD AND BUILD LAB
490 Z=0:GOSUB810:PRINT@576,STRING$(32,191);:PRINT@640,STRING$(32
,143);
500 P=610-L:IFP/2<>INT(P/2)THENP=P-1
510 PRINT@P-2,STRING$(L+2,32);
520 PRINT@P,CHR$(23);STRING$(L,"-");
530 REM  BEGIN GUESSING LETTERS
540 FORX=65TO90:A$(X)=CHR$(X):NEXTX
550 PRINT@704,"WHAT'S YOUR LETTER?";
560 PRINT@898,STRING$(30,176);
570 REM PRINT ALPHABET
580 PRINT@964,"";:FORX=65TO90:PRINTA$(X);:NEXTX
590 CU=744:PRINT@768,"('/' TO GUESS WORD)";
600 A$="":PRINT@CU,CU$;
610 A$=INKEY$:IFA$=""THEN610
620 IFA$="*"THEN290
630 IFA$="/"THEN980
640 A=ASC(A$)
650 IFA<65ORA>90THEN600
660 PRINT@CU,A$;
670 IFA$(A)=" "THENPRINT@768,"YOU ALREADY USED ";A$;"!";:GOSUB96
0:GOTO590
680 S=0:NC=0:FOR X=1TOL
690 IFW$(X)=A$THENW(X)=1:S=S+1
700 IFW(X)=1THENNC=NC+1
710 NEXTX
720 IFNC=LTHEN740
730 IFS<>0THEN740ELSEIFZ>=7THEN900ELSEPRINT@768,CHR$(30);:PRINT@
768,"THE MONSTER GETS ";:POKEML,0:POKEML+15,12:Z8=USR(0):GOSUB81
0:GOSUB960:A$(A)=" ":GOTO580
740 PRINT@P,"";
750 FORX=1TOL
760 IFW(X)=1THENPRINTW$(X);:GOTO780
770 PRINT"-";
780 NEXTX:A$(A)=" ":IFNC=LTHEN1010
790 POKEML,0:POKEML+15,13:Z8=USR(0):GOSUB960:GOTO580
800 REM LAB AND BODY PARTS
810 Z=Z+1:ONZGOTO820,850,860,870,880,890,840,900
820 PRINT@0,STRING$(32,131);:PRINT@512,STRING$(32,176);:FORX=0TO
8:PRINT@64*X,CHR$(191);:PRINT@64*X+62,CHR$(191);:NEXT:PRINT@72,"
F R A N K E N S T E I N";
830 FORX=260TO516STEP64:PRINT@X,STRING$(4,191);:PRINT@X+48,STRIN
G$(4,191);:NEXT:PRINT@198,CHR$(170);:PRINT@246,CHR$(149);:PRINT@
332,CHR$(133);:PRINT@370,CHR$(138);:RETURN
840 PRINT"HIS HEAD!";:FORX=64TO68:FORY=8TO10:SET(X,Y):NEXTY,X:SE
T(65,11):RETURN
850 PRINT"HIS BODY!";:FORX=61TO70:FORY=12TO14:SET(X,Y):NEXTY,X:F
ORX=63TO68:FORY=15TO16:SET(X,Y):NEXTY,X:FORY=17TO18:SET(65,Y):NE
XT:RETURN
860 PRINT "HIS ARM!";: FORX=71TO74:SET(X,12):NEXT:FORY=12TO15:SE
T(76,Y):NEXT:RETURN
870 PRINT"HIS OTHER ARM!";:FORX=57TO61:SET(X,12):NEXT:FORY=12TO1
5:SET(56,Y):NEXT:RETURN
880 PRINT"HIS LEG!";:FORY=19TO24:SET(61,Y):NEXT:FORX=60TO63:SET(
```

Program continues

✓See List of Advertisers on page 322

"Guess-a-Word isn't easy. The computer will show you how many letters are in the mystery word; your task is to discover the word."

the data begins at line 1780 and runs to the end. The number of words is determined by the value of the variable NW at the end of line 60. If you decrease the list or change it, count your words and set the value of NW accordingly.

When creating your own list, be very careful. Since the list is used for all three games, you must be sure that only one word can be spelled using those letters. For example, you can't use the word live because you can also spell evil using those letters.

The Machine Language Sound Routine

Machine language sound routines normally send the values 1 and 0 alternately out the cassette port (255) with a vibration count to create sound. This program uses double size characters—CHR$(23); as soon as you send the 1 or 0 out that port, it flips the double size characters to single size and wrecks the screen image.

However, if you send 8 and 9 instead of 1 and 0, the double size stays double size:

```
        LD    C,0        ;VIBRATION COUNT
                         POKED BY PROGRAM
SOUND   LD    A,8
        OUT   (255),A    ;ON TO SPEAKER
        LD    B,C        ;THE VIBRATION COUNT
TON     DJNZ  TON        ;B = B - 1 IFB>0 THEN
                         REPEAT THIS LINE
        LD    A,9
        OUT   (255),A    ;OFF TO SPEAKER
        LD    B,C        ;VIBRATION COUNT
                         AGAIN
TOFF    DJNZ  TOFF       ;B = B - 1 IFB>0 THEN
                         REPEAT THIS LINE
        DEC   C          ;C = C - 1 LOWER VI-
                         BRATION COUNT
        JR    NZ,SOUND   ;IFC>0 VIBRATE AT NEW
                         VALUE OF C
        LD    A,8
        OUT   (255),A    ;RESET PORT VALUE
        RET              ;RETURN TO BASIC
                         PROGRAM
```

By continually lowering the value of the C register, we create an ascending whoop. The starting value of the whoop gets POKEd by the Basic program where needed. When we want a descending whoop, change DEC C in line 10 to INC C. One curious note: using 8 and 9 for sound rather than 1 and 0 means we have to send 8 out the port again or there is a faint clicking sound shortly after the regular tone.

Whenever you're entering a program with machine language routines in it, it's a good idea to save partially completed copies before running the section with the user calls. If you entered anything incorrectly, you just might lose the program completely! ■

Program continued

```
X,24):FORX=61TO64:SET(X,18):NEXT:RETURN
890 PRINT"HIS OTHER LEG!";:FORY=19TO24:SET(69,Y):NEXT:SET(72,24)
:FORX=66TO70:SET(X,18):NEXT:RETURN
900 FORX=1TO15:PRINT@200,ZP$;:POK
EML,30:POKEML+15,13:Z8=USR(0)::PRINT@200,STRING$(23,32);:POKEML,
240:POKEML+15,12:Z8=USR(0):NEXT:FORX=64TO68:FORY=8TO10:SET(X,Y):
NEXTY,X:PRINT@1022,"";CHR$(23);:SET(65,11)
910 PRINT@704,CHR$(30);:PRINT@768,CHR$(30);:PRINT@704,"THE WORD
WAS ";H$;:FORTI=1TO250:NEXT:PRINT@72,"  THE MONSTER LIVES!!!   "
;
920 FORTI=1TO6:FORY=9TO12:RESET(76,Y):RESET(56,Y):NEXTY:FORY=12T
O15:SET(76,Y):SET(56,Y):NEXTY:POKEML,230:POKEML+15,12:Z8=USR(0)
930 FORY=12TO15:RESET(76,Y):RESET(56,Y):NEXTY:FORY=9TO12:SET(76,
Y):SET(56,Y):NEXTY:POKEML,100:POKEML+15,13:Z8=USR(0):NEXTTI
940 PRINT@832,"PLAY AGAIN (Y) OR GO TO MENU (*)";:GOTO1020
950 REM CLEAR MIDDLE SECTION OF SCREEN FOR INPUT
960 FORTI=1TO300:NEXTTI:FORCL=768TO832STEP64:PRINT@CL,CHR$(30);:
NEXTCL:RETURN
970 REM GUESS WORD
980 PRINT@768,"WHAT IS THE WORD";:INPUTW$:IFW$="*"THEN290
990 IFW$=H$THEN1010
1000 POKEML,0:POKEML+15,12:Z8=USR(0):PRINT@832,"WRONG! THERE GOE
S ";:GOSUB810:GOSUB960:GOTO590
1010 POKEML,0:POKEML+15,13:Z8=USR(0):PRINT@832,"CORRECT!  PLAY A
GAIN (Y) OR":PRINT"GO TO  MENU (*)";
1020 Y$=""
1030 Y$=INKEY$:IFY$=""THEN1030
1040 IFY$="*"THENCLS:GOTO310
1050 IF Y$="Y"THENCLS:GOTO390ELSEPRINT@832,CHR$(30);:GOTO1010
1060 REM ========== GAME OF JUMBLES ==========================
1070 J$="":GOSUB410
1080 FORX=1TOL
1090 Q=RND(L):IFW$(Q)=""THEN1090
1100 J$=J$+W$(Q):W$(Q)=""
1110 NEXTX:IFJ$=H$(H)THEN1070
1120 REM UNSCRAMBLE WORD
1130 PRINT@340,CHR$(23);J$;
1140 PRINT@448,"WHAT IS THE WORD ";:PRINT@704,"('*' FOR MENU)";
1150 FORX=1TO9:U$(X)="":NEXT
1160 CU=596:PRINT@CU,CU$;
1170 A$=""
1180 A$=INKEY$:IFA$=""THEN1180
1190 IFA$="*"THEN290
1200 A=ASC(A$):IFA=13THENPRINT@CU,CB$;:GOTO1250
1210 IFA<>8THEN1230ELSEPRINTA$;:IFCU>596THENX=X-1:CU=CU-2
1220 PRINT@CU,CU$;:GOTO1170
1230 IFA<65ORA>90THEN1170
1240 X=X+1:PRINT@CU,A$;:CU=CU+2:PRINT@CU,CU$;:U$(X)=A$:GOTO1170
1250 A$="":FORLE=1TOX:A$=A$+U$(LE):NEXT
1260 IFA$=H$(H)THENPRINT@704,"CORRECT!";CHR$(30);:R=R+1:POKEML,0
:POKEML+15,13:Z8=USR(0):GOTO1280
1270 PRINT@704,"WRONG, THE WORD WAS ";H$(H);"!";:W=W+1:POKEML,0:
POKEML+15,12:Z8=USR(0)
1280 PRINT@960,"(ENTER '*' FOR MENU.)";:J$="":PRINT@832,"HIT ENT
ER TO CONTINUE";:INPUTEN$:CLS:IFEN$="*"THEN310
1290 PRINT@0,CHR$(23);"RIGHT";R;TAB(9);"WRONG";W;TAB(18);"SCORE
";:PRINT USING "###.## %";100*(R/(R+W));:PRINT@64,STRING$(32,131
);:GOTO1070
1300 REM ========== GAME OF GUESS-A-WORD =====================
1310 GOSUB410
1320 FORX=1TO9:U(X)=0:NEXT:TR=0:FORX=65TO90:A(X)=1:NEXTX
1330 REM PRINT BOARD
1340 PRINT@770,STRING$(30,131);
1350 PRINT@962,STRING$(30,131);
1360 P=160-L:IFP/2<>INT(P/2)THENP=P-1
1370 GOSUB1710:GOSUB1660
1380 FORX=1TO9:U$(X)="":NEXT:PRINT@448,CHR$(30);:PRINT@384,CHR$(
23);"WHAT IS YOUR GUESS? ";:CU=424:PRINT@CU,CU$;:X=0:PRINT@512,"
('/' = QUIT, '*' = MENU)";
1390 A$=""
```

Program continues


How's your love life?
INTERLUDE #99 THE ULTIMATE EXPERIENCE...
A little dull around the edges? Routine? Predictable? Boring? Maybe all it needs is a little Interlude. Interlude is the most stimulating computer game ever conceived. It combines a computer interview, an innovative programming concept, and a one-of-a-kind manual to turn your love life into exciting, adventurous, delicious fun!
Interlude is: romantic... playful... outrageous... a fantasy. Interlude is: ■ A Bed of Roses (Interlude #1) ■ Mata Hari (Interlude #49) ■ The Chase (Interlude #7) ■ Rodeo! (Interlude #71) ■ The King and I (Interlude #60) ■ Some Enchanted Evening (Interlude #84) ■ Caveman Caper (Interlude #82) ■ From Here to Ecstasy (Interlude No. 30) ■ Satin Dreams (Interlude #72).
More than 100 Interludes are included in the program. Most are described in detail in the accompanying manual, but several surprise Interludes are buried in the program awaiting that very special time when your interview says you're ready. (When you learn secret Interlude #99, your love life may never again be the same!) Interlude can give you experiences you'll never forget. Are you ready for it?
287
Interlude™
The Ultimate Experience.
INTERLUDE, Dept. 80, 10635 Richmond Avenue, Houston, Texas 77042. I'm really ready. Send my Interlude today.
Apple II (16K)* ☐ Cassette ($18.95) ☐ Diskette ($21.95) ☐ Diskette — Pascal or DOS 3.3 ($21.95)
TRS-80 (Model 1, 16K)** ☐ Cassette ($18.95) ☐ Diskette ($21.95)
Poster ☐ 20"x 24" reproduction of this ad without ad copy ($4.95—includes shipping charges)
Available for immediate shipment.
Please enclose your check payable to INTERLUDE or complete the charge information:
Add $1.75 for shipping and handling.
☐ MASTERCARD ☐ VISA All charge customers must sign here.
Account No. Expiration date MasterCard Bank Code
CHARGE CUSTOMERS: Order by phone toll-free! **1-800-231-5768 Ext. 306** (Texas: 1-800-392-2348 Ext. 306)
Name Age
Address
City State Zip
*Apple II is a registered trademark of Apple Computers, Inc. **TRS-80 is a registered trademark of Radio Shack, a Tandy Co.

Program continued

```
1400 A$=INKEY$:IFA$=""THEN1400
1410 IFA$="*"THEN290
1420 IFA$="/"THEN1590
1430 A=ASC(A$)
1440 IFA=13THENPRINT@CU,CB$;:GOTO1490
1450 IFA<>8THEN1470ELSEPRINTA$;:IFCU>424THENX=X-1:CU=CU-2
1460 PRINT@CU,CU$;:GOTO1390
1470 IFA<65ORA>90THEN1390
1480 X=X+1:PRINT@CU,A$;CU$;:CU=CU+2:U$(X)=A$:GOTO1390
1490 FORX=1TOL:U(X)=0:IFU$(X)<>""THENAL=ASC(U$(X)):A(AL)=0
1500 IFW$(X)=U$(X)THENW(X)=1:U(X)=2
1510 NEXT
1520 FORX=1TOL:FORY=1TOL:IFU(X)<>0THEN1560
1530 IFU$(X)<>W$(Y)THEN1550
1540 IFW(Y)=0ANDU(X)=0THENW(Y)=2:U(X)=5
1550 NEXTY
1560 NEXTX
1570 GOSUB1710:FORX=1TOL:IFW(X)<>1THEN1640ELSENEXTX
1580 PRINT@512,"YOU HAVE GUESSED THE WORD";CHR$(30);:POKEML,0:PO
KEML+15,13:Z8=USR(0):GOTO1600
1590 PRINT@512,"THE WORD WAS  ";H$(H);CHR$(30);:POKEML,0:POKEML+
15,12:Z8=USR(0)
1600 PRINT:INPUT"PLAY AGAIN (Y) OR GO TO MENU (*)";Y$:CLS:Y$=LEF
T$(Y$,1)
1610 IFY$="*"THEN310
1620 IFY$="Y"THEN1310
1630 PRINT@576,STRING$(32,32);:PRINT@576,"";:GOTO1600
1640 GOSUB1660:FORX=1TOL:W(X)=0:NEXT:PRINT@384,STRING$(32,32);:G
OTO1380
1650 REM ALPHABET
1660 PRINT@838,"";
1670 FORX=65TO90:IFA(X)=0THENPRINT" ";:GOTO1690
1680 PRINTCHR$(X);
1690 NEXT:RETURN
1700 REM UPDATE GUESS-A-WORD
1710 PRINT@0,"NUMBER OF TRIES =";TR;:TR=TR+1
1720 PRINT@P,"";:FORX=1TOL
1730 IFW(X)=1THENPRINTW$(X);:GOTO1750
1740 PRINT"-";
1750 NEXTX:PRINT@256,STRING$(32,32);:PRINT@256,"CORRECT BUT WRON
G PLACE-";
1760 FORX=1TOL:IFU(X)=5THENPRINTU$(X);
1770 NEXTX:FORX=1TO9:U(X)=0:NEXT:RETURN
1780 DATA FREEDOM,TONGUE,ANTLER,LANTERN,CHAIR,VANDAL,CLOWN,PRESI
DENT,BALMY,FIXTURE,FEATURE,PRANCE,BELT,CURVE,SECOND,STATUE,CORNE
R,BLINK,HAIRY,PLEASANT,PHEASANT,THUNDER,TORNADO,WEATHER,WHETHER,
HURRICANE,TEMPLE,FACTORY
1790 DATA BLUBBER,MIDGET,DWARF,FRAUD,GORILLA,ANTELOPE,GIRAFFE,FE
NCE,RACCOON,NORMAL,TOOTH,BUTTON,COFFEE,WATER,BUFFALO,COMET,SOLAR
,PANIC,NEEDLE,HYDRANT,FAUCET,TOILET,STADIUM,EMPATHY,SYMPATHY,CIT
ATION,CUPBOARD,AUTO
1800 DATA VEHICLE,TURNPIKE,GRANDIOSE,OFFENSIVE,CABINET,PRACTICE,
STROBE,SLIDING,NECKLACE,GHOST,CROP,JEWELRY,TEMPERATE,SOCKET,LOCK
ET,POCKET,WRENCH,SENSATION,STUPID,GARDENER,COFFIN,LITERATE,FASHI
ON,GROOVE,METEORITE
1810 DATA ASTEROID,PLANET,CHEMICAL,MOTOR,SCREEN,POLICE,MACHINERY
,HAYSTACK,CHIME,MOVEMENT,ISLAND,DIGIT,GLACIER,BASEBALL,HOCKEY,TE
NNIS,FOOTBALL,SKATING,ANIMAL,LEGEND,FINAL,PLAIN,GOOD,BOOK,GRAPH,
RULER,CACTUS,CRAFTY,RADIO
1820 DATA SLANG,CLOSET,PLANT,ASTRONOMY,RECORD,PICTURE,BECKON,ELE
PHANT,ARCADE,SPLAT,PORCH,GEOLOGY,CEILING,BASEMENT,BATHROOM,MONKE
Y,BIRD,EXPANSION,KITCHEN,LIBRARY,METRIC,BRICK,MOOSE,HUMAN,NATURA
L,CERTAIN,MULCH,THRIFTY
1830 DATA EFFICIENT,HANDLE,TRUNK,STORMY,GALAXY,ENTRANCE,EDITION,
LEAF,CRUTCH,LIGHT,FLIGHT,SLEIGHT,RIVET,CARROT,SENTENCE,MAGAZINE,
PEBBLE,PENCIL,PLANK,WINDOW,FROZEN,FLEX,BALANCE,CLOCK,VIDEO,HELME
T,TWENTY,PERCENT,INTEREST
1840 DATA LENGTH,COMPUTER,RACK,SNEEZE,MOUTH,POSTERIOR,DEPOSIT,BA
NK,SCHOOL,COLLEGE,COLLAGE,DENIZEN,FLAME,POROUS,TORPEDO,SLIPPER,S
OCK,SHIRT,CALCULATE,VEGETABLE,PEPPER,CLONE,STAND,RATIO,JACKET,VE
ST,VESTIBULE,WHEEL
1850 DATA PALMISTRY,BARBELL,UNIVERSE,SKIING,JUMPY,OBESE,ABSTRACT
,CHAMPION,ALLOWANCE,HOSPITAL,INSURANCE,DIRECTOR,PRODUCER,PORTION
,NOTICE,MOVIE,FLIMSY,CRAZY,CRAFTILY,MAGICAL,PROCEED,GALLEY,LEGAL
,RIVER,WATERFALL,CASCADE
1860 DATA INTENSE,PRETEXT,FORMATION,ROTATION,DISCOUNT,MUFFLER,PH
LEGM,PARENTAL,MONEY,WEALTHY,PLAUSIBLE,DESERT,CHART,HAWK,FAMOUS,C
AMERA,SLINKY,POSITIVE,NEGATIVE,TYRANT,HAPPEN,PRINCESS,BELLY,SUNS
ET,MOUNTAIN,VALLEY,BLANKET
1870 DATA PANTHER,TAXATION,FURNACE,VESSEL,FRAME,MUFFIN,HAMBURG,T
URBAN,EDUCATION,FLUSH,USEFUL,WHIM,CHEETAH,KANGAROO,OSTRICH,RAMIF
Y,CLOAK,BLISS,PYRAMID,CYLINDER,SCHISM,CHASM,METHOD,MENTION,BATTE
RY,CHOWDER,TENT,SLEEVE
1880 DATA CLOTHING,ADJUST,MAJESTIC,CHAMBER,STEWARD,CHOMP,SHIMMER
,DEFLATE,GROWTH,FERTILIZE,BANISH,BLEMISH,HORROR,FINGER,HISTORY,L
OOSEN,RIGHT,TOGETHER,SEPARATE,COMB,PROGRAM,ESTIMATE,CALLOUS,DIAG
ONAL
```

We've lowered the

No, the low CompuServe $5.00 per hour charge hasn't changed, but the world of information we offer is growing steadily.

Of CompuServe's 22 large computer systems, 10 are housed here in our Dublin, Ohio Computer Center.

More Computer Power

The number of CompuServe customers keeps growing—now more than 8,000 strong. The more customers we have, the more computer power and access lines we dedicate to our Information Service. Our 22 mainframe computer systems provide sufficient power and capacity to serve thousands more.

More News

In addition to the Associated Press, we've added more news. Read The New York Times and all the news that's fit to print, including Broadway show and movie reviews, stories from Pulitzer Prize-winning writers and more. We are continually adding new features to the information service, so check our "What's New" section regularly to see what other information is available.

TRS-80™ Color Computer and Videotex™

The CompuServe Information Service is sold in Radio Shack® Stores and, in addition to all the major brands of computers and terminals, can be used with the new TRS-80 Color Computer and Videotex terminal. Our special software formats the information in easy-to-read pages. Go forward, backward, anywhere in the CompuServe Information Service databases—even directly to a page of information.

Big System Reliability

When we say we're reliable, we mean it. In fact CompuServe computers were up and running for a 99.6 percent reliability factor last year. We've also built in many "invisible" features like data error detection and retransmission—all those technical enhancements which add up to the CompuServe Information Service being ready to use when you're ready to use it.

Shoot spuds in space down the tubers.

Space Potatoes

Christopher Brumme
Miller Abel
The Current Loop Suite 1990
190 South King St.
Honolulu, HI 96813

Perusing an old copy of *80 Microcomputing* one day, my partner and I happened upon a reference to the game of Space Potatoes. We were completely entranced; what could such a sport entail? After hours searching old newspaper clippings and discarded listings we were none the wiser.

We finally realized we would have to write our own. There must be enemy rockets manned by hostile vegetables from another dimension. We would be surrounded at our combat console, verniers and dials vying for our attention. As the attacking starch balls popped into our universe from their hyperspatial tube, we would fry them with our positron-injection beams. With these parameters in mind, we used the inside-out approach to program design, and immediately began coding!

With or Without Ketchup

When you—a seasoned space cadet—blast off to your assigned patrol, the fleet admiral will ask what difficulty you wish to undertake. Even if you choose a masochistic rating of 150, those conniving critters can concentrate their force elsewhere, leaving a mere sackful to attack your quadrant. After receiving your orders, you will find your scope filling up with french fry candidates. Maneuver them into your cross-hairs using the U, D, L and R keys, then give them the juice with the F key.

If you miss, they turn their eyes on you, causing heavy damage, which will be registered on your LED scale. Should one of those tricky 'taters veer off your screen, they will take repeated shots at your flanks with their spud guns. You have only a few seconds to get them back onto your screen.

Fuel is used up as you fire your thrusters or guns. You must become acquainted with maneuvering your vessel—the U key moves your cross-hairs up, and therefore, the veggies down. And if you run out of time, the whole galaxy will fall under their photosynthetic sway. If you sustain too much damage or run out of fuel, your ship will float, a lifeless hulk, ready for boarding by those terrible tubers.

When you run this program you will be confronted with the message SPACE POTATOES flashing before your eyes. This animated display is achieved by quickly switching between 32 and 64-character modes. Line 20 does successive Outs through port 255; an eight will convert to 32-character mode, a two converts back to 64-character mode. A timing loop of about 1/6 of a second slows down the mode switching. We found that anything faster than this can be positively hypnotizing.

At the end of the line we use the INKEY$ function to see if the user is ready to play. If you press any key, the INKEY$ function terminates the loop of line 20.

Idaho, Sweet or Yam

If you find this game a little difficult, you can make a few easy alterations. Line 860 determines the chances of the Space Potato evading you for a given game turn. (Game turn: one round of the program cycle which checks to see if you have entered a new move.) As the program is written, it has one chance in 10 of choosing one of the four directions. Since it may choose the direction in which it is already going, the odds are actually a little lower. Simply replace the 10 in line 860 with a higher number to make things easier.

You might also try to only allow the potato to avoid you in two directions rather than

"If you sustain too much damage or run out of fuel, your ship will float, a lifeless hulk, ready for boarding by those terrible tubers."

four. Line 870 picks which direction the Potato is going to travel. MV$ ( ) contains the four legal directions for a potato or a human—U, D, L and R. One direction is chosen using the random number generator built into Basic. This direction is put into C$. C$ normally contains the last key you pressed, allowing you to keep moving in the last direction chosen until either you or the potato change direction. Change the four in line 870 to a two and the potato will only be able to use two of the four legal moves for evasion, up or down.

The difficulty rating you choose determines, to within a randomly determined accuracy, the number of potatoes you must shoot down. Choosing fewer potatoes effectively gives you more fuel, time and allowable damage in which to shoot down each critter.

If you have a sound effects generator on your TRS-80, or an audio amplifier hooked onto your cassette interface, you might like to customize the program for sound. Line 650 is a good place to start, adding beeps for various alarms.

If your hardware is more sophisticated and you want some positron-injection beam noises, look at line 910. This code produces those diagonal lines of force which fly up the screen when you press the fire key. Put your GOSUB before the " :NEXT I" to make a sound for every two pixels. GOSUB at the beginning of the line will produce a single sound for the whole firing process. If you want a sound of rending tuber tissues to signify a hit, put a single sound at the beginning of line 700; if you wish to build the sound as the potato is fried, put your GOSUB before the ":NEXT I" in line 700.

You might like to try one final modification, if you have the lowercase modification for the Electric Pencil. You have probably noticed that if you run Basic with your machine set to lowercase you get a very strange character set. Code 7, or a G, looks rather like a space invader from the popular video game. Change the 140 in line 270 to a seven, and the Space Potato is transformed. If you like the look of that, you can let your imagination run wild. There are left and right lightning bolts to emulate positron-injection beam fire (they replace the 150 and 169 in line 980) and you can use those twisted arrows to point to a Potato when he is over the center of the crosshairs.

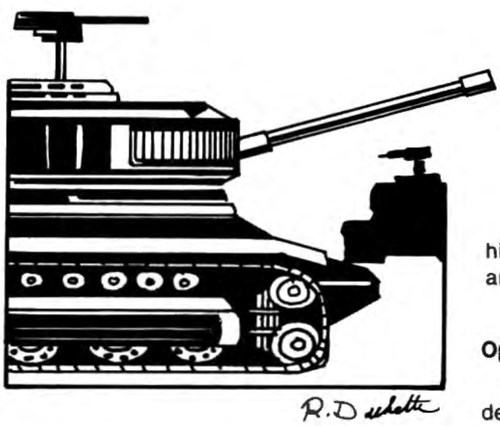
The biggest limitation of this game is its speed. We gained some increases by removing a lot of the REMarks and by using multiple statement lines wherever feasible. Short of rewriting in machine code, the only other alternative seems to be a Basic compiler. ■

A cassette of this game is available from the authors.

Program Listing 1

```
10 CLS:FORI=1TO17:PRINT:PRINT"S P A C E   P O T A T O E S";:NEXT
I
20 FORI=1TO50:NEXTI:OUT255,8:FORI=1TO50:NEXTI:OUT255,2:A$=INKEY$
:IFA$=""THEN20
30 DEFINT A-Z:DEFSNG D,G,F:DIMMV$(4),Z9(14,2):GOSUB 940
40 CLS:PRINT@340,"";:INPUT"DIFFICULTY";DI:CLS
50 Y0=0:Y9=14:X0=0:X9=28:RF=0:OD=0:OF=30
60 MV$(1)="U":MV$(2)="D":MV$(3)="L":MV$(4)="R"
70 C$=""
80 GOSUB190'           INIT SCREEN
90 GOSUB 720'          PLACE CRAFT
100 M$=INKEY$:IFM$=""THENM$=C$ELSE F9=F9-G1:IFM$<>"F"THENC$=M$
110 IFM$="F" GOSUB 320:GOTO 170
120 IFM$="U" Y3=1 : GOSUB 350:GOTO170
130 IFM$="D" Y3=-1 : GOSUB350:GOTO170
140 IFM$="L" X3=1 : GOSUB350:GOTO170
150 IFM$="R" X3=-1 : GOSUB350
160 FORI=1TOW9:NEXTI
170 GOSUB 460
180 GOTO100
190 FORI=15360TO16383:POKEI,191:NEXTI
200 FORI=0TO14:PRINT@(I*64),STRING$(29,32);:NEXTI
210 FORI=15374TO15551STEP64:POKEI,191:NEXTI
220 FORI=16142TO16332STEP64:POKEI,191:NEXTI
230 PRINT@448,STRING$(8,140);TAB(21)STRING$(8,140);
240 PRINT@96," TIME ";:PRINT@224," HITS ";:PRINT@352," MISS ";:P
RINT@480," FUEL ";:PRINT@672," DAMAGE ";
250 PRINT@544,STRING$(30,136);:PRINT@736,STRING$(30,32);
260 T9=999 : F9=30 : D9=0 : H9=0 : M9=0 : C9=INT(3*RND(DI))
270 HF=0 : MF=0 : SF=0 : A9=140 : G1=.1 : LF=0
280 GOSUB 500
290 RETURN
300 REM X+Y ---> POKE
310 P1=15360+Y1*64+X1:RETURN
320 REM FIRE
330 GOSUB 910 : IF X1=14 AND Y1=7 GOSUB 670 ELSE GOSUB 790
340 RETURN
350 IF SF=0 THEN POKE P1,B9
360 X1=X1+X3
370 Y1=Y1+Y3
380 X3=0:Y3=0
390 IF ((Y1>Y9 OR Y1<Y0) OR (X1>X9 OR X1<X0)) THEN SF=1 : RETURN
400 SF=0
410 GOSUB 310
420 B9=PEEK(P1)
430 IF B9=140 THEN A9=136
440 POKE P1,A9
450 RETURN
460 IFHF=1THENIFC9>0THENGOSUB720:GOTO500ELSELF=2:GOTO500
470 IFMF=1THENTF=2:GOSUB840:GOTO500
480 IFSF=1THENTF=1:GOSUB840:GOTO500
490 REM UPDATE SCREEN
500 A9=140 : T9=T9-1 : IFT9<=0 THEN LF=1
510 IF D9>=30 THEN LF=1
520 IF F9<=0 THEN LF=1
530 IFLF>0THEN890
540 PRINT@102,T9;TAB(41);
550 PRINT@230,H9;" ";
560 PRINT@358,M9;" ";
570 IFINT(OF)=INT(F9)THEN590
580 PRINT@INT(544+OF)," ";:OF=F9
590 IFINT(OD)=INT(D9)THEN610
600 PRINT@(INT(736+D9)),CHR$(140);:OD=D9
610 GOSUB 860:HF=0:MF=0
620 IF RF>0 THEN 650
630 IF D9>25 OR F9<5 OR T9<75 THEN RF=1
640 RETURN
650 IF RF=1 THEN PRINT@940," <RED> ";:RF=2 ELSE PRINT@940,STRING
$(7,191);:RF=1
660 RETURN
670 REM HIT
680 H9=H9+1
690 C9=C9-1
700 FORI=128TO191:POKE15822,I:NEXTI:POKE 15822,32
710 HF=1:RETURN
720 REM PLACE CRAFT
730 X1=RND(20)+4:Y1=RND(10)+2
740 GOSUB310
750 B9=PEEK(P1)
760 POKE P1,A9
770 C$=MV$(RND(4))
780 RETURN
790 REM MISS
800 M9=M9+1
810 MF=1
820 RETURN
830 REM HIT BACK
840 D9=D9+(RND(3)+TF)/5
850 RETURN
860 IF RND(10)<>1 THEN RETURN
870 C$=MV$(RND(4))
880 RETURN
890 PRINT@203."";:IF LF=1 THEN PRINT"YOU LOSE";:ELSE PRINT "YOU
WIN";
900 A$=INKEY$:IF A$<>" " THEN 900 ELSE 40
910 FOR I=1 TO 14 : POKE Z9(I,1),Z9(I,2) : NEXT I : POKE 15822,1
88
920 FOR I=1 TO 14 : POKE Z9(I,1),32 : NEXT I : POKE 15822,32
930 RETURN
940 ' SET UP FIRE TRAIL
950 L=16256:R=16284
960 FOR I=1 TO 14 STEP 2
970 Z9(I,1)=L : Z9(I+1,1)=R
980 Z9(I,2)=150 : Z9(I+1,2)=169
990 L=L-62 : R=R-66 : NEXT I
1000 RETURN
```

Preschoolers shoot'em up.



Tank

Dan L. Rice
138 Savannah Drive West
Bead, DE 19701

I think the preschool population is being ignored in the area of computer games. This point was recently brought to my attention by my preschooler, Keri. My 10 year old and I would spend countless hours playing computer games that Keri could not understand. Since I wanted her to enjoy our family computer as much as the rest of us do, I set out to design a game for preschoolers.

I decided the game must provide a challenge, be easily operated with a minimum of key usage and its rules should be easily understood and remembered.

The game I designed was named Tank by the youngest computer nut in my household, and judging by its acceptance, I succeeded in meeting my objectives.

Game Objective

The player has five turns to move a tank through an obstacle course to destroy five targets. If an obstacle is hit, the player loses one turn. The player wins if he or she destroys the five targets before hitting five obstacles.

Game Description

Five targets, represented by X's, are randomly placed on a field covered with obstacles (represented by graphics characters). The number of obstacles is controlled by the player's response to the difficulty level of play desired. By using the up, down, left and right arrow keys, the player moves the tank, represented by an H, around the obstacles to overlay each target. If he or she hits an obstacle, the player loses one turn, and the tank is repositioned on the field.

Operation

The player is asked what difficulty level is desired; easy, medium, or hard, to which E, M, or H is the response. The playing field is then filled with randomly selected graphics characters, less for the easy play level, more for the hard level. The five targets are placed randomly on the field, and the tank is randomly positioned around the field until any key is pressed. This signals the tank to stop and the game to begin.

If your computer system includes an expansion interface, the amount of time it takes each player to destroy the five targets is recorded. This will give you time scores to compare with other players.

Modification

The program is written so almost every variable can be modified to your own specifications. In order to change the number of enemy targets, you change the value of EN in statement 90. To change the number of turns the player gets, change the value of YR in statement 100. To change the target character from the X, change the value of TG in statement 110 to the decimal ASCII value of the desired character. To change the tank character from H, change the value of TK in statement 120 to the decimal ASCII value of the desired character. To eliminate the timing facility if you do not have expansion interface, set T$ equal to NO in statement 130. Finally, to change the number of graphics characters put on the field for each difficulty level, change the values assigned to DF in statements 720, 730 and 740.

I hope you enjoy this game. I think you will find there is a bit of preschooler in all of us, as you fight the kids off in order to continue playing the game yourself.■

Program Listing

```
10 ' ********************
20 ' ***** T A N K *****
30 ' ********************
40 ' *** WRITTEN  BY ***
50 ' *** DAN L. RICE ***
60 ' ********************
70 CLS:RANDOM:' CLEAR SCREEN AND RESEED RANDOM NUMBER GENERATOR
80 CLEAR 50: ' CLEAR STORAGE
90 EN=5: ' NUMBER OF ENEMY TARGETS
100 YR=5: ' NUMBER OF PLAYER TRIES
110 TG=88:' TARGET CHARACTER
120 TK=72: ' TANK CHARACTER
130 T$=" ": ' T$="NO" IF NO EXPANSION INTERFACE
140 GOSUB680: ' GET DIFFICULTY FACTOR
150 ' SET UP THE PLAYING FIELD
160 FORI=1TODF
170 SH=RND(63):LO=RND(1023)
180 LO=LO+15360
190 IF PEEK(LO)<>32THEN170
200 POKE LO,SH+128
210 NEXTI
220 ' SET UP THE TARGETS
230 FORI=1TOEN
240 X=RND( 960):IFPEEK(X+15424)<>32THEN240
```

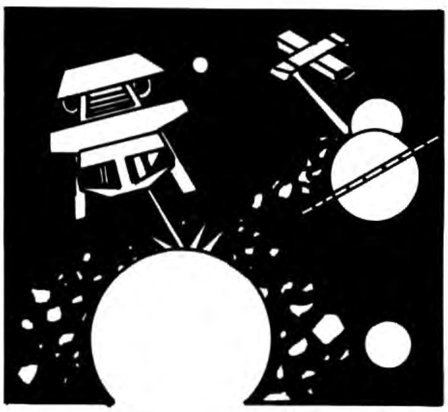
Program continues

Listing continued

```
250 POKEX+15424,TG:NEXTI
260 ' WAIT FOR SIGNAL TO START
270 H1=RND(1023):IFPEEK(H1+15424)<>32THEN270
280 H=H1+15424
290 Q$=INKEY$:IFQ$<>""THEN310
300 POKEH,TK:FORI=1TO50:NEXTI:POKEH,32:GOTO270
310 POKEH,TK
320 ' START THE GAME
330 IFT$="NO"THEN360
340 IFT1<>0THEN360
350 T$=RIGHT$(TIME$,5):T1=VAL(LEFT$(T$,2))*60+VAL(RIGHT$(T$,2))
360 PRINT @ 30,"ENEMY " EN " YOU " YR;
370 S$=INKEY$
380 IFT$="NO"THEN410
390 T$=RIGHT$(TIME$,5):T9=VAL(LEFT$(T$,2))*60+VAL(RIGHT$(T$,2))
400 PRINT @50,T9-T1;
410 IFS$=""THEN370
420 F=H
430 X=ASC(S$)
440 IFX=91 THENH=H-64:GOTO490
450 IFX=10 THENH=H+64:GOTO490
460 IFX=9  THENH=H+1:GOTO490
470 IFX=8  THENH=H-1:GOTO490
480 GOTO370
490 IFH<15360THENH=15360:GOTO530
500 IFH>16383THENH=16383:GOTO530
510 IFPEEK(H)=32THEN550
520 IFPEEK(H)=TGTHEN560
530 POKEF,32:FORI=1TO100:POKEH,32:POKEH,42:NEXTI:POKEH,32
540 YR=YR-1:IFYR=0THEN590  ELSE270
550 POKEF,32:POKEH,TK:GOTO360
560 POKEF,32:FORI=1TO100:POKEH,32:POKEH,TK:NEXTI
570 EN=EN-1:IFEN=0THENGOTO590
580 GOTO360
590 CLS:PRINTCHR$(23);:IF EN=0THEN 620
600 PRINT @06,"SORRY, YOU LOST";
610 GOTO650
620 PRINT @06,"I SURRENDER, YOU WIN";
630 IFT$="NO"THEN650
640 PRINT @70,"IT TOOK YOU " T9-T1 "SECONDS.";
650 PRINT@134,"PRESS ENTER TO CONTINUE"
660 Q$=INKEY$:IFQ$=""THEN660
670 GOTO70
680 PRINTCHR$(23);TAB(6)"ENTER -E- FOR EASY":PRINTTAB(6);"ENTER
-M- FOR MEDIUM"
690 PRINTTAB(6); "ENTER -H- FOR HARD"
700 Q$=INKEY$:IFQ$=""THEN700
710 CLS
720 IF Q$="E" THEN DF=100:RETURN
730 IF Q$="M" THEN DF=200:RETURN
740 IF Q$="H" THEN DF=300:RETURN
750 GOTO680
```

Be fruitful and colonize.

Space Empires



James L. Smith
1310 Acorn
Rogers, AR 72756

The date is 2501 A.D. Mankind is building colony ships to burst forth, explore, and colonize the universe. As Admiral-in-Chief you have until 2020 A.D. to explore and colonize 19 star systems; only 20 years. This is accomplished by spending megacredits each year (one turn = one year) to build colonies and move your space fleet to the next star system. Each turn is divided into phases as listed below.

Status Report

Each turn begins with a status report that tells how many megacredits you have earned from the previous year. Within a ten percent adjustment, you will receive five megacredits for each explored system, and 15 megacredits for each system you colonize. Megacredits left over from the previous year will be included. The status report includes the number of systems you have colonized, explored, and systems remaining to be explored.

Movement Phase

Each game, a different universe (playing grid) is randomly created. On the first turn you begin with one colony, where your fleet is based. Each turn that you move to a star system (•), it becomes an explored system; E. It will cost you seven megacredits each time you move. To move your fleet on the 10 by 10 grid you must type the y-axis first (the left hand number), then a comma, then type the x-axis and hit Enter. There are some movement limitations: You can only move two spaces during a turn, you must land on a star system, and you cannot move outside the universe (playing grid). If you make a wrong move you are still charged seven megacredits, and your fleet returns to your last base. Hints: remember, you only have twenty turns to explore and colonize nineteen systems. You can only afford one mistaken move per game. Plan ahead so you don't waste time by landing on systems you already own.

During future turns, various symbols will appear on the playing grid. O means a colony. E means an explored system. An asterisk * means an unexplored system. S shows the location of your ship.

Emergency Situations

On even numbered game turns, you will have to meet and cope with emergency situations. The cost of meeting each emergency will be given. However, you may gamble and spend less to conserve megacred-

Program Listing. Space Empires

```
5 CLS: PRINT@ 140, "* * * SPACE EMPIRES * * *"
6 PRINT TAB(22) "BY"
7 PRINT TAB(16) "JAMES L. SMITH"
8 PRINT:PRINT "THIS IS A GAME OF SPACE EXPLORATION. PLA
     YERS MUST MOVE"
9 PRINT "THEIR STAR FLEETS CORRECTLY AND ECONOMIZE WISE
     LY"
10 PRINT"TO COLONIZE ALL 20 STAR SYSTEMS AND WIN THE GA
     ME"
12 PRINT: PRINT"THIS GAME IS CURRENTLY FOR SALE ON A RO
     YALTY"
14 PRINT "AND / OR CASH BASIS.  CONTACT ME FOR AN AGREE
     MENT"
15 PRINT "J. L. SMITH---1310 ACORN---ROGERS ARK  72756-
     -636-6109"
16 PRINT: INPUT "ARE YOU READY? HIT ENTER FOR THE FUN!"
     ; U
18 CLEAR 500
20 CLS: T=2500: A$="#,###.##"
25 S$="C": RANDOM
30 DIM G$(10,10)
35 PRINT @ 206, "------ONE MOMENT PLEASE------"
37 PRINT @ 270, "WE ARE CREATING THE UNIVERSE"
40 G$(1,1)="*"
43 C8=1:C9=1
46 FOR I = 1 TO 19
```

Program continues

"This is how you win the game. You must build colonies as quickly as you can and pyramid your revenue."

its. If your gamble does not work, or you spend nothing, you must suffer the consequences.

Possible emergencies are:

1.) Fleet repairs and supplies: If you neglect this one, your ship does not move next turn.

2.) Revolts: Star Systems require certain expenditures of megacredits to maintain. If you gamble and lose or pay nothing, then the system is in revolt. You do not receive revenue from systems in revolt. Explored systems will always revolt first. If there are no explored systems, then a colony will revolt. Colonies that revolt must be rebuilt. If you have no more systems left, you lose the game.

3.) Budget Cuts: Eventually, you must deal with the bureaucracy. The world council will cut your budget by 1/6. This is not so bad early in the game but later it can mean bad news.

4.) Natural Disasters: At one point during the game, if you are unlucky, a colony will be destroyed and revert to unexplored status. It will have to be rebuilt.

Build Colonies

This is how you win the game. You must build colonies as quickly as you can and pyramid your revenue. Only explored systems may be colonized. Since it costs 90 megacredits to build a colony, your first build should occur on turn six or seven. You can build none, one, or more than one colony per turn if you have the loot. There is a slight pause in the program here before the next game turn occurs.

Space Empires is never a sure win. Like life, there are many variables. If you have developed a good strategy it will increase your chances. I could give you a few hints but that would spoil the fun. Develop your own! ■

5 - 30	Prepare for the game
35- 80	Create the Universe (playing grid)
100-220	Compute the revenue for present turn
230-325	Print the Status Report
340-525	Print the grid and call for movement
535-700	Test movement accuracy and move fleet
710-890	Determine and print emergencies
940-1060	Build Colonies
1070-1110	Test for win/lose or return for next turn
2000-2090	Subroutine for fleet emergencies
2500-2699	Bring systems out of revolt
2710-2799	Test for system where fleet landed
2800-2999	Winners display
3000-3060	Reduce colonies

Table 1. Program Structure

Program listing continues

```
48 H9=RND(3): V9=RND(3)
50 IF H9=1 GOTO 60
53 IF H9=2 AND C8<9 THEN D8=2: GOTO 60
56 IF H9=3 AND C8>2 THEN D8=-2: GOTO 60
58 GOTO 48
60 IF V9=1 GOTO 70
62 IF V9=2 AND C9<9 THEN D9=2: GOTO 70
64 IF V9=3 AND C9>2 THEN D9=-2: GOTO 70
70 C8=C8+D8: C9=C9+D9
72 IF G$(C8,C9)="*" THEN D8=0: D9=0: GOTO 48
74 IF I=19 THEN G$(C8,C9)="C": I1=C8: J1=C9: GOTO 78
76 G$(C8,C9)="*"
78 D8=0: D9=0
80 NEXT I
100 T=T+1: Z=9: Y=0
110 FOR I = 1 TO 10
120 FOR J = 1 TO 10
125 IF G$(I,J)="." THEN 180
130 IF G$(I,J)="*" THEN S1=S1+1: GOTO 180
140 IF G$(I,J)="E" THEN E1=E1+1: GOTO 180
150 IF G$(I,J)="C" THEN C1=C1+1: GOTO 180
160 IF G$(I,J)="S" THEN I1=I: J1=J: GOSUB 2710: GOTO 18
    0
170 IF G$(I,J)="R" THEN R1=R1+1
180 NEXT J
190 NEXT I
200 ET=E1*5: CT=C1*15
205 G1=RND(20): G1=G1/100 + .90
210 G2=RND(20): G2=G2/100 +.90
220 ET=ET*G1: CT=CT*G2
230 CLS: PRINT "CIRCA";T;"A.D.";
240 PRINT TAB(25) "FEDERATION STATUS REPORT": PRINT: PR
    INT
245 PRINT TAB(10) "UNEXPLORED STAR SYSTEMS"; S1
250 PRINT TAB(10) "SYSTEMS IN REVOLT      "; R1: PRINT
260 PRINT TAB(10) "REVENUE FROM";E1;"EXPLORED SYSTEMS";

270 PRINT USING A$; ET
280 PRINT TAB(10) "REVENUE FROM";C1;"COLONY SYSTEMS  ";

290 PRINT USING A$; CT
293 PRINT TAB(10) "BALANCE FORWARD                 ";
294 PRINT USING A$; TC
295 TC=TC+ET+CT
300 PRINT: PRINT TAB(10) "TOTAL FEDERATION REVENUE
    ";
310 PRINT USING A$; TC: PRINT
320 PRINT TAB(10)"REPORT COMPLETE - HIT ENTER";
325 INPUT U: CLS
330 IF X=1 GOTO 700
335 IF (TC+.005)<7 PRINT "SORRY COMMANDER - YOU'RE SHOR
    T OF FUNDS - NO MOVEMENT": GOTO 700
340 PRINT "CIRCA: ";T;"A.D.";
350 PRINT TAB(22)"SPACE EMPIRES";
360 PRINT TAB(42)"MEGACREDITS";
370 PRINT USING A$;TC: PRINT
380 FOR J = 1 TO 10
390 PRINT TAB(Z) J;
400 Z=Z+4
410 NEXT J
420 Z=0: Z=5: Y=1
430 FOR I = 1 TO 10
440 PRINT CHR$(10) TAB(Z) Y;
450 Z=Z+1
460 FOR J = 1 TO 10
470 Z=Z+4
475 IF G$(I,J)="" PRINT TAB(Z) ".";: GOTO 490
480 PRINT TAB(Z) G$(I,J);
490 NEXT J
500 Z=0: Z=5: Y=Y+1
510 NEXT I
520 PRINT: PRINT: PRINT TAB(9) "FLEET MOVEMENT PHASE (V
    ,H)";
525 INPUT V,H: CLS
535 IF V=I1 AND H=J1 PRINT "NO FLEET MOVEMENT THIS PHAS
    E - AS ORDERED": GOTO 700
536 IF V>10 OR V<1 PRINT "YOUR FLEET WAS ORDERED OUT OF
     THE UNIVERSE": GOTO 1140
537 IF H>10 OR H<1 PRINT "YOUR FLEET WAS ORDERED OUT OF
     THE UNIVERSE": GOTO 1140
```

Program continues

Program listing continues

```
540 TC=TC-7
550 IF G$(V,H)="" GOTO 680
560 IF V>I1 THEN I2=V-I1: GOTO 580
570 I2=I1-V
580 IF H>J1 THEN J2=H-J1: GOTO 600
590 J2=J1-H
600 IF I2>2 GOTO 680
610 IF J2>2 GOTO 680
617 IF S$="*" THEN S$="E"
620 G$(I1,J1)=S$
625 S$=G$(V,H)
630 G$(V,H)="S"
635 IF S$="E" OR S$="*" THEN E1=E1+1
637 IF S$<>"*" PRINT "YOU HAVE LANDED ON ONE OF YOUR SY
    STEMS": GOTO 670
640 PRINT "YOU HAVE SUCCESSFULLY EXPLORED ANOTHER SYSTE
    M"
670 GOTO 700
680 PRINT: PRINT"SORRY COMMANDER. AN INCORRECT SET OF C
    OORDINATES"
690 PRINT" WAS ORDERED.  YOUR FLEET RETURNS TO PLANET F
    ALL !"
700 PRINT: IF X=1 CLS: PRINT"YOUR FLEET IS STILL IN POR
    T": GOSUB 2000
710 IF R1>0 GOSUB 2500
720 IF INT(T/2)<>T/2 GOTO 900
740 PRINT: PRINT TAB(15) "* * * EMERGENCY SITUATION * *
    *": PRINT
750 G4=RND(6):
755 IF G4=1 AND X=1 OR G4=2 AND X=1 GOTO 750
757 IF G4=3 AND E1=0 AND C1=0 OR G4=4 AND E1=0 AND C1=0
     GOTO 750
758 IF G4=6 AND B3=1 OR G4=6 AND C1<=1 GOTO 750
760 ON G4 GOTO 770,780,790,800,810,825
770 PRINT "YOUR FLEET NEEDS FUEL AND SUPPLIES": GOSUB 2
    000: GOTO 900
780 PRINT "YOUR FLEET IS IN NEED OF REPAIRS THIS YEAR"
785 GOSUB 2000: GOTO 900
790 PRINT "A STAR SYSTEM NEEDS MEDICAL AID": GOTO 840
800 PRINT "A STAR SYSTEM IS IN REVOLT": GOTO 840
810 PRINT "THE WORLD COUNCIL DECREASES YOUR PRESENT ALL
    OTMENT OF"
814 PRINT "MEGACREDITS BY 1/6 FOR THIS YEAR": TC=INT(TC
    -(TC*.16))
816 PRINT " YOUR CURRENT BALANCE IS";TC;"MEGACREDITS.":
     GOTO 900
825 PRINT " ONE OF YOUR COLONIES HAS BEEN DESTROYED BY
    A NATURAL"
827 PRINT "DISASTER.  IT IS REDUCED TO EXPLORED STATUS.
    "
829 GOSUB 3000: B3=B3+1: GOTO 900
840 R1=R1+1
850 IF E1=1 AND S$="E" OR E1=1 AND S$="*" THEN E1=E1-1:
     S$="R": GOSUB 2500: GOTO 900
853 IF E1=0 GOTO 882
855 FOR I = 1 TO 10
860 FOR J = 1 TO 10
863 IF G$(I,J)="E" THEN E1=E1-1: GOTO 890
870 NEXT J
880 NEXT I
881 PRINT "ERROR IN REVOLT SEQUENCE AT 881": STOP
882 IF C1=1 AND S$="C" THEN S$="R": C1=C1-1: GOSUB 2500
    : GOTO 900
883 FOR I = 1 TO 10
884 FOR J= 1 TO 10
885 IF G$(I,J)="C" THEN C1=C1-1: GOTO 890
887 NEXT J
888 NEXT I
889 PRINT "ERROR IN REVOLT SEQUENCE AT 889": STOP
890 G$(I,J)="R":  GOSUB 2500
900 PRINT: PRINT TAB(15) "+ + + BUILD COLONIES + + +"
910 PRINT: PRINT"IT COST 90 MEGACREDITS TO BUILD A COLO
    NY."
920 PRINT "YOU HAVE";TC;"MEGACREDITS REMAINING."
930 PRINT "YOU HAVE";E1;"SYSTEMS TO COLONIZE AT PRESENT
    ."
940 INPUT "HOW MANY COLONIES DO YOU WISH TO BUILD"; B
945 IF B=0 GOTO 1060
950 B1=B*90
960 IF B1>(TC+.005) PRINT "THAT EXCEEDS YOUR BUDGET": G
    OTO 940
964 IF B>E1 PRINT "YOU CAN ONLY BUILD";E1;"COLONIES": G
    OTO 940
970 TC=TC-B1
980 IF E1=1 AND S$="E" OR E1=1 AND S$="*" GOTO 1052
990 FOR I = 10 TO 1 STEP -1
1000 FOR J = 10 TO 1 STEP -1
1010 IF G$(I,J)="E" GOTO 1040
1020 NEXT J
1030 NEXT I
1032 PRINT"ERROR IN BUILD COLONIES AT 1032": STOP
1040 G$(I,J)="C": L1=L1+1: C1=C1+1: E1=E1-1
1050 IF B>L1 GOTO 980
1051 GOTO 1060
1052 S$="C": L1=L1+1: C1=C1+1: E1=E1-1
1054 IF B>L1 GOTO 980
1060 Z2=0: PRINT "NUMBER OF COLONIES BUILT:"; L1
```

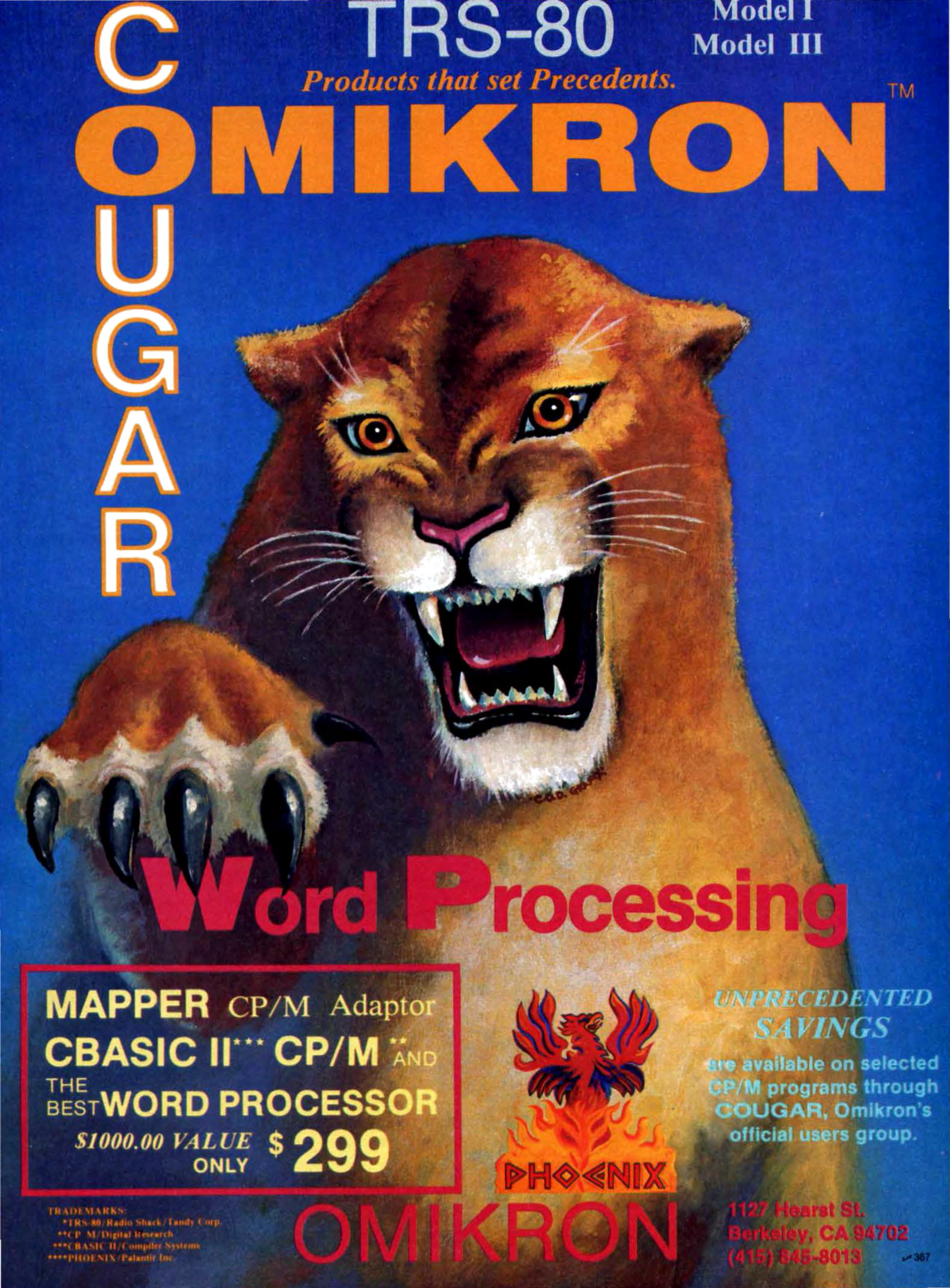
Program continues

✓See List of Advertisers on page 322

Program listing continues

```
1070 IF C1=>20 GOTO 1110
1080 IF T=2520 GOTO 1140
1085 IF E1=0 AND C1=0 PRINT "SORRY - YOU HAVE LOST THE
     GAME": GOTO 1140
1087 IF TC<0 PRINT "SORRY - YOU HAVE LOST THE GAME": GO
     TO 1140
1090 S1=0: E1=0: C1=0: R1=0: CT=0: J2=0: B1=0: ET=0: L1
     =0: I2=0
1100 GOTO 100
1110 CLS: GOSUB 2800
1120 PRINT: PRINT "YOU WIN THE ASIMOV EXPLORATION AWARD
      FOR COLONIZING ALL SYSTEMS"
1130 GOTO 1160
1140 PRINT: PRINT "* * OUT OF 20 SYSTEMS YOU COLONIZED"
     ;C1;"STAR SYSTEMS. * *"
1150 PRINT"HOWEVER, YOU DID NOT REACH YOUR GOAL.  SORRY
     , YOU LOOSE."
1160 INPUT "DO YOU WISH TO PLAY AGAIN (Y/N)"; Q$
1170 IF Q$="Y" GOTO 18
1180 END
2000 G1=RND(20): G1=(G1/100+.90)*20: AE=INT(G1)
2005 PRINT "TO REACTIVATE YOUR FLEET YOU CAN SPEND";AE;
     "MEGACREDITS."
2010 PRINT "YOU MAY SPEND LESS, BUT IT WILL BE A GAMBLE
     ."
2020 PRINT "YOU CURRENTLY HAVE";TC;"MEGACREDITS."
2030 INPUT "HOW MUCH DO YOU WISH TO SPEND."; P
2032 IF P=0 GOTO 2070
2033 IF P>(TC+.005) PRINT "THAT EXCEEDS YOUR BUDGET.":
     GOTO 2030
2035 TC=TC-P
2040 IF P=>AE GOTO 2080
2050 P1=P/AE: G3=RND(0)
2060 IF G3<P1 GOTO 2080
2070 X=1: PRINT"SORRY, YOUR FLEET IS STILL DRY DOCKED."
     : GOTO 2090
2080 X=0: PRINT "YOUR FLEET IS NOW OPERATIVE."
2090 P1=0: AE=0: RETURN
2499 END
2500 PRINT:PRINT "NUMBER OF SYSTEMS IN REVOLT:";R1
2505 G2=RND(20): G2=(G2/100+.90)*20: AE=INT(G2)
2510 PRINT "YOU CAN SPEND UP TO";AE;"MEGACREDITS ON EAC
     H ONE."
2520 PRINT "YOU MAY GAMBLE BY SPENDING LESS ON EACH ONE
     ."
2525 PRINT "YOU HAVE";TC;"MEGACREDITS REMAINING."
2530 INPUT "HOW MUCH DO YOU WISH TO SPEND ON ONE SYSTEM
     "; P
2532 IF P=0 GOTO 2570
2533 IF P>(TC+.005) PRINT"THAT EXCEEDS YOUR BUDGET.": G
     OTO 2530
2535 TC=TC-P
2540 IF P>=AE GOTO 2580
2550 P1=P/AE: G3=RND(0)
2560 IF G3<P1 GOTO 2580
2570 PRINT "SORRY, YOUR SYSTEM IS STILL IN REVOLT.": GO
     TO 2660
2580 IF S$="R" THEN R1=R1-1: S$="E": E1=E1+1: GOTO 2633

2585 FOR I = 1 TO 10
2590 FOR J = 1 TO 10
2600 IF G$(I,J)="R" GOTO 2630
2610 NEXT J
2620 NEXT I
2625 PRINT "ERROR IN SUBROUTINE 2500": STOP
2630 R1=R1-1: G$(I,J)="E": E1=E1+1
2633 PRINT "YOUR SYSTEM IS BACK TO NORMAL."
2635 IF R1=0 GOTO 2660
2640 INPUT "DO YOU WISH TO SAVE ANOTHER SYSTEM (Y/N)";
     Q$
2650 IF Q$="Y" THEN P1=0: GOTO 2530
2660 P1=0: AE=0: RETURN
2699 END
2710 IF S$="C" THEN C1=C1+1: GOTO 2730
2715 IF S$="R" THEN R1=R1+1: GOTO 2730
2720 E1=E1+1
2730 RETURN
2799 END
2800 PRINT TAB(30) "#"
2810 PRINT TAB(30) "#"
2820 PRINT TAB(30) "#"
2839 PRINT TAB(28) "@@@@@"
2840 PRINT TAB(26) "@@@@@@@@@"
2850 PRINT TAB(26) "@@@@@@@@@"
2860 PRINT TAB(26) "@@@@@@@@@"
2870 PRINT TAB(26) "@@@@@@@@@"
2880 PRINT TAB(24) "@@@@@@@@@@@@@"
2890 PRINT TAB(22) "@@@@  '''''  @@@@"
2900 PRINT TAB(19) "@@@@@@@  '''''  @@@@@@@"
2910 PRINT TAB(28) "''''''"
2995 RETURN
2999 END
3000 FOR I = 1 TO 10
3010 FOR J = 1 TO 10
3020 IF G$(I,J)="C" GOTO 3050
3030 NEXT J
3040 NEXT I
3045 PRINT "ERROR IN LINE 3045": STOP
3050 G$(I,J) = "E": C1=C1-1: E1=E1+1
3060 RETURN
```



Model I
Model III
TRS-80
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

4K and Level II.

Simon

Dave McGlumphy
4429 Paula Lane
Red Bank, TN 37415

I wrote this program to simulate the electronic game Simon on a TRS-80 Level II 16K computer, and later adapted it for use on a Level II 4K machine. I wanted to keep the program small so it could load quickly into 4K. I used several programs which have appeared in previous issues of *80 Microcomputing* to assist me.

Taking out the remark statements makes the program more difficult to understand, but this article should clear up the logic. I assume that owners of Level II machines have heard of Simon and understand that you win by echoing the same blocks the computer flashes for you.

Line 0 sets the memory size to 4K minus the amount of memory needed by the Super Sound program which is POKEd into memory by lines 1 through 3. When you print the memory you'll find only a small amount left even if you are using a 16K machine. Line 4 sets all variables with names beginning with the letters C through E and G through Z to integers to let the program run faster. Line 5 defines all the variables that begin with the letters B or F as strings. I used five elements of the string array B to print a block of light on the screen, and five elements of the string array F to print a frame of light which would contain a number.

Line 6 clears the screen, then sets the computer to print large characters by the CHR$(23) command. Lines 7 through 11 give minimal directions to the player, and line 12 waits for the player's response. Since the player need type only one character, I chose to use the INKEY$ function to eliminate the need for the enter key. The computer won't continue until the player presses a number from one to nine.

Line 13 prints the player's selection on the screen. Line 14 provides audible feedback to the player with two sounds, high pitched and low pitched. (See page 130 of the May, 1980, issue of *80 Microcomputing* for "Super Sound" and page 110 of the April issue for how to modify your CTR-41 to produce sound.) Line 15 assigns the numeric value of the string I$ to the variable named DI which stands for DIfficulty. Line 16 puts 11 graphic characters into the string element F(1). The graphic character is the complete block of six pixels. Line 17 creates a portion of the frames by making the string element F(2) equal to a complete graphic block followed by nine spaces followed by another graphic block. Line 18 completes the building of a frame. The frame is printed on five lines of the screen. Line 19 builds a block by putting the first element of the frame array, F, into all five elements of the block array.

Then comes the problem portion of the program. The array P contains the problem to be solved as soon as line 20 executes. Lines 21 and 22 print headings on the screen. Line 23 prints four frames by changing N from one to four and performing line 56. N tells the subroutine at line 56 which frame to draw. Line 24 is a delay loop to improve the appearance of the program. Line 25 establishes the number of repetitions required to beat the computer, establishes a For. . .Next loop to display the correct number of elements from the problem array, and changes the problem array each time if the user has selected a difficulty level greater than five. Difficulty levels one through five repeat the sight and sound sequences for each repetition, whereas the higher difficulty levels change the sequence in each round of play.

Line 26 sets the variable N to the Lth element of the problem array and then per-

Program Listing 1.

```
0 POKE16561,224:POKE16562,79:CLEAR200
1 AD=20451:HI=INT(AD/256):POKE16527,HI:POKE16526,AD-HI*256
2 FORI=ADTOAD+28:READDT:POKEI,DT:NEXTI
3 DATA 205,127,10,62,1,14,0,237,91,61,64,69,47,230,3,179,211,255
,13,40,4,16,246,24,242,37,32,241,201
4 DEFINTC-E,G-Z
5 DEFSTRB,F
6 CLS:PRINTCHR$(23)" DAVE MCGLUMPHY'S VERSION OF"
7 PRINT@128,"S I M O N   W I T H   S O U N D";
8 PRINT@384,"WHAT LEVEL OF DIFFICULTY WOULD"
9 PRINT"YOU LIKE TO TRY?"
10 PRINT@576,"1 IS THE EASIEST AND 9 IS THE   HARDEST."
11 PRINT@768,"TYPE A NUMBER FROM 1 TO 9. ";
12 I$=INKEY$:IFI$<"1"ORI$>"9"THEN12
13 PRINTI$;
14 SS=USR(4096+20):SS=USR(4096+40)
15 DI=VAL(I$)
16 F(1)=STRING$(11,191)
17 F(2)=CHR$(191)+STRING$(9,128)+CHR$(191)
18 F(3)=F(2):F(4)=F(2):F(5)=F(1)
19 FORJ=1TO5:B(J)=F(1):NEXTJ
20 FORJ=1TO10:P(J)=RND(4):NEXTJ
21 CLS:PRINT"D A V E   M C G L U M P H Y ' S   V E R S I O N  O
F  S I M O N"
22 PRINT"COPYRONG 1980                                    DIFFICULTY
 LEVEL";DI
23 FORN=1TO4:GOSUB56:NEXTN
```

Program continued

"I wrote this program because I couldn't afford to buy a $20 gizmo to do the same thing . . ."

forms the subroutine at line 57, which prints a large block of light on the screen at position N. Lines 27 and 28 set the variables SC (Sound Code) and DU (Duration) to produce the desired sounds when the subroutine at line 59 is performed. Line 29 is a delay loop to improve the program's appearance. Line 30 causes the block at position N to be changed to a numbered frame.

Line 31 causes an extra frame to be displayed on the screen to provide visual and audio distraction, increasing the difficulty of the game. Line 32 terminates the For. . . Next loop which displays the problem for the player. Line 33 ensures that if the player accidentally presses a key too soon, it won't be counted as part of his answer.

Lines 34–40 get the player's answer or reply. Line 35 ensures that the player enters a valid response. Line 36 displays a block at position N as the player requests. Line 37 checks to see if the player answered in the correct sequence, and if he didn't, a distinctive losing sound is produced and program control is sent to line 48 which tells the player that he lost.

Lines 38 and 39 set variables to produce appropriate sounds for successful guesses. Line 40 terminates the player's input routine. Line 41 tells the player how far he's gotten, and line 42 delays program execution before allowing the next portion of the problem to be displayed. Line 43 terminates the mainline portion of the program. Lines 44–50 tell the player whether he won or lost and ask if he wants another game. Line 51 waits for a Y or N to be keyed. If the player keys Y, a two-pitched sound is produced and the computer runs the program again. If the player types N, the program plays a little tune by executing lines 53–55, and then quits.

Line 56 is the subroutine which actually draws a frame on the screen and then numbers the frame from one to four. Line 57 is similar to line 56, but it draws a block of light on five lines instead of a numbered frame. Both lines 56 and 57 use the variable N to determine which of the four frames or blocks to print, and where to print them. Line 58 prints a frame with a random number from one to four at a random location below the frames or blocks produced by lines 56 and 57. This is to distract the player. The last line, line 59, produces sounds based on the level of difficulty, sound code and duration given to it by the calling lines in the mainline portion of the program.

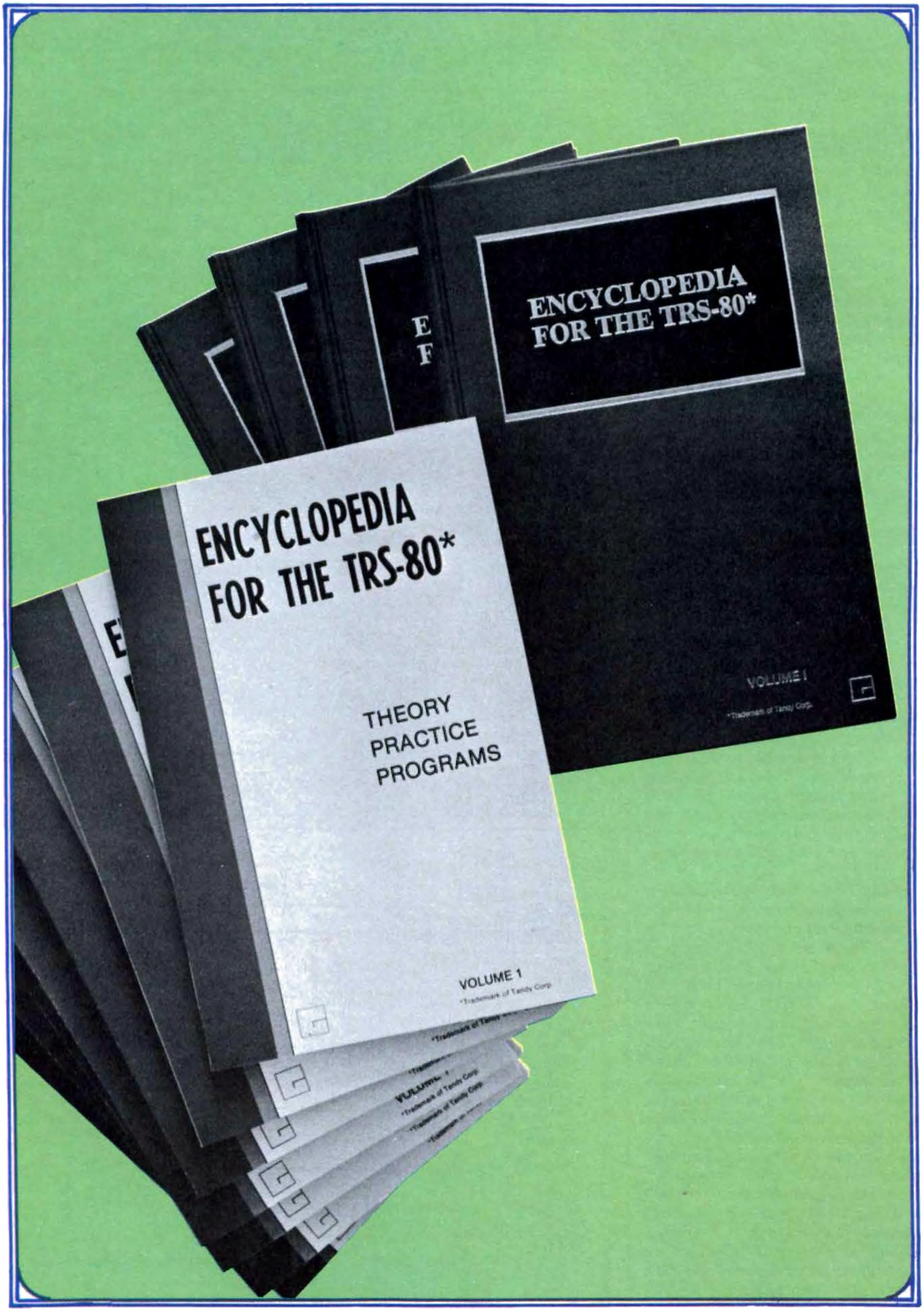
I wrote this program because I couldn't afford to buy a $20 gizmo to do the same thing when I already had a computer handy. I'll be delighted to respond to questions you may have if you choose to write and send a SASE. I would especially like to hear from someone who has written his own version of Merlin.■

```
24 FORN=1TO400:NEXTN
25 FORJ=1TO8:FORL=1TOJ:IFDI>5THENP(L)=RND(4)
26 N=P(L):GOSUB57
27 SC=8-N
28 DU=64*256:W=0:GOSUB59
29 FORK=1TO100-5*N:NEXTK
30 GOSUB56
31 IFDI=5ORDI=9THENS=RND(50)+640:GOSUB58:GOSUB59:FORK=1TO100:NEX
TK:PRINT@704,CHR$(31)
32 NEXTL
33 I$=INKEY$:I$=""
34 FORL=1TOJ
35 I$=INKEY$:IFI$<"1"ORI$>"4"THEN35
36 N=VAL(I$):GOSUB57
37 IFN<>P(L)THENFORM=1TO100:SS=USR(288):SS=USR(384):NEXTM:GOTO48

38 SC=8-N:DU=6*256:W=10:GOSUB59
39 GOSUB56
40 NEXTL
41 PRINT@208,"NUMBER OF SUCCESSFUL ECHOES =";J;
42 FORL=1TO400:NEXTL
43 NEXTJ
44 CLS:PRINTCHR$(23);
45 PRINT"      CONGRATULATIONS!!"
46 PRINT@192,"    YOU BEAT THE COMPUTER."
47 FORJ=1TO10:SS=USR(4096+20):SS=USR(4096+40):SS=USR(4096+60):NE
XTJ:GOTO49
48 PRINT@768,"SORRY, BUT YOU LOST."
49 PRINT
50 PRINT"WOULD YOU LIKE TO TRY AGAIN?";
51 I$=INKEY$:IFI$="Y"THEN52ELSEIFI$="N"THEN53ELSEGOTO51
52 CLS:SS=USR(4096+20):SS=USR(4096+40):RUN
53 FORJ=1TO4:READDU,PI:GOSUB55:NEXTJ:CLS:END
54 DATA 40,60,20,55,30,60,40,50
55 SS=USR(256*DU+PI):RETURN
56 FORQ=1TO5:PRINT@240+Q*64+N*17,F(Q);:NEXTQ:PRINT@436+17*N,N;:R
ETURN
57 FORQ=1TO5:PRINT@240+Q*64+N*17,B(Q);:NEXTQ:RETURN
58 FORQ=1TO5:PRINT@S+Q*64,F(Q);:NEXTQ:PRINT@S+196,RND(4);:RETURN

59 IFDI=1ORDI=6THENSS=USR(DU+SC*10)ELSEIFDI=2ORDI=7THENSS=USR(DU
+50)ELSEIFDI=3ORDI=8THENSS=USR(DU+SC*10+W)ELSESS=USR(DU+RND(30)+
20)
60 RETURN
```


✓See List of Advertisers on page 322


ENCYCLOPEDIA FOR THE TRS-80*
VOLUME I
ENCYCLOPEDIA FOR THE TRS-80*
THEORY
PRACTICE
PROGRAMS
VOLUME 1

The Newest Peripheral for your Microcomputer

The 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 can't realize the full potential of the TRS-80.

Now, you'd think that the 150 pages of articles published in **80 Microcomputing** each month would satisfy even the most voracious of TRS-80 users. But no! You've asked for even more information!

Our response to the clamor for additional information is the **Encyclopedia for the TRS-80**, a ten-volume reference work that is absolutely packed with 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. The articles and programs will be similar to those found in **80 Microcomputing**. Each book is full of material on programming techniques, business, games, tutorials, education, utilities, interfacing. . . you name it.

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, over a period of months. This means that each new volume will reflect the latest developments and discoveries, making this a **living** encyclopedia for TRS-80 users.

Volume 1 will be out in June. It's more than 270 pages in length. . . that's 270 pages of solid information. . . no advertising. Here's a sample of what's included in Volume 1:

- Four Graphics Methods—Improved techniques for dealing with the TRS-80's low resolution graphics.
- TTY Interface—Build an interface hookup a Teletype to a TRS-80
- The President Decides—A thrilling simulation that lets you make decisions as if you were President of the United States
- Measure Instructional Effectiveness—Many programs teach or tutor; this one helps determine the effectiveness of the lessons.
- The Invoice—A program for small businesses which produces custom invoices.
- Punch Out Your Disks—Double your disk storage in seven easy steps

The **Encyclopedia for the TRS-80** is available in two editions. The deluxe COLLECTOR'S EDITION is a handsome green and black hardcover volume, with the title stamped in gold and a colorful protective dust jacket provided. These durable volumes are perfect for school and library use and make a handsome addition to your personal library. They are available for $19.95 per volume. Also available is the more economical softcover edition. This contains all the information in the hardcover edition, but at a savings of over 45%. It is ruggedly bound for heavy use and has an attractive green, ivory, and gold cover. Available for $10.95 per volume. Both editions are in an easy-to-read 6" × 9" format which is perfect bound to remain flat when opened.

80 MICROCOMPUTING SPECIAL INTRODUCTORY OFFER

Order the entire set (the first ten volumes) of the **Encyclopedia for the TRS-80** and receive the tenth volume **FREE!**

PAY ONLY $164.00 for the first ten volumes of the deluxe hardcover COLLECTOR'S EDITION—a regular **$199.50** value

PAY ONLY $83.00 for the first ten volumes of the softcover edition—a regular **$109.50** value

PLUS—This special offer **includes** UPS shipping and handling charges (an additional savings of **$15.00**)

Beat inflation by buying the whole set now . . . we cannot guarantee these same single volume prices for future volumes. You cannot lose! Preview the first volume of the set and judge for yourself. If the **Encyclopedia for the TRS-80** does not meet your needs or expectations, just return the first volume within ten days and we will refund the entire amount of your purchase. If at any time you are dissatisfied, you may cancel the remaining volumes and receive a refund for the full value of those volumes. The books will be automatically shipped to you every 6 weeks via UPS. Take advantage of this offer now! You can't afford to miss out on the newest peripheral—the **Encyclopedia for the TRS-80.** Call your order in today TOLL FREE at 1-800-258-5473, or use the coupon below or the postage-paid postcard included in this magazine.

THE Encyclopedia for the TRS-80*

YES, please reserve for me a complete set of the first ten volumes of the **Encyclopedia for the TRS-80**.

___ I want to go first class so send me the deluxe hardbound COLLECTOR'S EDITION—a $199.50 value for **$164.00 postpaid**. A total savings of $34.95 over the single volume price! (Order #EN8100)

___ Send me the "blue collar" softcover edition—the same information at a substantially lower price, a $109.50 value for **$83.00 postpaid**! A total savings of $25.95 over the single volume price! (Order #EN8080)

YES, I want to start my Encyclopedia collection

___ Volume 1 of the deluxe hardcover COLLECTOR'S EDITON @ $19.95 (Order #EN8101)**

___ Volume 1 of the Softcover edition @ $10.95 (Order #EN8081)**

**(Please enclose $1.50 per volume for shipping and handling. All volumes will be shipped UPS if the complete street address is provided; otherwise shipment is by 4th class book rate.) Allow 4-6 weeks for delivery.

___Payment enclosed Please charge to my ___VISA ___Mastercard ___AmEx

Card# ______ ***MC Interbank#*** ______ ***Expires*** ______

Name ______

Address ______

City ______ ***State*** ______ ***Zip*** ______

Signature ______

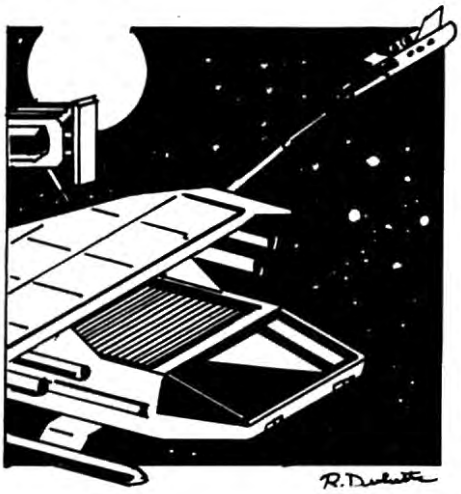
80 Microcomputing, Pine St., Peterborough, NH 03458

8008

*TRS-80 is a trademark of Tandy Corp.

Five ships in three dimensional space.

Star Guard



John Beringer
2729 West Sahara Apt. 2
Las Vegas, NV 89102

Star Guard is written in Level II Basic for 16K, and uses almost all available memory. It is an unusual space adventure because the player moves five ships independently of each other in a single game turn. Also, Star Guard is played in three dimensional space.

Options

The player can move any or all of his ships to perform several activities. If he sends out all his ships to seek and destroy the enemy, he leaves his star systems wide open for attack. He may choose to keep some ships home for protection, while sending out a scout or two to explore the surrounding star systems. This will weaken the total fighter force. All options have advantages and disadvantages.

Movement is measured along the X, Y and Z coordinates, or along an imaginary line extending from the X, Y and Z coordinates of one point to the X, Y and Z coordinates of another. The volume of space defined by the X, Y and Z coordinates and containing the stars referenced in Star Guard could be called known space. Sol, Earth's star, is in the center of known space (Sol's coordinates are 0, 0, 0). All other stars are measured from Sol; the coordinates for all other stars are the distances in light-years for each coordinate when viewed from an arbitrary point of reference. Fig. 1 shows one point of reference of known space.

To move from one point in space to another, you must know the ship's current position and destination. Coordinates show exactly where a ship is and pinpoint exactly where the destination star is. The starships in Star Guard are called jumpships because they can jump from one point to another without crossing the intervening space, making the transit almost instantaneously. There is, however, a maximum distance that jumpships may travel safely; this distance is six light-years. Beyond this distance, the jumpship risks overjump (jumping to an unknown point in space). When a ship overjumps, it is lost forever. The player has five jumpships in his fleet to use or lose.

Stargates are orbital fortresses which protect inhabited star systems. Once constructed, stargates become a permanent part of a star system—they cannot be moved from one star to another. Sol has a stargate. Earth's colony star system (chosen randomly by the computer for each game) also has a stargate. The object of the game is to protect your two stargates and to destroy all the enemy's jumpships or stargates. Retaining both stargates and at least one jumpship while destroying all the enemy's gates or ships is considered a definate victory. Retaining only one stargate and at least one jumpship is only a marginal victory. Losing both gates or all ships is a defeat.

The D'nim

The D'nim are ferocious interlopers who have invaded with the intention of becoming the dominant species in known space. The D'nim have already destroyed our colony of Lalande 21158, thus sparking Interstellar War I. The D'nim have two stargates (locations unknown) and five jumpships. The player will have to dispatch a force to locate and destroy either the D'nim gates or ships, or both.

The mechanics of ship jumping are simple: Enter the coordinates of the destination star as the computer asks for them. The distance to that star is then calculated and the player is asked if he wants to jump to the specified destination. If the distance is six light-years or less, there is no problem. On entering Y, the ship will be jumped to its destination. If the distance is greater than six light-years, a decision must be made to jump the entire distance (and risk overjump) or to jump only part of the distance. A jump up to 6.5 light-years can usually be made safely. As distance over 6.5 increases, so does the chance of getting lost by overjump. In some cases it may be desirable to risk overjump to reach a stargate that is likely to be attacked. Enter N to refuse the jump option and get a new destination prompt.

If you enter the current coordinates of the ship in question, it will not jump. Alternatively, enter 99, 99, 99 as the destination coordinates, answer Y when asked if destination is desired, and enter zero for distance to jump.

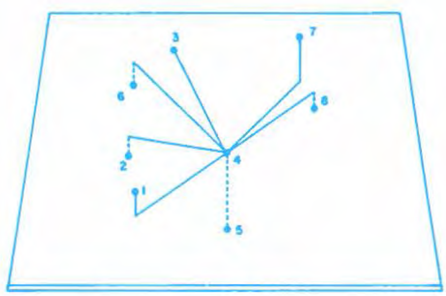


Fig. 1. Possible 3-D representation of the stars in the known space region of the game. This may help the player to visualize the three dimensions.

1. *Wolf 359*
2. *Innes' Star*
3. *Alpha Centuri group*
4. *Sol (Earth's star)*
5. *Epsilon Indi*
6. *Barnard's Star*
7. *Luyten*
8. *Ross 248*

Battle

The D'nim jumpships move in a similar manner, but will never risk overjump for any reason. The D'nim will be searching for your stargates while you are looking for theirs, and they will only search a system once, so eventually they will find all the player's gates. When the player's ships find a D'nim stargate, or when a D'nim ship finds a player's gate, or when the enemy ships find each other, a conflict results. A battle screen is drawn showing the opposing forces. Each D'nim ship is shown by an up arrow (↑), and each of the player's ships is represented by a plus sign (+). Stargates are represented by a zero in the center of the Battle Screen, and stars being fought for are shown as an asterisk (*) when there are no stargates in the star system. Since one ship acts as an observation post for the player, it will not be shown on the battle screen; the player will appear to have one less ship than he actually has.

The mechanics of battle are simple. A factor total is calculated for the ships and gates in conflict situations. The value of each ship is 10 units, while the value for a stargate is 15. The total may be broken down into an attack factor or defense factor. Defense represents use of shields, anti-missile devices, etc., to repel the attack of the enemy. Attack (offense) represents use of lasers, missiles, etc., to wear down the opponent's defenses and to destroy his ships. The D'nim defense is applied against the player's offense and the result determines how effective the attack was. The player's defense is applied against the D'nim offense, and the result determines how effective the D'nim attack was.

Retreat

Retreat is an option available to the player. The D'nim fight to the death. Retreat is not allowed while stargates are under attack. Under other circumstances, retreat costs all of the player's available power, half of which is alloted to defense while the other half powers the jump to a safe point in space. There is no power applied to the attack factor. It is possible for a ship to be destroyed while retreating, but this is preferable to losing the entire force.

Possibly the most interesting part of Star Guard is the conflict scenarios. When a battle is taking place, the warring ships explode when they are hit, and the screen jiggles to signify a hit on the player. The battles are the meat of the game; everything else is preparation.

When either side has lost all ships or gates, the game ends. By finding and destroying all D'nim ships and gates, it is possible to rid known space of these marauders and make life among the stars safe again for humans.■

Hint: By using a sheet of paper to keep track of ships and their destinations, and the star systems visited, it is possible to play a more effective game. (The computer shouldn't have to do everything. That would take some of the challenge out of the game.)

Program Listing

```
1 CLS:PRINTTAB(22);"STAR GUARD"
2 PRINT:PRINT"MOVE YOUR 5 JUMPSHIPS THRU 3-DIMENSIONAL SPACE TO
DEFEND":PRINT"YOUR 2 STARGATES AND TO FIND AND DESTROY THE RAMPA
GING":PRINT"D'MIN INVADERS!"
3 PRINT:PRINT"PRESS ANY KEY TO BEGIN"
4 IF INKEY$=""GOTO4
10 DIM STARS$(9)
15 DIM S1(31)
20 ST$(1) = "WOLF 359"
30 ST$(2) = "INNES STAR"
40 ST$(3) = "ALPHA CENTURI"
50 ST$(4) = "SOL"
60 ST$(5) = "EPSILON INDI"
70 ST$(6) = "BARNARDS STAR"
75 ST$(7) = "LUYTEN"
80 ST$(8) = "ROSS 248"
90 ST$(9) = "BLACK HOLE"
95 CLS
100 DIM CGATE(2,5), PGATE(2,5), CSHIP(5,5), PSHIP(5,5)
105 DIM SXYZ(9,3)
106 FOR N = 1 TO 9
107   FOR M = 1 TO 3 : READ SXYZ(N,M) : NEXT M
108 NEXT N
109 DATA -6,2,1
110 DATA 4,0,-8
111 DATA -1,-1,-4
112 DATA 0,0,0
113 DATA 0,-6,0
114 DATA 4,-3,-10
115 DATA 6,3,-2
116 DATA 6,-1,-1
117 DATA 99,99,99
118 PRINT "THE STARS IN OUR LOCAL GROUP" : PRINT : PRINT "STAR N
AME"," X, Y, Z LOCATION RELATIVE TO SOL"
119 FOR N = 1 TO 8 : PRINT STARS$(N),SX(N,1);SX(N,2);SX(N,3) : NE
XT : INPUT "PRESS 'ENTER' TO CONTINUE. OK";KEY
120 PGATE(1,1) = 1 : FOR N = 3 TO 5 : PGATE(1,N) = 0 : NEXT
125 N = RND(8) : IF N = 4 THEN GOTO 125 ELSE FOR M = 3 TO 5 : PG
ATE(2,M) = SXYZ(N,M - 2) : NEXT
126 PGATE(1,2) = 4 : PGATE(2,1) = 2 : PGATE(2,2) = N
130 CLS
140 PRINT "GATE NBR","NAME"
150 PRINT "1","SOL","HOME STAR SYSTEM" : PRINT "2",STARS$(PGATE(2
,2)),"COLONY STAR SYSTEM": FOR N = 1 TO 500: NEXT
157 INPUT"PRESS 'ENTER' TO GO ON. OK";KEY:CLS
160 D=RND(5)
165 FORN=1TOD
170 PSHIP(N,1)=N:FORM=3TO5:PSHIP(N,M)=PGATE(1,M):NEXTM
180 NEXTN
195 FORN=DTO5
200 PSHIP(N,1)=N:FORM=3TO5:PSHIP(N,M)=PGATE(2,M):NEXTM
210 NEXTN
214 FOR N = 1 TO 2
215   SUN = RND(8)
225   IF SUN = PGATE(1,2) OR SUN = PGATE (2,2) THEN GOTO 215
230   CGATE(N,1) = N : CGATE(N,2) = SUN
235   FOR M = 3 TO 5 : CGATE(N,M) = SXYZ(SUN,M - 2) : NEXT M
240 NEXT N
242 IF CGATE(1,2) = CGATE(2,2) GOTO 214
245 FOR N = 1 TO 5
250   GATE = RND (2)
255   CSHIP(N,1) = N : CSHIP(N,2) = GATE
260   FOR M = 3 TO 5 : CSHIP(N,M) = CGATE(GATE,M) : NEXT M
265 NEXT N
270 DIM XPLRD(9), CDEST(5), SCREEN(5)
275 FOR N = 1 TO 8 : XPLRD(N) = 0 : NEXT
280 FOR N = 1 TO 5 : CDEST(N) = 99 : NEXT
282 FOR N = 1 TO 2: XPLRD(CGATE(N,2))=1: NEXTN
285 SCT = 1
286 FIND = 0
287 FLEET = 0
288 RDY = 0
290 FLEET = RND(5) : IF FLEET < 3 GOTO 290
300 CLS:PLOST=0:CLOST=0
301 FOR N=1 TO 5
302 IF PSHIP(N,1)=0 PLOST=PLOST+1
301 FOR N=1 TO 5
302 IF PSHIP(N,1)=0 PLOST=PLOST+1
303 IF CSHIP(N,1)=0 CLOST=CLOST+1
304 NEXT N
305 IF CLOST>4 OR PLOST>4 GOTO 407
306 PRINT "GATE/SHIP                  STATUS"
310 FOR N = 1 TO 2
315   IF PGATE(N,1) = 0 PRINT STARS$(PGATE(N,2)), "DESTROYED" :
       GOTO 325
320   PRINT STARS$(PGATE(N,2)), "INTACT AND BATTLE WORTHY"
325 NEXT N
327 FSHIPS=0
330 FOR N = 1 TO 5
340   IF PSHIP(N,1) = 0 PRINT "JUMPSHIP "; N, "LOST OR DESTROYED
 IN BATTLE" : GOTO 350
345   PRINT "JUMPSHIP "; N, "INTACT AT "; PSHIP(N,3); ", "; PSHI
P(N,4); ", "; PSHIP(N,5);:FORP=1TO8:IFPSHIP(N,3)=SX(P,1)ANDPSHIP
(N,4)=SX(P,2)ANDPSHIP(N,5)=SX(P,3)THENPRINTST$(P):GOTO347ELSENEX
TP:PRINT" "
347   FSHIPS=FSHIPS+1
350 NEXT N
352 IF FSHIPS < 1 GOTO 408
355 FGATES = 0
357 EGATES = 0
360 FOR N = 1 TO 2
365   IF PGATE(N,1) <> 0 THEN FGATES = FGATES + 1
370   IF CGATE(N,1) <> 0 THEN EGATES = EGATES + 1
375 NEXT N
377 IF FGATES = 0 AND EGATES = 0 THEN CLS :PRINT "ALL FRIENDLY A
ND ENEMY GATES ARE DESTROYED" : PRINT "tHIS WAR IS A STANDOFF" :
 END
380 IF FGATES = 0 THEN CLS : PRINT "ALL OUR GATES ARE DESTROYED!
" : PRINT "THE HUMAN RACE IS ENSLAVED FOREVER!" : END
385 IF EGATES = 0 THEN CLS: PRINT "ALL ENEMY GATES ARE DESTROYED
```

Program continues

Program continued

```
" : PRINT "THE HUMAN STARFLEET IS VICTORIOUS... THE STARS ARE OU
RS!" : END
387 PRINT
390 ESHIPS = 0
395 FOR N = 1 TO 5 : IF CSHIP(N,1) <> 0 THEN ESHIPS = ESHIPS + 1

397 NEXTN
400 PRINT "ENEMY STARGATES REMAINING TOTAL "; EGATES
405 PRINT "ENEMY JUMPSHIPS REMAINING TOTAL "; ESHIPS
406 INPUT"PRESS 'ENTER' TO GO ON. OK";KEY
407 IF ESHIPS=0 AND FSHIPS=0 THEN PRINT"ALL HUMAN AND D'NIM SHIP
S ARE DESTROYED": PRINT"THIS WAR IS A DRAW": PRINT"BUT WE WILL M
EET AGIAN SOON...": END
408 IFFSHIPS=0CLS:PRINT"ALL OUR SHIPS HAVE BEEN DESTROYED!":PRIN
T"THE D'NIM ARE ON THE WAY!":PRINT:END
409 IFESHIPS=0CLS:PRINT"ALL D'NIM SHIPS HAVE BEEN DESTROYED!":PR
INT"AS THEY HAVE NO MEANS TO CONTINUE THE WAR, THEY SURRENDER
    UNCONDITIONALLY!":END
410 PRINT : PRINT "SCANNING FOR ENEMY PRESENSE..."
420 FOR N = 1 TO 5
425    FOR M = 1 TO 2
427    IF CGATE(M,1)=0 OR PSHIP(N,1)=0 GOTO 432
430    IF PSHIP(N,3) = CGATE(M,3) AND PSHIP(N,4) = CGATE(M,4) AND
 PSHIP(N,5) = CGATE(M,5) THEN GOSUB 5000 : CLS
432    IF PGATE(M,1)=0 OR CSHIP(N,1)=0 GOTO 440
435    IF CSHIP(N,3) = PGATE(M,3) AND CSHIP(N,4) = PGATE(M,4) AND
 CSHIP(N,5) = PGATE(M,5) THEN GOSUB 5100 : CLS
436    REM GOSUB GATE/SHIP CONFLICT
440    NEXT M
445 NEXT N
447 IF CGATE(1,1)=0 AND CGATE(2,1)=0 THEN GOTO 355
448 IF PGATE(1,1)=0 AND PGATE(2,1)=0 THEN GOTO 355
450 FOR N = 1 TO 5
452 IF PSHIP(N,1)=0 GOTO 490
455    FOR M = 1 TO 5
457    IF CSHIP(M,1)=0 OR PSHIP(N,1)=0 GOTO 485
460    IF PSHIP(N,3) <> CSHIP(M,3) GOTO 485
465    IF PSHIP(N,4) <> CSHIP(M,4) GOTO 485
470    IF PSHIP(N,5) <> CSHIP(M,5) GOTO 485
475    GOSUB 5200 : CLS
480    REM SHIP/SHIP CONFLICT
485    NEXT M
490 NEXT N
494 PRINT "SCANNING ENDS": FOR N=1TO500: NEXTN
495 CLS
496 PLOST=0:CLOST=0
497 FOR N = 1 TO 5
498 IF PSHIP(N,1)=0 THEN PLOST = PLOST + 1
499 IF CSHIP(N,1)=0 THEN CLOST = CLOST + 1
500 NEXT N
501 FSHIPS = 5 - PLOST:ESHIPS = 5 - CLOST
505 IF FSHIPS < 1 OR ESHIPS < 1 GOTO 407
550 FOR N = 1 TO 5
559 LOST = 0
560    IF PSHIP(N,1) = 0 PRINT:PRINT "-- SHIP "; N; "LOST OR DEST
ROYED IN BATTLE" :PRINT: LOST = LOST + 1: GOTO 810
565    CLS : PRINT "STAR", "SPATIAL LOCATION - X, Y, Z CO-ORDINAT
ES" : FOR M = 1 TO 8 : PRINT STAR$(M), SXYZ(M,1); SXYZ(M,2); SXY
Z(M,3) :   NEXT M
570    PRINT@ 640, "FOR JUMPSHIP "; N; " AT "; PSHIP(N,3); PSHIP(
N,4); PSHIP(N,5)
575    INPUT "ENTER DESTINATION X C0-ORDINATE. ";X : X = INT(X)
580    INPUT "ENTER DESTINATION Y CO-ORDINATE. ";Y : Y = INT(Y)
585    INPUT "ENTER DESTINATION Z CO-ORDINATE. ";Z : Z = INT(Z)
590    DX = X - PSHIP(N,3)
595    DY = Y - PSHIP(N,4)
600    DZ = Z - PSHIP(N,5)
605    DIST = 0
610    DEST = SQR(DX * DX + DY * DY + DZ * DZ)
620    PRINT "DESTINATION IS "; DEST; " LIGHT-YEARS AWAY"
622 PRINT@ 960,"DO YOU WISH THIS DESTINATION";: INPUTA$: IF A$<>
 "Y" AND A$ <> "N" THEN GOTO 622 ELSE IF A$ = "N" GOTO 565
625    IF DEST<1 GOTO 635
627    IF DEST<6.1 GOTO 730
630    INPUT "ENTER DISTANCE TO JUMP. " ; DIST
635    IF DIST <= 0 PRINT "SHIP "; N; "DID NOT JUMP" : GOTO 810
640    IF DIST <= 6 GOTO 720
650    REM ELSE OVERJUMP
660    OVER = DIST - 3
670    LOST = INT(SQR(RND(OVER)))
680    IF LOST = 1 GOTO 720
690    PRINT "SHIP "; N; " LOST BY OVERJUMP"
700    PSHIP(N,1) = 0
710    GOTO 810
720    IF DEST > DIST + .5 GOTO 762
730    PSHIP(N,3) = X
740    PSHIP(N,4) = Y
750    PSHIP(N,5) = Z
760    GOTO 800
762    IF DX = 0 AND DY = 0 AND DZ = 0 THEN DIST = 0 : GOTO 635
764    IF DX = 0 GOTO 772
766    IF DX > 0 THEN DX = DX - 1 ELSE IF DX < 0 THEN DX = DX + 1

768    GOSUB 7000
770    IF HIGH = 0 GOTO 788
772    IF DY = 0 GOTO 780
774    IF DY > 0 THEN DY = DY - 1 ELSE IF DY < 0 THEN DY = DY + 1

776    GOSUB 7000
778    IF HIGH = 0 GOTO 788
780    IF DZ = 0 GOTO 762
782    IF DZ > 0 THEN DZ = DZ - 1 ELSE IF DZ < 0 THEN DZ = DZ + 1

784    GOSUB 7000
786    IF HIGH = 1 GOTO 762
788    PSHIP(N,3) = PSHIP(N,3) + DX
790    PSHIP(N,4) = PSHIP(N,4) + DY
792    PSHIP(N,5) = PSHIP(N,5) + DZ
800    PRINT "SHIP "; N; " REPORTS SUCCESSFUL JUMP TO "; PSHIP(N,
3); PSHIP(N,4); PSHIP(N,5)
810    INPUT"PRESS 'ENTER' TO CONTINUE. OK"; KEY
815 NEXT N
816 IF LOST > 4 THEN FSHIPS = 0 : GOTO 409
```

Program continues

Orange Micro

"THE COMPUTER PRINTER SPECIALISTS"

UP TO 25% DISCOUNTS! — SAME DAY SHIPMENT!

NEW CENTRONICS 739 (RADIO SHACK LINE PRINTER IV)

With Graphics and Word Processing Print Quality

• 18 x 9 dot matrix; suitable for word processing • Underlining • proportional spacing • right margin justification • serif typeface • 80/100 CPS • 9½" Pin Feed/Friction feed • Reverse Platen • 80/132 columns • Top of form

CENTRONICS 739-1 (Parallel) .... (List $995) **$ Call**
CENTRONICS 739-3 (Serial) .... (List $1045) **$ Call**

ANADEX

Dot Graphics, Wide Carriage

• 11 x 9 dot matrix; lower case descenders • Dot resolution graphics • Bi-directional, logic seeking • Up to 200 CPS • RS 232 Serial & Parallel • Forms control • X-ON/X-OFF • Up to 6 part copy.

ANADEX 9501 .... (List $1650) **$1350**

AXIOM

Dot Graphics, Low Price

• Dot addressable graphics • 70 CPS • 12 characters per inch • 80 columns • adjustable tractors • 3 part copies • Manufactured by Sieko • Parallel, RS232 serial, IEEE-488 interfaces available.

AXIOM GP-80M .... (List $399) **$349**

IDS PAPER TIGERS

Dot Resolution Graphics, quality print, speed

• 7 wire printhead (445); 9 wire printhead (460) with lower case descenders • Over 150 CPS • bi-directional, logic seeking (460) • 8 character sizes; 80-132 columns • Adjustable tractors • High-resolution dot graphics • Proportional spacing & text justification (460).

IDS 445G 7 wire printhead, graphics .... (List $895) **$ 750**
IDS 460G 9 wire printhead, graphics .... (List $1394) **$1150**
IDS 560G 9 wire, wide carriage, graphics ...... (List $1794) **$1400**

CALL FOR FREE CATALOG

(800) 854-8275

CA, AK, HI (714) 630-3322

At Orange Micro, we try to fit the right printer to your application. Call our printer specialists for free consultation.

EPSON MX80/MX70

Low-Priced Professional Print Quality

• 9 x 9 dot matrix • Lower case descenders • 80 CPS • Bidirectional, Logic seeking • 40, 66, 80, 132 columns per line • 64 special graphic characters: TRS-80 Compatible • Forms handling • Multi-pass printing • Adjustable tractors

We also carry a full line of Epson Accessories.

EPSON MX80 .... (List $645) **$Call**
EPSON MX 70 Dot graphics, 5 x 7 matrix .... (List $450) **$Call**
EPSON MX100 wide carriage .... (List $995) **$Call**
GRAPPLER™ Apple graphics parallel interface and cable **$165**

MX80/70 FRICTION FEED KIT

Uses installable kit for single sheets.
Easy 15 minute installation. **$ 75**

ANACOM

Low Cost, High Speed, Wide Carriage

• 9 x 9 dot matrix • Lower case descenders • Wide carriage • Adjustable tractors to 16" • 150 CPS, Bidirectional, Logic Seeking

ANACOM 150 .... (List $1350) **$ Call**

NEC SPINWRITER

High Speed Letter Quality

• 55 CPS • Typewriter quality • Bidirectional • Plotting • proportional spacing.

5510-5 RO, Serial, w/tractors .... (List $2995) **$2675**
5530-5 RO, Parallel, w/tractors .... (List $2970) **$2650**

TELEVIDEO CRT'S

AT DISCOUNT PRICES!

TVI 910
TVI 912C
TVI 920C
TVI 950

Please Call Toll Free
Prices are too low to advertise

QUANTITY PRICING AVAILABLE

PRINTERS

MALIBU 165 wide carriage, graphics, letter quality .. (List $2495) **$ 1975**
QUME 5/45 typewriter quality .... (List $2905) **$ 2559**
VISTA-C. ITOH Starwriter, letter quality ...... (List $1895) **$ Call**

INTERFACE EQUIPMENT

EPSON ACCESSORIES .... **$ Call**
ORANGE INTERFACE for Apple II parallel interface board & cable .... **$ 110**
MICROTRONICS Atari parallel interface .... **$ 69**
TRS-80 CABLES to keyboard or Exp. interface .... **$ Call**
NOVATION D-CAT direct connect modem .... **$ Call**

RETAIL PRINTER STORES:
3150 E. La Palma, #I, Anaheim, CA.
13604 Ventura Bl., Sherman Oaks, CA.
Store Hours: M-F 10-6 Sat. 10-4

✓296

Phone orders WELCOME; same day shipment. Free use of VISA & MASTERCARD. Personal checks require 2 weeks to clear. Manufacturer's warranty included on all equipment. Prices subject to revision.

Orange Micro, Inc.
3150 E. La Palma, Suite I
Anaheim, CA 92806

Program continued

```
817 PRINT "SENSORS DETECT D'NIM SHIP MOVEMENT.  WAIT...": ESHIPS
 = 0
830 FOR N = 1 TO 5: IF CSHIP(N,1) <> 0 THEN ESHIPS=ESHIPS+1
832 NEXTN
835 IF ESHIPS < 1 GOTO 1220
836 IF ESHIPS<FLEET THEN FLEET = ESHIPS: RDY = 0
840 IF RDY = 1 GOTO 1020
850 N = 1: RDY = 1
860 IF N > 5 GOTO 1000
870 IF CSHIP(N,1) = 0 N = N + 1 : GOTO 860
880 SCT = N
885 REM GOT SCOUT
890 IF N<5 THEN M=N+1 ELSE FLEET=1:RDY=1:GOTO 1020
895 ESHIPS=FLEET
900 IF CSHIP(M,1)=0 GOTO 980
910 DX=CSHIP(SCT,3)-CSHIP(M,3)
920 DY=CSHIP(SCT,4)-CSHIP(M,4)
930 DZ=CSHIP(SCT,5)-CSHIP(M,5)
940 DIST=SQR(DX[2 + DY[2 + DZ[2)
950 IF DIST<1 THEN RDY=RDY+1: GOTO980
960 IF DIST<7 THEN FOR P=3 TO 5: CSHIP(M,P)=CSHIP(N,P): NEXTP: R
DY=RDY+1: GOTO 980
970 DIST=7 : IDX=M: GOSUB 7100
980 ESHIPS=ESHIPS-1: M=M+1
990 IF ESHIPS>0 AND M<=5 GOTO 900
1000 IF RDY => FLEET THEN RDY=1 ELSE RDY=0
1010 GOTO 1215
1020 CDIST=999
1030 FOR N = 1 TO 8
1040 IF XPLRD(N)=1 GOTO 1080
1050 DIST=SQR((CSHIP(SCT,3)-SXYZ(N,1))[2 + (CSHIP(SCT,4)-SXYZ(N,
2))[2 + (CSHIP(SCT,5)-SXYZ(N,3))[2)
1060 IF DIST<1 THEN XPLRD(N)=1 : GOTO 1080
1070 IF DIST<CDIST THEN CDIST=DIST: CDEST(3)=SXYZ(N,1): CDEST(4)
=SXYZ(N,2): CDEST(5)=SXYZ(N,3)
1080 NEXTN
1085 DIST=CDIST
1090 ESHIPS=FLEET: N=SCT
1100 IF DIST>7 THEN DIST=7 : GOTO 1160
1120 FOR M = 3 TO 5: CSHIP(N,M)=CDEST(M): NEXTM
1130 N=N+1: ESHIPS=ESHIPS-1
1140 IF N<=5 AND ESHIPS>0 GOTO 1120
1150 GOTO 1215
1160 DX=CDEST(3)-CSHIP(SCT,3)
1170 DY=CDEST(4)-CSHIP(SCT,4)
1180 DZ=CDEST(5)-CSHIP(SCT,5)
1190 IDX=N: GOSUB 7100
1200 N=N+1: ESHIPS=ESHIPS-1
1210 IF N<=5 AND ESHIPS>0 THEN IDX=N: GOSUB 7240: GOTO 1200
1215 GOTO 1220
1220 GOTO 300
4999 END
5000 EGATES = 1 : FGATES = 0 : X = CGATE(M,3) : Y = CGATE(M,4) :
 Z = CGATE(M,5) : GOTO 5210
5100 EGATES = 0 : FGATES = 1 : X = PGATE(M,3) : Y = PGATE(M,4) :
 Z = PGATE(M,5) : GOTO 5210
5200 EGATES=0:FGATES=0:X=PSHIP(N,3):Y=PSHIP(N,4):Z=PSHIP(N,5)
5210 ESHIPS = 0 : FSHIPS = 0
5215 FOR ZZ=1 TO 31 : S1(ZZ)=0 : NEXT ZZ
5220 FOR P = 1 TO 5
5230    IF CSHIP(P,3) = X AND CSHIP(P,4) = Y AND CSHIP(P,5) = Z A
ND CSHIP(P,1) <> 0 THEN ESHIPS = ESHIPS + 1
5240    IF PSHIP(P,3) = X AND PSHIP(P,4) = Y AND PSHIP(P,5) = Z A
ND PSHIP(P,1) <> 0  THEN FSHIPS = FSHIPS + 1
5250 NEXT P
5260 CLS : PRINT "ALERT!  CONFLICT AT";
5274 A=0
5275 FOR B = 1 TO 8
5276 IF SX(B,1)=X AND SX(B,2)=Y AND SX(B,3)=Z THEN A = B: GOTO 5
278
5277 NEXTB
5278 IF A = 0 THEN PRINT@20,X,Y,Z ELSE PRINT@20,STAR$(A)
5280 PRINT
5290 IF FGATES <> 0 PRINT "YOU HAVE A STARGATE"
5300 PRINT "YOU HAVE "; FSHIPS; "JUMPSHIPS IN THE CONFLICT VOLUM
E"
5310 PRINT
5320 IF EGATES <> 0 PRINT "THE D'NIM HAVE A STARGATE"
5330 PRINT "THE D'NIM HAVE "; ESHIPS; "JUMPSHIPS IN THE CONFLICT
 VOLUME"
5340 EPWR = EGATES * 15 + ESHIPS * 10
5440 FPWR = FGATES * 15 + FSHIPS * 10
5450 PRINT "YOU HAVE "; FPWR; "UNITS OF POWER FOR OFFENSE/DEFENS
E" : PRINT "THE D'NIM HAVE "; EPWR; "UNITS OF POWER"
5500 PRINT"PRESS 'ENTER' TO BEGIN THE BATTLE";
5505 A$=INKEY$: IF A$="" THEN 5505
5510 CLS : PRINT TAB(25) "BATTLE SCREEN"
5512 FOR V=2 TO 125 : SET (V,3) : SET (V,41) : NEXT V
5514 FOR V=4 TO 40 : SET (2,V) : SET (3,V) : SET (124,V) : SET (
125,V) : NEXT V
5516 REM DRAW BORDER
5517 ZZ=0
5520 FOR P = 1 TO 20
5540 GOSUB 7300
5550   PRINT@ D,".";
5560 NEXT P
5570 IF EGATES>0 OR (FGATES>0 AND FSHIPS>0) PRINT@ 480,"0"; ELSE
 IF FGATES>0 AND FSHIPS<1 PRINT@ 480,"*"; ELSE IF A><0 PRINT@ 48
0,"*";
5572  R=FSHIPS - 1
5573 IF R>0 THEN LET R=R-1 : GOSUB 7300 :  PRINT@D,"+"; : GOTO 5
573
5580 ZZ=25
5590 IF ESHIPS=0 GOTO 5635
5600 FOR S=1 TO ESHIPS : GOSUB 7300 : PRINT@ D,"["; : NEXT S
5635 LET D=480 : GOSUB 7300
5640 CHIT=0: PHIT=0
5660 EPWR=EGATES*15+ESHIPS*10
5665 FPWR=FGATES*15+FSHIPS*10
5667 PRINT@ 896, CHR$(29);CHR$(30);
5670 PRINT@ 896,"POWER =";FPWR;"            ";: PRINT@ 910,"RETRE
AT? Y OR N";:INPUT RT$:IF RT$ = "Y" GOTO 6100 ELSE IF RT$ >< "N"
 GOTO 5667
```

Program continues

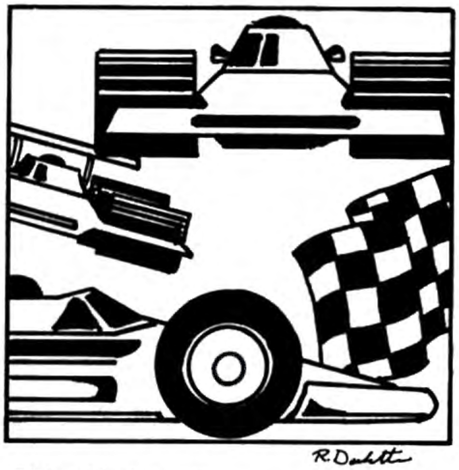
Program continued

```
5675 PRINT@910,CHR$(30);"UNITS FOR OFFENSE";:INPUT FOFF : IF FOF
F<0 PRINT@910,CHR$(30);"CANNOT HAVE < 0 OFFENSE";:FOR P=1TO500:N
EXTP:GOTO 5675
5680 IF FOFF > FPWR PRINT@896,CHR$(29); CHR$(30); "OFFENSE CANNO
T EXCEED TOTAL POWER";: FOR P=1 TO 300: NEXTP: GOTO 5667
5690 FD = FPWR-FOFF
5700 PRINT@896,CHR$(29);CHR$(30);"OFFENSE =";FOFF;"DEFENSE =";FD
;: FOR P=1 TO 500: NEXTP
5710 EATT=RND(EPWR): ED=EPWR-EATT
5730 DOFF=FOFF-ED: DFF=EATT-FD
5750 FOR P = 1 TO 5
5760 IF DOFF < 5 GOTO 5790
5770 IF CSHIP(P,1) = 0 GOTO 5790
5780 IF CSHIP(P,3) = X AND CSHIP(P,4) = Y AND CSHIP(P,5) = Z THE
N CSHIP(P,1) = 0: ESHIPS=ESHIPS-1: DOFF=DOFF-10 ELSE GOTO 5790
5783 R=1:CH=CH+1:GOSUB7500:S=26:GOSUB7400
5790 NEXTP
5795 PRINT@ 896, CHR$(29);CHR$(30);
5800 FOR P = 1 TO 5
5810 IF DFF < 5 GOTO 5840
5820 IF PSHIP(P,1)=0 GOTO 5840
5830 IF PSHIP(P,3)=X AND PSHIP(P,4)=Y AND PSHIP(P,5)=Z THEN PSHI
P(P,1)=0: FSHIPS=FSHIPS-1:GOSUB7600:PRINT@896,"ONE OF OUR SHIPS
HAS BEEN HIT!";:FORS=1TO500:NEXTS:DFF=DFF-10:PHIT=PHIT+1:S=21:GO
SUB 7400:PRINT@ 896, CHR$(30);:FOR S=1 TO 100:NEXT S
5840 NEXTP
5855 IF DOFF < 7 GOTO 5870
5860 IF EGATES > 0 THEN CGATE(M,1)=0: DOFF=0: EGATES=0:FORA1=1TO
4:PRINT@896,CHR$(30);:FORA2=1TO60:NEXTA2:PRINT@896,"THE D'NIM ST
ARGATE IS HIT!";:FORA2=1TO60:NEXTA2:NEXTA1:PRINT@ 480,"*";: FOR
S = 1 TO 30: NEXTS: PRINT@ 480," ";: FOR S = 1 TO 550 : NEXTS
5870 IF DFF < 7 GOTO 5900
5880 IFFGATES>0THENPGATE(M,1)=0:DFF=0:FGATES=0:GOSUB7600:PRINT@8
96,"OUR STARGATE IS HIT!";:FORS=1TO500:NEXTS:PRINT@480,"*";: FOR
 S = 1 TO 30: NEXTS: PRINT@ 480," ";:PRINT@960,"YOU HAVE LOST ST
ARGATE ";ST$(PGATE(M,2));"!";: FOR S = 1 TO 550 : NEXTS
5900 IF EGATES = 0 AND ESHIPS = 0 GOTO 5999
5910 IF FGATES = 0 AND FSHIPS = 0 GOTO 5999
5980 PRINT@ 896,"YOU LOST";PHIT;"JUMPSHIPS. THE D'NIM LOST";CHIT
;"JUMPSHIPS";: FOR S=1 TO 500: NEXTS
5990 GOTO 5660
5999 CLS: PRINT TAB(20);"CONFLICT RESOLVED":PRINT
6005 FOR P=1 TO 2
6010  IF PGATE(P,3)=X AND PGATE(P,4)=Y AND PGATE(P,5)=Z AND PGAT
E(P,1)=0 PRINT"YOU LOST OUR STARGATE AT ";ST$(PGATE(P,2));
6020  IF CGATE(P,3)=X AND CGATE(P,4)=Y AND CGATE(P,5)=Z AND CGAT
E(P,1)=0 PRINT"WE DESTROYED THE D'NIM STARGATE AT ";ST$(CGATE(P,
2));
6040 NEXTP
6050 PRINT
6060 IF PHIT=0 AND FSHIPS=0 GOTO 6070 ELSE PRINT"YOU LOST";PHIT;
"SHIPS.";FSHIPS;"SURVIVED.":PRINT
6070 IF CHIT=0 AND ESHIPS=0 GOTO 6099 ELSE PRINT"WE DESTROYED";C
HIT;"D'NIM JUMPSHIPS!";ESHIPS;"SURVIVED.":PRINT
6099 INPUT "PRESS 'ENTER' TO GO ON";Z$:RETURN
6100 IF FGATES >< 0 PRINT@910,"GATES CANNOT BE ABANDONED. PRESS
'ENTER'";:INPUT Z$:GOTO 5675
6110 PRINT@896,CHR$(29);CHR$(30);"RETREAT USES ALL PWR. DEFENSE
REQUIRES ";FPWR/2;"  ENGINES NEED ";FPWR-FPWR/2
6120 LOST = (RND(EPWR)-(FPWR/2))/10:LOST=INT(LOST):INPUT "PRESS
'ENTER' TO GO ON";Z$
6125 CLS:PRINTTAB(20);"CONFLICT RESOLVED":PRINT:P=0
6130 IF LOST <= 0 GOTO 6200
6135 P=P+1
6140 IF P>5 THEN FOR P=1 TO 500:NEXT P:GOTO 6050
6142 IF PSHIP(P,1)=0 GOTO 6135
6145 IF PSHIP(P,3)=X AND PSHIP(P,4)=Y AND PSHIP(P,5)=Z THEN PSHI
P(P,1)=0:FSHIPS=FSHIPS-1:PHIT=PHIT+1:LOST=LOST-1:PRINT"SHIP";P;"
 DESTROYED IN RETREAT":GOTO 6130
6150 GOTO 6130
6200 P=P+1
6210 IF P>5 THEN FOR P=1 TO 500:NEXT P:GOTO 6050
6220 IF PSHIP(P,3)=X AND PSHIP(P,4)=Y AND PSHIP(P,5)=Z THEN PSHI
P(P,3)=X+1:PRINT"SHIP";P;" RETREATED TO ";X+1,Y,Z
6230 GOTO 6200
7000 IF SQR(DX * DX + DY * DY + DZ * DZ) > DIST THEN HIGH = 1  E
LSE HIGH = 0
7099 RETURN
7100 IF DX = 0 AND DY = 0 AND DZ = 0 THEN GOTO 7299
7120 IF DX = 0 GOTO 7160
7130 IF DX > 0 THEN DX = DX - 1 ELSE IF DX < 0 THEN DX = DX + 1
7140 GOSUB 7000
7150 IF HIGH = 0 GOTO 7240
7160 IF DY = 0 GOTO 7200
7170 IF DY > 0 THEN DY = DY - 1 ELSE IF DY < 0 THEN DY = DY + 1
7180 GOSUB 7000
7190 IF HIGH = 0 GOTO 7240
7200 IF DZ = 0 GOTO 7100
7205 IF DZ > 0 THEN DZ = DZ - 1 ELSE IF DZ < 0 THEN DZ = DZ +01
7210 GOSUB 7000
7230 IF HIGH = 1 GOTO 7100
7240 CSHIP(IDX,3) = CSHIP(IDX,3) + DX
7250 CSHIP(IDX,4) = CSHIP(IDX,4) + DY
7260 CSHIP(IDX,5) = CSHIP(IDX,5) + DZ
7299 RETURN
7300 LET V=RND(60)
7310 IF V<3 GOTO 7300
7320 LET W=RND(12)
7330 IF W<2 GOTO 7320
7340 LET D=W*64 + V
7345 IF D=480 D=480+RND(5)
7350 LET ZZ=ZZ+1
7360 LET S1(ZZ)=D
7399 RETURN
7400 FOR A1=0 TO 4
7410 IF S1(S+A1)=0 GOTO 7420 ELSE PRINT@ S1(S+A1),"*"; : FOR A2=
1 TO 30 : NEXT A2 : PRINT@ S1(S+A1)," "; : S1(S+A1)=0 : GOTO 743
0
7420 NEXT A1
7430 RETURN
7500 FOR A1=1 TO 4:PRINT@896,CHR$(30);:FOR A2=1 TO 60:NEXT A2:PR
INT@896,"D'NIM SHIP HIT!";:FOR A2=1 TO 60:NEXT A2:NEXT A1
7510 RETURN
7600 FORA1=1TO10:OUT255,0:FORA2=1TO5:NEXTA2:OUT255,0:FORA2=1TO5:
NEXTA2:NEXTA1:PRINT@896,CHR$(30);
7610 RETURN
```


✓See List of Advertisers on page 322

Flat out and fourth to go.

Formula 80



Richard A. Sprague
900 15th St.
Neillsville, WI 54456

It's five o'clock at last!" you shout, as you grab your coat and run from the office. Ordinarily you'd rush straight home, but not tonight.

Step onto the track and into your waiting sportscar. Your fingers caress the leather steering wheel as you start the engine. Instantly, 450 horses roar as you rev the motor. Slowly loosen the clutch; the wheels spin hungrily as you feed the carburetor. The track rushes to meet you as the speedometer plays its reluctant catch-up game.

Nothing can compare with the feeling of racetrack driving. The thrill of high speed racing can grab you almost as powerfully as the addiction many of us have to high speed programming.

While the smell of the open road might be more enticing than the smell of an open keyboard, Racecar II gets the point across. It's simple to punch into your system in ten minutes, small enough to fit into 4K, and flexible enough to challenge the beginner and expert equally.

The Scene

Beginning with a simple display of a car in the middle of a race track, the car slowly gains speed (to the beat of real-time physics) as it accelerates. The track ahead of the car constantly changes through tricky curves and unnerving gyrations. To keep the car on the road, you have to keep your fingers on the arrow keys. (The left arrow is used for turning left; the right arrow for right turns.)

All too quickly, the road makes an unexpected swerve and you find your car crashing into the side; your score drops and you must get safely back on the road. The constant fight against road and score, as well as the special automatic provisions for beginners, makes the game highly addicting.

Racecar II invokes an automatic pilot routine for every beginning player. The routine ensures that the car will remain tied to the center of the road until one of the arrow keys is pushed. In other words, it's always entirely your fault when you crash.

The computer increments your score based on three variables: The amount of time on the clock; the score you've accumulated so far; and whether or not you've crashed. A crash means that your score will be cut by 30 percent. Nothing is scored or lost during the auto-pilot mode.

Operation and Documentation

When the TRS-80 gives its Ready prompt, try this: LIST the program, then hit Enter over and over. The words seem to move upward on the screen, part of the standard TRS-80 function called automatic scrolling. Racecar II capitalizes on this ability by placing characters at predetermined places on the screen, then doing scrolling operations. The characters move upward so fast that your eye mixes them together and it looks like a real track.

Additions, Deletions, Improvements

With a short program like Racecar II, the memory left in the machine makes improving the game almost as much fun as playing it. Fortunately, improvements are not only easy, but natural.

You may first want to change execution speed. Beginners may find the graphics too fast. The simple remedy is the insertion listed in Fig. 1.

If the display is too slow, the speed can be increased by combining statements into multiple statement lines. Or, you could define Z as an integer by adding the lines shown in Fig. 2. Also, the auto-mode could be improved for faster execution.

Variable	Definition
A$	Draws railings
C$	Draws the car (To change, see Fig. 4)
C	Determines car placement on the track. During auto-pilot, this is set at eight spaces right of the left railing.
E	Acceleration variable, initialized at 1024 as it begins any accelerating process.
G	Increment variable
M	When this is a one auto-pilot is set; anything else is manual overdrive.
P	Road variable: When negative, the road turns left; when positive, road turns right. Must be initialized positive.
S	Score variable: cannot be an integer.
T	The built-in clock, equal to the number of scrolls in a game.
X	Cannot be an integer: this variable indirectly determines where the next road section will be.
Y	Never an integer, this variable randomly decides how far the road will swing.
Z	Also must not be an integer. This is the most important variable: decides where the track and car will be positioned relative to predetermined screen elements. See Fig. 2.

Table 1. Variable Chart

"It looks and behaves just like a video game and it always provides a challenge. Besides, its gas mileage is unbeatable."

Insert this line for slower graphics:

145 FOR G = 1 TO 100 : NEXT

(The "100" may be changed for faster or slower speed.)

Fig. 1

Insert and change these lines for faster execution:

```
8   DEFINT Z
10  X9 = SIN(X)*12 : T = T + 1
15  Z = CINT(X9)
```

Fig. 2

Insert these statements to add the fast-turn option:

```
135 IF PEEK(15350) = 65 THEN C = C + 4 : M = 0
137 IF PEEK(15350) = 33 THEN C = C - 4 : M = 0
```

Fig. 3

Here are some combinations you might want to try as values for C$ if you don't like my car:

```
C$ = CHR$(151) + CHR$(143) + CHR$(171)
C$ = CHR$(156) + CHR$(143) + CHR$(172)
C$ = CHR$(181) + CHR$(188) + CHR$(186)
C$ = CHR$(157) + CHR$(188) + CHR$(174)
C$ = ">" + CHR$(143) + "<"
```

Fig. 4

If speed is no problem, but your coordination is, try inserting the lists shown in Fig. 3. With this option, you can have a sharper turn capability on your car whenever you hold down the shift key.

Car size and shape is easily altered by changing the arguments in the C$ declaration in Line 6. Fig. 4 diagrams some of the changes you might want to try. The railing can also be altered by changing A$. The most obvious change here is to add more blocks (and width) to the road with another " + CHR$(191)" in Line 6.

Scoring can get to be frustrating. You can change the format or delete the process altogether by deleting lines 40, 290, and the first half of 310. Finally, change line 320 to "320 RUN". You will now be competing against the clock only.

My experience with Racecar II shows its greatest use to be a demonstration program. It loads quickly from tape and instantly from disk; the rules are almost self-explanatory (perfect for small children and computer ignorants). It looks and behaves just like a video game and it always provides a challenge.

Besides, its gas mileage is unbeatable. ■

```
1 REM   RACECAR 2: BY RIK SPRAGUE AND JOHN SVETLIK
2 REM   TRS-80 L2-4K
3 REM   0898 BYTES   04JU80
4 REM
5 CLS:CLEAR50:DEFINTA-W:DEFSNGS
6 A$=CHR$(191)+CHR$(191)+CHR$(191):C$=CHR$(134)+CHR$(143)+CHR$(1
37)
7 P=1:C=10:E=1024
10 Z=SIN(X)*12:T=T+1
20 IFM=1THEN70
30 PRINT@980+(C-66),"        ";
40 S=S/10+T/10
50 PRINT@980+C,C$;
60 GOTO90
70 PRINT@988+Z,C$;
80 PRINT@920+Z,STRING$(11,32);
90 PRINT@980+Z,A$
100 PRINT@1000+Z,A$;
105 IFE<10THEN120
110 FORG=1TOE:NEXT:E=E/1.414
120 IFPEEK(15350)=0THENM=1:C=Z+8:ELSEM=1
130 IFPEEK(15350)=64THENC=C+2:M=0
140 IFPEEK(15350)=32THENC=C-2:M=0
150 IFT<15THEN10
160 IFRND(5)=1THENP=P*-1
170 X=X+Y*P:IFRND(15)=1THENGOSUB200
180 IFM=0THENIFC<ZORC>Z+20THENGOSUB250
190 GOTO10
200 ONRND(2)GOTO210  ,230
210 Y=.3
220 RETURN
230 Y=-.3
240 RETURN
250 REM  CRASH  SUBROUTINE
260 PRINT@980+C,STRING$(15,A$)
270 PRINT"CRAAASSSSHHHHHH!!!!!!!";
280 PRINT:PRINT"TIME="T
290 PRINT"SCORE="S;
300 FORG=0TO700:NEXT
310 S=S*.7:E=1024
320 GOTO10
```

Program Listing 1

```
0 REM   RACECAR:  A SWEAT-RIK PROGRAM
1 REM   TRS-80 L2 4K+
2 REM   1021 BYTES   JL79, JU80
3 REM
4 CLS:CLEAR50:DEFINTA-W
5 HO=32:J=HO
6 GOSUB360  :'  RANDOMIZE OPENING
10 GOTO260
20 REM  HERE'S WHAT MAKES IT TURN
30 Z=SIN(.9*X)*15*P
40 '
50 REM  IS ONE OF THE ARROWS PUSHED?
60 IFPEEK(15350)=32HO=HO-2
70 IFPEEK(15350)=64HO=HO+2
80 A4=A3:A3=A2:A2=A1:A1=Z+23
90 IFTIME<5A4=23
100 '
110 REM  DRAW  LEFT RAILING
120 PRINTTAB(Z+23)CHR$(124)CHR$(191);
130 '
140 REM  DRAW RACE CAR
150 PRINT@768+(HO),CHR$(134)CHR$(143)CHR$(137);
160 Y=702+J
170 PRINT@Y,"   "CHR$(133)"     ";:'  CLEAR "SHADOW"
180 J=HO
190 '
200 REM  DRAW RIGHT RAILING
210 PRINT@1001+Z,CHR$(191)CHR$(124)
220 R=A4:S=A4+18
230 '
240 REM  HORRORS!  HE CRASHED!!!
250 IFHO>SORHO<RPRINT"CRASH!!!!!!!":PRINT"TIME ="TIME;:GOTO340

260 IFTIME<10GOTO320
270 ONRND(2)GOTO280  ,290
280 X=X+.3
290 X=X-.3
300 '
310 REM INCREMENT THE TIMER
320 TIME=TIME+1
330 GOTO30
340 FORX=1TO750:NEXT
350 GOTO4
360 ONRND(2)GOTO370  ,380
370 P=-1:RETURN
380 P=1:RETURN
390 END
```

Program Listing 2

A printer that shouldn't be overlooked.

Epson's MX-80

Epson MX-80 Printer
Epson America, Inc.
Torrance, CA
$645

Tony Gitt
11260 Overland #10-B
Culver City, CA 90230

Do you need a printer for your TRS-80? There's a new printer distributed in the United States by Epson America, and like the ads say, "If you just bought another printer, boy, are you gonna be sorry."

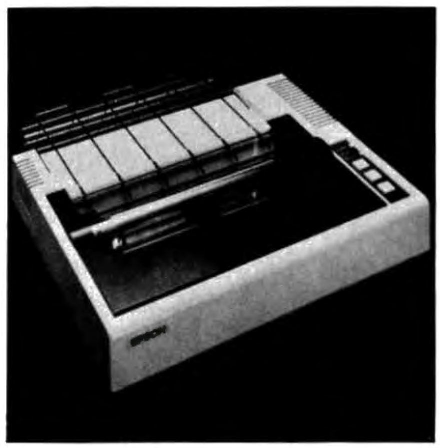
The Epson MX-80 Printer became available in the United States during the last quarter of 1980. The printer is made in Japan by Shinshu Seiki Company, Ltd., and if the design, quality and price of this printer are an indication of things to come, all I can say is that American printer manufacturers had better watch out.

What's an Epson?

The Epson MX-80 is an impact dot matrix printer that prints all 96 ASCII characters in a 9 by 9 dot format, and 64 graphics block characters in a 6 by 12 dot format. It prints normal characters in an 80-column format at 80 characters per second (CPS). It also prints a condensed or half-width set of characters in a 132-character per line format, and an expanded or double-width set of characters in both 40 or 66-character per line format.

The printer has a Centronics parallel interface, and connects to the TRS-80 Model I through the parallel printer port on the expansion interface. It uses 9 1/2-inch by 11-inch fan-fold tractor feed paper, which tears down to 8 1/2-inch by 11-inch sheets when the tractor holes are removed after printing.

The printer is housed in an attractive ivory-colored case. The power switch is located on the right side. The top front, right corner contains on-line, form feed and line feed switches, as well as power, ready and no paper indicator lamps. The printer measures 14.7 inches wide by 12 inches deep by 4.2 inches high, and weighs 12.1 pounds. That's relatively small and light for a printer, but the MX-80 is surprisingly rugged, which is a compliment to its designers.



Epson MX-80

ABCDEFGHIJKLMNOPQRSTUVWXYZ 01234567890 CONDENSED MODE
ABCDEFGHIJKLMNOPQRSTUVWXYZ 01234567890 NORMAL MODE
EXPANDED MODE
ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456
ABCDEFGHIJKLMNOPQRSTUVWXYZ 01234567890 NORMAL-EMPHASIZED MODE
EXPANDED-EMPHASIZED MODE
ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456
ABCDEFGHIJKLMNOPQRSTUVWXYZ 01234567890 CONDENSED-DOUBLE MODE
ABCDEFGHIJKLMNOPQRSTUVWXYZ 01234567890 NORMAL DOUBLE
EXPANDED-DOUBLE MODE
ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456
ABCDEFGHIJKLMNOPQRSTUVWXYZ 01234567890 NORMAL-EMPHASIZED-DOUBLE MODE
EXPANDED-EMPHASIZED-DOUBLE MODE
ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456

Fig. 1. Print Mode Samples

Reader Service for facing page ✓426.→

"The most impressive thing about the MX-80 is the quality of its print."

The most impressive thing about the MX-80 is the quality of its print. Dot matrix printers have always had several advantages over other types of printers, mainly in printing speed and mechanical simplicity. One of their disadvantages has always been the print quality. Since dots have some finite spacing between them, you can easily distinguish them from normal type. In addition, part of the characters which are normally rounded appear ragged on a dot matrix printer.

While this is perfectly acceptable for some applications, other applications require typewriter or what has become known as Selectric print quality. (Selectric refers to the IBM Selectric typewriters, which have interchangeable print heads, and provide very high quality print.) Many have been interfaced to computers and have become a kind of de facto standard when referring to print quality.

The first reasonably priced dot matrix printers that were suitable for the home

Character Size	Normal	Emphasized	Double Print	Double Print-Emphasized
	Enhancement Modes			
Normal 80 CPL	X	X	X	X
Condensed 132 CPL	X		X	
Expanded 40 CPL	X	X	X	X
Condensed Expanded GG CPL	X		X	

Table 1. Printing Modes

Program Listing 1

```
30000 '********* EPSON PRINTER MODE SELECTION PROGRAM *******
30010 'DISK NAME = PRNTMODE/BAS
30020 'LAST REVISION: 12/07/80
30030 CLS
30040 PRINT:PRINT"SELECT PRINTER MODE"
30050 PRINT
30060 PRINTTAB(10);"(1) COMPRESSED CHARACTERS (132 CHARS/LINE)"
30070 PRINTTAB(10);"(2) NORMAL CHARACTERS (80 CHARS/LINE)"
30080 PRINT:INPUT"ENTER NUMBER FOR CHARACTER MODE SELECTION";Z
30090 ONZGOSUB30390,30410
30100 IFZ=1THENM$="COMPRESSED (132 CHARS/LINE)"
30110 IFZ=2THENM$="NORMAL (80 CHARS/LINE)"
30120 CLS:PRINT:PRINT"PRINTER IS IN ";M$;" MODE"
30130 PRINT:PRINT"SELECT PRINT EMPHASIS MODE"
30140 PRINTTAB(10);"(1) NORMAL"
30150 PRINTTAB(10);"(2) DOUBLE PRINTING"
30160 PRINTTAB(10);"(3) EMPHASIZED"
30170 PRINTTAB(10);"(4) DOUBLE EMPHASIZED"
30180 PRINT:INPUT"ENTER NUMBER FOR ENHANCED PRINTING MODE SELECTION";Y
30190 PRINT:IFZ=1ANDY=3THENGOTO30196ELSEGOTO30192
```

Program continues

Program continued

```
30192 IFZ=1ANDY=4THENGOTO30196ELSE30210
30196 PRINT"YOU CANNOT SELECT 'EMPHASIZED' FOR COMPRESSED CHARACTERS"
30200 PRINT:INPUT"PRESS ENTER TO CONTINUE";Z$:GOTO30120
30210 ONYGOSUB30450,30470,30490,30510
30220 IFY=1THENE$="NORMAL PRINTING"
30230 IFY=2THENE$="DOUBLE PRINTING"
30240 IFY=3THENE$="EMPHASIZED PRINTING"
30250 IFY=4THENE$="DOUBLE EMPHASIZED PRINTING"
30260 CLS:PRINT:PRINT"MODES SELECTED:"
30270 PRINTTAB(18);M$
30280 PRINTTAB(18);E$
30290 PRINT:PRINT"SELECT LINE SPACING"
30300 PRINTTAB(10);"(1) 1/6 INCH (NORMAL)"
30310 PRINTTAB(10);"(2) 1/8 INCH"
30320 PRINTTAB(10);"(3) 7/72 INCH"
30330 PRINTTAB(10);"(4) 1/4 INCH (DOUBLE SPACING)"
30340 PRINTTAB(10);"(5) SPECIFY SPACING IN 1/72 INCH INCREMENTS"
30350 PRINT:INPUT"ENTER NUMBER FOR LINE SPACING SELECTION";V
30360 ON VGOSUB30530,30550,30570,30590,30610
30380 END
30390 POKE14312,15
30400 RETURN
30410 POKE14312,18:POKE14312,20
30420 RETURN
30430 POKE14312,18:POKE14312,14
30440 RETURN
30450 POKE14312,27:POKE14312,70:POKE14312,27:POKE14312,72
30460 RETURN
30470 POKE14312,27:POKE14312,70:POKE14312,27:POKE14312,71
30480 RETURN
30490 POKE14312,27:POKE14312,72:POKE14312,27:POKE14312,69
30500 RETURN
30510 POKE14312,27:POKE14312,69:POKE14312,27:POKE14312,71
30520 RETURN
30530 POKE14312,27:POKE14312,65:POKE14312,140:POKE14312,27:POKE14312,50
30540 RETURN
30550 POKE14312,27:POKE14312,65:POKE14312,137:POKE14312,27:POKE14312,48
30560 RETURN
30570 POKE14312,27:POKE14312,65:POKE14312,135:POKE14312,27:POKE14312,49
30580 RETURN
30590 POKE14312,27:POKE14312,65:POKE14312,146:POKE14312,27:POKE14312,50
30600 RETURN
30610 INPUT"ENTER NUMBER OF 1/72 INCH INCREMENTS YOU WISH TO SPACE";W
30620 POKE14312,27:POKE14312,65:POKE14312,(128+W):POKE14312,27:POKE14312,50
30630 RETURN
```

Program Listing 2

```
10 '********* EPSON EXERCISER ********
20 'DISK NAME = EPEXCRSR/BAS
30 'LAST REVISION: 12/19/80
40 LPRINT:LPRINTTAB(25);"EPSON PRINTER MODE TESTS"
50 LPRINTCHR$(18);:LPRINTCHR$(20);:LPRINTCHR$(27)"F";:LPRINTCHR$
(27)"H";:
60 LPRINTCHR$(15);
70 GOSUB560  :LPRINT"CONDENSED MODE"
80 LPRINTCHR$(18);
90 LPRINT"NORMAL MODE":GOSUB510
100 LPRINTCHR$(14);:LPRINT"EXPANDED MODE"
110 LPRINTCHR$(14);
120 GOSUB560
130 LPRINTCHR$(27)"E";
140 LPRINT"NORMAL-EMPHASIZED MODE":GOSUB510
150 LPRINTCHR$(14);:LPRINT"EXPANDED-EMPHASIZED MODE"
160 LPRINTCHR$(14);:GOSUB560
170 LPRINTCHR$(27)"F";
180 LPRINTCHR$(27)"G";
190 LPRINTCHR$(15);
200 LPRINT"CONDENSED-DOUBLE PRINT MODE":GOSUB560
210 LPRINTCHR$(18);
220 LPRINT"NORMAL DOUBLE PRINT":GOSUB510
230 LPRINTCHR$(14);:LPRINT"EXPANDED-DOUBLE PRINT MODE"
240 LPRINTCHR$(14);
250 GOSUB560
260 LPRINTCHR$(27)"E";
270 LPRINT"NORMAL-EMPHASIZED-DOUBLE PRINT MODE":GOSUB510
280 LPRINTCHR$(14);:LPRINT"EXPANDED-EMPHASIZED-DOUBLE PRINT MODE
"
290 LPRINTCHR$(14);
300 LPRINTCHR$(27)"G";
310 LPRINTCHR$(27)"E";
320 GOSUB560
330 LPRINTCHR$(27)"F"CHR$(27)"H";
340 LPRINT"NORMAL LINE SPACING"
350 GOSUB580  :LPRINT" 1/6INCH"
360 GOSUB580  :LPRINT" 1/6INCH"
370 LPRINTCHR$(27)CHR$(48);
380 LPRINT"1/8 INCH LINE SPACING"
390 GOSUB580  :LPRINT" 1/8INCH"
400 GOSUB580  :LPRINT" 1/8INCH"
410 LPRINTCHR$(27)"1";
420 LPRINT"7/72 INCH LINE SPACING"
430 GOSUB580  :LPRINT" 7/72INCH"
440 GOSUB580  :LPRINT" 7/72INCH"
450 LPRINT:LPRINTCHR$(27)"A"CHR$(146)CHR$(27)"2";
460 LPRINT"DOUBLE SPACING (1/4 INCHES)"
470 GOSUB580  :LPRINT" DOUBLE LINE SPACING"
480 GOSUB580  :LPRINT" DOUBLE LINE SPACING"
490 LPRINTCHR$(27)"A"CHR$(140)CHR$(27)"2"
500 END
510 FORX=161TO223
520 LPRINTCHR$(X);
530 NEXTX
540 LPRINT:LPRINT:LPRINT"ABCDEFGHIJKLMNOPQRSTUVWXYZ 01234567890"
:LPRINT
550 RETURN
560 LPRINT"ABCDEFGHIJKLMNOPQRSTUVWXYZ 01234567890":LPRINT
570 RETURN
580 LPRINT"ABCDEFGHIJKLMNOPQRSTUVWXYZ 01234567890";
590 RETURN
```

"The new generation of dot matrix printers eliminates the no descenders problem."

computer market utilized a matrix seven dots high and five dots wide. This type of matrix had limitations because the dots were relatively far apart, making the unprinted space between them fairly visible; and the matrix didn't go below the letter baseline—they couldn't print descenders. Descenders are the tails of letters like y and j.

The new generation of dot matrix printers eliminates the no descenders problem, and greatly reduce the visible spaces between the dots, since the dots are closer together. The Centronics 747 was the first new printer to appear with these capabilities. More recently the Epson MX-80 appeared.

The Epson MX-80 utilizes a 9 by 9 dot character matrix. The nine-dot height allows the dots to be placed below the baseline. The dots are spaced so close together they almost overlap but some small spaces are still visible.

Filling in the Spaces

Epson has several solutions for filling in the spaces. The first is to switch on the double-printing or double-striking mode. In this mode the printer prints a line, then rolls the paper up 1/216 inch and prints the line again. The result is that almost all the visible spaces between the dots are filled in when the character is imprinted the second time, just slightly offset from the first imprint of the character. Double printing reduces the printing speed by 50 percent to 40 CPS for a full line of 80 characters.

A second solution is to switch to the emphasized-printing mode. In this mode the printer makes a stronger impact on the paper as each dot is struck. This also causes a 50 percent speed reduction.

You can turn on the double-printing mode and the emphasized-printing mode together, which results in a very heavy letter imprint with no spaces between the dots. The speed in this mode is reduced 75 percent to 20 CPS.

Considering the various character sizes and enhancement modes, there are 10 different printing modes in which the Epson can be operated. Table 1 lists these modes. There is no emphasized mode or double print emphasized mode for condensed characters, because condensed character dots are already so close together that striking harder would smudge the printout and probably damage the print head. Samples of the print modes are shown in Fig. 1. Lowercase characters and condensed-expanded are not shown, but the printer does print them. Listing 1 is the program used to obtain the printouts of Fig. 1.

The result of the enhanced printing modes is clean, nicely formatted print. It still isn't Selectric quality, but it is some of the best dot matrix printing around.

Printing Speed

Normal speed for printing an unenhanced character on the Epson MX-80 is 80 CPS. Since the normal mode is 80 characters per line, the printer prints a full line in one second. However, if there are less than 80 characters in the line, the printer can print the line faster if the print head doesn't have to travel all the way across the paper. Epson has taken this into account by designing their printer with bi-directional print head control, as well as the ability to sense when a line is only partially full. In that case the printer stops at the last character and line feeds to the next line. This can result in average printing speeds that are faster than

ASCII Char.	Decimal Code	Function
SO	14	Expanded width cha.
DC 4	18	Cancel expd width
SI	15	Condensed wdth char
DC 2	18	Cancel condsd char
ESC E	27,29	Emphasized mode
ESC F	27,70	Cancel emphasized
ESC G	27,71	Double printing
ESC H	27,72	Cancel double prnt
ESC O	27,48	1/8" line spacing
ESC 1	27,49	7/72" line spacing
ESC 2	27,50	1/16" line spacing
ESC A	27,65	Variable line spacing

Note: ESC 2 must be used after ESC A to execute variable line spacing.

Table 2. Printer Control Codes

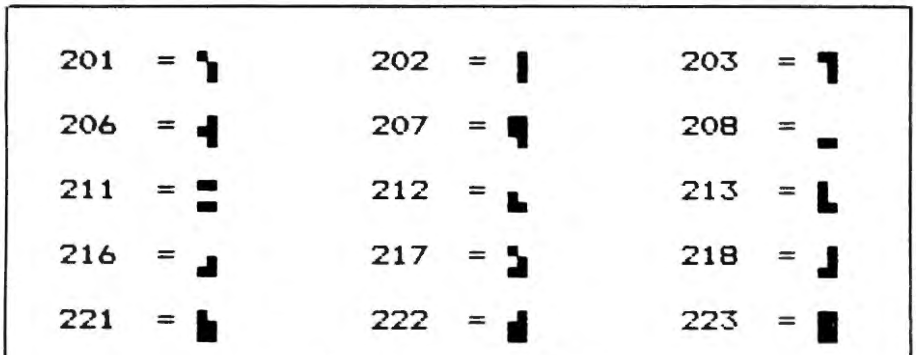


Fig. 2. Graphics Characters.

```
NORMAL LINE SPACING
ABCDEFGHIJKLMNOPQRSTUVWXYZ 01234567890l/6INCH
ABCDEFGHIJKLMNOPQRSTUVWXYZ 01234567890l/6INCH
1/8 INCH LINE SPACING
ABCDEFGHIJKLMNOPQRSTUVWXYZ 01234567890l/8INCH
ABCDEFGHIJKLMNOPQRSTUVWXYZ 01234567890l/8INCH
7/72 INCH LINE SPACING
ABCDEFGHIJKLMNOPQRSTUVWXYZ 01234567890?/72INCH
ABCDEFGHIJKLMNOPQRSTUVWXYZ 01234567890?/72INCH
DOUBLE SPACING (1/4 INCHES)

ABCDEFGHIJKLMNOPQRSTUVWXYZ 01234567890DOUBLE LINE SPACING

ABCDEFGHIJKLMNOPQRSTUVWXYZ 01234567890DOUBLE LINE SPACING
```

Fig. 3. Line Spacing.

"The whole manual, with cartoon illustrations and step-by-step examples, is excellent!"

the normal 80 CPS, or 40 or 20 CPS if the enhanced printing modes are selected.

Graphics

Another capability built into the Epson MX-80 adds to its versatility, and nicely complements the Model I's capabilities: the ability to print 64-block graphics characters. A printout of these graphics in the emphasized mode is shown in Fig. 2. The graphics may be emphasized, double printed, or both.

The MX-80 normally allows 1/6 inch between lines, equivalent to 12 vertical dots. The processor within the printer will also accept commands to allow 1/8 inch (nine vertical dots) between lines, or 7/72 inch (seven vertical dots) between lines. In addition to those built-in line spacing options, you can program any amount of line spacing you desire.

Controlling the Printer

All printing capabilities are conveyed to the printer by a series of control codes. Table II lists the codes, with the exception of the vertical and horizontal tabs.

To get the codes into the printer from the TRS-80, address the printer and then send the code. In assembly language you would output the hexadecimal control code or codes to the printer at memory address 37E8 (H).

In Basic there are several ways to output the codes to the printer. The most straightforward is to use the LPRINT statement. By LPRINTing the decimal code for the ASCII control character in an ASCII control character string (CHR$), the proper code is sent to the printer to go into the compressed character mode (132 characters per line): LPRINT CHR$(15).

CHR$ is the character string Basic function and 15 is the decimal code for the ASCII character SO. The printer would print the line in the compressed character mode. To cancel the compressed character mode, type: LPRINT CHR$(18). This line sends the ASCII character DC 2 (decimal code 18) to the printer and cancels the compressed character mode.

When two control codes are necessary to command a function, such as in the case of emphasized printing (ESC E), there are several ways in Basic to send the codes to the printer: LPRINT CHR$(27) CHR$(69), or LPRINT CHR$(27) "E", or POKE 14312,27 : POKE 14312,69.

Use of the POKE command avoids an unwanted linefeed/carriage return at the end of a Basic program since the LPRINT command automatically sends a LF/CR. The POKE command will avoid registration problems with preprinted forms of fanfold paper.

One mode needs further explanation. When using the expanded-character mode, upon receiving a carriage return/line feed the printer is designed to return to whatever print mode was previously selected prior to selecting the expanded-character mode. The other modes stay selected until a new mode is selected.

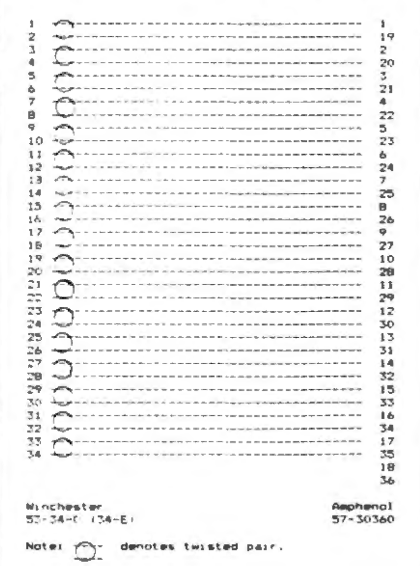


Fig. 4.

Selection Program

There are many control code combinations that can be sent to the printer. The Epson printer mode selection program (Listing 2) provides a simple selection capability in Basic. It uses POKE statements to avoid unwanted line spacing.

The program first asks you to select compressed or normal characters. (It doesn't provide for expanded characters since they would print for only one line.) If you want to use the expanded-character mode you will have to write the control code statements into your program for each line that you want to print with expanded characters.

It next asks you to select an enhancement mode. Finally, it asks you to select line spacing.

Testing the Printer

By holding the line-feed button down while turning on the power switch, the printer will run through its paces, rapidly filling up about two and one-half sheets of paper. The built-in test does not print the various enhancement modes or the variable line spacing modes.

Connecting the Printer to the TRS-80

Unfortunately, the Epson MX-80 does not come with a cable that will interface it to the TRS-80 Model I expansion interface. The printer uses a 36-pin D-type connector, and the expansion interface utilizes a 34-pin card-edge connector. Epson recommends that twisted pair wire be used. Fig. 4 shows the cable interconnections and the connector part numbers. I built my own cable for $15.

I made my cable twelve feet long, and have no noise pick-up problems. The MX-80 manual says that a Radio Shack cable (number 26-4402) will work, it is priced at $29.

WOW—What a Manual!

The MX-80 comes with a 40-page paperback-size manual. It adequately explains how to unpack and use your printer, and has only a few Japanese to English translation errors. Shortly after I got my printer I heard a rumor that Epson had another manual. A phone call to Epson produced a pleasant surprise. The new manual arrived in the mail the very next day. Epson hired David A. Lien, author of *The TRS-80 Level I User's Manual,* to write a manual for the MX-80.

The result is a 107-page, full-sized manual that explains almost everything you might want to know about your MX-80, in a simplified language even the most unsophisticated computer user can easily understand. The whole manual, with cartoon illustrations and step-by-step examples, is *excellent*! The only thing I miss in the manual is an explanation of how to use POKE statements to avoid extra line feeds when commanding the printer with control codes, and a schematic and parts list of the printer.

I looked very carefully at the competition before buying the MX-80. After using the printer for two months, there are only three things I have found lacking:

1. Friction paper feed capability
2. Proportional printing capability
3. Automatic form feed after printing a programmable number of lines

The Centronics 747 has items one and two, but after trying one I felt the print quality wasn't as good as the Epson, and the price was significantly higher. The Base 2 printer has item 3, but the print quality doesn't compare with the MX-80, and the price is higher.

And what about price? I ordered my printer in late October of 1980 from Patio Computer in North Hollywood, CA. At that time the list price of the MX-80 was $645, but I found that most dealers were discounting the MX-80 when they had them. I had to wait six weeks for a new shipment from Japan.

I can only tell you that I am very happy with my Epson MX-80 printer. For the price, you just can't beat the quality. ■

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.

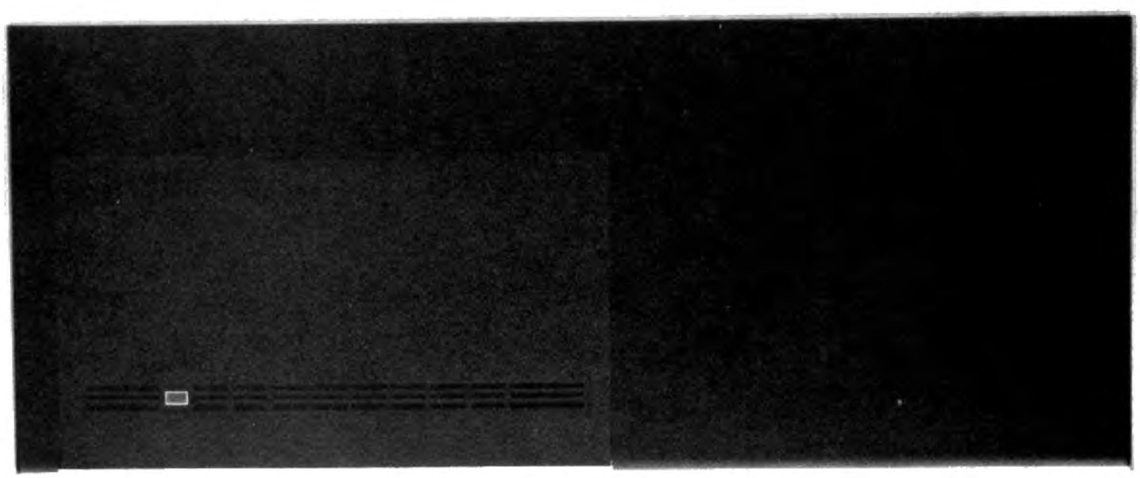
Introductory Offer To TRS80 Users.

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. ✓26
669 Giraudo Drive, San Jose, CA 95111
(408) 629-2283



laredo systems inc.

MICROCOMPUTER TECHNOLOGY INCORPORATED

SUMMER SALE

CLEARANCE SPECIAL
SHUGART SA400 (TF-3)

35 track disk drive includes power supply and chassis.

LIMITED SUPPLY

$289

FREE FULL YEAR WARRANTY

with purchase of new 40 TK.
TANDON DISK DRIVE
comes complete with power supply, chassis, cable & DOS PLUS
ONLY $409

DOUBLE DENSITY SPECIAL!

MTI Doubler Pkg. includes doubler and microsystems DOS PLUS 3.3D, run single and double density, plus a box of Verbatim diskettes.

$275

DISK STORAGE SPECIAL

Buy 5 boxes of diskettes at $30.00 ea. and get a $30 Value Flip-Sort

FREE

LEXICON II ACOUSTIC MODEM SPECIAL CLOSEOUT

$119

Memory Kit (16K)	$43.00
AC Isolator (6 socket)	$49.95
Disk Head Cleaner	$19.95
Diskettes Verbatim (10) hard ring 5" s.s	$30.00

Disk Drive Sale!

with power supply and chassis

Shugart SA400L, 40 trk.	$319
Pertec FD200, 40 trk.	$349
TF-5 MPI B51, 40 trk.	$349
TF-11 Tandon, 40 trk.	$309
TDH-1 Dual sided drive, 35 trk.	$419
TF-3M Drive Sys. 3 Shugart 40 trk. (1 cabinet)	$599

SOFTWARE SALE

DOS+ 3.3	$99
DOS+ 3.3D	$99
Disk & System Diagnostic	$49

INTERFACES - MOD I

Interface 16K	$349
Interface 32K	$419

MONTHLY SPECIAL - NEW FOR TRS-80®

TF-15 80 TRACK DISK DRIVE

Double your capacity. 200K bytes of storage. Complete with power supply and chassis.

$479

TF-17 DUAL 80 TRACK DISK DRIVE

Quadruple your capacity. 400K bytes of storage. Complete with power supply and chassis.

$659

SYSTEMS — MOD I

Interface 32K, TF-11 disk drive cable and DOS 3.3

$799

MOD I Disk Expansion Systems

- 2 TANDON TF-11 .... $618
- 1 Two-Drive Cable $26
- 1 Expansion Interface 32K $419
- 1 DOS+ 3.3 .... $99

TOTAL LIST PRICE $1162

SPECIAL PRICE ONLY **$1029**

★ BARE DRIVES FOR ANY MICROCOMPUTER

Pertec FD200	$282	FD250	$359
Shugart SA400 35 trk.	$269	SA400L 40 track	$279
MPI B51	$279	B52	$349
MPI B91	$399	B92	$525
TANDON 40 trk.	$259	TANDON 80 trk.	$339
TANDON D/40 trk.	$339	TANDON D/80 trk.	$504

MICROCOMPUTER TECHNOLOGY, INC.

3304 W. MACARTHUR, SANTA ANA, CA 92704
★ PHONE (714) 979-9923 ★ TELEX #678401TAB IRIN ✓28

®TRS-80™ IS A REGISTERED TRADEMARK OF TANDY CORPORATION

MasterCard VISA

ALL PRICES CASH DISCOUNTED
FREIGHT FOB FACTORY
ASK FOR FREE CATALOG

✓See List of Advertisers on page 322

The last piece of software you'll ever need.

Being of Sound Algorithm

Carol S. Weinberg
11 Locust St.
Closter, NJ 07624

This program will help you write a simple will. It may encourage those who ordinarily would not write any will, and guide those who might improvise without adequate research.

The program guides you through the traditional sections of a will and asks you to think about your estate and the future security of people in your life. It will also help you update your will conveniently, economically, and as often as changing circumstances dictate.

Warning

You should first be aware of some limitations: I am a researcher, not an attorney. The content and format of the will and the information contained in these instructions are based on dependable sources, however. But estate and inheritance laws vary state to state: The person who writes his or her own will cannot assume it will be as flawless as it would be if it were designed by an attorney. The value of the final document will depend on how carefully you evaluate your personal circumstances, the terminology you use writing the will, the complexity of your estate, and some simple common sense.

As a first time user of the program, consider submitting the final will to an attorney for review. The attorney's fee should be nominal (since the will has already been written) and worth the additional security of legal approval.

The program is flexible, but like any standardized format, has certain limitations. For this reason, it is best for *simple* wills only. A simple will need not be short, or limited to one or two beneficiaries: Simplicity is determined by your circumstances. If your circumstances are unusual or complex, you should have an attorney draft your will.

Complex circumstances may include considerations like the following:

- Your estate exceeds $175,000
- Your children or any of your beneficiaries require extraordinary care (e.g., if they are disabled)
- You wish to leave someone out of your will who would ordinarily be viewed as a legitimate heir (e.g., disinherit a spouse or child); or you show a disproportionate preference for one person compared with others who have the same or a higher kinship status
- You wish to name more than one guardian to care for your children and administer the property you leave to them; or
- You want to set up a trust

Preparation

Do a thorough inventory of your assets and work up a net worth statement. Include in it cash and any property with tangible monetary value (real estate, bank accounts, stocks, insurance policies, jewelry, works of art, etc.) which you own in whole or in part. Now, list any property of sentimental value you would like to pass on to specific individuals. Determine what you want to give and to whom. It's a good idea to leave something to your spouse, any children you may have, and any grandchildren, even if it is only nominal. (Some states require that you do so.)

Record the full legal names of your potential beneficiaries, and check spelling. Finally, be sure that the total value of your bequests does not exceed the total value of your estate *less* any taxes or fees and burial expenses which you estimate will be incurred when your estate is settled.

The Program

The program, written in Basic for a 16K TRS-80 Level II, should run on any line printer, including those which print only 32 or 40 characters per line. The will is not programmed to print out on separate pages, however. If you have a long will and cannot use continuous roll paper in your printer, make adjustments to the program, or control your printer manually.

It's a good idea to make a dry run so that you can be prepared to answer all the questions that are presented, and become familiar with the printed format, as well as the limitations and capabilities of the program.

"The executor sees that the requests you make are carried out. He or she should be trustworthy. . ."

Executor

The executor is the person who will see that the requests you make in your will are carried out. He or she should be trustworthy, young enough to survive you, and competent. Often the executor is the person who will inherit the major part of your estate, but this need not be so. The executor can be a friend of the family, an attorney, or even an institution (such as your favorite bank). Whoever or whatever you choose, discuss this matter with them first and obtain their agreement.

The program permits you to enter the names of two alternative executors. This should not be overlooked, it is not unusual for the person you named first to default (due to death, disablement, refusal, etc.). Additionally, the program allows you to specify whether or not you want the executor to be bonded. If your executor(s) are trustworthy, waive this option. Bonding entails additional expense. You should be aware that, in some states, an executor will be required to furnish a bond at the discretion of the probate judge even if this option is waived.

The wording of the executor section of the will reads, "I designate my (relationship), (name of person), executor. . ." All other sections read similarly. Adjectives can be included in the relationship description (e.g., "my dear and competent friend, John Doe"). A descriptive phrase may also be included along with the name (e.g., "my attorney, John Doe, who has served me faithfully for 30 years"). Using such phrases helps personalize your will, further identify the person intended, and explain the reasons behind your actions. Try to keep them short, however.

Guardian

If you have any minor children, a guardian will look after them and their property, acting as a surrogate parent, until they are legally of age. The program assumes that if you have a surviving spouse, he or she will perform this function.

Besides your spouse, the program allows you to name up to three persons as potential guardians for your children—a primary guardian (secondary in the case that you have a spouse), and two additional guardians in the case that your spouse/or primary guardian do not survive long enough to complete their responsibility. The program allows you to sepcify whether or not you want the guardian(s) to be bonded. Again, if the people you elected are trustworthy, you can waive this option.

Specific (Noncash) Gifts

Noncash gifts are intended for people you want to remember but do not intend to bequeath the major part of your estate. The gifts may be of monetary or sentimental value (e.g., shares of stock, a wedding ring, a photo album, etc.). Any item you do not specify here will be considered part of your residuary estate (any property which remains after all specific bequests are made) and will be inherited by the person you specify as beneficiary.

Describe each item you list specifically, to avoid confusion with other items you own. For example, if you list an original oil painting and you own several, describe the

Sample Printout

```
** PROGRAM AUTOMATICALLY ADJUSTS LINE LENGTH TO THAT ALLOWED BY YOUR
   PRINTER WITHOUT SPLITTING WORDS AT ENDS OF LINES. **

                              Last Will of            ** TITLE AUTO-
                              Walter Rich             MATICALLY CENTERED **

                  I, Walter Rich, of One Sutton
               Place, New York, New York, state      ** STANDARD TEXT **
               that this is my last will,
               revoking all previous wills.
               1. Executors:
                  I designate my bank, First
               Federated, executor of this will.
                  I designate my brother, Will       ** ALLOWS SPECIFICA-
               Rich, as alternative executor,       TION OF TWO ADDITIONAL
               in case my bank, First Federated,    EXECUTORS **
               is unable or unwilling to act as
               executor, or ceases to do so.
                  I request that the executor        ** OPTION TO BOND
               be bonded.                            OR NOT **
** OPTIONAL,   2. Guardians:
DEPENDING ON      If my spouse, Wanda Rich,          ** OPTIONAL, DEPEND-
EXISTENCE OF   dies before me or at the same         ING ON EXISTENCE OF
MINOR CHIL-    time, I designate my mother,          SPOUSE **
DREN **        Althea Rich, guardian of the
               person and property of my
               children until they are legally
               of age.
                  I designate my brother, Will       ** ALLOWS SPECIFICA-
               Rich, as alternative guardian,        TION OF TWO ALTERNA-
               in case my mother, Althea Rich,       TIVE GUARDIANS **
               is unable or unwilling to act as
               guardian or ceases to do so.
                  I do not require that the          ** OPTION TO BOND
               guardian be bonded.                   OR NOT **
** OPTIONAL,   3. Specific Gifts:
DEPENDING ON      I leave my ten thousand            ** GIFTS MAY BE DES-
WISH TO LEAVE  (10,000) shares of Tandy Corp.        CRIBED AS SPECIFICALLY
SPECIFIC       common stock to my mother,            AS NECESSARY **
GIFTS (UP TO   Althea Rich.
TEN MAY BE        If that person dies before         ** OPTION FOR ALTER-
SPECIFIED) **  me or at the same time, this          NATIVE BENEFICIARY
               gift is to become part of my          (NOT TAKEN HERE) **
               residuary estate.
** OPTIONAL,   4.  General Gifts:
DEPENDING ON       I give $2,000 yearly to the       ** FLEXIBILITY IN
WISH TO LEAVE  person who cares for my pets,         LIMITING AMOUNT AND
CASH GIFTS     Rover and River, as long as           DETERMINING BENEFI-
(UP TO TEN     either shall live. If that per-       CIARY **
MAY BE SPE-    son dies before me or at the
CIFIED)        same time, this gift is to
               become part of my residuary
               estate.
                  I give ten thousand dollars        ** ALLOWS USE OF DES-
               ($10,000) to my adoring mother,       CRIPTIVE TERMS TO
               Althea Rich, to help provide          PERSONALIZE WILL AND
               for her aging years.                  EXPLAIN ACTION TAKEN **
                  If that person dies before
               me or at the same time, I give        ** OPTION FOR ALTER-
               ten thousand dollars ($10,000)        NATIVE BENEFICIARY
               to the General Wildlife Fund.         (TAKEN HERE) **
               5. Residuary Estate:
                  I will the remainder of my
               property to my amorous wife,.
               Wanda Rich.
** ALLOWS UP      If that person (those per-         ** MULTIPLE BENEFI-
TO THREE AL-   sons)-die before me or at the         CIARY ALLOWED, WITH
TERNATIVE      same time, I will this property       STIPULATION OF PRO-
BENEFICIARIES ** to my brother, Will Rich, and       PORTIONING OF ESTATE **
               my mother, Althea Rich, to be
               apportioned respectively as
               follows:  30%,70%.
               6. Taxes:
```

Sample continues

All taxes and government fees related to the settlement of my estate are to be paid out of the residuary estate. 7. Executor's Options: In order to carry out the settlement of my estate, I confer upon my executor full power to sell, lease, mortgage, or otherwise liquidate the assets in my estate.	** STANDARD TEXT **
My executor should adjust the number of shares of securities I leave as specific gifts to account for all stock dividends, splits, or other changes in the form of the stock occurring after the date of this will.	** OPTIONAL, DEPENDING ON RELEVANCE **
Signed: ---------------------- Date: ------------------------- Witnesses: According to Walter Rich's wish, we all assembled on the date shown above to witness the signing of this will. With all of us there at the same time, Walter Rich signed it and described it as his last will.	** STANDARD TEXT **
Signed: ---------------------- Address: ---------------------- ----------------------	** SPACE FOR THE SIGNATURES OF THREE WITNESSES **

Program Listing

```
10 '*** SIMPLE WILL IN BASIC, AUGUST 14, 1980 ***'
30 'WRITTEN FOR TRS 80 LEVEL II, 16K WITH LINE PRINTER'
40 CLS:PRINT:PRINT:PRINT:PRINT:PRINTCHR$(23)"        CUS
   TOMIZED WILL":FOR Z=1 TO 1000:NEXT Z
50 CLS:PRINT:PRINT:PRINT"THIS IS A PROGRAM WHICH WILL C
   USTOMIZE A SIMPLE WILL AND PRINT  IT OUT ON YOUR L
   INE PRINTER, READY FOR  YOUR SIGNATURE AND THOSEOF
    THREE WITNESSES."
160 PRINT:PRINT"FOR BACKGROUND INFORMATION, SEE AN ARTI
    CLE ENTITLED
170 PRINT"'WHAT YOU SHOULD KNOW ABOUT WILLS,' PUBLISHED
     IN
180 PRINT"CONSUMER REPORTS, JULY 1980, ON WHICH THIS PR
    OGRAM
190 PRINT"IS BASED."
200 PRINT:PRINT"HIT 'ENTER' WHEN READY TO PROCEED.":INP
    UT Z
210 CLS:PRINT:PRINT:PRINT:PRINT
215 CLEAR.CLEAR 2500:DEFSTR A-T
220 PRINT"BE SURE YOUR LINE PRINTER IS HOOKED UP AND RE
    ADY. THEN ENTER   MAXIMUM NUMBER OF CHARACTERS PE
    R LINE (32, 40 OR 80) OF WHICH   YOUR LINE PRINTER
     IS CAPABLE.":INPUT V
245 CLS:PRINT:PRINT"*** ENCLOSE INFORMATION IN QUOTATIO
    N MARKS ***"
250 PRINT:PRINT:INPUT "YOUR FULL LEGAL NAME";T
260 PRINT:PRINT:INPUT "AND COMPLETE ADDRESS";AD
270 CLS:PRINT:PRINT:PRINT:INPUT"AND WHAT IS YOUR SEX (M
    /F)";SX
275 CLS:PRINT:PRINT:PRINT:PRINT:PRINTCHR$(23)"
     EXECUTORS":FOR Z=1 TO 1000:NEXT Z
320 GOSUB 9000:PRINT"WHO DO YOU WISH TO NAME AS EXECUTO
    R OF YOUR ESTATE? (TYPE
330 PRINT"IN THE RELATIONSHIP OF THE INDIVIDUAL/INSTITU
```

Program continues

```
     TION TO YOU,
340 PRINT"AND THE FULL NAME OF THE INDIVIDUAL/INSTITUTI
     ON, SEPARATING
350 PRINT"EACH WITH A COMMA, E.G., 'MY BROTHER, JOHN A.
     DOE' OR
360 PRINT"'MY BANK, FIRST NATIONAL OF WEEHAUKEN.'  YOU
     MAY USE
361 PRINT"DESCRIPTIVE WORDS AND PHRASES IF YOU LIKE."
365 INPUT E1
370 GOSUB 9000
380 PRINT"AND WHO IS TO BE THE EXECUTOR IN THE EVENT TH
     AT THE FIRST
390 PRINT"NAMED INDIVIDUAL IS UNWILLING OR UNABLE TO CA
     RRY OUT THOSE
400 PRINT"RESPONSIBILITIES? (TYPE RELATIONSHIP, NAME AN
     D ANY DESCRIPTIVE  PHRASES YOU LIKE)":INPUT E2
401 IF E2="NONE" THEN 440
410 PRINT:PRINT"AND WHO IS TO BE THE EXECUTOR IN THE EV
     ENT THAT BOTH THE
420 PRINT"FIRST AND SECOND INDIVIDUALS ARE UNWILLING OR
      UNABLE TO CARRY
430 PRINT"OUT THESE RESPONSIBILITIES? (TYPE RELATIONSHI
     P, NAME, AND       OPTIONAL DESCRIPTIVE PHRASES)":
     INPUT E3
440 CLS:PRINT:PRINT:PRINT:PRINT"DO YOU WISH THE EXECUTO
     RS TO BE BONDED?":PRINT:PRINT"(IF YOU TRULY TRUST
     THE INDIVIDUALS YOU NAMED, BONDING IS AN"
450 PRINT"UNNECESSARY EXPENSE.)":INPUT EB
460 CLS:PRINT:PRINT:PRINT:PRINT:PRINTCHR$(23)"
      GUARDIANS":FOR Z=1 TO 1000:NEXT Z
490 CLS:PRINT:PRINT:PRINT:INPUT"HOW MANY CHILDREN UNDER
      18 YEARS OF AGE DO YOU HAVE, IF ANY";Y
500 IF Y=0 THEN 730
510 CLS:GOSUB 9000:PRINT:PRINT"DO YOU HAVE A SPOUSE WHO
      WOULD BE RESPONSIBLE FOR YOUR
520 PRINT"CHILDREN AFTER YOUR DEMISE?  IF SO, ENTER NAM
     E OF SPOUSE;":PRINT"IF NOT, ENTER 'NONE'."
540 INPUT SP
550 IF SP="NONE" THEN 570
560 PRINT:PRINT:PRINT:PRINT:PRINT"IF YOUR SPOUSE SHOULD
      DIE BEFORE YOU OR AT THE SAME TIME...":FOR Z=1 TO
     1500:NEXT Z
570 GOSUB 9000:PRINT"WHO DO YOU WISH TO APPOINT AS GUAR
     DIAN FOR YOUR MINOR CHILDREN?"
580 PRINT"(ENTER RELATIONSHIP AND NAME OF INDIVIDUAL, S
     EPARATED BY A
590 PRINT"COMMA. E.G., 'MY SISTER, JANE E. DOE'.)  YOU
     MAY INSERT"
600 PRINT"DESCRIPTIVE PHRASES IF YOU LIKE.
610 INPUT G1
640 GOSUB 9000:PRINT"AND WHO DO YOU APPOINT GUARDIAN IN
      CASE THE INDIVIDUAL YOU
650 PRINT"NAMED FIRST IS UNWILLING OR UNABLE TO CARRY O
     UT THOSE RESPON-
660 INPUT"SIBILITIES";G2
665 IF G2="NONE" THEN 700
670 PRINT:PRINT"AND WHO DO YOU APPOINT IN CASE THE FIRS
     T AND SECOND NAMED
680 PRINT"INDIVIDUALS ARE UNWILLING OR UNABLE TO CARRY
     OUT THOSE RE-
690 INPUT"SPONSIBILITIES";G3
700 CLS:PRINT:PRINT:PRINT:PRINT"DO YOU WISH THE GUARDIA
     NS TO BE BONDED?  (IF YOU TRULY TRUST
710 PRINT"THESE INDIVIDUALS, BONDING IS AN UNNECESSARY
     EXPENSE.)
720 INPUT GB
730 CLS:PRINT:PRINT:PRINT:PRINT:PRINTCHR$(23)"        SP
     ECIFIC GIFTS":PRINT:PRINT"(YOU MAY SPECIFY UP TO 1
     0 GIFTS)":FOR Z=1 TO 1500:NEXT Z
830 FOR Z=1 TO 10:GOSUB 9000
835 PRINT"===== GIFT #";Z;"======"
840 PRINT"DESCRIBE GIFT":INPUT D(Z)
845 IF D(1)="NONE" THEN 920
850 IF Z>1 AND D(Z)="NONE" THEN 890
860 PRINT:PRINT"AND THE NAME OF THE BENEFICIARY?  (ENTE
     R RELATIONSHIP AND NAME
```

Program continues

See List of Advertisers on page 322

"Cash gifts are intended for people to whom you do not intend to bequest the major part of your estate."

subject and artist to distinguish it from others.

The program allows you to list up to 10 bequests of specific gifts, with one alternative beneficiary for each. If you do not list an alternative beneficiary, the program will specify that the gift become part of your residuary estate should the primary beneficiary die befor you. You may specify any number of gifts to a beneficiary at one time when the program asks for this information. Multiple gifts to one person count as only one bequest in the program if you state them simultaneously.

If you indicate a bequest of shares of stock, the program will automatically print a statement directing your executor to adjust the shares to reflect any changes which affect the value of the stock (stock dividends, splits, etc.) which may have occured since the date of the will.

General (Cash) Gifts

Cash gifts are also intended for people to whom you do not intend to bequest the major part of your estate. Any cash left after your general gifts are made will be part of your residuary estate.

Include in this section the proceeds of life insurance policies, if you purchased the policies yourself, even if the beneficiaries have already been stipulated on the policies.

The program allows you to list up to ten bequests, with one alternative beneficiary for each. If you do not list an alternative, the program will specify that the gift is to become part of your residuary estate should the primary beneficiary not survive you.

Residuary Estate

This is the part of your estate which remains after all other gifts have been made. It is usually the major part of the estate and the beneficiary named is the most important to the person writing the will. The program allows you to name more than one primary beneficiary and to specify the way you want the estate to be divided among them.

Two alternative beneficiary designations may be listsed in case the primary does not survive you. These also allow for multiple names and proportioning the estate.

Taxes and Executor's Options

These final sections tend to be standard in wills. The first provides that taxes and fees related to the settlement of your will be paid from the residuary estate. The second gives your executor the power needed to administer the estate properly.

In front of three witnesses, you must de-

```
870 PRINT"OF INDIVIDUAL, SEPARATED BY A COMMA.  YOU MAY
     USE DESCRIPTIVE   PHRASES IF YOU LIKE.)":INPUT D1
    (Z)
875 PRINT:PRINT"IF THIS PERSON DOES NOT SURVIVE YOU, WH
    O DO YOU NAME AS
876 PRINT"THE NEXT BENEFICIARY?":INPUT D2(Z)
880 NEXT Z
890 CLS:PRINT:PRINT:PRINT:PRINT:PRINT"DO ANY OF YOUR GI
    FTS CONSIST OF STOCKS WHOSE VALUE
900 PRINT"MIGHT CHANGE DUE TO SPLITS, STOCK DIVIDENDS,
    ETC.?"
910 INPUT SK
920 CLS:PRINT:PRINT:PRINT:PRINT:PRINTCHR$(23)"
     CASH GIFTS":PRINT:PRINT"(YOU MAY SPECIFY UP TO 10
     GIFTS)":FOR Z=1 TO 1500:NEXT Z
1000 FOR Z=1 TO 10:GOSUB 9000:PRINT"===== GIFT #";Z;"==
     ===="
1010 PRINT:PRINT"IF CASH, SPELL OUT AMOUNT AND ENCLOSE
     DOLLAR FIGURE IN PARENTHE-SES, E.G., 'FIVE HUNDRED
      DOLLARS ($500)'.  IF INSURANCE,"
1011 PRINT"SPECIFY POLICY, E.G., 'PROCEEDS OF FEDERAL I
     NSURANCE POLICY'.":INPUT C(Z)
1015 IF C(Z)="NONE" THEN 1070
1020 PRINT:PRINT"THE RELATIONSHIP AND NAME OF THE INDIV
     IDUAL TO WHOM YOU ARE
1030 PRINT"MAKING THIS GIFT? (SEPARATE RELATIONSHIP AND
      NAME WITH A COMMA. USE DESCRIPTIVE PHRASES IF YOU
      LIKE.)":INPUT C1(Z)
1040 PRINT:PRINT"AND THE ALTERNATIVE BENEFICIARY SHOULD
      THAT PERSON DIE BEFORE
1050 INPUT "YOU"; C2(Z)
1060 NEXT Z
1070 CLS:PRINT:PRINT:PRINT:PRINT:PRINTCHR$(23)"        R
     ESIDUARY ESTATE":FOR Z=1 TO 1000:NEXT Z
1080 GOSUB 9000:PRINT"ENTER THE RELATIONSHIP(S) AND THE
      NAME(S) OF THE PERSON(S) TO
1090 PRINT"WHOM YOU WISH TO LEAVE THE REMAINDER OF YOUR
      ESTATE. SEPARATE
2000 PRINT"RELATIONSHIP(S) AND NAME(S) WITH COMMAS. USE
      DESCRIPTIVE TERMS  IF YOU LIKE.":INPUT R1
2010 PRINT:PRINT"IF MORE THAN ONE PERSON IS INVOLVED, I
     NDICATE HOW THE
2020 PRINT"ESTATE IS TO BE PROPORTIONED. USE PERCENTAGE
     S, TYPING
2030 PRINT"THEM IN THE SAME ORDER AS YOU DID THEIR NAME
     S. IF THERE
2040 PRINT"IS ONLY ONE BENEFICIARY, JUST HIT 'ENTER.'":
     INPUT P1
2050 GOSUB 9000:PRINT"WHO IS TO BE THE BENEFICIARY IN T
     HE CASE THAT THE PERSON(S)
2060 PRINT"YOU FIRST NAMED DIE BEFORE YOU? (ENTER RELAT
     IONSHIP(S)
2070 PRINT"AND NAME(S), SEPARATED BY COMMAS.)":INPUT R2
     :PRINT
2075 IF R2="NONE" THEN 3040
2080 PRINT"IF MORE THAN ONE PERSON IS INVOLVED, INDICAT
     E HOW THE
2090 PRINT"ESTATE IS TO BE PROPORTIONED. IF ONLY ONE, J
     UST HIT 'ENTER.'"
3000 INPUT P2
3010 GOSUB 9000:PRINT"AND WHO IS TO BE THE BENEFICIARY
     IN THE CASE THAT ALL OF
3020 PRINT"THE PERSONS PREVIOUSLY NAMED DIE BEFORE YOU?
      (ENTER RELATION-
3030 PRINT"SHIP(S) AND NAME(S), SEPARATED BY COMMAS.)":
     INPUT R3:PRINT:IF R3="NONE" THEN 3040
3031 PRINT"IF MORE THAN ONE PERSON IS INVOLVED, INDICAT
     E HOW THE
```

Program continues

```
3032 PRINT"ESTATE IS TO BE PROPORTIONED. IF ONLY ONE, J
     UST HIT 'ENTER'."
3033 INPUT P3
3040 CLS:PRINT:PRINT:PRINT:PRINT:PRINT"IS PRINTER ON-LI
     NE AND READY TO GO?"
3050 PRINT:PRINT"HIT ENTER WHEN READY TO PROCEED.":INPU
     T Z
3060 FOR Z=1 TO (V-12)/2:LPRINT" ";:NEXT Z:LPRINT"LAST
     WILL OF"
3062 FOR Z=1 TO INT((V-LEN(T))/2):LPRINT" ";:NEXT Z:LPR
     INT T
3070 LPRINT:A="    I, "+T+" OF "+AD+", STATE THAT THIS I
     S MY LAST WILL, REVOKING ALL PREVIOUS WILLS.":GOSU
     B 10000
3085 LPRINT" 1 . EXECUTORS:"
3090 A="    I DESIGNATE "+E1+", EXECUTOR OF THIS WILL.":
     GOSUB 10000:IF E2="NONE" THEN 3105
3091 A="    I DESIGNATE "+E2+", AS ALTERNATE EXECUTOR, I
     N CASE "+E1+", IS UNABLE OR UNWILLING TO ACT AS EX
     ECUTOR, OR CEASES TO DO SO."
3095 GOSUB 10000
3100 IF E3="NONE" THEN 3105
3101 A="    IN THE CASE THAT BOTH OF THESE PERSONS ARE U
     NABLE OR UNWILLING TO ACT AS EXECUTOR, OR CEASE TO
      DO SO, I DESIGNATE "+E3+", AS EXECUTOR OF MY WILL
     ."
3102 GOSUB 10000
3105 IF EB="NO" THEN 3110
3106 A="    I REQUEST THAT THE EXECUTOR BE BONDED.":GOSU
     B 10000:GOTO 3114
3110 A="    I DO NOT REQUIRE THAT THE EXECUTOR BE BONDED
     .":GOSUB 10000
3114 IF Y=0 THEN 3187
3115 LPRINT" 2 . GUARDIANS: "
3130 IF SP="NONE" THEN 3145
3135 A="    IF MY SPOUSE "+SP+", DIES BEFORE ME OR AT TH
     E SAME TIME, I DESIGNATE "+G1+", GUARDIAN OF THE P
     ERSON AND PROPERTY OF MY CHILDREN UNTIL THEY ARE L
     EGALLY OF AGE.":GOSUB 10000
3140 GOTO 3150
3145 A="    I DESIGNATE "+G1+", AS GUARDIAN OF THE PERSO
     N AND PROPERTY OF MY CHILDREN UNTIL THEY ARE LEGAL
     LY OF AGE.":GOSUB 10000
3150 IF G2="NONE" THEN 3173
3156 A="    I DESIGNATE "+G2+", AS ALTERNATE GUARDIAN, I
     N CASE "+G1+", IS UNABLE OR UNWILLING TO ACT AS GU
     ARDIAN, OR CEASES TO DO SO.":GOSUB 10000
3160 IF G3="NONE" THEN 3173
3171 A="    ALTERNATIVELY, IF NEITHER OF THESE PERSONS S
     URVIVES LONG ENOUGH TO CARRY OUT THIS RESPONSIBILI
     TY, I DESIGNATE "+G3+", AS GUARDIAN OF MY CHILDREN
     ."
3172 GOSUB 10000
3173 IF GB="NO" THEN 3180
3175 A="    I REQUEST THAT THE GUARDIAN BE BONDED.":GOSU
     B 10000:GOTO 3187
3180 A="    I DO NOT REQUIRE THAT THE GUARDIAN BE BONDED
     .":GOSUB 10000
3187 IF D(1)="NONE" THEN 3221
3188 IF Y>0 THEN 3189 ELSE LPRINT " 2. ";:GOTO 3190
3189 LPRINT " 3. ";
3190 LPRINT"SPECIFIC GIFTS: "
3195 FOR Z=1 TO 10
3200 IF Z>1 AND D(Z)="NONE" THEN 3221
3205 A="    I LEAVE MY "+D(Z)+" TO "+D1(Z)+".":GOSUB 100
     00
3210 IF D2(Z)<>"NONE" THEN 3215
3211 A="    IF THAT PERSON DIES BEFORE ME OR AT THE SAME
```

Program continues

```
     TIME, THIS GIFT IS TO BECOME PART OF MY RESIDUARY
     ESTATE.":GOSUB 10000:GOTO 3220
3215 A="   IF THAT PERSON DIES BEFORE ME OR AT THE SAME
     TIME, I LEAVE MY "+D(Z)+" TO "+D2(Z)+".":GOSUB 10
     000
3220 NEXT Z
3221 IF C(1)="NONE" THEN 3252
3222 X=2
3223 IF Y>0 THEN X=X+1
3224 IF D(1)<>"NONE" THEN X=X+1
3225 LPRINTX; ". GENERAL GIFTS: "
3230 FOR Z=1 TO 10
3235 IF Z>1 AND C(Z)="NONE" THEN 3252
3240 A="   I GIVE "+C(Z)+" TO "+C1(Z)+".":GOSUB 10000
3245 IF C2(Z)<>"NONE" THEN 3250
3246 A= "   IF THAT PERSON DIES BEFORE ME OR AT THE SAM
     E TIME, THIS GIFT IS TO BECOME PART OF MY RESIDUAR
     Y ESTATE.":GOSUB 10000:GOTO 3251
3250 A="   IF THAT PERSON DIES BEFORE ME OR AT THE SAME
     TIME, I GIVE THAT AMOUNT TO "+C2(Z)+".":GOSUB 100
     00
3251 NEXT Z
3252 X=2:IF Y>0 THEN X=X+1
3253 IF D(1)<>"NONE" THEN X=X+1
3254 IF C(1)<>"NONE" THEN X=X+1
3260 LPRINTX;". RESIDUARY ESTATE:
3265 IF P1<>"" THEN 3270 ELSE A="   I WILL THE REMAINDE
     R OF MY PROPERTY TO "+R1+".":GOSUB 10000:GOTO 3279
3270 A="   I WILL THE REMAINDER OF MY PROPERTY TO "+R1+
     ", TO BE APPORTIONED RESPECTIVELY AS FOLLOWS: "+P1
     +".":GOSUB 10000
3279 IF R2="NONE" THEN 3320
3280 IF P2<>"" THEN 3295 ELSE A="   IF THAT PERSON (THO
     SE PERSONS) DIE BEFORE ME OR AT THE SAME TIME, I W
     ILL THIS PROPERTY TO "+R2+".":GOSUB 10000:GOTO 330
     4
3295 A="   IF THAT PERSON (THOSE PERSONS) DIE BEFORE ME
     OR AT THE SAME TIME, I WILL THIS PROPERTY TO "+R2
     +", TO BE APPORTIONED RESPECTIVELY AS FOLLOWS: "+P
     2+".":GOSUB 10000
3304 IF R3="NONE" THEN 3320
3305 IF P3<>"" THEN 3315 ELSE A="   IF THE PRECEDING PE
     RSONS DIE BEFORE ME OR AT THE SAME TIME, I WILL TH
     IS PROPERTY TO "+R3+".":GOSUB 10000:GOTO 3320
3315 A="   IF THE PRECEDING PERSONS DIE BEFORE ME OR AT
     THE SAME TIME, I WILL THIS PROPERTY TO "+R3+", TO
     BE APPORTIONED RESPECTIVELY AS FOLLOWS: "+P3+".":
     GOSUB 10000
3320 LPRINT X+1;
3325 LPRINT". TAXES:
3330 A="   ALL TAXES AND GOVERNMENT FEES RELATED TO THE
     SETTLEMENT OF MY ESTATE ARE TO BE PAID OUT OF THE
     RESIDUARY ESTATE.":GOSUB 10000
3339 LPRINTX+2;
3340 LPRINT". EXECUTOR'S OPTIONS:
3345 A="   IN ORDER TO CARRY OUT THE SETTLEMENT OF MY E
     STATE, I CONFER UPON MY EXECUTOR FULL POWER TO SEL
     L, LEASE, MORTGAGE, OR OTHERWISE LIQUIDATE THE ASS
     ETS IN MY ESTATE.":GOSUB 10000
3360 IF SK="YES" THEN 3365 ELSE 3391
3365 A="   MY EXECUTOR SHOULD ADJUST THE NUMBER OF SHAR
     ES OF SECURITIES I LEAVE AS SPECIFIC GIFTS TO ACCO
     UNT FOR ALL STOCK DIVIDENDS, SPLITS, OR OTHER CH
     ANGES IN THE FORM OF THE STOCK OCCURRING AFTER THE
     DATE OF THIS WILL."
3370 GOSUB 10000
3391 IF V<>80 THEN 3400
3392 LPRINT:LPRINT:GOSUB 3500:LPRINT"SIGNED:"
3393 GOSUB 3500:LPRINT"        -------------------------
     --------"
3394 LPRINT:LPRINT:GOSUB 3500:LPRINT"DATE:"
3395 GOSUB 3500:LPRINT"         -----------------------
     --------"
3397 GOTO 3410
3400 LPRINT:LPRINT"SIGNED:"
3401 LPRINT"          ------------------------"
```

Program continues

"They should know you, be established in the community, and be young enough to survive you."

```
3402 LPRINT:LPRINT"DATE:"
3403 LPRINT"      --------------------------"
3410 LPRINT:LPRINT"WITNESSES:
3420 A="   ACCORDING TO "+T+"'S WISH, WE ALL ASSEMBLED
     ON THE DATE SHOWN ABOVE TO WITNESS THE SIGNING OF
     THIS WILL.":GOSUB 10000
3421 IF SX="M" THEN A="   WITH ALL OF US THERE AT THE S
     AME TIME, "+T+" SIGNED IT AND DESCRIBED IT AS HIS
     LAST WILL."
3422 IF SX="F" THEN A="   WITH ALL OF US THERE AT THE S
     AME TIME, "+T+" SIGNED IT AND DESCRIBED IT AS HER
     LAST WILL."
3425 GOSUB 10000
3440 FOR Z=1 TO 3:LPRINT:IF V<>80 THEN 3450
3441 LPRINT"SIGNED:                                  ADD
     RESS:"
3442 LPRINT"       --------------------------------
     -------------------------------"

3443 LPRINT:LPRINT:LPRINT"
                    ------------------------------
     -"
3445 GOTO 3460
3450 LPRINT"SIGNED:"
3451 LPRINT"       ------------------------"
3452 LPRINT:LPRINT"ADDRESS:"
```

Program continues

clare that this is your last (i.e., most recent) will, and sign and date the will in their presence. The witnesses do not need to read the will, but they should know you, be established in the community, and be young enough to survive you. None of the witnesses should be mentioned in your will as beneficiaries. Have the witnesses sign the will and record their addresses in the presence of yourself and each other. This will allow them to bear witness to each other's signature, if necessary. If their handwriting is illegible, print their names and addresses directly underneath the handwritten information. Although the program permits you to print as many copies as you desire, sign only one and mark the others COPY.

If your will is printed on more than one page, number each page in the top right hand corner and indicate how many pages the will runs, e.g., "page one of four pages." Initail each page, and have each witness initial each page, next to the page number notation. Be sure that the last page shows part of the test of the will—not just the required signatures.

"After completing your new will, destroy all previous wills. . . don't make any hand written corrections. . . ."

Storage of Will

Put the will in a place safe from loss, theft or fire which will be accessible after your death. Do not use your own safe deposit box unless you are certain your state allows someone else to enter it immediately after your death.

Give your executor the following information in writing:

- Location of your will and how to gain access to it
- Burial and funeral instructions
- Description of unusual situations in your will, if any, and explanation of the reasons behind them
- Your mother's maiden name and your father's name

After completing you new will, destroy all previous wills and their copies. Don't make any handwritten corrections to the will. The advantage of having this program is that you don't have to do that—if there is something you wish to change, just run it through again, and have the new version witnessed and signed. ■

```
3453 LPRINT"        -----------------------"
3454 LPRINT:LPRINT"         -----------------------"
3460 NEXT Z
3465 CLS:PRINT:PRINT:PRINT"DO YOU WISH TO PRINT AN ADDI
     TIONAL COPY OF THIS WILL?":PRINT"IF SO, ADVANCE PA
     PER AND ENTER 'Y'. IF NOT, ENTER 'N'."
3470 INPUT A:IF LEFT$(A,1)="Y" THEN 3060
3480 END
3500 FOR Z=1 TO 40:LPRINT" ";:NEXT Z:RETURN
8999 END
9000 CLS:PRINT:PRINT"    *** ENCLOSE INFORMATION IN QUOT
     ATION MARKS      ***"
9010 PRINT"    *** ENTER 'NONE' IF ANSWER IS NONE OR NO
     OTHERS ***"
9020 PRINT:PRINT:RETURN
9025 END
10000 U1=LEN(A):U4=U1:U1=INT(U1/V)+2:A=A+" "
10010 FOR U2=V-1 TO V/2 STEP -1
10020 B=MID$(A,U2,1):IF B=" " THEN 10030 ELSE NEXT
10030 U3=U3+1:B1(U3)=LEFT$(A,U2):IF U3=U1 THEN 10050
10035 IF U4<=U2+1 THEN 10050
10040 A=RIGHT$(A,U4-U2+1)
10045 U4=U4-U2:GOTO 10010
10050 FOR U5=1 TO U3:IF B1(U5)="" THEN 10060 ELSE LPRIN
     T B1(U5):NEXT
10060 U3=0:FOR U5=1 TO U1:B1(U5)="":NEXT:RETURN
10070 END
```


✓See List of Advertisers on page 322

Some ham software.

Morse Resource Part I

Allan S. Joffe W3KBM
1005 Twining Road
Dresher, PA 19025

As a ham radio operator, I often use my TRS-80 to help me transmit Morse code. The programs presented may be readily changed to suit your special needs. Portions of one program may also be combined with ideas from another to produce software for specialized applications.

Morse Code

The Morse code consists of groupings of signals of two basic durations. A dot is the basic interval of time and a dash is three times the length of a dot. Transmitted in groups, these time intervals make up the alphabet, numerals and punctuation.

We use two variables to transmit Morse code: a speed variable, determining how fast we send our code, and a vocabulary variable, determining how many characters we need to support. The speed variable is open. I prefer to stay at a transmitted speed of about 15 words per minute (wpm). If I drop below this speed, I still transmit each character at the same rate and the wpm count drops by the amount of time left between words. This makes for uniform ear contact at any speed.

The computer will control the relay (your transmitter or code oscillator). The examples show the use of the cassette control relay as output port control. For extended use you should build or buy a control interface and save your cassette relay for its intended use.

Common Threads

These programs make liberal use of string functions. ASCII value representations are used for alphanumeric characters, punctuation and a few control functions.

A dot is represented by an S (short unit time). A dash is represented by an L (long unit time). The interval between characters is represented by an E.

Thus, the character B would look like "LSSSE" and the question mark would look like "SSLLSSE." Every character is a combination of S (dot) and L (dash) plus a final E as the spacing interval generator. They include the open and close quote marks because they are string representations.

We will also use the INKEY$, a buffer concept, and various common functions of the Basic interpreter.

Character	Program Form	ASCII Value
A	SLE	65
B	LSSSE	66
C	LSLSE	67
D	LSSE	68
E	SE	69
F	SSLSE	70
G	LLSE	71
H	SSSSE	72
I	SSE	73
J	SLLLE	74
K	LSLE	75
L	SLSSE	76
M	LLE	77
N	LSE	78
O	LLLE	79
P	SLLSE	80
Q	LLSLE	81
R	SLSE	82
S	SSSE	83
T	LE	84
U	SSLE	85
V	SSSLE	86
W	SLLE	87
X	LSSLE	88
Y	LSLLE	89
Z	LLSSE	90
0	LLLLLE	48
1	SLLLLE	49
2	SSLLLE	50
3	SSSLLE	51
4	SSSSLE	52
5	SSSSSE	53
6	LSSSSE	54
7	LLSSSE	55
8	LLLSSE	56
9	LLLLSE	57
?	SSLLSSE	63
,	LLSSLLE	44
.	SLSLSLE	46
/	LSSLSE	47

Morse Character Directory Chart

Line Analysis

Line 5 clears enough string space to accommodate the program. Line 10 allows you to input directions and A$, which becomes your message string as you type. Lines 15 and 20 are program mechanics which, along with line 25, allow the CODE$ to be processed.

Line 30 works as follows: There are 40 If. . .Then statements which represent the 26 letters of the alphabet, 10 numerals and four punctuation marks. Line 30 separates these into two groups. The first group is the alphabet, as all ASCII numerical representations of the alphabet are greater than 64. The second group is the numeral and punctuation group, all of whose ASCII values are less than 64. This allows a two-tier search of the If. . .Then statement list which gives the program a bit more running speed.

Line 35 starts the character directory, with 26 lines that represent all the letters of the alphabet. Line 300 is the start of the completion of the character list. It contains numerals zero through nine and the four punctuation marks supported by the program. Lines 1000 through 1040 perform the dissection of CODE$ so that each element of CODE$ can be operative via the group of subroutines that follow the end statement in line 1050.

Line 5000 starts out with TM = 0. This statement is needed as a remedy for the following program anomaly. If the very first character that you type into A$ has an S as the first element in CODE$, then the S becomes elongated in time so that it sounds

"The Morse Code consists of groupings of signals. . .transmitted in groups, these time intervals make up the alphabet, numerals and punctuation."

like a dash instead of a dot. TM = 0 takes care of this problem. The next statement in this line, OUT 255,4, turns the cassette relay on. We then have delay loop TM which holds it on for the desired length of our dot. OUT 255,0, opens the cassette relay which completes the dot duration. Then we have the required return which allows the next element in the particular CODE$ being dissected to be processed.

Line 6000 is the subroutine that determines the length of our dash and except that it does not need the TM = 0 statement, operates in a fashion similar to line 5000. Line 7000 processes the length of time between characters. It is a simple delay loop with no relay operation function.

Limitations

Two limitations in this program are: When the Basic interpreter hits a comma in a string, it displays the error message, Extras Ignored. When you type your message, type a quotation mark as the first character. This will eliminate the error message.

The TRS-80 string length limit is 255 characters. If you try to input more characters the computer refuses to accept them. A concatenated string will violate the character limit.

This is not a big hassle since CW transmissions on the ham bands abound in abbreviations.

A Program Bonus

It would be nice to be able to enter one special character which would send a message commonly used, such as a general call to see if someone wants to talk. In ham Morse terms it might look like this: CQ CQ CQ DE W3KBM W3KBM CQ CQ CQ CQ DE W3KBM W3KBM W3KBM K, which tells the listening world that W3KBM is looking for someone to talk to.

The K at the end of the message says that W3KBM is through with his call for a contact and is now tuning across the band for a response.

We can do this in our program by inserting line 12 into Program Listing 1.

```
12 IF A$ = "#"GOSUB 8000
```

Now insert the subroutine in line 8000.

```
8000 A$ = "CQ CQ CQ DE W3KKBM
W3KBM W3KBM K":
Return
```

Now when you type in # when the program asks you to type your message, it will send the general call message. You are limited only by the number of symbols on the keyboard that are not used as part of the character directory.

Speeding It Up

Program Listing 2 has the same limitations and advantages as Listing 1. But its approach is different, not using the large list of If. . .Then statements. This makes the program run faster.

In addition to a Clear statement, we also use a DIM statement when the code character directory is entered. In every case the subscript is the ASCII value of the character (see Table 1).

The next program is different from the first two. It uses the INKEY$ function to transmit a character as soon as it is entered. This program handles the comma problem without any special treatment and removes the string limitation of 255 characters.

It is an excellent program for teaching Morse code. If you key a code practice oscillator, you see and hear what you send in real time.

Program 3

INKEY$ continuously scans the character list. When a key is pressed, that character is printed on the screen and passed on to the program lines starting with line 830. The character directory is entered using a series of If. . .Then statements. Line 20 is used if you want to erase the screen; merely Enter # and the screen will be clean except for the heading Type Message. A small portable radio placed near the computer while the program is running will let you listen to the character list being scanned.

It takes a certain rhythm on the keyboard to keep a continuous flow of code going. Enter any character and when you hear it starting to come out, hit the next key. You do not have to wait for the current character to be completed as the program acts like an automatic key with self-completing characters. With some typing skill and practice you can send very nice code. You have full control of between-word spacing, while the program does the rest.■

```
5  CLEAR 300
10 CLS: INPUT"ENTER MESSAGE";A$
15 J = LEN(A$)
20 X1 = X1 + 1
25 CODE$ = MID$(A$,X1,1)
30 IF ASC(CODE$)<64 GOTO 255
32 REM THE NEXT SERIES OF 26 LINES ENTERS THE ALPHABET, ONLY A
   FEW OF THE LINES WILL BE SHOWN AS THEY FOLLOW A DEFINITE PATTERN.
35 IF CODE$ = "A" THEN CODE$ = "SLE"
40 IF CODE$ = "B" THEN CODE$ = "LSSSE"
45 IF CODE$ = "C" THEN CODE$ = "LSLSE"
50 IF CODE$ = "D" THEN CODE$ = "LSSE"
55 IF CODE$ = "E" THEN CODE$ = "SE"
 .
 .
 .     After the 26 letters of the alphabet are so entered, we enter numerals and punctuation starting
 .     with line 255. . . . . . . .
 .
 .
 .
255 IF CODE$ = "0" THEN CODE$ = "LLLLLE"
260 IF CODE$ = "1" THEN CODE$"SLLLLE"
265 IF CODE$ = "2" THEN CODE$ = "SSLLLE"
 .
 .
 . we complete the numerals and then insert our four punctuation marks starting with line 300
300 IF CODE$ = "." THEN CODE$ = "SLSLSLE"
305 IF CODE$ = "," THEN CODE$ = "LLSSLLE"
310 IF CODE$ = "?" THEN CODE$ = "SSLLSSE"
315 IF CODE$ = "/" THEN CODE$ = "LSSLSE"
320 REM Now we come to the part of the program that turns our entries into a form that allows control of the
    cassette relay so that it may be used to key a code oscillator if you so desire. Remember the caution about
    using this relay for controlling loads in excess of its normal load, namely, the cassette motor!
1000 LN = LEN(CODE$)
1010 FOR X = 1 TO LN
1020 IF MID$(CODE$,X,1) = "S" GOSUB 5000 ELSE IF MID$(CODE$,X,1) = "L" GOSUB 6000 ELSE IF MID$
     (CODE$,X,1) = "E" GOSUB 7000
1030 NEXT X
1035 IF X1 = J THEN X 1 = 0: GOTO 10
1040 IF X1 < J THEN GOTO 20
1050 END

5000 TM = 0: OUT 255,4:FOR TM = 1 TO 20:NEXT TM: OUT 255,0:RETURN
6000 OUT 255,4:FOR TM = 1 TO 40:NEXT TM: OUT 255,0:RETURN
7000 FOR TM = 1 TO 20:NEXT TM:RETURN
```

Program Listing 1.

```
10   CLEAR 300:CLS
15   DIM A$(100)  :REM CHARACTER TABLE FOLLOWS
20   A$(65)="SLE"
25   A$(66)="LSSSE"
30   A$(67)="LSLSE"
40   A$(68)="LSSE"
 .
  .       The entire character directory is entered in this
   .      fashion.  Just follow the pattern as shown
    .
     .
      .
500   INPUT "TYPE IN MESSAGE";A$
510   LN = LEN(A$)
520   FOR J = 1 TO LN
530   Z$ = MID$(A$,J,1)
540   X = ASC(Z$)
550   FOR R = 1 TO LEN(A$(X))
660   IF MID$(A$(X),R,1) = "S" GOSUB 2000
670   IF MID$(A$(X),R,1) = "L" GOSUB 3000
680   IF MID$(A$(X),R,1) = "E" GOSUB 4000
690   NEXT R,J
700   GOTO 500
710   END

2000 TM=Ø:OUT 255,4: FOR TM = 1 TO 20:NEXT TM:OUT 255,Ø:RETURN
3000 OUT 255,4:FOR TM = 1 TO 40:NEXT TM:OUT 255,Ø:RETURN
4000 FOR TM = 1 TO 20:NEXT TM:RETURN
```

Program Listing 2.

```
5  CLS:PRINT "TYPE MESSAGE"
10 A$ = INKEY$:PRINT A$;
20 IF A$ = "#" GOTO 5
30 IF A$ = "A" THEN A$="SLE"
35 IF A$ = "B" THEN A$="LSSSE"
40 IF A$ = "C" THEN A$="LSLSE"
45 IF A$ = "D" THEN A$="LSSE"
50 IF A$ = "E" THEN A$="SE"
60 .
    .
     .
      .
       .
        .    Enter rest of character directory following the
             pattern shown.

830 LN = LEN(A$)
850 FOR X = 1 TO LN
870 CODE$ = MID$(A$,X,1)
890 IF CODE$ = "S" GOSUB 1010
910 IF CODE$ = "L" GOSUB 1030
930 IF CODE$ = "E" GOSUB 1050
950 NEXT X
970 GOTO 10
990 END

1010 OUT 255,4: FOR TM = 1 TO 20:NEXT TM:OUT 255,0:RETURN
1030 OUT 255,4: FOR TM = 1 TO 40:NEXT TM:OUT 255,0:RETURN
1050 FOR TM = 1 TO 20:NEXT TM: RETURN
```

Program Listing 3.

Plug-in, Turn-on, Connect!

The Microconnection

The Microconnection
The Micro Peripheral Corp.
Mercer Island, WA
$250

Fred Blechman
7217 Bernadine Ave.
Canoga Park, CA 91307

If you have a Radio Shack TRS-80 Model I, II or III and don't have a modem yet, you don't know what you're missing!

The term "modem" is a contraction of "modulate-demodulate", which is electronic jargon for (loosely defined) talking and listening. Unlike conventional radio transmissions, however, modem signals are not voice or music signals impressed on carrier frequency. Modems communicate over the telephone lines, using frequency-shifting to transmit and receive binary digits (bits) in the standardized ASCII code.

Various transmission standards exist, but the so-called Bell System 103 standard is the most generally used, with sending frequencies of 1070 Hz and 1270 Hz, and receiving frequencies of 2025 Hz and 2225 Hz. The transmission rate is expressed in baud (which can be thought of as bits-per-second). The most common rate is 300 baud, which translates to about 30 characters per second for serial mode.

Modems are the circuitry that codes and decodes the signals for use with the phone line. You also need a source of power for the modem, a software program to allow the computer to communicate in ASCII, and an interface from computer-to-modem and modem-to-phone-line. If you use Radio Shack components, you would need the following: 26-1171 Telephone Interface II Acoustic Modem, 26-1146 Communications Software, 26-1145 RS-232C Interface Board and the 26-1140 Expansion Interface. This amounts to about $600!

The Microconnection

When I became aware of this bits-and-pieces approach to adding modem capability to a TRS-80, I decide to wait until some enterprising people would find a way to put this all into one package. Well, Radio Shack has done it with their 26-5000 TRS-80 Videotex Terminal which costs about $400. The Videotex Terminal includes keyboard, modem, built-in software and 4K of memory. It attaches to the antenna terminals of a color or black and white television set, and plugs into the phone line. However, it is not a computer.

Micro Peripheral's Microconnection

Another approach, far less expensive and more versatile if you already have a Model I, II or III TRS-80, is the Microconnection, from Microperipheral Corp. This modem is about $250.

It plugs directly into the TRS-80 (even a Level I 4K!), the phone line and the 117 VAC line, and converts your TRS-80 into a terminal capable of accessing the expanding world of telecommunications. An RS-232 output socket built into the Microconnection allows you to run your serial printer for permanent copy of your communications—both incoming and outgoing. It's like having your own teletypewriter!

The Microconnection is *directly* coupled to the phone line, rather than *acoustically,* so you have silent transmissions with increased line sensitivity and reduction in transmission errors. A big plus is that the Model I Microconnection plugs directly onto the card-edge at the back of the keyboard unit, so you don't need an expansion interface or RS-232C serial interface board! Six screw-terminals on the back of the Microconnection allow you to couple the modem to ham radio equipment, a tape recorder for ASCII communication or a taped ASCII recording.

Making the Connection

I got my Microconnection on a Friday, and spent part of the weekend reading the

"Seeing the screen "talking" to me from a computer at the other end of a long-distance phone line was amazing."

manual and trying to work up the courage to hook it up. Although there are no modifications required to the TRS-80, I was hesitant about connecting to the phone line and the keyboard unit. Although the Microconnection is FCC approved, and comes with a standard modular plug, I could envision my confusion causing phone problems. And confused I was! Although the documentation was good, and even had photos to show how the unit is installed, it didn't really explain what was happening. No schematic or theory of operation is included, and I had learned to be careful about mistreating my '80. I was uncomfortable about connecting things without having any idea what I was doing.

In addition to my concern was that I already had two other peripherals attached to the TRS-80 keyboard slot—an Exatron Stringy Floppy tape unit and a Radio Shack Printer Interface Cable. How many things could I hang on the keyboard unit?

"The Microconnection is directly coupled to the phone line, rather than acoustically, so you have silent transmissions with increased line sensitivity."

After a couple of hours of reading the documentation, I took a deep breath and went to it.

I decided to remove the printer cable and Stringy Floppy from the keyboard, just to be sure that nothing would affect operation of the Microconnection. I turned the computer off, and pushed the Microconnection 40-pin card edge connector (at the end of a 12-inch ribbon cable) onto the card edge at the rear of the keyboard unit. (If I had an expansion interface, I would have used the screen printer card edge.) No problem.

Next, I plugged in the transformer that drops 117 VAC to 12 VAC as input to the Microconnection's built-in power supply. A red LED on the top of the Microconnection lights to tell you that the power is on. Remember to plug in the transformer before the software is loaded or the program will either not work or crash to Memory Size?

Connecting to the phone line was easy. The Microconnection comes with a modular phone plug at the end of a six-foot telephone cord. All I needed was a modular duplex adapter (Radio Shack 279-357 or equivalent). This allowed me to connect both the Microconnection and the telephone to the wall modular phone jack. If you have the older four-prong telephone plugs and jacks, various outlets sell adapters. There are two pushbuttons on the front of the Microconnection. Make sure the left one is in the voice position or your phone line will be affected.

The Software

After everything is connected, turn on the computer and load the cassette software terminal program, S80. I encountered a problem at this point and after several unsuccessful tries, I finally made a good load (and a backup copy) using a marvelous utility tape called Nuload by Marigold Associates. The screen identified the program and asked, after a short delay, for an input of F for full duplex or H for half duplex. I typed F and the screen went blank, with only the cursor in the upper left hand corner.

Following the documentation, I tested the Microconnection and software by disconnecting the duplex plug from the wall jack. The phone and Microconnection were connected together, but not to the phone line. Next, I put the right hand duplex/simplex switch in the simplex mode and put the voice/data switch in the data position. The carrier LED flashed on and then off. As I pressed the keyboard keys, the letters or numbers appeared on the screen and the carrier LED lit up. Eureka! It worked!

Using the Microconnection

Now what? I couldn't try to get on the national Compuserve network, since my membership had not come through yet. I checked a list of 118 public bulletin boards provided by Microperipheral. Since I had a TRS-80 I looked for Forum-80, Info-80 or Comm-80 bulletin boards, since no one had told me I could contact any bulletin board—even Apple, PET or Heath! I plugged the modular duplex adapter back into the phone jack, put the voice/data switch on voice and the duplex/simplex switch in duplex (for two-way modem communication). Much as I hated to make a few long-distance calls to check out the operation of the Microconnection, I didn't know any better—so I made three of them—Seattle, Fullerton (CA) and St. Louis.

Seeing the screen "talking" to me from a computer at the other end of a long-distance phone line was amazing. ■

✓See List of Advertisers on page 322

More on the use of a macro and manufacturer-independent applications programs.

A Macro Processor For Basic—Part II

J. Alan Olmstead
J. Olmstead Financial Engineering Systems
3843 West Street, Moritz Lane
Phoenix, AZ 85023

Last month's article described a macroprocessor which, in a sense, makes the computer self-programming, and accordingly, more cost-effective. Those using the JOFES macroprocessor have discovered that creating 1,000 lines of error-free Basic programming per day is normal.

The macroprocessor is only an incidental part of a much larger approach to computer optimization. Without the logical function libraries drawn upon by means of macro calls, you can do nothing with the processor. The libraries provide the real value because they deliver the final product.

Programmers should be able to ignore the structure of Basic as it has been defined by any manufacturer, and use macro library modules to perform *logical* programming functions which remain the same on computers of all makes and models. Whenever a new computer model is selected, macro libraries are expanded to include code specific to the native (and, therefore, most efficient) mode of that particular computer. When high-level application programs are respecialized through these expanded libraries, they come out fully rewritten and running perfectly the first time.

Logical Programming

The concept of logical programming originates in the logical/physical input and output control utilities provided by manufacturers of large-scale computers as programming aids to their customers. The logical program module is used to define the concept of the action desired, and the physical program module carries out that idea through specific commands for that particular model.

Proceeding from the logical/physical division of labor, one arrives at a concept of programming which avoids any form of Basic command which is peculiar to any computer manufacturer or individual model within a manufacturer's line. For example, the Radio Shack Model I command for a substring search is: A = INSTR(A$,"SAM"). The value "SAM" is compared against each incremental three-byte substring of A$, beginning with position one and ending at LEN(A$) – 2. If "SAM" is found anywhere in A$, the variable A contains the position number in A$ corresponding to the S in "SAM".

This function does not exist at all in the Northstar Basic, for example, and it exists only in an altered form in other makes and models. If an application program written for the TRS-80 Model I contained the INSTR function at a physical level, that is, actually using the INSTR command peculiar to Radio Shack, that program could not be recast for running on other makes and models of computers, unless it were manually reprogrammed.

To make the applications program conversion free, the substring search function should be defined as a logical function, to be performed by a call to a macro library module:

```
1000 'MACRO = SERCH X,A,A$,B$,TRS801
1010 'ENDCALL
```

Interpretation of this macro call reads: "Beginning at the position number in A, search the string A$ for the first incidence of the substring B$. If found, place the beginning location number in X, otherwise set X to zero and return to the user. Generate object code for the TRS-80 Model I."

Because the code applies only to the macroprocessor, which is computer-independent, this application coding will not change in future uses except for the substitution of a different object computer name.

The logical function library module would, by further example, provide alternate forms, depending upon what object computers had been included in the original design and subsequent extensions.

```
1 'MACRO = SERCH
2 'BOOL ( (&&05) = (TRS801) )
3 &&01 = INSTR(&&02,&&03,&&04)
```

ABS	GOTO	STR$
ASC	IF...THEN	VAL
CHR$	INT	Integer/real numerics to 99,999,999.99
DATA	LEN	
DIM	MID$	String variables to 256 bytes
END	READ	
FOR...NEXT...STEP	REM and	Variable names to six positions including
GOSUB-RETURN	RESTORE	A-Z, 0-9, $@#%&!

Math and relational operators are the same as in interpreter Basic; all other commands not listed here are called from logical function libraries.

Table 1. List of Recommended Basic Command Words

> *"The macroprocessor is only an incidental part of a much larger approach to computer optimization."*

```
 4 'ENDBOOL
 5 'BOOL ( )&&05) = (IBM5110) )
 6 &&01 = IDX(STR(&&03,&&02,LEN(&&03) - &&02 + 1),
   &&04)
 7 IF &&01 = 0 GOTO 10
 8 &&01 = &&01 + &&02 - 1
 9 'ENDBOOL
10 RETURN
11 'ENDMAC
```

In the example, the macro module "SERCH" will provide accurate code for two quite different computers, because the 'BOOL tests exclude coding based upon the object computer model name. The applications programs which call this macro module are never affected by a change in the name of the object computer.

Avoiding Errors

In addition to the INSTR function, there are several inherently locked-in features in TRS-80 Model I code. The most troublesome is the If...Then...Else compound command, with the conflicting interpretation of an implied Else where it is not expressly coded. Furthermore, the Else statement is illegal in many computer Basics; the balance of the line must be dealt with in the event of a Then test failure; and the Then operand is optionally treated as either a redundant GOTO or a dependent command clause. If compilers and translators fail, it surely will be in handling the If...Then...Else statement.

This can be solved by never using the Else clause, but rather, defining an "else condition" as a drop-through to the next line. It is also recommended that the object of the Then clause be only a line number, making Then redundant for GOTO.

The Onerror/Resume pair should definitely be programmed from macro library modules, since error trapping is handled uniquely in nearly every computer model.

The STRING$ function similarly does not exist in most Basics, even though it is a handy command.

The LEFT$ and RIGHT$ functions are Microsoft variations of the substring function omitted from other systems. The MID$ command always has a counterpart, but the method of execution can be quite different.

For example, the Microsoft command "MID$(A$,10,5)" means "five positions beginning from position number ten." The IBM 5110/5120 counterpart, "STR(A$,10,5)", means exactly the same thing. But the Northstar equivalent is "A$(10,14)", indicating that the second argument is the name of the ending position rather than the length of the substring measured from its beginning position.

Safe programming procedure here uses a simplified programming technique and depends upon the object computer translation utilities provided with the macroprocessor. Always use the MID$ command rather than LEFT$ and RIGHT$, and use fully resolved arguments, not computed arguments, for the beginning point and length. Thus, even without the macroprocessor's translator, a simple ten-line Basic program can automatically convert every MID$ command in the program in no more than a minute or so. The following illustrates:

Undesirable Form	Desirable Form
A$ = MID$(B$,INT(X/Y), LEN(X$))	W = X / Y Z = INT(W) L = LEN(X$) A$ = MID$(B$,Z,L)

Many interpreters, and especially compilers, cannot function with consistent accuracy when the input source code is much more complex than these syntactical constructions.

Though it is permissible to omit the second (length) variable in Radio Shack Basic, do not do so. Many other computer Basics cannot handle it at all, and some try with intermittent results.

Finally, do not use a string variable as its own working space when shortening it to a

SEQOPN	Open a sequential file
SEQRD	Read a sequential file
SEQWR	Write a sequntial file
SEQRWD	Rewind a sequential file to beginning
SEQPOS	Position a sequential file to logical record "n"
SEQCLS	Close a sequential file
ISQOPN	Open an indexed sequential file
ISQRD	Read an indexed sequential file record by key
ISQWR	Write an indexed sequential file record by key
ISQCLS	Close an indexed sequential file
RNDOPN	Open a random access file
RNDRD	Read a random access file record
RNDWR	Write a random access file record
RNDCLS	Close a random access file

Table 2. Logical Functions in the Disk Library

✓See List of Advertisers on page 322

"Keep the specification of entry and exit conditions at the most general level."

portion of its original content. For example, A$ = MID$(A$,10,5): The substring is first placed into a working location created temporarily, then the newly formed short string is returned to the variable A$. IBM Basic in particular executes a veritable slaughter job on this command format, apparently attempting to use the same physical locations of memory as sending, work and receiving fields all in one. Designate general purpose working variables in numeric and string form—for example, the X and X$ combinations. The command sequence X$ = MID$(A$,10,5):A$ = X$ never fails, and X$ is always available for transitional use.

Numeric Constants

Providing four different internal notation methods (integer, real, single and double-precision) for computers used principally in business application environments is simply absurd. Any compiler with the common sense of an abacus will provide simple 10-digit BCD (Binary Coded Decimal) notation, with the decimal point initialized between the second and third significant digits (99,999,999.99). The decimal point will float freely at run time, and a SETDEC command (set decimal point) could be provided to position the decimal at a desired rounding position. Ten-digit BCD takes up only five positions of memory (99 99 99 99 99H) and the Z-80's DAA (Decimal Adjusted Arithmetic) hardware command makes math operations simple.

Wherever possible, Basic programs should be written using only integer and real number types. Upon conversion for another computer, these types will adapt into 10-digit BCD notation with ease and accuracy.

As seen in Table 1, business or commercially oriented Basic is quite economical, once all the manufacturer-specific command words are removed. Functions which do not appear in the Table must be provided from the logical function libraries, of which there are six specific classes: (1) disk I/O, (2) screen/keyboard I/O, (3) line printer I/O, (4) serial I/O, (5) general utilities, and (6) user-designed proprietary functions.

PTROPN	Open the printer file
PRINT	Print a line or line segment
PTRCK	Check printer readiness status
PTRINI	Initialize printer ROM features
PTRCLS	Close the printer file

Table 3. Logical Functions in the Line Printer Library

Entry and exit conditions to and from the main program and modules are defined at the simplest possible level. Taking the INSTR macro call (illustrated above) as an example, the entry conditions for the module "SERCH" require that only two numeric variables, two string variables, and the name of the object computer be specified. Each element is assigned a specific function with expressed or implied rules.

For example, it doesn't matter if X is cleared to zero on entry, but A$ must be at least as long as or longer than B$, and A must be an integer value greater than zero and equal to or less than the length of A$ minus the length of B$. If the macro "SERCH" has been expanded to test one or more of these entry conditions before attempting the execution of the substring search, it would also contain provisions for returning error codes to the calling application program. Upon exit, a single condition is expected: the condition of X equal to zero or greater than zero, but less than the length of A$ minus the length of B$ plus one.

Keep the specification of entry and exit conditions at the most general level which may be fulfilled by even the most elemental computer. The exchange of data between the main program and the macro module it calls should never depend on the availability of instructions like PEEK and POKE; many computer Basics do not offer any such commands. Automatic conversion from one computer to another in such a case would not be possible—even worse, the design concept of the application program would have to be redrawn before reprogramming could begin.

Arguments

Arguments included in disk function calls include: name of the disk controller;

CLRSCR	Initialize the screen and keyboard
SCROLU	Scroll-up
SCROLD	Scroll-down
SCROLP	Set scroll protected region
CLRLIN	Clear to end-of-line
CLRFRM	Clear to end-of-frame
STROBE	Interrogate keyboard once for pressed key
KEYIN	Receive typed-in line and edit

Table 4. Logical Functions in the Screen-Keyboard Library

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.**

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 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

123 LOCUST STREET LANCASTER, PENNSYLVANIA 17602
Phone 717/291-1116

✓278

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation.
FCC Registration Number: A909KE-68171-DM-N

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

THE PROGRAM STORE
Washington, DC
800/424-2738

STEVENS RADIO SHACK
Phoenixville, PA
800/345-6279

SIMUTEK
Tucson, AZ
800/528-1149

VISA · master charge

✓See List of Advertisers on page 322

" By its nature, a compiler cannot cope with the breadth and depth of code which makes application programs manufacturer-specific."

size of the disks; density of disk data (single, double, quad); number of tracks per disk, sectors per track, blocks per sector, and block length in bytes; number of drives on line and number of lowest drive (zero or one); port address of disk controller; and access mode for the current logical call (In, Out, Both).

Arguments included in screen/keyboard function calls include: model name of console unit; number of vertical screen lines; number of horizontal screen positions; port address of screen; port address of keyboard; decimal value of cursor character; decimal values for various decorative graphics.

Arguments included in line printer function calls include: model name of the printer; number of vertical lines per page; number of horizontal positions per line; port address of printer controller; page prompting indicator; decimal value for various control characters.

Typical arguments included in serial I/O function calls include: model name of serial controller; name of modem, if used; name of protocol, if used; port address of controller; default baud rate; default word length; default stop bits; and default parity, if any.

Although 40-60 separate macro subroutines is not a small writing endeavor, it is not all that burdensome, when you consider the effort required to convert an applications program from one computer to another, and the growing inclination that cross-compilation into machine language is the minimum acceptable form for commercially oriented software. Moreover, large portions of each new program will not have to be documented: The documentation for the macro library modules is written once for all future references.

By its nature, a compiler cannot cope with the breadth and depth of code which makes applications programs manufacturer-specific and, therefore, highly labor-intensive during conversion. Whether applications programs are intended to execute under the interpreter or after compilation, it is recommended that their manufacturer-specific routines be called from logical function libraries by a Basic language macroprocessor. ■

JOFES precoded macro libraries are available from J. Olmstead Financial Engineering Systems.

SERIAL	Initialize the serial I/O controller
RACK	Request acknowledgement
ACK	Send acknowledgement
WRACK	Wait for request for acknowledgement
NACK	Send negative acknowledgement
RECEIV	Receive a line
XMIT	Transmit a line
SEREND	Close serial I/O controller
PLOT	Draw a line from x1,y1 to x2,y2
PENUP	Lift pen off paper
PENDWN	Position pen onto paper ready to draw
PENSEL	Select pen number 1,2,3,4
GRID	Set X/Y grid increments
FONT	Select character font
SERPRT	Print in graphics mode
LJUST	Left-justify data in field
RJUST	Right-justify data in field
SPREAD	Typeset line in field
BOX	Draw rectangle
CENCEN	Center data vertically and horizontally in box

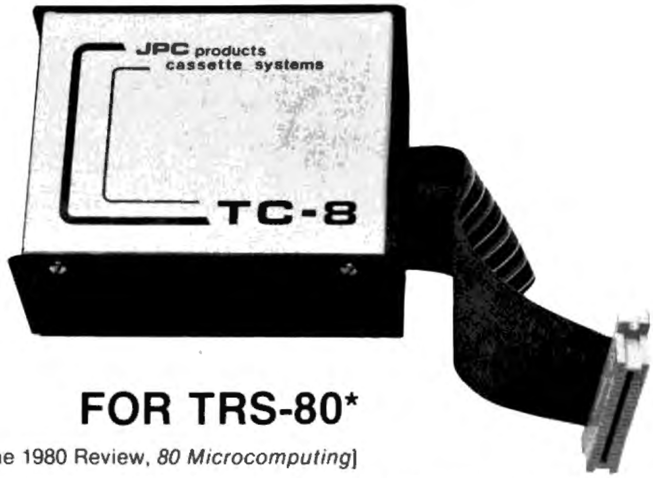
Table 5. Logical Functions in the Serial I/O (Communications) Library

FLUD	Create a string of a given number of a given byte
SERCH	Search for a substring within a larger string
BCDASC	Convert BCD to ASCII
ASCBCD	Convert ASCII to BCD
ASCHEX	Convert ASCII hex address to pure hex
HEXASC	Convert pure hex to ASCII form
FORMAT	Convert a named BCD field to ASCII and format it in preparation for printing or display
ERTRAP	General error trapping location
ERRMSG	Generate error messages
ERLERX	Emergency error exit to be used when user's error exit subroutine is not operational

Table 6. Logical Functions in the Utility Library

Poor Man's Floppy

HIGH SPEED CASSETTE SYSTEM



FOR TRS-80*

Now the widely acclaimed JPC Cassette System is available for your TRS-80* computer. The price is only $90.00

[Reprint of June 1980 Review, *80 Microcomputing*]

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 322

DON'T SAY WE DIDN'T WARN YOU!

ASYLUM!

ASYLUM Is now in stock and ready for immediate shipment. This is it! The ultimate adventure.

ASYLUM is the most sophisticated, sinister, challenging 3-D graphics adventure ever written! You are placed in an asylum for DEATHMAZE survivors. Your only goal: ESCAPE!

3-D PERSPECTIVE GRAPHICS!

ASYLUM has over 1200 locations! Hallways recede into the screen, *as though you are actually there!* Doors open and close! Beds, desks, and even guards and other inmates are drawn on the screen! And graphics are generated instantaneously! Simply press an arrow key to move or turn, and you are moved. Absolutely *no* waiting!

ADVANCED LANGUAGE INTERPRETER!

English language commands may be issued at any time from a vocabulary of almost 300 words. Commands may include complete statements. "GET EVERYTHING ON THE DESK EXCEPT THE MATCHES" is perfectly legal!

CHALLENGING, CHILLING!

To whet your imagination: the Hall of Doors, Nicholson the axe murderer, a revolving door, the time machine, the guru of nails, the old roadster in the drag strip, the ancient catapult room, plus dozens of other problems and a reasonable amount of built-in, subtle hints.

ATARI 400 COMPUTER TO BE AWARDED TO ESCAPEE!

Will anyone escape? Who knows? Intelligence, cunning and patience will be necessary. Send us your correct solution postmarked no later than November 15th. On November 30th we will draw from the entrants for an ATARI 400 16K color computer!

NO HINTS WILL BE GIVEN
OVER THE PHONE!

ASYLUM 16K cassette	**$14.95**
ASYLUM 32K disk	**$19.95**

TRS-80 LEVEL II,
MODEL I AND MODEL III ONLY.

Med Systems Software
P.O. Box 2674-T Chapel Hill, NC 27514
(919) 933-1990 ✓128

ASYLUM MAY ALSO BE PURCHASED
FROM THE PROGRAM STORE,
HW ELECTRONICS (HOBBY WORLD),
AND DEALERS WORLDWIDE.

Serial I/O.

80 to RS-232 to TI-810

Morris Herman
503 Rosario Drive
Santa Barbara, CA 93110

The TI-810 printer is a multi-copy impact printer that utilizes a microprocessor to optimize bi-directional printing. I interfaced it to my TRS-80, using the Radio Shack RS232C interface.

The speed of the printer is 150 cps (characters per second) using a 9 by 7 dot matrix. The printer also features adjustable tractor feed, switch selectable automatic line feed on carriage return, switch selectable auto perforation skipover control, six or eight lines per inch, switch selectable parity, baud rates from 110 to 9600, software or hardware-programmable vertical form, and a standard RS232 interface. My printer also has a compressed character printing option which allows 132 characters in an eight-inch line length, a battery-backed vertical forms control program that is not lost at power down, and a full set of ASCII characters. In addition the printer features automatic self-test, a function that checks electronic and mechanical parts of the printer.

I received my Radio Shack RS232C interface before the TI-810 printer. However, with a loaned copy of the TI-810 Operator's Manual I proceeded to read and compare the serial interface requirements. It was apparent that this interface was not of the plug and run variety. Every signal from the serial interface (with the exception of signal and protective grounds) is defined as an input or output in the RS232C interface and the TI-810 similarly. That is, an input signal to the printer is defined as an input signal to the RS232C interface also. For example, the Receive Data line of the TI-810 was on pin 3 whereas the TRS-80 was transmitting data on the Transmit Data line—pin 2. It turns out that this is the only signal that can be switched using a double-pole, double-throw switch provided in the RS232C interface.

Important signals like Request to Send, Data Set Ready, Data Terminal Ready, etc. are defined to be in conflict in the two documents. The reason for this is that the RS232C configures the TRS-80 as a data terminal, while the TI-810 is also configured as a data terminal. There would have been no problem had the RS232C interface configured the TRS-80 as a data communication device. Table 1 shows these conflicts in more detail.

As far as the TI-810 is concerned, the important signals are:

- Data from TRS-80 on Receive Data—pin 3.
- Positive EIA level on Data Set Ready—pin 6.
- Positive EIA level on Carrier Detect—pin 8.

Status signals from TI-810 are:

- Printer busy on Reverse Channel—positive EIA level = printer not busy—pin 11.
- Printer on-line on Data Terminal Ready—positive EIA level = printer on-line—pin 20.

As far as the TRS-80 RS232C interface, the important signals are:

- Data to printer on Transmit Data—pin 2.
- Printer busy on Clear to Send—positive EIA level = printer not busy—pin 5.
- Printer on-line on Data Set Ready—positive EIA level = printer on-line—pin 6.

(See Table 2 to construct a crossover cable.)

EIA Levels

For timing and control, the signal is considered to be on when the voltage on the line is more positive than three volts (with respect to Signal Ground); and is considered to be off when the voltage is more negative than minus three volts (with respect to Signal Ground). The on condition corresponds to a TTL logic zero (0) and the off condition corresponds to a TTL logic one (1). All these level conversions are performed by the RS232C interface.

Transmission of data is performed over a single line using an asynchronous-bit-serial technique. This means that there is no clock line to indicate when the next bit arrives. However, both the printer and the RS232C interface know the baud rate or bit rate and parity type (even or odd) at which the bits are sent. The printer knows by the pencil switch settings while the interface is programmed by the driver. Needless to say, the two settings must match for correct operation of the interface. The two EIA levels play an important part in the data transmission. The start bit on the line is denoted by a negative-going EIA level. The next eight bits are output serially, using positive EIA levels for zero (0) and negative EIA levels for one (1). The stop bit is then appended as a positive EIA level that is sustained until the next start bit of the next character is sent. To ensure against errors due to different clock circuits, the actual baud rate is multiplied by 16 and data lines are sampled in the middle of each bit. This allows a differential of one-sixteenth of the baud rate with no data loss.

The Program

At this point, I looked at the software required to run the printer using the RS232C interface. The sample program in the RS232C manual was not suitable for my needs in addition to being in error. The word description does not match the program listing. Somebody must have fixed the program, but the new listing did not get into the

"The interface initialization and operation should be automatic."

manual. I made the following decisions concerning the RS232C interface operation:

- The baud rate should be fixed at 2400 regardless of the interface switch setting.
- Word lengths should be seven bits with one even parity bit plus one start bit and one stop bit.
- Check printer on-line, printer busy and RS232C transmit register empty before outputting a character.
- The interface initialization and operation should be automatic.

To accomplish the first three objectives, I wrote a machine-language program that changed the printer device control block (DCB) in the TRS-80 RAM to point to my new RS232C interface driver. The DCB is used by Basic or any other program as a pointer to the driver program for that particular device. The character to be output is in register C and a Call instruction is executed to the address in locations 4026 and 4027. In addition, the program initializes the interface with the desired baud rate, word length, parity, start and stop bits. At this point, the program returns to the system. Along with the initialization part, the actual character output driver is made resident in memory and is pointed to by the DCB. (See Program Listing 1.)

The program originates at address FF00 (hex) since I have a 48K RAM TRS-80. For other size machines, this can be changed to 7F00 for 16K, and BF00 for 32K machines. In that case, make sure to change line 170 to load 7F or BF into the memory location pointed to by HL. Lines 120-170 in the program change the DCB to point to the new driver.

The OUT E8 instruction resets the UART in the RS232C interface. The A4 value sent by the OUT EA instruction sets up even parity, seven-bit word length, one-stop bit, enables Transmit Data and sets up positive EIA levels on Request to Send and Data Terminal Ready. The last two signals will translate through the crossover cable to signals Carrier Detect and Data Set Ready for the TI-810 printer. For other printers, these requirements may change. Remember that a zero in the UART control register (see Table 6 of the RS232C manual), will set up a positive EIA level on that line. Conversely, a one will set up a negative EIA level. It may also be necessary to change the crossover cable wiring, in order to set up EIA levels. Only one thing is constant, and that is, as Table 6 of the RS232C manual shows, the interface can only affect the signals on pins 4 and 20 of TRS-80 side (i.e., Request to Send and Data Terminal Ready signals respectively).

The AA value sent with the out E9 instruction sets up the 2400-baud rate. This baud rate is optimum for the TI-810 since the printer operates at 150 characters per second which converts to a 1500-baud rate. The next lower baud rate of 1200 will idle the printer, while at 2400 baud the printer will be operating a maximum rate. For printers that operate at lower baud rates, see the RS232C manual for Table 5 which describes the baud rate settings. Remember to use two nibbles even though only the transmit channel is used (i.e., 66H for 600 baud).

At this point, the initialization is complete and the program jumps to location 402D which is the entry point to TRSDOS without initialization. Now let's take a look at the actual character output driver part of the program which starts at location FF1C in my version. The IN E8 instruction inputs the modem status register. This register will contain the status of the printer on lines Clear to Send and Data Set Ready in bits seven and six respectively. Remember that the crossover cable translates these signals to lines Reverse Channel and Data Terminal Ready, which contain the busy and on-line status of the printer. We are looking for positive EIA levels on these signals, and therefore our driver will check for zero in these bits. If a signal is non-zero, we loop back to check the lines. This will result in a hang-up of the TRS-80, if the printer is not ready. Once the printer is readied, the driver will continue.

The IN EA instruction inputs the UART status register which contains error indicators and a bit that indicates if the previous output character has been sent to the printer. This information is reflected in the transmitter-holding-register-empty bit 6. We are looking for a one to indicate that the register is empty and loop back if a zero is found.

Finally, we take the character to be output and load it into the A register, output it with the OUT EB instruction and perform a RET instruction to exit and return to the calling program.

If you look at the RS232C interface manual, you will note that the driver adds a line feed after every carriage return. I have not included this feature in order to allow for more flexibility. The TI-810 has a pencil switch that can generate a line feed automatically after every carriage return. For the Electric Pencil, in order to underline text, it is necessary that the two controls be output separately. It is easier to change a pencil switch than to do a driver change.

To automate the execution of the above program, I put it into a disk file and set up the Auto feature of TRSDOS to automatically execute this program on power-up. The actual code of the program was first assembled with the Radio Shack editor/assembler and the resultant code was input into the TRS-80 using the Debug utility of TRSDOS as follows:

- Boot up the TRSDOS.
- After the display of the DOS Ready message appears type Debug.
- Hit the Break key. This will result in Debug being loaded and started.
- Use the M command to set up the starting address of the program. In my case, I typed MFF00 followed by a space.
- Enter the hexadecimal values of the program followed by a space. The space increments the program counter so that the next value can be entered.
- After all values have been entered, exit modify-memory mode by typing Enter.
- Exit Debug by using the G402D command.

At this point the program is in memory ready to be made a disk file. The Dump utility of TRSDOS will be used to create that file.

TRS-80	PIN #	DESCRIPTION	TI-810
—	1	Protective Ground	—
Output	2	Transmit Data	Output
Input	3	Receive Data	Input
Output	4	Request to Send	Output
Input	5	Clear to Send	Input
Input	6	Data Set Ready	Input
—	7	Signal Ground	—
Input	8	Carrier Detect	Input
Not Used	11	Reverse Channel	Output
Output	20	Data Term Ready	Output
Input	22	Ring Indicator	Not Used

Table 1. Pin Assignments

TRS-80 FEMALE DB25S		TI-810 MALE DB25P
1—Protective Gnd	to	1—Protective Gnd
2—Transmit Data	to	3—Receive Data
3—Receive Data	to	2—Transmit Data
4—Request to Send	to	8—Carrier Detect
8—Carrier Detect	to	4—Request to Send
6—Data Set Ready	to	20—Data Terminal Ready
20—Data Terminal Ready	to	6—Data Set Ready
7—Signal Gnd	to	7—Signal Gnd
5—Clear to Send	to	11—Reverse Channel

Table 2

"I have not found it necessary to protect my driver in upper memory."

```
               00050 ;          LISTING 1
               00100 ;          TRS-80 SERIAL RS232-C INTFC DRVR
FF00           00110            ORG     0FF00H
FF00 212540    00120            LD      HL,4025H       ;PRINTER DCB ADDR
FF03 3602      00130            LD      (HL),2         ;DCB TYPE
FF05 212640    00140            LD      HL,4026H
FF08 361C      00150            LD      (HL),1CH       ;LS BYTE OF DRVR ADDR
FF0A 212740    00160            LD      HL,4027H
FF0D 36FF      00170            LD      (HL),0FFH      ;MS BYTE OF DRVR ADDR
FF0F D3E8      00180            OUT     (0E8H),A       ;RESET UART
FF11 3EA4      00190            LD      A,0A4H         ;SET UP INTFC
FF13 D3EA      00200            OUT     (0EAH),A
FF15 3EAA      00210            LD      A,0AAH         ;SETUP,BAUD RATE(2400)
FF17 D3E9      00220            OUT     (0E9H),A
FF19 C32D40    00230            JP      402DH          ;EXIT TO TRSDOS
FF1C DBE8      00240 DRIVER     IN      A,(0E8H)       ;INPUT PRNTR STAT
FF1E CB77      00250            BIT     6,A            ;CHECK PRNTR ONLINE
FF20 20FA      00260            JR      NZ,DRIVER
FF22 CB7F      00270            BIT     7,A            ;CHECK PRNTR BUSY
FF24 20F6      00280            JR      NZ,DRIVER
FF26 DBEA      00290 STAT       IN      A,(0EAH)       ;INPUT UART STAT
FF28 CB77      00300            BIT     6,A            ;CHECK TX REGIS. EMPTY
FF2A 28FA      00310            JR      Z,STAT
FF2C 79        00320            LD      A,C            ;MOVE CHARAC. IN A REGIS.
FF2D D3EB      00330            OUT     (0EBH),A       ;OUTPUT TO UART
FF2F C9        00340            RET                    ;RETURN
0000           00350            END
```

Program Listing 1. Serial RS232C Interface Driver.

Make sure the write-protect tab is off the diskette before trying to write on it. Type the following:

DUMP RS232/CMD (START = X'FF00',END = X'FF2F',TRA = X'FF00')

For other size RAM machines, change the Dump parameters accordingly. Finally, the RS232 program is made executable on power-up by typing: AUTO RS232.

For a non-disk TRS-80, you have to generate a System program to accomplish the same purpose. This program will differ from the program in Program Listing 1 in two respects: It is not automatic; the program has to be loaded and run after entering Basic by using the System command; and the reentry to Basic in a non-disk system is accomplished by jumping to location 1A19. You can use the Radio Shack editor/assembler or the RSM/2 monitor to create the System tape. Remember to change line 230 to perform a jump to 1A19.

I have not found it necessary to protect my driver in upper memory by using a Memory Size entry. If you find that your driver gets wiped out by Basic, enter the decimal equivalent of the starting address of the driver minus one. (For 32K and 48K machines, subtract 65536 from the actual address.)

Your system is now ready to use an RS232 printer, but the work is not done yet. You can use LPRINT and LLIST, but if you have other software packages such as the editor/assembler, RSM2 monitor, cassette version of the Electric Pencil, etc., you will quickly discover that the printer will not work. The problem is that those software packages use their own print routines that do not utilize the DCB in locations 4026 and 4027. It is then necessary to find the start of the print driver and change it by patching in a jump to the RS232 driver in upper memory. I have found that I can use the RSM/2D monitor from Small Systems Software to hunt for the memory-mapped address of the printer (37E8) and then disassemble the code around it to find the start of the driver and patch a C31CFF to jump to my driver. If the original driver does not have the character to be output in register C, it is also necessary to save the current contents of C, move character to C, call the RS232 driver, and upon return restore C to its original value. What you have to do depends entirely on the way that driver was written.

The disk version of Electric Pencil includes such a driver with a minor problem. Since the authors could not know which printer is interfaced with the TRS-80, the driver does not check to see if the printer is busy or on-line. This deficiency limits the printer operation to 1200 baud.

There is another solution to this general problem and that is to use a parallel interface to the TI-810. It turns out that the TI-810 has a Centronics-compatible interface. This is the interface that Radio Shack uses for its printers. But alas, the interface is not perfect for my TI-810 since the original parallel driver in ROM Basic does not pass form control characters such as form feed (12) and vertical tab (11). It seems the driver is written for a printer with no forms control and the driver performs that function by counting lines and using a line count of 66 (unless modified) in RAM. To modify this driver, it is necessary to write another driver, relocate in upper memory and change the DCB to point to the new driver. With all of your experience in serial interfacing, you should be able to do that job very easily. ■

✓See List of Advertisers on page 322

At last! Effortless data statements from machine language to Basic.

Datagen

Dan & Cass Lewart
12 Georjean Drive
Holmdel, NJ 07733

A TRS-80 programmer will often need to use Basic and machine language (System) progams together. A Basic program is great for writing complicated routines with lots of logic or with arithmetic calculations. But it is too slow for tasks such as generating musical tones or for sorting large arrays. A machine language program, though more difficult to write, will do all those jobs quickly and efficiently and will get around many limitations of Basic. For this reason, sophisticated programs frequently consist of a Basic program performing some of the chores and then calling one or more machine language routines to do the rest.

Storing and Loading of Basic/Machine Language Programs

The Basic and machine language parts of a combined program can be separately stored and then loaded with Level II Basic CSAVE, CLOAD and System commands. However, this approach is both inconvenient and time consuming, because two or more programs have to be loaded everytime the program is to be run. Therefore, several methods for storing and loading Basic and machine language programs as a single module were developed. Such methods are described in the Basic Level II Manual (Pages 8/8-8/12), and in this and other magazines.

The approach in the Basic manual consists of storing a machine language program by reading a string of data statements with decimal equivalents of the machine language program, POKEing it into memory, and executing with a USR call from a Basic driver program. The difficulty with this approach is that the values in the

```
Memory Size? 32255 (enter)
Radio Shack Level II Basic
Ready
>CLOAD (Program in Fig. 4) (enter)
Ready
>SYSTEM (enter)
*? MUSIC (enter)
*? DATGEN (enter)
*? / (enter)
First Address? H7000 (enter)
Last Address? H7014 (enter)
First Line? 300 (enter)
Increment? 10 (enter)
Ready
>LIST (enter)
```

Fig. 1. Dialog with computer to execute DATGEN (user entries are underlined)

Program Listing 1. Assembled Z80 Source Code of DATGEN

```
0002                BREAK   EQU     2           ;BREAK KEY BIT
01C9                CLS     EQU     1C9H        ;CLEAR SCREEN
1C90                CPHLDE  EQU     1C90H       ;COMPARE HL AND DE
0088                DATA    EQU     88H         ;BASIC TOKEN FOR DATA
0001                DGT     EQU     1           ;DIGIT FLAG ON
40F9                END     EQU     40F9H       ;PTR TO BASIC PROGRAM END
1AF8                FIXPTR  EQU     1AF8H       ;CALCULATE LINE POINTERS
1D78                GETDGT  EQU     1D78H       ;SET C FOR NUMERIC CHAR
1AEF                INIT    EQU     1AEFH       ;RETURN TO BASIC
1BB3                INPSTR  EQU     1BB3H       ;INPUT STRING FROM KB
000A                LENGTH  EQU     10          ;BYTES PER BASIC LINE
0000                NODGT   EQU     0           ;DIGIT FLAG OFF
28A7                OUTSTR  EQU     28A7H       ;DISPLAY STRING
3840                ROW7    EQU     3840H       ;BREAK KEY ROW
7E00                        ORG     7E00H       ;32256 DECIMAL
7E00 CDC901         DATGEN  CALL    CLS         ;CLEAR SCREEN
7E03 21247F                 LD      HL,PRMPT1   ;POINT TO FIRST PROMPT
7E06 CDA37E                 CALL    GETINT      ;INPUT FIRST ADDRESS
7E09 225B7F                 LD      (MACH),HL   ;STORE IT
7E0C 21327F                 LD      HL,PRMPT2   ;POINT TO SECOND PROMPT
7E0F CDA37E                 CALL    GETINT      ;INPUT LAST ADDRESS
7E12 22577F                 LD      (LAST),HL   ;STORE IT
7E15 213F7F                 LD      HL,PRMPT3   ;POINT TO THIRD PROMPT
7E18 CDA37E                 CALL    GETINT      ;INPUT FIRST BASIC LINE#
7E1B 22597F                 LD      (LINE),HL   ;STORE IT
7E1E 214A7F                 LD      HL,PRMPT4   ;POINT TO FOURTH PROMPT
7E21 CDA37E                 CALL    GETINT      ;INPUT INCREMENT
7E24 22557F                 LD      (INC),HL    ;STORE IT
7E27 DD2AF940               LD      IX,(END)    ;2+ADDR OF PROGRAM END
7E2B DD2B                   DEC     IX          ;CORRECT IX TO POINT
7E2D DD2B                   DEC     IX          ;TO END OF PROGRAM
7E2F 3EFF           NEWLIN  LD      A,0FFH      ;ANYTHING > 0
7E31 CD1E7F                 CALL    PUTCHR      ;FILL LINE POINTER
7E34 CD1E7F                 CALL    PUTCHR      ;TEMPORARILY
7E37 2A597F                 LD      HL,(LINE)   ;BASIC LINE#
7E3A 7D                     LD      A,L         ;LINE# LSB
7E3B CD1E7F                 CALL    PUTCHR      ;ADD TO PROGRAM
7E3E 7C                     LD      A,H         ;LINE# MSB
7E3F CD1E7F                 CALL    PUTCHR      ;ADD TO PROGRAM
7E42 3E88                   LD      A,DATA      ;DATA TOKEN
7E44 CD1E7F                 CALL    PUTCHR      ;ADD TO PROGRAM
7E47 CD177F                 CALL    PUTSPC      ;ADD SPACE TO PROGRAM
7E4A 3E0A                   LD      A,LENGTH    ;BYTES PER LINE
7E4C 32547F                 LD      (BYTE),A    ;STORE IT
```

Program continues

"A Basic program. . . is too slow for tasks such as generating musical tones or for sorting large arrays."

data statements have to be computed from the hex equivalents and manually entered into the program. This can be a very awkward, error-prone procedure.

A more exotic method is to store a Basic and a machine language program as a single machine language program and then load the combination program with the System command. Another method is to store the machine language program by imbedding it in a character string in the Basic program. The problem with these two methods is that the combined Basic/machine programs are difficult to modify and list due to special non-printable characters imbedded in the character strings.

DATGEN Does It All

The machine language program DATGEN simplifies the straightforward approach in the Basic manual by automatically generating data statements equivalent to a machine language program stored in memory. DATGEN will take a specified section of the computer memory, translate it from hex into decimal, and then append it as nicely formatted data statements to your Basic program. The assembler source code listing of DATGEN and the equivalent Basic program that will POKE DATGEN into memory are shown in Program Listings 1 and 2.

The unique feature of the DATGEN program is that it actually writes Basic program lines (DATA. . .) and puts them into proper slots in the computer memory to make them indistinguishable from Basic statements entered directly from the keyboard.

How to Use DATGEN

Enter DATGEN into computer memory either by assembling the source Listing 1 or by running the Basic Listing 2 (the data statements were, of course, generated by DATGEN itself). Then the first time around store DATGEN on tape with the T-Bug command 'P7E00 7F53 7E00 DATGEN' or a similar command if you have a different monitor than T-Bug. Before calling DATGEN, enter the Basic driver and the machine language program to be translated into data statements into computer memory via the CLOAD and System commands, if the programs have been previously stored on tape. The machine language program can be entered manually by means of T-Bug or another monitor program, or even by POKEing it manually, if the program is not too long.

A typical sequence of operations would be as follows:

- Protect DATGEN by providing Memory Size = 32255 (16K system).
- CLOAD or enter manually your Basic driver program.

```
7E4F 2A5B7F    NEWBYT  LD     HL,(MACH)      ;ADDRESS OF MACHINE BYTE
7E52 46                LD     B,(HL)         ;MACHINE BYTE
7E53 0E00              LD     C,NODGT        ;NO DIGITS YET
7E55 1664              LD     D,100          ;HUNDRED'S DIGIT
7E57 CD057F            CALL   DIGIT          ;ADD ASCII TO PROGRAM
7E5A 160A              LD     D,10           ;TEN'S DIGIT
7E5C CD057F            CALL   DIGIT          ;ADD ASCII TO PROGRAM
7E5F 0E01              LD     C,DGT          ;PUT DIGIT, NOT SPACE
7E61 1601              LD     D,1            ;UNIT'S DIGIT
7E63 CD057F            CALL   DIGIT          ;ADD ASCII TO PROGRAM
7E66 ED5B577F          LD     DE,(LAST)      ;ADDRESS OF LAST BYTE
7E6A CD901C            CALL   CPHLDE         ;COMPARE WITH MACH ADDR
7E6D 3022              JR     NC,DONE        ;JUMP IF LAST BYTE DONE
7E6F 23                INC    HL             ;POINT TO NEXT BYTE
7E70 225B7F            LD     (MACH),HL      ;STORE IT
7E73 21547F            LD     HL,BYTE        ;BYTES LEFT IN LINE
7E76 35                DEC    (HL)           ;ONE LESS
7E77 2807              JR     Z,FINLIN       ;JUMP IF LINE DONE
7E79 3E2C              LD     A,','          ;SEPARATE DATA
7E7B CD1E7F            CALL   PUTCHR         ;ADD TO PROGRAM
7E7E 18CF              JR     NEWBYT         ;DO NEXT MACHINE BYTE
7E80 AF        FINLIN  XOR    A              ;END OF LINE
7E81 CD1E7F            CALL   PUTCHR         ;ADD TO PROGRAM
7E84 2A597F            LD     HL,(LINE)      ;LINE#
7E87 ED4B557F          LD     BC,(INC)       ;INCREMENT
7E8B 09                ADD    HL,BC          ;NEXT LINE#
7E8C 22597F            LD     (LINE),HL      ;STORE IT
7E8F 189E              JR     NEWLIN         ;DO NEXT LINE
7E91 AF        DONE    XOR    A              ;ZERO
7E92 0603              LD     B,3            ;THREE OF THEM
7E94 CD1E7F    OUTZER  CALL   PUTCHR         ;ADD TO PROGRAM
7E97 10FB              DJNZ   OUTZER         ;THREE "0"'S
7E99 DD22F940          LD     (END),IX       ;STORE PROGRAM END PTR
7E9D CDF81A            CALL   FIXPTR         ;CALCULATE LINE POINTERS
7EA0 C3EF1A            JP     INIT           ;RETURN TO BASIC
7EA3 E5        GETINT  PUSH   HL             ;SAVE PROMPT ADDRESS
7EA4 DDE1              POP    IX             ;IN IX
7EA6 DDE5      MESS    PUSH   IX             ;RESTORE PROMPT ADDRESS
7EA8 E1                POP    HL             ;TO HL
7EA9 CDA728            CALL   OUTSTR         ;DISPLAY PROMPT
7EAC CDB31B            CALL   INPSTR         ;INPUT STRING
7EAF 3A4038            LD     A,(ROW7)       ;BREAK KEY ROW
7EB2 CB57              BIT    BREAK,A        ;BREAK KEY PRESSED?
7EB4 2804              JR     Z,NOBRK        ;JUMP IF NOT
7EB6 E1                POP    HL             ;FIX STACK
7EB7 C3007E            JP     DATGEN         ;START OVER
7EBA 110000    NOBRK   LD     DE,0           ;INTEGER=0
7EBD CD781D            CALL   GETDGT         ;GET NEXT CHAR FROM INPUT
7EC0 3824              JR     C,DEC          ;IF DIGIT, BASE 10
7EC2 2832              JR     Z,FINISH       ;IF CR, RETURN 0
7EC4 FE48              CP     'H'            ;IS IT H(EX)?
7EC6 20DE              JR     NZ,MESS        ;NO, PROMPT AGAIN
7EC8 CD781D            CALL   GETDGT         ;GET HEX DIGIT
7ECB 380C      HEX     JR     C,NUMBER       ;JUMP IF 0-9
7ECD 2827              JR     Z,FINISH       ;RETURN IF CR
7ECF FE41              CP     'A'            ;LESS THAN "A"?
7ED1 38D3              JR     C,MESS         ;PROMPT AGAIN IF IT IS
7ED3 FE47              CP     'F'+1          ;GREATER THAN "F"?
7ED5 30CF              JR     NC,MESS        ;PROMPT AGAIN IF IT IS
7ED7 D607              SUB    7              ;CONVERT A-F
7ED9 E5        NUMBER  PUSH   HL             ;SAVE INPUT BUFFER PTR
7EDA FDE1              POP    IY             ;IN IY
7EDC EB                EX     DE,HL          ;HL IS INTEGER
7EDD 29                ADD    HL,HL          ;MULTIPLY
7EDE 29                ADD    HL,HL          ;INTEGER
7EDF 29                ADD    HL,HL          ;BY
7EE0 29                ADD    HL,HL          ;10H
7EE1 CDF87E            CALL   ADDCHR         ;ADD CHAR TO INTEGER
7EE4 18E5              JR     HEX            ;LOOK AT NEXT CHARACTER
7EE6 E5        DEC     PUSH   HL             ;SAVE INPUT BUFFER PTR
7EE7 FDE1              POP    IY             ;IN IY
7EE9 62                LD     H,D            ;COPY INTEGER
7EEA 6B                LD     L,E            ;TO HL
7EEB 19                ADD    HL,DE          ;MULTIPLY
7EEC 29                ADD    HL,HL          ;INTEGER
7EED 19                ADD    HL,DE          ;BY
7EEE 29                ADD    HL,HL          ;10
7EEF CDF87E            CALL   ADDCHR         ;ADD CHAR TO INTEGER
7EF2 38F2              JR     C,DEC          ;JUMP IF 0-9
7EF4 20B0              JR     NZ,MESS        ;PROMPT AGAIN IF NOT CR
7EF6 EB        FINISH  EX     DE,HL          ;MOVE INTEGER TO HL
7EF7 C9                RET                   ;RETURN
7EF8 D630      ADDCHR  SUB    '0'            ;CONVERT TO HEX
7EFA 1600              LD     D,0            ;ZERO MSB
7EFC 5F                LD     E,A            ;LSB=CHAR
7EFD 19                ADD    HL,DE          ;ADD CHAR
7EFE EB                EX     DE,HL          ;RESTORE INTEGER TO DE
7EFF FDE5              PUSH   IY             ;RESTORE INPUT BUFFER PTR
7F01 E1                POP    HL             ;TO HL
```

Program continues

"In case of error, press the Break key and the prompts will reappear."

- Load the machine language program to be appended to your Basic program with the System command.
- Load the DATGEN program with the System command, hit / (enter).
- Answer all prompts so that the data statement line numbers are higher than line numbers in your Basic driver program.
- List the newly combined program to make sure it is errorless.
- Store it with the CSAVE command and/or run it.

DATGEN will issue the following prompts:
First Address—Is the first location of the machine language program in decimal or hex (if in hex proceed with H);
Last Address—Last location, as above;
First Line—First line number for data statements, should not interfere with your Basic program;
Increment—Line number increment of data statements.

In case of error, press the Break key and the prompts will reappear. After you answer the last prompt, DATGEN will run, append the data statements and then return to Basic with Ready.

```
7F02 C3781D            JP      GETDGT           ;GET NEXT CHAR AND RETURN
7F05 78       DIGIT    LD      A,B              ;MACHINE BYTE REMAINDER
7F06 1E2F              LD      E,'0'-1          ;ASCII OF DIGIT
7F08 1C       DIV      INC     E                ;NEXT DIGIT
7F09 92                SUB     D                ;DECREMENT PLACE VALUE
7F0A 30FC              JR      NC,DIV           ;AGAIN IF NOT NEGATIVE
7F0C 82                ADD     A,D              ;UNDO EXTRA SUBTRACTION
7F0D 47                LD      B,A              ;SAVE REMAINDER
7F0E CB41              BIT     0,C              ;CHECK TO SEE IF DIGIT
7F10 200B              JR      NZ,NOSKIP        ;ADD DIGIT IF C IS 1
7F12 3E30              LD      A,'0'            ;ASCII 0
7F14 BB                CP      E                ;COMPARE WITH DIGIT
7F15 2004              JR      NZ,NOSPC         ;ADD DIGIT IF NONZERO
7F17 3E20     PUTSPC   LD      A,' '            ;LEFT JUSTIFY WITH SPACE
7F19 1803              JR      PUTCHR           ;ADD IT TO PROGRAM
7F1B 0E01     NOSPC    LD      C,DGT            ;SET DIGIT FLAG
7F1D 7B       NOSKIP   LD      A,E              ;CHAR
7F1E DD7700   PUTCHR   LD      (IX+0),A         ;PUT A IN BASIC PROGRAM
7F21 DD23              INC     IX               ;POINT TO NEXT BYTE
7F23 C9                RET                      ;RETURN
7F24 46       PRMPT1   DEFM    'FIRST ADDRESS'  ;FIRST PROMPT
7F31 00                DEFB    0                ;END OF STRING
7F32 4C       PRMPT2   DEFM    'LAST ADDRESS'   ;SECOND PROMPT
7F3E 00                DEFB    0                ;END OF STRING
7F3F 46       PRMPT3   DEFM    'FIRST LINE'     ;THIRD PROMPT
7F49 00                DEFB    0                ;END OF STRING
7F4A 49       PRMPT4   DEFM    'INCREMENT'      ;FOURTH PROMPT
7F53 00                DEFB    0                ;END OF STRING
0001          BYTE     DEFS    1                ;BYTES LEFT IN LINE
0002          INC      DEFS    2                ;LINE# INCREMENT
0002          LAST     DEFS    2                ;ADDR OF LAST MACH BYTE
0002          LINE     DEFS    2                ;CURRENT LINE#
0002          MACH     DEFS    2                ;CURRENT MACH BYTE ADDR
7E00                   END     DATGEN           ;FINIS
```

Example

We want to execute within a Basic program a short routine generating either a 1500 Hz or a 2500 Hz tone on the tape recorder jack (Port 255). We will use the machine language program called Music shown in Listing 3. To execute this machine language program, a Basic driver program such as the one shown in Program 4 is necessary. The driver also issues prompts for a 1500 or 2500 Hz tone (pressing 'X' turns the tone off).

The data statements will be generated by DATGEN with a dialogue like the one in Program Listing 5. We assume that you have a TRS-80 Level II Basic with 16K of memory and that DATGEN, the tone generating program Music, and the Basic driver program have all been stored on tape. User's responses are underlined. The resulting program with appended data statements is shown in Program Listing 6. It now can be saved on cassette. Running the program will automatically POKE Music into memory.

In the above example the machine language program is so short the use of DATGEN cannot really be justified. However, for longer programs DATGEN can save a lot of time and trouble.

Finally a note of caution: If the machine language program to be translated into data statements is located in the same area in memory as DATGEN (7E00H-7F53H), then before loading DATGEN move your program out of the way, below DATGEN, with a simple PEEK/POKE loop. ■

```
10 DEFINT B,C
20 A=0
30 FOR B=32256 TO 32595
40      READ C:POKE B,C:A=A+C
50 NEXT
60 IF A<>35038 PRINT "ERROR IN DATA STATEMENT":END
70 PRINT"DONE, STORE WITH T-BUG 'P7E00 7F53 7E00 DATGEN'"
80 DATA 205,201,  1, 33, 36,127,205,163,126, 34
90 DATA  91,127, 33, 50,127,205,163,126, 34, 87
100 DATA 127, 33, 63,127,205,163,126, 34, 89,127
110 DATA  33, 74,127,205,163,126, 34, 85,127,221
120 DATA  42,249, 64,221, 43,221, 43, 62,255,205
130 DATA  30,127,205, 30,127, 42, 89,127,125,205
140 DATA  30,127,124,205, 30,127, 62,136,205, 30
150 DATA 127,205, 23,127, 62, 10, 50, 84,127, 42
160 DATA  91,127, 70, 14,  0, 22,100,205,  5,127
170 DATA  22, 10,205,  5,127, 14,  1, 22,  1,205
180 DATA   5,127,237, 91, 87,127,205,144, 28, 48
190 DATA  34, 35, 34, 91,127, 33, 84,127, 53, 40
200 DATA   7, 62, 44,205, 30,127, 24,207,175,205
210 DATA  30,127, 42, 89,127,237, 75, 85,127,  9
220 DATA  34, 89,127, 24,158,175,  6,  3,205, 30
230 DATA 127, 16,251,221, 34,249, 64,205,248, 26
240 DATA 195,239, 26,229,221,225,221,229,225,205
250 DATA 167, 40,205,179, 27, 58, 64, 56,203, 87
260 DATA  40,  4,225,195,  0,126, 17,  0,  0,205
270 DATA 120, 29, 56, 36, 40, 50,254, 72, 32,222
280 DATA 205,120, 29, 56, 12, 40, 39,254, 65, 56
290 DATA 211,254, 71, 48,207,214,  7,229,253,225
300 DATA 235, 41, 41, 41, 41,205,248,126, 24,229
310 DATA 229,253,225, 98,107, 25, 41, 25, 41,205
320 DATA 248,126, 56,242, 32,176,235,201,214, 48
330 DATA  22,  0, 95, 25,235,253,229,225,195,120
340 DATA  29,120, 30, 47, 28,146, 48,252,130, 71
350 DATA 203, 65, 32, 11, 62, 48,187, 32,  4, 62
360 DATA  32, 24,  3, 14,  1,123,221,119,  0,221
370 DATA  35,201, 70, 73, 82, 83, 84, 32, 65, 68
380 DATA  68, 82, 69, 83, 83,  0, 76, 65, 83, 84
390 DATA  32, 65, 68, 68, 82, 69, 83, 83,  0, 70
400 DATA  73, 82, 83, 84, 32, 76, 73, 78, 69,  0
410 DATA  73, 78, 67, 82, 69, 77, 69, 78, 84,  0
```

Program Listing 2. This program will POKE the program DATGEN into memory.

```
0A7F          00100 GETARG  EQU     0A7FH   ;GET ARG FROM BASIC
0001          00110 HI      EQU     1
00FF          00120 PORT    EQU     0FFH    ;CASSETTE PORT
3808          00130 ROW4    EQU     3808H   ;"X" KEY
7000          00140         ORG     7000H   ;=28672 DECIMAL
7000 CD7F0A   00150 START   CALL    GETARG  ;GET VALUE FROM USR(X)
7003 0E01     00160         LD      C,HI    ;TEMP. STORAGE
7005 3E03     00170 SPKR    LD      A,3
7007 A9       00180         XOR     C       ;FLIP HI TO LO, LO TO HI
7008 4F       00190         LD      C,A
7009 D3FF     00200         OUT     (PORT),A;OUTPUT
700B 45       00210         LD      B,L     ;LOAD DELAY LOOP
700C 10FE     00220 DEL     DJNZ    DEL     ;WASTE TIME
700E 3A0838   00230         LD      A,(ROW4);CHECK FOR "X"
7011 1F       00240         RRA
7012 30F1     00250         JR      NC,SPKR ;CONTINUE
7014 C9       00260         RET             ;RETURN TO BASIC
7000          00270         END     START
```

Program Listing 3. Assembled Z80 source code of the demonstration program Music

```
10 'DATGEN DEMO PROGRAM FOR TRS-80 BY DAN AND CASS LEWART
20 'JUNE 16, 1980
30 'LET PROGRAM SET MEMORY SIZE
40 POKE 16562,111:POKE 16561,254:CLEAR 50
50 CLS:DEFINT A-Z
60 'POKE FREQUENCY GENERATING PROGRAM (FIG.3)
70 FOR A=28672 TO 28692:READ B:POKE A,B:NEXT
80 'SET USR(L) VECTOR
90 POKE 16527,112:POKE 16526,0
100 'PROMPT FOR TONE #1 (1500 HZ) OR #2 (2500 HZ)
110 CLS:PRINT@465,"TONE NUMBER (1 OR 2)?"CHR$(14);
120 C$=INKEY$
130 'PROMPT TILL YOU GET "1" OR "2"
140 IF C$<>"1" AND C$<>"2" THEN 120
150 PRINT C$;CHR$(15);
180 'DETERMINE ARGUMENT FROM THE FOLLOWING APPROXIMATION:
190 'L = INT(69000/FREQUENCY - 4.1)
200 'L(1500) = 41   L(2500) = 23
210 IF C$="1" THEN L=41
220 IF C$="2" THEN L=23
230 'CALL FREQUENCY GENERATING ROUTINE, TO EXIT PRESS "X"
240 D=USR(L)
250 GOTO 110
```

Program Listing 4. Driver program for Music without data statements

```
10 'DATGEN DEMO PROGRAM FOR TRS-80 BY DAN AND CASS LEWART
20 'JUNE 16, 1980
30 'LET PROGRAM SET MEMORY SIZE
40 POKE 16562,111:POKE 16561,254:CLEAR 50
50 CLS:DEFINT A-Z
60 'POKE FREQUENCY GENERATING PROGRAM (FIG.3)
70 FOR A=28672 TO 28692:READ B:POKE A,B:NEXT
80 'SET USR(L) VECTOR
90 POKE 16527,112:POKE 16526,0
100 'PROMPT FOR TONE #1 (1500 HZ) OR #2 (2500 HZ)
110 CLS:PRINT@465,"TONE NUMBER (1 OR 2)?"CHR$(14);
120 C$=INKEY$
130 'PROMPT TILL YOU GET "1" OR "2"
140 IF C$<>"1" AND C$<>"2" THEN 120
150 PRINT C$;CHR$(15);
180 'DETERMINE ARGUMENT FROM THE FOLLOWING APPROXIMATION:
190 'L = INT(69000/FREQUENCY - 4.1)
200 'L(1500) = 41   L(2500) = 23
210 IF C$="1" THEN L=41
220 IF C$="2" THEN L=23
230 'CALL FREQUENCY GENERATING ROUTINE, TO EXIT PRESS "X"
240 D=USR(L)
250 GOTO 110
300 DATA 205,127, 10, 14,  1, 62,  3,169, 79,211
310 DATA 255, 69, 16,254, 58,  8, 56, 31, 48,241
320 DATA 201
```

Program Listing 5. Driver from Listing 4 with appended data statements

H&E COMPUTRONICS INC.

•••EVERYTHING FOR YOUR TRS-80™•••

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

BUSINESS PAC 100

NEW

★ All orders processed within 24-Hours
★ 30-Day money back guarantee on all Software

100 Ready-To-Run Business Programs

(ON CASSETTE OR DISKETTE).....Includes 110 Page Users Manual.....5 Cassettes (Or Diskettes)
Inventory Control.....Payroll.....Bookkeeping System.....Stock Calculations.....
Checkbook Maintenance.....Accounts Receivable.....Accounts Payable.....

BUSINESS 100 PROGRAM LIST

1	RULE78	Interest Apportionment by Rule of the 78's
2	ANNU1	Annuity computation program
3	DATE	Time between dates
4	DAYYEAR	Day of year a particular date falls on
5	LEASEINT	Interest rate on lease
6	BREAKEVN	Breakeven analysis
7	DEPRSL	Straightline depreciation
8	DEPRSY	Sum of the digits depreciation
9	DEPRDB	Declining balance depreciation
10	DEPRDDB	Double declining balance depreciation
11	TAXDEP	Cash flow vs. depreciation tables
12	CHECK2	Prints NEBS checks along with daily register
13	CHECKBK1	Checkbook maintenance program
14	MORTGAGE/A	Mortgage amortization table
15	MULTMON	Computes time needed for money to double, triple, etc.
16	SALVAGE	Determines salvage value of an investment
17	RRVARIN	Rate of return on investment with variable inflows
18	RRCONST	Rate of return on investment with constant inflows
19	EFFECT	Effective interest rate of a loan
20	FVAL	Future value of an investment (compound interest)
21	PVAL	Present value of a future amount
22	LOANPAY	Amount of payment on a loan
23	REGWITH	Equal withdrawals from investment to leave 0 over
24	SIMPDISK	Simple discount analysis
25	DATEVAL	Equivalent & nonequivalent dated values for oblig.
26	ANNUDEF	Present value of deferred annuities
27	MARKUP	% Markup analysis for items
28	SINKFUND	Sinking fund amortization program
29	BONDVAL	Value of a bond
30	DEPLETE	Depletion analysis
31	BLACKSH	Black Scholes options analysis
32	STOCVAL1	Expected return on stock via discounts dividends
33	WARVAL	Value of a warrant
34	BONDVAL2	Value of a bond
35	EPSEST	Estimate of future earnings per share for company
36	BETAALPH	Computes alpha and beta variables for stock
37	SHARPE1	Portfolio selection model-i.e. what stocks to hold
38	OPTWRITE	Option writing computations
39	RTVAL	Value of a right
40	EXPVAL	Expected value analysis
41	BAYES	Bayesian decisions
42	VALPRINF	Value of perfect information
43	VALADINF	Value of additional information
44	UTILITY	Derives utility function
45	SIMPLEX	Linear programming solution by simplex method
46	TRANS	Transportation method for linear programming
47	EOQ	Economic order quantity inventory model
48	QUEUE1	Single server queueing (waiting line) model
49	CVP	Cost-volume-profit analysis
50	CONDPROF	Conditional profit tables
51	OPTLOSS	Opportunity loss tables
52	FQUOQ	Fixed quantity economic order quantity model

	NAME	DESCRIPTION
53	FQEOWSH	As above but with shortages permitted
54	FQEOQPB	As above but with quantity price breaks
55	QUEUECB	Cost-benefit waiting line analysis
56	NCFANAL	Net cash-flow analysis for simple investment
57	PROFIND	Profitability index of a project
58	CAP1	Cap. Asset Pr. Model analysis of project
59	WACC	Weighted average cost of capital
60	COMPBAL	True rate on loan with compensating bal. required
61	DISCBAL	True rate on discounted loan
62	MERGANAL	Merger analysis computations
63	FINRAT	Financial ratios for a firm
64	NPV	Net present value of project
65	PRINDLAS	Laspeyres price index
66	PRINDPA	Paasche price index
67	SEASIND	Constructs seasonal quantity indices for company
68	TIMETR	Time series analysis linear trend
69	TIMEMOV	Time series analysis moving average trend
70	FUPRINF	Future price estimation with inflation
71	MAILPAC	Mailing list system
72	LETWRT	Letter writing system-links with MAILPAC
73	SORT3	Sorts list of names
74	LABEL1	Shipping label maker
75	LABEL2	Name label maker
76	BUSBUD	DOME business bookkeeping system
77	TIMECLCK	Computes weeks total hours from timeclock info.
78	ACCTPAY	In memory accounts payable system-storage permitted
79	INVOICE	Generate invoice on screen and print on printer
80	INVENT2	In memory inventory control system
81	TELDIR	Computerized telephone directory
82	TIMUSAN	Time use analysis
83	ASSIGN	Use of assignment algorithm for optimal job assign.
84	ACCTREC	In memory accounts receivable system-storage ok
85	TERMSPAY	Compares 3 methods of repayment of loans
86	PAYNET	Computes gross pay required for given net
87	SELLPR	Computes selling price for given after tax amount
88	ARBCOMP	Arbitrage computations
89	DEPRSF	Sinking fund depreciation
90	UPSZONE	Finds UPS zones from zip code
91	ENVELOPE	Types envelope including return address
92	AUTOEXP	Automobile expense analysis
93	INSFILE	Insurance policy file
94	PAYROLL2	In memory payroll system
95	DILANAL	Dilution analysis
96	LOANAFFD	Loan amount a borrower can afford
97	RENTPRCH	Purchase price for rental property
98	SALELEAS	Sale-leaseback analysis
99	RRCONVBD	Investor's rate of return on convertable bond
100	PORTVAL9	Stock market portfolio storage-valuation program

☐ CASSETTE VERSION	$99.95
☐ DISKETTE VERSION	$99.95
☐ TRS-80* MODEL II VERSION	$149.95

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

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

24 HOUR ORDER LINE
(914) 425-1535

H & E COMPUTRONICS INC.

•••EVERYTHING FOR YOUR TRS-80™•••

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

SMALL BUSINESS SYSTEMS GROUP

COORDINATED BUSINESS SYSTEMS

★ *All orders processed within 24-Hours*
★ *30-Day money back guarantee on all Software*
★ *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*

VISA® master charge

FACTS ABOUT THE S.B.S.G. BUSINESS PACKAGES

1. **S.B.S.G.** is a sophisticated Business Software System designed for the serious businessman.
2. Each of the **S.B.S.G. Business Modules** may be purchased separately...or you may purchase the entire coordinated business system.
3. Modules purchased separately do not coordinate with the General Ledger (although for the standard **S.B.S.G.** fee, the user may upgrade his individual modules for the coordinated system).
4. Foolproof, Step-By-Step procedures are supplied, planned and documented for the **First-Time Computer User.** All programs are self-explanatory, telling the user what is required at every step.
5. Programs are written in **BASIC** and the source code listing is supplied for those users who decide to modify the original system.
6. A complete users manual is supplied with each module.
7. Demo Data diskettes are supplied with sample data.
8. **S.B.S.G.** has an In-House staff that can answer questions and problems related to the proper use of the **S.B.S.G. Business System** (on the telephone or through the mail).
9. First-Time Computer Owners Note-Instructions are provided for entering state payroll withholding tables. There is an additional charge if you prefer to have **S.B.S.G. Programmers** insert the correct data.
10. Minimum system requirement is 2-drives to run any single module.
11. Minimum system requirement is 3-drives to run the coordinated business system (AR-AP-GL) or (AR-AP-GL with PAYROLL).
12. Minimum system requirement is 4-drives to run the extended coordinated system (AR-AP-GL-PR and INVENTORY/INVOICING).
13. The **A. OSBORNE & ASSOCIATES** business manuals are provided **FREE** with each order (they may be purchased separately at $20 per manual).
14. The INVENTORY and INVOICING modules are original programs written by **S.B.S.G.**
15. Each module can be purchased as independent modules to run on a 2 or more drive system except INVOICING.
16. Memory requirement is 48K for the MODEL-I and 64K for the MODEL-II.
17. All **S.B.S.G. BUSINESS SYSTEMS** may be upgraded up to 4-disk drives. No data is ever lost during an upgrade. There is a standard **S.B.S.G.** charge for all upgrades.

ACCOUNTS PAYABLE

The accounts payable system receives data concerning purchases from suppliers and produces checks in payment of outstanding invoices. In addition, it produces cash management reports. This system aids in tight financial control over all cash disbursements of the business. Several reports are available and supply information needed for the analysis of payments, expenses, purchases and cash requirements. All A/P data feeds General Ledger so that data is entered into the system just once. These programs were developed 5 years ago for the Wang micro-computer and have been tested in many environments since then. The package has been converted to the TRS-80™ and is now well documented, on-line, interactive micro-computer system with the capabilities of (or exceeding many larger systems).

CAPABILITIES:

- ★ menu driven; easy to use; full screen prompting and cursor control
- ★ invoice oriented; everything revolves around the invoice; handles new invoice or credit memo or debit memo
- ★ invoce information recorded; invoice #, description, buyer, check register #, invoice date, age date, amount of invoice, discount (in %), freight, tax ($), total payable
- ★ transaction print and file maintenance procedures insure accuracy
- ★ flexible check calculation procedure; allows checks to be calculated for a set of vendors-or-for specific vendors
- ★ program prints your checks; contiguous computer checks with your company letterhead can be purchased from SBSG
- ★ reports include (samples on back):
  - • open item listing/closed item listing - both detail and summary
  - • debit memo listing/credit memo listing
  - • aging
  - • check register report (to give an audit trail of checks printed)
  - • vendor listing and vendor activity (activity of the whole year)
- ★ fully linked to GENERAL LEDGER; each invoice can be distributed to as many as five (5) different GL accounts; system automatically posts to cash and A/P accounts

ACCOUNTS RECEIVABLE

The objective of a computerized A/R system is to prepare accurate and timeley monthly statements to credit customers. Management can generate information required to control the amount of credit extended and the collection of money owed in order to maximize profitable credit sales while minimizing losses from bad debts. The programs composing this system were developed 5 years ago, especially for small businesses using the Wang Microcomputer. They have been tested in many environments since then. Each module can be used stand alone or can feed General Ledger for a fully integrated system.

CAPABILITIES:

- ★ menu driven; easy to use; full screen prompting and cursor control
- ★ invoice oriented; invoices can be entered before ready for billing, when ready for billing, after billing or after paid
- ★ allows entry of new invoice, credit memo, debit memo, or change/delete invoice
- ★ allows for progress payment
- ★ transaction information includes:
  - • type of A/R transaction
  - • customer P.O. #
  - • description of P.O.
  - • billing date
  - • general ledger account number
  - • invoice amount
  - • shipping/transportation charges
  - • tax charges
  - • payment
  - • progress payment information
  - • transaction print & file maintenance procedures insure accuracy
- ★ customer statements printed; computer statements with your company letterhead can be purchased from SBSG
- ★ reports include: (samples on back)
  - • listing of invoices not yet billed
  - • open items (unpaid invoices)
  - • closed items (paid invoices)
  - • aging
- ★ fully linked to General Ledger; will post to applicable accounts; debit A/R, credits account you specify

H & E COMPUTRONICS INC.

•••EVERYTHING FOR YOUR TRS-80™•••

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

PAYROLL

Payroll invoices many complex calculations and the production of reports and documents, many of which are required by government agencies. It is an ideal candidate for the computer. With this Payroll system in-house, you can promptly and accurately pay your employees and generate accruate documents/reports to management, employees, and appropriate government agencies concerning earnings, taxes, and other deductions. The package has been converted to the TRS-80™ and is now a well documented, op-line, interactive, micro-computer system with the capabilities of (or exceeding) many larger systems.

CAPABILITIES:

- ★ performs all necessary payroll tasks including:
  - file maintenance, pay data entry and verification
  - computation of pay and deduction amounts
  - printing of reports and checks
- ★ can handle salaried and hourly employees
- ★ employees can receive:
  - hourly or salary wage
  - vacation pay
  - holiday pay
  - piecework pay
  - overtime pay
- ★ employees can be paid using any combination of pay types (except, hourly cannot receive salary and salary cannot receive hourly)
- ★ special non-taxable or taxable lump sums can be paid regularly or one time (bonus, reimbursements, etc)
- ★ health and welfare deductions can be automatically calculated for each employee
- ★ earnings-to-date are accumulated and added to permanent records; taxes are computed and deducted: US income tax, Social Security tax, state income tax, other deductions (regular or one time)
- ★ paychecks are printed; computer checks with your company letterhead can be purchased from SBSG
- ★ calculations are accumulated for; employee pay history, 941A report, W-2 report, insurance report, absentee report
- ★ fully linked to General Ledger. Each employee's payroll information can be distributed to as many as (12) twelve different GL accounts; system automatically posts to cash account

INVENTORY CONTROL/INVOICING

- ★ **ISAM** (Indexed Sequential Access Method) eliminates the necessity for time consuming sort.
- ★ Pre-Allocated Files for IMMEDIATE update and inquiry capabilities.
- ★ Fast Disk storage and retrieval.
- ★ Inventory Master Record includes...class...SKU...Division...Retail... Cost...Beginning Balance...Period Sale Units...Period Receipts...On Order...On Hand...Minimum Reorder Point...Recommended Reorder Amount...Vendor Number...Period Sale Dollars...YTD Sale Units...YTD Sale Dollars.
- ★ Calculated and Displayed Formulas include...Gross Margin ($)... Gross Margin (%)...Gross Margin ROI (%)...Average Inventory Retail ($)...Average Inventory Cost ($)...Turn-Over (%).
- ★ Reports Generated include...Master File Listing...Class Description Listing...Transaction Audit Trail...Minimum Reorder Point by Vendor...Retail Price List...Retail & Cost Price List...Period Sales Report ...Year to Date Sales Report...Stock Status (Screen or printer output) ...Commission Report (for salesmen and buyers).
- ★ Transaction Types include...Sales, Vendor Receipts...Vendor Orders...Customer Returns...Vendor Returns...Transfer Stock.

GENERAL LEDGER

The General Ledger accounting system consolidates financial data from other accounting subsystems (A/R, A/P, Payroll, direct posting) in an accurate and timely manner. Major reports include the Income Statement and Balance Sheet and a "special" report designed by management. The beauty of this General Ledger system is that it is completely user formatted. You "customize" the account numbers, descriptions, and report formats to suit particular business requirements. These programs were developed 5 years ago for the Wang micro-computer and have been tested in many environments since then. The package has been converted to the TRS-80™ and is now a well documented, on-line, interactive micro-computer system with the capabilities of (or exceeding) many larger systems.

CAPABILITIES:

- ★ more than 200 chart of accounts can be handled
- ★ account number structure is user defined and controlled
- ★ more than 1,750 transactions may be entered via:
  - direct posting; done by hand; validated against the account file before acceptance
  - external posting; generated by A/R, A/P, Payroll or any other user source
- ★ data is maintained and reported by:
  - month
  - quarter
  - year
  - previous three quarters
- ★ reports (samples on back) include:
  - trial balances
  - income statement
  - balance sheet
  - special accounts reports and more.....
- ★ user formats reports with the following designated as you wish:
  - titles
  - headings
  - account numbers
  - descriptions
  - subtotals
  - totals
  - skip lines
  - skip pages
- ★ up to eight levels of totals - fully user designated
- ★ menu driven; easy to use; full screen prompting and cursor control

COMPUTRONICS
MATHEMATICAL APPLICATIONS SERVICE

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

24 HOUR ORDER LINE
(914) 425-1535

PRICING	MOD-1 VERSION	MOD-II VERSION	MOD-III VERSION
ACCOUNTS RECEIVABLE	$125	$225	$199.95
ACCOUNTS PAYABLE	$125	$225	$199.95
GENERAL LEDGER	$125	$225	$199.95
PAYROLL	$125	$225	$199.95
INVENTORY	$175	$275	$199.95
INVOICING	$150	$250	$199.95
COORDINATED INVENTORY/INVOICING ACCOUNTS RECEIVABLE	$449	$749	$599.95
COORDINATED AR-AP-GL	$375	$675	$599.95
COORDINATED AR-AP-GL with PAYROLL	$495	$899	$799.95
EXTENDED COORDINATED AR-AP-GL INVOICING/INVENTORY with PAYROLL	$799	$1299	$1199.95

✓See List of Advertisers on page 322

Why baffle them with Boolean when you can guide them with graphics?

Alternate Course

Michael A. Duffin
1507 East Ave.
Berwyn, IL 60402

A number of years ago I took a course called "An Introduction to Data Processing". It was the dullest, most useless course that I have ever taken.

Most introduction courses to data processing at that particular point in time were for people who worked or were planning to work in data processing. There wasn't a course for the layman. These earlier courses encouraged the fallacy that people who enter data processing must have a strong background in math or science. These courses also tried to touch on every concept of data processing without ever giving the student a chance to use these concepts. Most data processing courses at the time were what I considered GI/GO courses (garbage in/garbage out).

The material was presented by the instructor, memorized by the student and then regurgitated by the student on a test. In retrospect, it is not difficult to understand why many people chose not to continue in data processing after taking an introductory course.

Last summer the Assistant Dean at Morton College, Cicero, IL, asked me if I would be interested in teaching an "Introduction to Data Processing" course which utilized newly acquired TRS-80s. Although my background as a systems programmer qualified me to teach this type of course, I didn't know Basic and I didn't know a TRS-80 from a can opener.

I purchased a TRS-80 and began writing programs to learn the fundamentals of Basic. These could be used as examples in a classroom environment.

A Different Approach

Learning Basic and learning about the TRS-80 was the easy part of the task. There was still the problem of avoiding the pitfalls that would make my class similar to the introductory classes that I had taken. My first step was to talk to other instructors who had used the TRS-80 in this type of course. They mentioned many techniques for tying the concepts of the TRS-80 operating system into other operating systems. However, when it came to teaching Basic, I found that most of them had their students running math, science or business problems. As a former math teacher I knew that many people have a fundamental fear of mathematics and that before some people can show their potential in mathematics they must overcome this fear. Some people have a similar fear of computers; to illustrate this, the following is an episode that occurred shortly after I bought my TRS-80.

A group of people were visiting me one evening and, like anyone with a new toy, I tried to show off my new TRS-80. I showed them a checkbook balancing program and a loan appropriations program I had acquired.

No one seemed very excited. At this point I was quite sure that there was no way that I would ever convince anyone who wasn't predisposed to data processing that it was an acceptable field.

As a last ditch effort I loaded a very simple game that made use of some of the TRS-80's graphics. A complete change came over the group. They crowded around the machine and everyone wanted to play. I didn't get to use my TRS-80 for the rest of the evening, but I did discover a less frightening method for teaching the fundamentals of Basic—*graphics*.

Before the class began I decided upon four objectives that I wanted my class to attain:

- The first objective was to alleviate the fear that many people have about computers.
- Second, I wanted the students to learn only about data processing. I wanted to avoid lengthy discussions concerning principles in math, science or business. This does not mean that I avoided these concepts entirely. Many of these concepts are directly related to data processing, such as number systems (i.e., binary, octal, decimal and hexadecimal) and the Order of Operations used for mathematical interpretations of formulas. However, I wanted to avoid presenting algorithms or formulas for their own sake.

• Third, I wanted to promote creativity and logic and dispel the fallacy that only someone with a mathematical mind can program a computer.

• Fourth, I wanted the students to be able to write simple programs in Basic and finally, I wanted the students to have a rudimentary knowledge of data processing concepts. That is, I wanted the students to be able to explain the fundamentals of a computer to a novice.

We used graphics extensively (although not exclusively). By using graphics, the students could immediately see if something was wrong with their programs and thus could correct the instruction or group of instructions, regardless of whether the student used PRINT@, POKE or Set to create their figures. I provided students with sample problems to illustrate various instructions. I then asked the students to logically deduce how these instructions worked before they were discussed in class. (This not only promoted common sense but in certain cases the deductions were unbelievably creative.)

The students were encouraged to work in teams. Some instructors might argue that this encourages cheating and that one of the students would end up doing all the work. This might be true if every student had a fixed teammate. However, I also encouraged different people to work together at different times. I knew that students working together could communicate concepts to each other more clearly than an instructor could.

Students were required to keep their programs on tape. This gave their work permanence and allowed for the creation of larger, more sophisticated programs. Unfortunately, it also showed them some of the frustrations of programming, such as accidental erasures or machine failure.

The TRS-80 and the Basic instruction set were used to illustrate the general concepts of data processing such as memory, firmware, software, etc.

Space Ship Game

I used Program Listing 1 to show my students that computers can be fun. For most of them, playing with this game was their first hands-on experience with a computer.

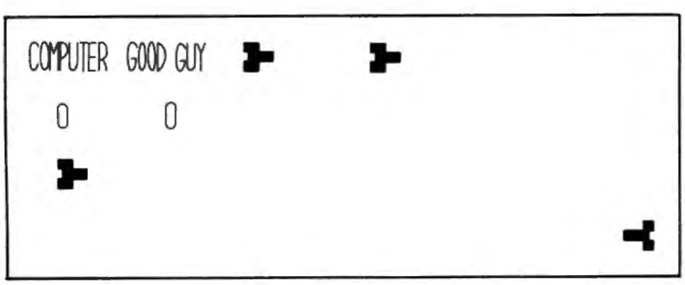
This game utilizes some of the graphics capabilites of the TRS-80. It was designed on a 16K Level II machine. I believe the only modification required to run it on a Level I machine is to replace the single quotation mark with the word Remark.

The object of this game is to shoot enemy ships (controlled by the computer) before they shoot you. The user's ship is illustrated in Fig. 1. The ships the computer controls are illustrated in Fig. 2.

A brief description of the variables used in the program is given in lines 12 through 82.

Fig. 1

Fig. 2



Screen Display of Space Ship Game

When the program starts, directions for the control of the user's ship are given. The first few times you run this game there may not be enough time to read and understand the instructions. Thus, you may want to increase the upper limit in the For...Next loop in lines 140 and 180 to a number larger than 2500. The option that I gave my students was to hit the Break key when instructions came on the screen and then when they were ready, the students were instructed to type in CONT and then hit the Enter key.

When the game begins, three enemy ships appear at the top of the screen and your ship appears somewhere on the right of the screen. The enemy ships are moving and shooting when they appear. The user's ship remains stationary until a number between 1 and 4 is typed. Once the number is entered the ship continues to move in that direction until directed to stop (enter the number 5) or move in another direction (enter a number between 1 and 4). If the user decides to shoot (by hitting the number 6) the shot moves in the same direction the ship is moving.

The direction of the shot is controlled by the value of P in the subroutine between lines 2500 and 2900. The actual shot being fired is controlled in the subroutine between lines 3000 and 3199. This subroutine also prints Hit on the screen if an enemy ship is hit. The POKE instruction is used to print Hit (line 3110) to avoid the automatic upward movement of the screen when a Print or PRINT@ instruction is printed on the bottom line of the screen. This subroutine (lines 3000 to 3199) also increments the counter for the user's hits (D) at line 3100.

Once an enemy ship is destroyed, lines 3144 to 3150 determine how many additional enemy ships will be built. The maximum number of enemy ships is 30 (controlled by the variable L and line 3150).

There are a few things about the program which aren't obvious until you've played it for a while. One thing is that the enemy ships cannot be permanently damaged by the shots of another enemy ship. The enemy ships regenerate any damage caused by their allies. Another thing is that if the user accidentally runs into an enemy ship or comes within the range of their shields (i.e., one space on any side of an enemy ship) the enemy is credited with a hit.

If the user is hit, then his/her ship will again appear somewhere on the right side of the screen moving in the same direction as before it was hit. Since two objects cannot appear in the same space at the same point in time, the enemy is credited with a hit if the user's ship tries to materialize in the space occupied by an enemy ship.

The main body of the program is contained between lines 186 and 350. It makes extensive use of subroutines to control the particulars of the program. Although subroutines slow down the program (especially if there are 30 ships on the screen) maintenance on the program is much easier since a function or group of functions is contained within specific lines of code.

Line 186 contains initial values for a number of variables. Of particular interest is the array W. By specifying additional elements such as W(4) = −64; W(5) = 65; or W(6) = 63 the enemy ships can be given the capability to move in a variety of different directions. The values specified for W on line 186 cause the ship to move to the right—W(0), down—W(1), stay in the same position—W(2) or move to the left—W(3). If additional values such as W(4) (move up), W(5) (diagonal, top to bottom, left to right) or W(6) (diagonal, top to bottom, right to left) are used, then the first part of line 4300, AB = RND(3) will have to be modified. This instruction randomly picks the direction the ship will move and thus the number in the RND instruction must reflect the largest number of array elements.

Line 188 goes to the subroutine that prints the score in the upper left of the screen.

Lines 190 to 194 go to the subroutine to build the enemy (i.e., the computer's) ships. Lines 3250 to 3390 build the computer's ships and also go to the subroutine that builds the user's ship (lines 1000 to 2904).

The variable T determines if a new ship must be built for the user (lines 1000 to 1080). Line 2020 goes to the appropriate section of code to move the ship indicated by the value of INKEY$. The If statements between lines 2100 and 2430 avoid trying to POKE values that are outside of the range of 15360 to 16383. These are the values which reflect the positions on the screen. POKEing outside of this range may cause the program to be wiped out.

Line 230 goes to subroutine 3400 which erases the current enemy ship. Subroutine 4200 (entered from line 232) determines how many positions the ship will move and what direction the ship will move. It also determines if the enemy ship has crashed into the user's ship.

Lines 234 to 280 erase (subroutine 3400) and rebuild (subroutine 3300) the enemy ship. The If statements determine if the enemy ships have reached the end of the screen. When they have, a new ship is started (subroutine 3250) and the process is continued.

Lines 294 to 320 determine if the enemy ship is going to shoot (subroutine 3600). Lines 3600 to 3790 cause the enemy ship to shoot and determine if the user's ship has been hit.

A detailed description of each of the subroutines is included below.

User's Ship Routine

The routine for the user's ship first determines if a new ship must be built based on the value of variable T. If a new ship must be built (i.e., T = 0) then the starting position of the ship is randomly determined by lines 1020 to 1031. The ship is built in line 1060. If the routine is used to move an existing ship (i.e., T = 1), the existing ship is erased first. If a key (1 through 6) has been depressed since the last time this routine was entered then the appropriate move routine is entered via line 2020. If a key has not been depressed, line 2030 will keep the ship moving in the same direction it had been moving.

Lines 2100 through 2430 control the various directions of the user's ship. The If statements used in these lines keep the ship from going outside the boundaries (i.e., 15360 through 16383) of the screen. Lines 2500 through 2890 control the movements of the user's shot. The variable P is based on the last direction the ship was moving (i.e., N), thus the shot moves in the same direction as the ship. Finally, the ship is built and another subroutine is entered to determine if the user's ship has hit an enemy ship.

User's Shot Control

The value of the variable M is set between lines 2500 through 2890. When the routine to control the user's shot is entered, line 3000 first determines that the value is within the range of the screen. Line 3004 checks to see if M is blank. If it is blank then an asterisk is written on the screen and we Return. Line 2530 in the calling subroutine causes this process to be repeated 30 times. If the value of M is not blank then the address of M is checked against the addresses of all of the enemy ships (lines 3020 to 3040). If they don't match, we return to the point of origin. If M does match a value of one of the enemy ships, the following happens. The counter is incremented (line 3100). The enemy ship is blanked (line 3102). Hit is printed and erased (lines 3110 to 3132). A new enemy ship is built to replace the old one (line 3142). From one to three new enemy ships are built (lines 3144 to 3160).

Build Enemy Ship, Determine if User is Hit

A random location at the top of the screen is selected by lines 3260 and 3270. Line 3280 causes the new enemy ship to avoid overwriting the score. The address of the enemy ship is then checked against the value of the users ship. If they are the same then the enemy's score is incremented and T is set to 0 so that a new user's ship will be built (line

✓See List of Advertisers on page 322

3307). Regardless of whether the address of the enemy's ship equals the address of the user's ship, lines 3310 through 3320 print and erase the word Bam. Next, the user's ship is incremented or rebuilt and finally the enemy ship is built (line 3380).

Control Enemy Shots, Increment Score

The enemy ship always shoots from a top to bottom, left to right diagonal (line 3620). It shoots a dash (—) (line 3640). The user's ship shoots an asterisk. Once the enemy has finished shooting and if he hasn't hit anything, then the enemy ship is moved (line 3662). If the user's ship is hit the computer's score is incremented (line 3700) and Hit is printed and erased (lines 3720 and 3750).

Direction of Enemy Ship

Line 4200 sets a value of U between one and 11. This is the amount of spaces that an enemy ship will move. This value is used between lines 234 and 280 in the main part of the program. Then the area surrounding the user's ship is checked. If these are blank then the routine returns to the point of origin. If the area is not blank then the user's ship is blanked (line 4430), Bam is printed and erased, the score is incremented and T is set to zero so that the user's ship will be regenerated.

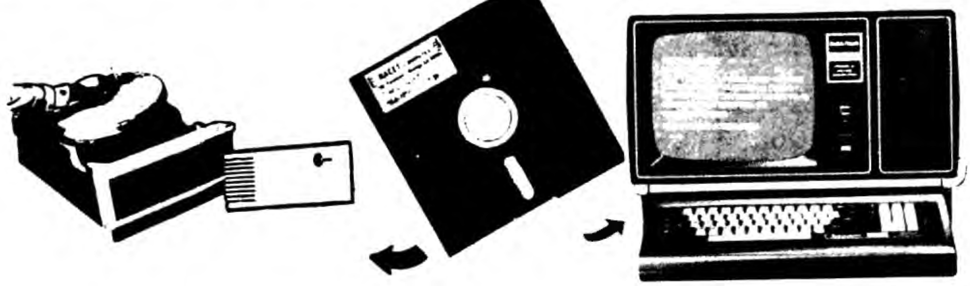
Needless to say, my students were not given such a lengthy explanation of this program at the time. In later classes, however, they did learn many of the fundamentals used in this program.■

Program Listing 1. Space Ship Game

```
2 '*******************************************************
    *'
4 '*'
6 '*    VIDEO SPACE SHIP GAME '
8 '*'
10 '******************************************************
    *****'
12 '          LIST OF VARIABLES USED '
14 ' A= COUNTER IN FOR NEXT LOOPS'
15 ' AB= RANDOM NUMBER USED TO DESCRIBE NEXT ARRAY ELEM
    ENT'
16 ' B= RANDOM VALUE FOR COMPUTER SHOT'
18 ' CD= COUNTER IN FOR NEXT LOOP'
20 ' D= NUMBER OF HITS BY GOOD GUY'
22 ' E= NUMBER OF HITS BY COMPUTER '
24 ' F= LOCATION WHERE HIT IS PRINTED'
26 ' G= '
28 ' H= PRINTS AND ERASES "BAM" 3 TIMES'
30 ' I= USED TO DETERMINE RANDOM STARTING POSITION OF G
    OOD GUYS SHIP'
31 ' IN= DIRECTION OF CURRENT ENEMY SHIP'
32 ' J= INDEX FOR COMPUTERS SHIPS'
34 ' K= POSITION OF COMPUTERS SHOT INDEXED BY J'
36 ' L= NUMBER OF COMPUTERS SHIPS--MAX OF 10'
38 ' M= POSITION OF GOOD GUYS SHOT'
40 ' N= LAST DIRECTION OF GOOD GUYS SHIP'
42 '     1= UP'
44 '     2=DOWN'
46 '     3=LEFT'
48 '     4=RIGHT'
50 '     5=STOP'
52 '     6=SHOOT'
54 ' O='
56 ' P= DIRECTION OF SHOT : INITIALLY DETERMINED BY N'
58 ' Q= COUNTER FOR HIT'
60 '     RANDOM NUMBER OF NEW ENEMY SHIPS--MAX =3'
62 ' R= USED TO CHECK IF 2 SHIPS HIT'
64 ' S='
66 ' T= INDICATES IF NEW GOOD GUY SHIP MUST BE BUILT'
68 '     0=YES : 1=NO'
70 ' U=  RANDOM NUMBER-NUMBER OF TIMES ENEMY SHIP WILL
    MOVE'
72 ' V= LOCATION WHERE BAM IS PRINTED'
74 ' W= ARRAY USED TO DETERMINE DIRECTION OF ENEMY SHIP
    '
76 ' X= POSITION OF COMPUTERS SHIP--INDEXED BY J'
78 ' Y= POSITION OF GOOD GUYS SHIP'
80 ' Z='
82 ' Y$= SAME VALUES AS N: INITIAL VALUE FROM INKEY$'
100 '*** ROUTINE TO WRITE MESSAGES AT START OF GAME ***
    '
105 CLS:PRINT:PRINT CHR$(23); "    INSTRUCTIONS"
110 PRINT "HIT 1 TO MOVE UP"
115 PRINT "HIT 2 TO MOVE DOWN"
120 PRINT "HIT 3 TO MOVE LEFT"
125 PRINT "HIT 4 TO MOVE RIGHT"
130 PRINT "HIT 5 TO STOP"
```

Program continues

RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes —



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

RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes —

RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes — RACET
— RACET computes — RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes —

Program continued

```
135 PRINT "HIT 6 TO SHOOT"
140 FOR I=1 TO 2500:NEXT I
145 CLS:PRINT:PRINT CHR$(23); "    INSTRUCTIONS"
150 PRINT "3 TO 30 ENEMY SHIPS WILL APPEAR"
155 PRINT "YOU MUST SHOOT THEM BEFORE"
160 PRINT "   THEY SHOOT YOU."
165 PRINT
170 PRINT "WHEN YOU SHOOT, YOUR SHOT WILL"
172 PRINT "  MOVE IN THE SAME DIRECTION"
174 PRINT "  YOU ARE MOVING."
176 PRINT "IF YOU ARE STOPPED, THE SHOT"
178 PRINT "  WILL GO UP."
180 FOR I=1 TO 2500: NEXT I
182 CLS
184 '*** INITIALIZTION ROUTINE ***'
185 ONERROR GOTO 10000
186 W(3)= -1:W(0)=1:W(1)=64:W(2)=0:IN=0:U=0:DIM K(32) :
     DIM X(32) :J=1 : L=3 : D=0 : M=0 : N=5 : T=0 : E=0
188 GOSUB 3800
190 FOR J=1 TO L
192 GOSUB 3250 : 'ROUTINE FOR ENEMY SHIP'
194 NEXT J
200 '*** ROUTINE TO MOVE COMPUTERS SHIPS ***'
210 GOSUB 3800
220 FOR J=1 TO L
230 GOSUB 3400 : 'ROUTINE TO ERASE LAST POSITION OF SHI
     P'
232 GOSUB 4200
233 IF IN=0 THEN U=1
234  FOR CD=1 TO U
235 GOSUB 3400
240 X(J)=X(J)+IN:'MOVE SHIP 1 DOWN AND 1 TO LEFT'
242 IF X(J) < 15360 THEN GOSUB 3250
250 'SHIP OFF SCREEN CREATE A NEW SHIP'
260 IF X(J) >= 16383 THEN GOSUB 3250 : GOTO 280
270 GOSUB 3300 : 'INCREMENT SHIP POSITION '
280 NEXT CD
294 IF E<D+2 THEN GOTO 310
296 B=RND(5) : 'ROUTINE TO INITIALIZE SHOOTING'
300 IF B <> 3 THEN GOTO 330
310 K(J) = X(J)+3
320 GOSUB 3600
330 NEXT J
340 GOSUB 1000
350 GOTO 200
1000 '*** ROUTINE FOR GOOD SHIP ***'
1010 IF T=1 THEN GOTO  2000
1020 I=RND(14) : I=I+1
1030 Y=63*I  +  (I-1):'SHIP STARTS AT RIGHT OF SCREEN'
1031 Y=Y+15360
1050 IF Y-2 <= 15360 THEN Y=16383
1056  T=1
1060 POKE Y.183 : POKE Y-1,140 : POKE Y-2,140:'BUILDS S
     HIP'
1070 FOR A=1 TO 100: NEXT A
2000 'ROUTINE TO MOVE GOOD GUY SHIP'
2005 POKE Y,32 : POKE Y-1,32 : POKE Y-2,32
2010 Y$=INKEY$
2020 ON VAL(Y$) GOTO 2100,2200,2300,2400,2890,2500
2030 ON N GOTO 2100,2200,2300,2400,2900,2500
2040 GOTO 2600
2100 '*** MOVE UP ***'
2105 Y=Y-64 : N=1
2110 IF Y<15360 THEN Y=Y+1023
2115 IF Y=15360 THEN Y=16383
2120 GOTO 2900
2200 ' *** MOVE DOWN ***'
2205 Y=Y+64 : N=2
2210 IF Y=16383 THEN Y=15360+192
2212 IF Y>16383 THEN Y=Y-1023
2214 IF Y<15378 THEN Y=Y+192
2220 GOTO 2900
2300 '*** MOVE LEFT ***'
2305 Y=Y-1 : N=3
2310 IF Y-2< 15360 THEN Y=16383
2320 GOTO 2900
2400 '*** MOVE RIGHT ***'
2410 Y=Y+1 : N=4
2420 IF Y>16383 THEN Y=15360+192
2430 GOTO 2900
2500 P=N : 'SHOT FIRED IN SAME DIRECTION AS MOVEMENT'
2504 G=5
2510 IF P=5 THEN P=1 : 'SHOOT UP IF NOT MOVING'
2520 M=Y-3 : 'NEW SHOT'
2530 FOR A=1 TO 30
2540 POKE M,32
2600 'DUMMY STATEMENT'
2610 ON P GOTO 2620,2640,2660,2680
2615 STOP
2620 '*** SHOOT UP ***'
2626 M=M-64
2632 GOTO 2800
2640 ' *** SHOOT DOWN ***'
2646 M=M+64
2652 GOTO 2800
2660 '*** SHOOT LEFT ***'
2666 M=M-1
2672 GOTO 2800
2680 '*** SHOOT RIGHT ***'
```

Program continues

Program continued

```
2685 IF M=Y-3 THEN M=Y
2686 M=M+1
2800 GOSUB 3000
2810 POKE M,32
2830 NEXT A
2840 M=0
2842  GOTO 2900
2890 N=5
2900 IF Y-2 <= 15360 THEN Y=16383
2902 POKE Y,183: POKE Y-1,140 : POKE Y-2,140
2903 GOSUB 4400
2904 RETURN
3000 IF M<15360 OR M > 16383 THEN M=0: GOTO 3199
3004 IF PEEK(M)=32 THEN POKE M,42 : GOTO 3199
3010 POKE M,42
3020 FOR J=1 TO L
3030 IF M=X(J) OR M=X(J)+1 OR M=X(J)+2 THEN GOTO 3100
3040 NEXT J
3050 GOTO 3199
3100 D=D+1 : GOSUB 3800
3102 GOSUB 3400
3104 'PRINTS "HIT"'
3110  POKE X(J),72:POKE X(J)+1,73:POKE X(J)+2,84
3130 FOR Q=1  TO 100:NEXT Q
3132 'BLANKS "HIT"'
3134 POKE X(J),32:POKE X(J)+1,32:POKE X(J)+2,32
3140 M=0
3142 GOSUB 3250
3144 Q=RND(2)
3146 Q=Q+1
3148 J=L+1 : L=L+Q
3150 IF L+1 >= 31 THEN L=30
3152 FOR A=J TO  L
3154 GOSUB  3250 :   J=J+1
3160 NEXT A
3170 GOSUB 3400
3199 RETURN
3250 '*** ROUTINE TO BUILD ENEMY SHIP ***'
3252 '***   ALSO DETERMINES IF SHIPS HIT EACH OTHER ***
     '
3260 I=RND(61)
3270 X(J)=15360+I
3280 IF X(J)<= 15377 THEN X(J)=X(J)+192
3300 R=X(J)
3302 FOR H=1 TO 3
3303 C=PEEK(R)
3304 IF C=32 THEN GOTO 3360
3305 IF C=32 THEN GOTO 3360
3307 IF R=Y-63 OR R=Y-64 OR R=Y-65 OR  R=Y+1 OR R=Y-3 O
     R  R=Y OR R=Y-1 OR R=Y-2 THEN E=E+1: GOSUB 3800: T
     =0
3308 IF R+3>16383 THEN R=16380
3309 'PRINTS "BAM"'
3310 POKE R,66:POKE R+1,65:POKE R+2,77
3312 FOR B=1 TO 50 : NEXT B
3318 'ERASES "BAM"'
3320 POKE R,32:POKE R+1,32:POKE R+2,32
3360 R=R+1
3370 NEXT H
3372 GOSUB 1000
3380 POKE X(J),187 : POKE X(J)+1,157 : POKE X(J)+2,140
3390 RETURN
3394 '***  BLANKS OUT SHIP ***'
3400 POKE X(J),32 : POKE X(J)+1,32 : POKE X(J)+2,32
3410 RETURN
3600 FOR A=1 TO 47
3610 POKE K(J),32
3620 K(J)=K(J)+65
3630 IF K(J)  >= 16383 THEN K(J)=0 : GOTO 3790
3640 POKE K(J),45
3650 IF K(J)=Y OR K(J)=Y-63 OR K(J)=Y-64 OR K(J)=Y-65 O
     R K(J)=Y OR K(J)=Y-1 OR K(J)=Y-2 THEN GOTO 3700
3660 NEXT A
3662 GOSUB 2000
3670 GOTO 3790
3700 E=E+1 : GOSUB 3800
3710 T=0
3720 'PRINTS "HIT"'
3730 POKE Y-2,72:POKE Y-1,73:POKE Y,84
3740 FOR Q=1 TO 100 : NEXT Q
3742 'BLANKS "HIT"'
3750 POKE Y,32:POKE Y-1,32:POKE Y-2,32
3790 RETURN
3800 PRINT@ 0,"COMPUTER  GOOD GUY"
3810 PRINT@ 131,E : PRINT@ 142,D
3820 RETURN
4000 GOTO 200
4200 U=RND(10) : U=U+1
4300 AB=RND(3) : IN = W(AB)
4400 IF Y=16383 THEN Y=15483
4402 IF Y<15423 OR Y>=16319  GOTO 4499
4404 IF PEEK(Y-3)=32 AND PEEK(Y+1)=32 AND PEEK(Y+64)=32
      AND PEEK(Y+63)=32 AND PEEK(Y+65)=32  AND PEEK(Y-6
     5)=32  AND PEEK(Y-64)=32 AND PEEK(Y-63)=32 THEN GO
     TO 4499
4420 GOSUB 3400
4430 POKE Y,32 : POKE Y-1,32 : POKE Y-2,32
4432 'PRINTS "BAM"'
4440 POKE Y-2,72:POKE Y-1,73:POKE Y,84
4450 FOR B=1 TO 100 : NEXT B
4452 IF T=0 THEN GOTO 4470
```

Turn your '80 into a table top teacher.

Quiz

Fred Olsen
4153 Arey Drive
San Diego, CA 92154

Education and training is a microcomputer application that each of us can enjoy. Quiz will help you realize the great potential as a teaching tool that is stored within your machine and should provide a few ideas for putting the teacher to work.

Teachers use flash cards to review a great range of subject material. A problem is displayed to students on the front of a card while the teacher sees the answer printed on the back of the card. The teacher provides reinforcement based upon student response to the problem. Flash cards are also highly effective as self-teachers when used in private study.

Flash cards led to development of a training device known as the tabletop teacher, a pinball type of machine with flashing lights, bells and buzzers. Questions with multiple-choice or true/false answers were prepared on cards. The correct answer was encoded on the back or edge of the card for interpretation by the machine. The device displayed a card and timed the wait for student response. The response was immediately graded. As each item was scored, the card was mechanically replaced by another from the supply of question cards stored within the device.

The New Tabletop Teacher

Programming the TRS-80 to perform the tasks of the tabletop teacher starts with a review of the previous description. We find that we will require means to prepare and store decks of question and answer cards, and present the prepared items and score responses.

The screen provides the page on which to prepare and edit each item for ultimate transfer to storage. Maximum item length is 1K (a full screen). The requirement will be to maintain a file of items much beyond the capacity of even the maximum TRS-80 memory expansion. All of us have cassette capability providing unlimited storage capacity. Long term storage requires tape utilization anyway, so let's look at the feasibility of tape as our storage medium for the decks of cards.

Tape storage and retrieval does not allow random access and is quite slow. Random access would be necessary if there were a requirement to shuffle the decks, but we can probably live without that feature. Cassette-oriented Basic doesn't seem to be capable of delivering 1K of data in a timely manner. The greatest time expense is the block leader required at 255-byte (maximum) intervals. (Leaders are time intervals introduced to allow the tape mechanism to reach operating speed.) If items are limited to 255 characters (one-quarter screen) we have no problem. However, one-quarter screen is not sufficient for this application. I therefore abandon Basic during the tape handling processes.

Brief assembly language procedures can move data directly from display memory to tape and from tape back to display memory in reasonable time. A block leader is still required, of course, but the block length is unlimited so one such delay will serve a full screen.

The Program

The Basic program and assembly language procedures presented here provide only the necessities to generate item decks and to administer the quiz. Notice that the machine language is provided in data form and is loaded into high memory by the Basic program. USR instructions provide the linkage between the procedures. The program listings are thoroughly remarked to ease understanding of the program operation. Asterisk, pound sign and up arrow are used to communicate directly with the program and therefore must not appear on the screen except at the point where the program is to take control. An asterisk indicates that the next character to be entered is the item answer. A pound sign marks the last char-

Program Listing 1

```
7531          00100          ORG     7531H
              00110 ;17 FEB 1980 - RECORD SCREEN - DEVELOPED FOR QUIZ
7531 3E00     00120 SAVE     LD      A,0               ;CASSETTE
7533 CD1202   00130          CALL    0212H             ;DEFINE DRIVE
7536 CD8702   00140          CALL    0287H             ;WRITE LEADER
7539 01003C   00150          LD      BC,3C00H          ;SET START POINT
753C 0A       00160 LOOP1    LD      A,(BC)            ;GET BYTE TO RECORD
753D CD6402   00170          CALL    0264H             ;RECORD BYTE
7540 03       00180          INC     BC                ;STEP TO NEXT BYTE
7541 FE23     00190          CP      23H               ;WAS IT #?
7543 C23C75   00200          JP      NZ,LOOP1          ;NO, LOOP
7546 CDF801   00210 EXIT     CALL    01F8H             ;YES, STOP RECORDER
7549 C39A0A   00220          JP      0A9AH             ;RETURN
754C 00       00230          NOP
754D 00       00240          NOP
754E 3E00     00250 READ     LD      A,0               ;CASSETTE
7550 CD1202   00260          CALL    0212H             ;DEFINE DRIVE
7553 CD9602   00270          CALL    0296H             ;FIND SYNC BYTE
7556 21003C   00280          LD      HL,3C00H          ;SET START POINT
7559 CD3502   00290 LOOP2    CALL    0235H             ;READ BYTE (DATA)
755C FE2A     00300          CP      2AH               ;IS IT *?
755E C26D75   00310          JP      NZ,DISP           ;NO, DISPLAY IT
7561 CD3502   00320          CALL    0235H             ;READ BYTE (ANSWER)
7564 6F       00330          LD      L,A               ;ANSWER INTO L
7565 2600     00340          LD      H,0               ;CLEAR H
7567 CD3502   00350          CALL    0235H             ;READ BYTE (#)
756A C34675   00360          JP      EXIT              ;BACK TO BASIC
756D 77       00370 DISP     LD      (HL),A            ;DISPLAY BYTE
756E 23       00380          INC     HL                ;INCR DISPLAY POINTER
756F C35975   00390          JP      LOOP2             ;NEXT BYTE
0000          00400          END
```

acter to be processed to or from tape. An up arrow (unshifted) tells the Basic program to end operator input and begin processing the screen memory onto tape. A triple pound sign is called for in the operating instructions primarily to cause the operator to glance at the screen one last time before handing control to the program by depressing the up arrow.

Operator instructions are provided on the screen at the start of each of the two segments which are menu selected. Quiz keeps the cursor on the screen during the preparation segment to help in editing the item. The item number that appears on the screen is for convenience only and can be erased, replaced, or relocated. The tab key (right arrow) is programmed to provide five spaces—a convenient indentation for formatting. Up arrow (shifted) and down arrow are handy to position the cursor. Down arrow (unshifted) performs the same as Enter (carriage return, line feed, erase line). Shifted up, down, right and left arrows move the cursor one space or line without erasing the character at the new position. The best way to learn to use the editing capability is to spend a few minutes driving the cursor around the screen entering or altering data.

Using Quiz

The sample item shown in Fig. 1 is in a typical format and shows the use of a title line, repositioned item number and an application. Prepare your own items from text material or use problems presented in a text.

Essay format can be handled by preparing the question as a note page. Note pages are marked by a space in the terminator where the program expects to find the correct response (*###[). The next page might then be prepared as an essay response and allow for self-scoring. The note page is also useful in presenting explanations and solutions. Consider using note pages to store text material. Note pages do not count as items in the programmed 10-item procedure, so you will be able to handle a lot of text material, the limit being the length of tape. When such material is later processed by the Read procedure, pages are advanced by simply depressing the space bar.

California Driver License Review

1. When two automobiles meet on a narrow mountain road, the right-of-way belongs to:

A. The Automobile Going Downhill.
B. The Automobile Going Uphill.
C. The Larger Automobile.
D. The Larger Driver. *B###[

Fig. 1. Sample Quiz Item Format

Challenges

Quiz, as stated earlier, is intended only as a starting point. There is plenty of room for improvement and refinement. Keep in mind that changes must be made in such a manner so that previously prepared decks will be compatible. All that really means is that you must not change the terminator format. The following improvement ideas have occurred to me and are planned for future versions.

- Variable length quizzes.
- Print correct response.
- Varying point weights.
- Timed response.
- Student name and score file.
- Double buffer input to accelerate next item display.
- Provide for large character display.
- Graphics.
- Link Quiz with other programs to save program-generated displays.
- Repeat key.■

Program Listing 2

```
100 ' Q U I Z -- MARCH 198Ø -- W6PNT
11Ø '***** UP ARROW APPEARS AS LEFT BRACKET IN THIS LISTING
12Ø ' MOVE THE MACHINE LANGUAGE PROGRAMS INTO HIGH MEMORY
13Ø FOR N=3ØØØØ TO 3ØØ69                                       'N IS DESTINATION
14Ø READ M                                                     'GET A BYTE
15Ø POKE N,M                                                   'STORE BYTE AT N
16Ø NEXT N                                                     'MOVE 7Ø BYTES
17Ø CLS : PRINT"                    ******  Q U I Z  ******
18Ø PRINT:PRINT"MAKE A SELECTION FROM THE FOLLOWING:"
19Ø PRINT:PRINT"          1  - WRITE A QUIZ
2ØØ PRINT:PRINT"          2  - TAKE A QUIZ
21Ø PRINT:PRINT"ENTER 1 OR 2
22Ø A$=INKEY$:IF A$="" THEN 22Ø
23Ø IF A$="2" GOTO 55Ø
24Ø IF A$<>"1"PRINT"1 OR 2 PLEASE !!!":GOTO22Ø
25Ø CLS : PRINT"YOU HAVE CHOSEN TO WRITE A QUIZ.
26Ø PRINT:PRINT"1.  THE SCREEN WILL CLEAR
27Ø PRINT"2.  PREPARE THE SCREEN AS YOU WISH.
28Ø PRINT"3.  TAB (RIGHT ARROW) SPACES FIVE PLACES.
29Ø PRINT"4.  * AND [ ARE RESTRICTED TO PROGRAM CONTROL.
3ØØ PRINT"5.  END AN ITEM WITH: *X###[ .  THE CORRECT RESPONSE
31Ø PRINT"    GOES AT 'X' (USUALLY A,B,C,D,T, OR F)
32Ø PRINT"6.  A PAGE OF NOTES TERMINATES WITH: * ###[ (X=BLANK)
33Ø PRINT"7.  THE PROGRAM WILL LOOP UNTIL YOU HAVE ENTERED TEN
34Ø PRINT"    ITEMS.  PAGES OF NOTES DO NOT COUNT AS ITEMS.
35Ø PRINT:PRINT"DEPRESS 'ENTER' WHEN READY
36Ø A$=INKEY$:IFA$=""THEN 36Ø                                  'WAIT
37Ø CLS : PRINT"PREPARE TAPE RECORDER. WRITE DOWN START POINT.
38Ø PRINT:PRINT"WHEN SET TO RECORD, DEPRESS 'ENTER'.
39Ø A$=INKEY$:IF A$=""THEN 39Ø                                 'WAIT
4ØØ CLS : PRINTCHR$(14)                                        'CURSOR ON
41Ø FOR N=1 TO 1Ø
                                              'ITEM COUNTER
42Ø PRINTN;".  ";                                              'PRINT ITEM #
43Ø A$=INKEY$:IF A$=""THEN 43Ø                                 'WAIT
44Ø IF ASC(A$)=9 A$="     "                                    'R ARROW,5 SPACES
45Ø IF K=1 AND A$=" "N=N-1                                     'NOTE IS NOT ITEM
46Ø IF A$="*" K=1                                              'SET NOTE FLAG
47Ø PRINTA$;                                                   'DISPLAY CHARACTER
48Ø IF A$<>"["THEN43Ø                                          '[=RECORD DISPLAY
49Ø POKE 16526,49 : POKE 16527,117 : X=USR(Ø)                  'CALL TAPE WRITE
5ØØ CLS : K=Ø                                                  'RESET NOTE FLAG
51Ø NEXT N                                                     'NEXT ITEM
52Ø PRINT"TEN ITEMS RECORDED.  WRITE DOWN THE TAPE COUNTER.
53Ø PRINT:PRINT"DEPRESS 'ENTER' WHEN READY.
54Ø A$=INKEY$:IF A$=""THEN 54ØELSE CLS:GOTO18Ø                 'WAIT & RESTART
55Ø CLS : PRINT"THIS IS THE QUIZ APPLICATION  --
56Ø PRINT:PRINT"SET UP THE TAPE RECORDER TO PLAY THE QUIZ.
57Ø PRINT:PRINT"     DEPRESS 'ENTER' WHEN READY.
58Ø A$=INKEY$:IF A$="" THEN 58Ø
59Ø K = Ø                                                      'CLEAR THE SCORE
6ØØ FOR N = 1 TO 1Ø                                            'PROCESS 1Ø ITEMS
61Ø CLS
62Ø POKE 16526,78 : POKE 16527,117 : X=USR(Ø)                  'CALL TAPE READ
63Ø A$=INKEY$:IF A$="" THEN 63Ø                                'WAIT (ANS IN X)
64Ø IF X=32 N=N-1 :GOTO 67Ø                                    'SPACE IS NOTES
65Ø IFASC(A$)=XPRINT@896,"CORRECT":K=K+1ELSEPRINT@896,"WRONG" 'SCORE RESPONSE
66Ø FOR M=1 TO 5ØØ: NEXT M                                     'HOLD ITEM SCORE
67Ø NEXT N                                                     'LOOP (DISPLAY X?)
68Ø CLS : PRINT:PRINT"    END QUIZ  "
69Ø PRINT:PRINT" YOUR SCORE IS";1Ø*K; " PER CENT":GOTO18Ø
7ØØ DATA 255,62,Ø,2Ø5,18,2,2Ø5,135,2,1,Ø,6Ø,1Ø,2Ø5,1ØØ,2,3,254
71Ø DATA 35,194,6Ø,117,2Ø5,248,1,195,154,1Ø,Ø,Ø,62,Ø,2Ø5,18,2
72Ø DATA 2Ø5,15Ø,2,33,Ø,6Ø,2Ø5,53,2,254,42,194,1Ø9,117,2Ø5,53,2
73Ø DATA111,38,Ø,2Ø5,53,2,195,7Ø,117,119,35,195,89,117,Ø,Ø,Ø,Ø
74Ø END
```

Yes, You Can Increase Your Programming Productivity

SNAPP II EXTENDED BASIC

SNAPP II EXTENDED BASIC A family of enhancements to the Model II BASIC interpreter. Part of the package originated with the best of APPARAT, INC.'s thoughts in implementing NEWDOS BASIC. The system is written entirely in machine language for SUPER FAST execution. The extensions are fully integrated into Model II BASIC, and require NO user Memory, and NO user disk space. The package is made up of the following six modules.

XBASIC—Six single key stroke commands to list the first, last, previous, next or current program line, or to edit the current line. Includes quick way to recover BASIC program following a NEW or system or accidental re-boot. Ten single character abbreviations for frequently used commands: AUTO, CLS, DELETE, EDIT, KILL, LIST, MERGE, NEW, LLIST, and SYSTEM.

XREF—A powerful cross-reference facility with output to display and/or printer. Trace a variable through the code. Determine easily if a variable is in use.

XDUMP—Permits the programmer to display and/or print the value of any or all program variables. Identifies the variable type for all variables. Each element of any array is listed separately.

XRENUM—An enhanced program line renumbering facility which allows specification of an upper limit of the block of lines to be renumbered, supports relocation of renumbered blocks of code, and supports duplication of blocks of code.

XFIND—A cross reference facility for key words and character strings, also includes global replacement of keywords.

XCOMPRESS—Compress your BASIC programs to an absolute minimum. Removes extraneous information; merge lines; even deletes statements which could not be executed. Typically saves 30-40% space even for programs without our REM statements! Also results in 7-10% improvement in execution speed.

ENTIRE PACKAGE FOR MODEL II $200
ENTIRE PACKAGE FOR MODEL III $125

SNAPP SCREEN

Another dependable product from Snapp, Inc. Using rigid plastic, Snapp, Inc. designed the ultimate green screen at the lowest price. The Snapp Screen installs in a Snapp with the use of pressure sensitive tape. Reduce user fatigue with the Snapp Screen.

DOSFIX

$15

A collection of patches to TRSDOS and BASIC to enhance their usability and function. Includes our well known BREAK7E patches and facilities to disable and verify detect which will increase average disk speed by 30%. Free with any Model II Software Package. Purchased separately. $15

SUPERSNAPP X

You can sort fast using your present facilities, or you can do it faster with Racets' superb facilities, or you can use the fastest: SUPERSNAPP X. The heart of SUPERSNAPP X is a SUPER FAST in-memory sort routine that has been benchmarked against everything on the market and beats them all . . . hands down.
SUPERSNAPP X is the most important component of SNAPP X EXTENDED BUILTIN FUNCTIONS which is a much needed set of additions to the Model II BASIC interpreter that will put time saving power at your fingertips. Let's compare (using random data) SUPERSNAPP X and Racet's GSF SORT for speed:

SORT	SUPERSNAPP X	RACET GSF
10,000 integers	39 seconds	59 seconds
5,000 Singles	22 seconds	34 seconds
2,000 Strings	10 seconds	15 seconds

SUPERSNAPP X is guaranteed to be the FASTEST in memory SORT on the market or your money back. With it you also get these EXTENDED BUILTIN FUNCTIONS: PEEK, PEEKW, POKE, POKEW, XDATS, XTIMS, ETIMS, FILES, AND THE SPECIAL SCMD (SNAPP__COMMAND). PLUS: open "E" Set SCROLL PROTECTION. ERASE all ARRAYS in one command. Specify size and Blink rate of CURSOR. LONG ERROR MESSAGES. Read from Video Screen Read. PEEK complete strings from memory. POKE complete strings to memory. convert upper case to lower case and vice-versa, turn complete screen off and on at will, extract largest or smallest values from user supplied list of numbers
We consider this next function as probably the most powerful addition ever made to Microsoft BASIC. **PRINT USING INTO A STRING.** The ability to arrange data into a string variable should perk your imagination.

Model II $100.00
Model III $75.00

SPOOLER Model I, Model II and Model III

Our workhorse! Unlike the one supplied with TRSDOS 2.0, ours requires no special knowledge or training on the part of the operator. Additionally, ours performs much better. On the Tandy SPOOLER, every time a disk is accessed, the printer stops dead! This package is available for Model I, in the TRSDOS/NEWDOS 80 versions, or for the Model II or Model III. Greatly enhances system performance when running typical business applications. Many applications have been benchmarked to run nearly TWICE AS FAST with the SPOOLER installed. Installs in minutes and no changes are required to your programs. The Model II version requires NO user memory. $100

SORTDIR

Locate directory entries with ease. SORTDIR will rearrange your directory entries in alphameric sequence on either a TRSDOS operating system or data diskette. The routine performs its function in seconds. No more wasted time hunting for your entries. Another time saver from SNAPPWARE. 30.00

XSCREEN

Supports the copying of the full video screen to the printer. Can be invoked by the operator with a keystroke, or from your program with a USR call. Requires NO user memory $30

AUTOFILE

Another big time saver from Snapp, Inc. Autofile offers all programmers much needed time saving tools to use when working with direct files. Eliminate tedious commands such as LSET's, MKIS's, MKDS's, MKSS's, CVD's, CVI's and CVS's. Even the FIELD statement has been eliminated. You won't have to guess as to where the FIELDed variable is. The ASC and CHR$ function references will be performed automatically.

MODEL II $75
MODEL III $60

AUTOMAP

Save time creating a formated screen with our Extended Basic Mapping Support system. The programmers task of communicating and displaying information with the user operator is completely automated with simple SEND and RECEIVE statement commands. Many different types of information can be sent and received from the operators with the use of a single command. Many hours of programming time are saved and cost efficiency is truly realized using Automap.

MODEL II $100
MODEL III $75

DOTWO

Allows programmer to build a list of any commands. All languages are supported. With DOTWO a programmer is now allowed to add any new commands of any kind and not be limited to subset restrictions. Anything that can be typed into the keyboard makes an acceptable command with DOTWO files. A programmer can virtually build an unlimited series of commands and not be limited to just TRSDOS commands. Included at no charge: BUILD/BAS which gives a programmer an easier method of creating and maintaining DOTWO or DO files. $50

MASTER / SLAVE

This software package was designed to support the transferring of files from one Model II to another, via direct connection or modem/phone line connection. ALL kinds of files, and baud rates up to 9600 are fully supported. Transfer files in either direction, even with the SLAVE Model II UNATTENDED! $100

HOSTII / TERMII

Allows remote control of a Model II from another Model II, or any ASCII terminal. Our Host system, unlike the one supplied with TRSDOS 2.0, supports accurate screen positioning on the Term station. Without this feature, formated displays appear on the screen looking like randomly placed garbage. Requires NO user memory! This system is designed to provide software support to our customer locations without ever leaving the office. Custom versions are now available for most nationally distributed terminals as a $25.00 option. Call for details. $50

Trace your roots by computer.

Family Relationships

Sil Horwitz
152 Mill Run Drive
Lake Mary, FL 32746

My wife is a genealogy fiend who finds the research fascinating. She usually considers my work with the computer in the same category as playing with model trains, but one day she brought me a card and asked, "Could this be programmed into that thing of yours?"

The card was a pyramid of sorts, which showed the relationship of one person to another from a common ancestor (progenitor). Following the lines of relationship was driving her crazy. Not knowing what I was getting into, and wanting to make the point that my TRS-80 was not a toy, I agreed to write a program so she wouldn't have to follow the lines down a card. It looked like a simple job. It wasn't.

The Program Listing follows the official designations for relatives to the sixth generation. (If you need more than that, the program is easily modified. My wife advises that more are rarely needed.) This program has made her life easy; she can now easily determine the relationship of a second cousin once removed to her great uncle!

The Program Once Removed

Lines 40 and 600 POKE in and out of video and hard copy printout (when requested) so the commands appear on the screen instead of messing up your printing. Lines 220 through 380 provide graphics to make it easier to visualize the generations from the common progenitor. Instructions are given in lines 400 through 560; these were made very concise, to fit on one page. Lines 550, 555 and 590 are traps to prevent the entry of unacceptable data (it does not trap incorrect data, such as calling a man F or putting a person in the wrong generation). Lines 640 to 2640 include many GOTOs for the branching required to show the relationships.

How to Find Old Aunt Tillie

It is most important that the proper common progenitor be selected. As an example, if you want to show the relationship of your uncle's children to each other, the common progenitor is your uncle. But, if you want to show the relationship of your uncle's child, Joan, to your daughter's child, John, the common progenitor is your grandfather. (Logic: Grampaw = 0, your father and your uncle both = 1, you and your uncle's child = 2, your daughter = 3, and your daughter's child = 4—Joan is John's first cousin twice removed.)

If you start at zero with the wrong person, every relationship will be wrong. There must be a direct lineage, and the common progenitor must be the first one you come to, working backwards from the relatives involved.

Fig. 1 shows the screen display for the example given above. If you don't want this to appear each time, change ELSE220 in line 50 to ELSE540.

As written, the program uses 4K bytes, with another 1K required for string manipulation. You could save enough space to get this into a 4K machine by eliminating the REMs, instructions (lines 420 to 520), and introductory material (lines 30, 100 to 110, 3000 to 5000). You could also modify the program for Level I by changing to a Set routine for the graphics in line 220, changing the @ to AT, using numerics instead of strings for the input (except A$ and B$), and eliminating line 130 and the POKE statements, as only the screen will be used. But it will work, and on a 4K machine, too!

And, unlike human genealogists, the computer won't go crazy trying to figure the relationships! ■

```
            ## 1 ####### 2 ####### 3 ####### 4 ####### 5 ####### 6
  0 #######
            ## 1 ####### 2 ####### 3 ####### 4 ####### 5 ####### 6
             GENERATIONS FROM COMMON PROGENITOR

TO ESTABLISH RELATIONSHIP, CONSIDER GENERATION OF FIRST INDIVIDUAL, AS
COMPARED WITH SECOND, USING NUMBERS TO INDICATE GENERATION FROM
NEAREST COMMON PROGENITOR. COMMON PROGENITOR IS CONSIDERED '0'—
COUNT FROM THERE.

AS REQUESTED, ENTER NAME, SEX (M/F), GENERATION #—IN THAT ORDER, WITH
EACH SEPARATED BY A COMMA AS SHOWN.

    ENTER NAME, SEX, GEN # OF FIRST INDIVIDUAL? JOAN,F,2
    ENTER NAME, GEN # OF SECOND INDIVIDUAL? JOHN,4_
```

Fig. 1.

Program Listing 1.

```
10 REM FAMILY RELATIONSHIP PROGRAM BY SIL HORWITZ, REV 3.6 80060
7
20 CLEAR1000
30 GOSUB3000:FORX=1TO8:GOSUB3040:NEXT:GOSUB3020:GOTO 100
40 POKE16414,88:POKE16415,4
50 Z=1:GOSUB70:PRINT@900,"TO CONTINUE, PRESS ENTER, OR '0' TO EN
D";:
INPUTZ:IFZ=0THENEND ELSE220
70 PRINT@832,STRING$(62,32);
80 PRINT@896,STRING$(62,32);:IFZ=1RETURNELSEGOTO 120
100 PRINT@333,"GENEALOGICAL RELATIONSHIP PROGRAM";
110 PRINT@464,"BY SIL HORWITZ, LAKE MARY, FL";
120 PRINT@900,"LINE PRINTER (L) OR SCREEN (S)?"
130 V$=INKEY$:IFV$="L"THEN 150 ELSEIFV$="S"THEN220ELSEGOTO 130
140 CLEAR1000
150 IF(V$="L")AND(PEEK(14312)>127)PRINT"LINE PRINTER NOT OPERATI
NG;
TURN ON BEFORE CONTINUING!!!":FORX=1TO1000:NEXT:Z=0:GOTO 80
220 CLS:PRINT@74,STRING$(64,140);:PRINT@202,STRING$(54,140);
240 FORY=4TO10:SET(20,Y):NEXT
260 PRINT@128," 0 ";
280 PRINT@76," 1 ";:PRINT@204," 1 ";
300 PRINT@86," 2 ";:PRINT@214," 2 ";
```

Program continues

Program continued

```
320 PRINT@96," 3 ";:PRINT@224," 3 ";
340 PRINT@106," 4 ";:PRINT@234," 4 ";
360 PRINT@116," 5 ";:PRINT@244," 5 ";
380 PRINT@126," 6 ";:PRINT@254," 6 ";
400 PRINT@271,"GENERATIONS FROM COMMON PROGENITOR"
420 PRINT:PRINTTAB(5)"TO ESTABLISH RELATIONSHIP, CONSIDER GENERA
TION OF"
440 PRINTTAB(5)"FIRST INDIVIDUAL, AS COMPARED WITH SECOND, USING
 NUMBERS"
460 PRINTTAB(5)"TO INDICATE GENERATION FROM NEAREST COMMON PROGE
NITOR."
480 PRINTTAB(5)"COMMON PROGENITOR IS CONSIDERED '0' - COUNT FROM
 THERE."
500 PRINT:PRINTTAB(5)"AS REQUESTED, ENTER NAME,SEX (M/F),GENERAT
ION # - IN"
520 PRINTTAB(5)"THAT ORDER, WITH EACH SEPARATED BY A COMMA AS SH
OWN."
540 PRINT:INPUT"ENTER NAME, SEX, GEN # OF FIRST INDIVIDUAL";A$,X
$,A
550 IF(X$<>"F")AND(X$<>"M")PRINT"USE ONLY 'F' OR 'M' FOR SEX":GO
TO540
555 IFA>6THEN590
560 INPUT"ENTER NAME, GEN # OF SECOND INDIVIDUAL";B$,B
590 IFA>6ORB>6CLS:PRINT@450,
"GEN #'S GREATER THAN 6 NOT ACCOMMODATED BY THIS PROGRAM":GOTO 4
0
600 IFV$="L"POKE16414,141:POKE16415,5
620 CLS:PRINT@390,A$+" IS "+B$+"'S ";
640 IFX$="F"GOTO700
660 IFX$="M"GOTO740
700 IFA=0THEN780
720 ONAGOTO1060,1320,1560,1840,2100,2400
740 IFA=0GOTO920
760 ONAGOTO1180,1480,1760,2020,2260,2600
780 IFB=0PRINT"COMMON PROGENITOR":GOTO 40    'A=0
800 ONBGOTO820,840,860,880,900,910
820 PRINT"MOTHER":GOTO 40
840 PRINT"GRANDMOTHER":GOTO 40
860 PRINT"GREAT-GRANDMOTHER":GOTO 40
880 PRINT"GREAT-GREAT-GRANDMOTHER":GOTO 40
900 PRINT"GREAT-GREAT-GREAT-GRANDMOTHER":GOTO 40
910 PRINT"GREAT-GREAT-GREAT-GREAT-GRANDMOTHER":GOTO 40
920 IFB=0PRINT"COMMON PROGENITOR":GOTO 40    'A=0
940 ONBGOTO960,980,1000,1020,1040,1050
960 PRINT"FATHER":GOTO 40
980 PRINT"GRANDFATHER":GOTO 40
1000 PRINT"GREAT-GRANDFATHER":GOTO 40
1020 PRINT"GREAT-GREAT-GRANDFATHER":GOTO 40
1040 PRINT"GREAT-GREAT-GREAT-GRANDFATHER":GOTO 40
1050 PRINT"GREAT-GREAT-GREAT-GREAT-GRANDFATHER":GOTO 40
1060 IFB=0PRINT"DAUGHTER":GOTO 40    'A=1
1080 ONBGOTO1100,1120,1140,1160,1170,1175
1100 PRINT"SISTER":GOTO 40
1120 PRINT"AUNT":GOTO 40
1140 PRINT"GREATAUNT":GOTO 40
1160 PRINT"GREAT-GREATAUNT":GOTO 40
1170 PRINT"GREAT-GREAT-GREATAUNT":GOTO 40
1175 PRINT"GREAT-GREAT-GREAT-GREATAUNT":GOTO 40
1180 IFB=0PRINT"SON":GOTO 40
1200 ONBGOTO1220,1240,1260,1280,1285,1290
1220 PRINT"BROTHER":GOTO 40
1240 PRINT"UNCLE":GOTO 40
1260 PRINT"GREATUNCLE":GOTO 40
1280 PRINT"GREAT-GREATUNCLE":GOTO 40
1285 PRINT"GREAT-GREAT-GREATUNCLE":GOTO 40
1290 PRINT"GREAT-GREAT-GREAT-GREATUNCLE":GOTO 40
1320 IFB=0PRINT"GRANDDAUGHTER":GOTO 40    'A=2
1340 ONBGOTO1360,1380,1400,1420,1440,1450
1360 PRINT"NIECE":GOTO 40
1380 PRINT"FIRST COUSIN":GOTO 40
1400 PRINT"FIRST COUSIN ONCE REMOVED":GOTO 40
1420 PRINT"FIRST COUSIN TWICE REMOVED":GOTO 40
1440 PRINT"FIRST COUSIN THRICE REMOVED":GOTO 40
1450 PRINT"FIRST COUSIN FOUR TIMES REMOVED":GOTO 40
1480 IFB=0PRINT"GRANDSON":GOTO 40    'A=2
1500 ONBGOTO1520,1380,1400,1420,1440,1450
1520 PRINT"NEPHEW":GOTO 40
1560 IFB=0PRINT"GREAT-GRANDDAUGHTER":GOTO 40    'A=3
1580 ONBGOTO1600,1400,1640,1660,1680,1700
1600 PRINT"GRANDNIECE":GOTO 40
1640 PRINT"SECOND COUSIN":GOTO 40
1660 PRINT"SECOND COUSIN ONCE REMOVED":GOTO 40
1680 PRINT"SECOND COUSIN TWICE REMOVED":GOTO 40
1700 PRINT"SECOND COUSIN THRICE REMOVED":GOTO 40
1760 IFB=0PRINT"GREAT-GRANDSON":GOTO 40    'A=3
1780 ONBGOTO1800,1400,1640,1660,1680,1700
1800 PRINT"GRANDNEPHEW":GOTO 40
1840 IFB=0PRINT"GREAT-GREAT-GRANDDAUGHTER":GOTO 40   'A=4
1860 ONBGOTO1880,1420,1680,1940,1960,1980
1880 PRINT"GREAT-GRANDNIECE":GOTO 40
1940 PRINT"THIRD COUSIN":GOTO 40
1960 PRINT"THIRD COUSIN ONCE REMOVED":GOTO 40
1980 PRINT"THIRD COUSIN TWICE REMOVED":GOTO 40
2020 IFB=0PRINT"GREAT-GREAT-GRANDSON":GOTO 40    'A=4
2040 ONBGOTO2060,1420,1680,1940,1960,1980
2060 PRINT"GREAT-GRANDNEPHEW":GOTO 40
2100 IFB=0PRINT"GREAT-GREAT-GREAT-GRANDDAUGHTER":GOTO 40   'A=5
2120 ONBGOTO2140,1440,1680,1960,2220,2240
2140 PRINT"GREAT-GREAT-GRANDNIECE":GOTO 40
2220 PRINT"FOURTH COUSIN":GOTO 40
2240 PRINT"FOURTH COUSIN ONCE REMOVED":GOTO 40
2260 IFB=0PRINT"GREAT-GREAT-GREAT-GRANDSON":GOTO 40   'A=5
2280 ONBGOTO2300,1440,1680,1960,2220,2240
2300 PRINT"GREAT-GREAT-GRANDNEPHEW":GOTO 40
2400 IFB=0PRINT"GREAT-GREAT-GREAT-GREAT-GRANDDAUGHTER":GOTO 40 '
A=6
2420 ONBGOTO2440,1450,1700,1980,2240,2540
2440 PRINT"GREAT-GREAT-GREAT-GRANDNIECE":GOTO 40
2540 PRINT"FIFTH COUSIN":GOTO 40
2600 IFB=0PRINT"GREAT-GREAT-GREAT-GREAT-GRANDSON":GOTO 40 'A=6
2620 ONBGOTO2640,1450,1700,1980,2240,2540
2640 PRINT"GREAT-GREAT-GREAT-GRANDNEPHEW":GOTO 40
3000 CLS:PRINT@128,"";
3020 PRINTSTRING$(63,"#"):RETURN
3040 PRINT"#";STRING$(61,32);"#":RETURN
5000 END
```


✓See List of Advertisers on page 322

Summarize yours and display the results.

Vital Statistics

C. Brian Honess
Management Science Department
College of Business Administration
University of South Carolina
Columbia, SC 29208

In this article we'll build a program which will summarize and display descriptive data value characteristics. Our program will compute and print the arithmetic mean, the range, and several other common descriptive statistics.

The first step is to decide on an upper limit for the number of data values to put through the descriptive statistics program. The upper limit will be a function of the number of storage locations your computer has in main memory and will also be a function of the type of problem you'll be using the program for. Set the upper limit keeping in mind that at the top of the program we have to save storage locations for the data values to be sure that each possible value has a reserved space.

Assume that we'll use the program for 100 or fewer data values. Add one to this number, and code a DIMension statement. Use X as the variable name for the data values:

```
10 DIM X(101)
```

One must be added to 100 because we reserve 100 storage locations for valid data values, and one storage location for an invalid data value called a "trailer" value. Trailer values signal the program when to stop reading in data values, allowing us to use the program for any number of values from 1 through 100.

To choose an appropriate trailer value you have to know something about the possible data values the program will process; the trailer value is selected from outside the range of valid data values. For example, if you are going to process examination scores that range between 0 and 100, inclusive, a suitable trailer value would be any number less than 0, or any number greater than 100.

It could well be that you don't have any idea what you'll be running through your descriptive statistics program at the time you write it. In this case, you could input or read a value for the trailer when the program is run. Any of the following program segments could be used:

```
20 PRINT "WHAT IS THE SMALLEST
   POSSIBLE DATA VALUE?"
30 INPUT S
40 T = S - 1
```

or

```
20 PRINT "WHAT IS THE LARGEST
   POSSIBLE DATA VALUE?"
30 INPUT L
40 T = L + 1
```

Or, since Basic allows Print and Input to be combined:

```
20 INPUT "WHAT IS THE LARGEST
   POSSIBLE DATA VALUE"; L
30 T = L + 1
```

Using the Read statement, you'd have to code an appropriate data statement before the program was run:

```
20 READ S
30 DATA 35   (or whatever your smallest
              value is)
40 T = S - 1
```

or

```
20 READ L
30 DATA 1234   (or whatever your largest
                value is)
40 T = L + 1
```

Let's assume we'll always be using the program for data values between 0 and 100, inclusive, so that −1 would be an appropriate trailer value. The next step is to code data statements with the various data values. We'll assume some exam scores and not bother with an Input or Read for the trailer value, but just attach it at the end of the list:

```
10 DIM X(101)
20 DATA 78,56,89,90,45,27,72,89,88,46
21 DATA 66,98,99,42,51,70,89,90,89,93, - 1
```

You don't need to know how many numbers you've got as long as you have less than the number you dimensioned for, and you don't have to sort them into any predetermined order.

The next step is to load the X vector (the multi-location X array) with data values. This can be done by putting a read statement inside a loop. Also keep

X										
78	56	89	45	…	93	-1	…			
1	2	3	4		20	21	98	99	100	101

Fig. 1. Sample X Vector

checking for the trailer value, so that you know when to exit the loop:

```
30 FOR I = 1 TO 101
40 READ X(I)
50 IF X(I) = - 1 THEN 70
60 NEXT I
70 N = I - 1
```

In line 30, the loop counter goes to 101, which is the number we earlier dimensioned X. If you change the DIM statement you must change this upper limit value for the loop counter. Line 50 checks to see if the value just read is the trailer value. If you determined the trailer value using an Input or Read statement, you'll have to change this line to:

```
50 IF X(I) = T THEN 70
```

Line 70 calculates the number of valid data values in the X vector. One must be subtracted from the value of I when the loop exit occurs so the trailer value won't be counted as a valid data value. The X vector, for our example, now looks like Fig. 1.

The variable N will be used throughout the remainder of the program as the upper limit of loop counters to avoid going all the way through the X vector to the dimensioned limit. In the example, for instance, we need deal with the first 20 storage locations of X only, since the other 81 don't contain valid data values.

Knowing N, we can add N data values and calculate the arithmetic mean. The formula for the arithmetic mean is:

$$\bar{X} = \sum_{i=1}^{n} x_i/n$$

$\bar{X}$ (read X-bar) is the mathematical symbol for the arithmetic mean. The formula simply says that you find X-bar by adding all the X values, from first to nth, and divide by n, the number of data values. The capital Greek S, or sigma simply means sum in mathematics. We'll initialize the variable SM to zero, enter a loop and find the sum of the N data values:

```
80 SM = 0
90 FOR I = 1 TO N
100 SM = SM + X(I)
110 NEXT I
```

Line 80 defines SM for use in line 100.

The arithmetic mean (henceforth AM) can now be easily calculated using:

```
120 AM = SM/N
```

Using the test data, SM = 1467 and the AM is 73.35. Print the mean and the number of valid data values so you can run and check your program. Use simple print statements:

```
130 PRINT "NO. OF DATA VALUES = "; N
140 PRINT "ARITHMETIC MEAN = "; AM
```

The Harmonic Mean

In order to illustrate a case in which the harmonic mean is desirable, consider the following example: Suppose you drive your car to a city 100 miles away, and you make the trip in one hour. You've averaged 100 miles per hour. Now, you turn around and make the return trip in two hours (maybe weighted down by your speeding tickets?), so your average speed for the return trip is 100/2 = 50 miles per hour.

What is your average speed for the entire trip? On the surface, it would seem that with an average speed of 100 mph going and 50 mph on the return, your average speed should be (100 + 50)/2 = 75 mph.

Using the formula for the arithmetic mean, this does indeed seem to be correct. But wait a minute...the total trip was 200 miles, and you were gone for three hours, so wasn't your average speed really 200/3 = 66.66 mph? In this case the arithmetic mean statistic failed us. Actually, we just selected the wrong statistic and should have used the harmonic mean. The formula for the harmonic mean is:

$$HM = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

The formula basically says we need to divide the number of data values by the sum of the reciprocals of the data values. Continuing with the example, the formula produces the correct answer:

$$HM = \frac{2}{\frac{1}{100} + \frac{1}{50}} = \frac{2}{.01 + .02} = \frac{2}{.03} = 66.\overline{66}$$

Now a problem involving investments. Suppose you have \$100 to invest in a particular stock each month. The price of the stock will probably vary, and you'll buy a different number of shares each month for your \$100 investment. If you want to calculate the price per share over a period of several months, the harmonic mean will produce the correct answer.

This problem is analogous to the average speed problem, in that the number of miles between two cities is constant, as is the total investment per month. The number of shares purchased in a given month is analogous to the miles per hour value for a particular lap of the trip.

A similar application would involve a constant gas tank capacity on your car. Suppose you have a 20 gallon gas tank, and you always refill it at exactly the point where you have run out of gas. If you always buy exactly 20 gallons but at a variety of prices per gallon, you'll find the harmonic mean useful for calculating the average price per gallon of gas.

Let SM serve as an accumulator for the sum of the reciprocals this time. The harmonic mean, (HM) can be calculated with this code:

```
150 SM = 0
160 FOR I = 1 TO N
170 SM = SM + 1/X(I)
180 NEXT I
190 HM = N/SM
```

We can then print it using:

```
200 PRINT "HARMONIC MEAN   = "; HM
```

If you run the program to this point, you should get a value of 64.9336 for the harmonic mean.

The Geometric Mean

The geometric mean is extremely useful in business applications for calculating the average percentage rate of growth of a variable over time. For example, assume you invest \$100 and it grows to \$108 by the end of a year. At the end of the second year you have \$120, at the end of the third year you have \$135, and at the end of the fourth year you have \$160. You can calculate the percentage of increase each of the years by looking at the ratio of growth for each particular year (see Table 1).

Suppose you want to find the average rate of growth over the four year period. You could find the arithmetic mean of the percentage increases—(8 + 11.1111 + 12.5 + 18.5185)/4 = 12.5314 percent. This seems a reasonable answer, but is it correct? An easy way to find out is to substitute values into the compound interest formula:

$$\text{Value} = \text{Principal}\ (1 + \text{rate})^{\text{No. of periods}}$$

$$V = 100(1 + 0.125314)^4 = 160.365$$

This is 37¢ more than the \$160 value shown in the table. The difference may not seem large, but we've used very small values—it would be much larger if we used a larger principal and/or a greater number of years. This error was not caused by round-off errors or some other computer-connected problem but by using the wrong statistic. Instead of the arithmetic mean, the geometric mean should have been used.

End of Year	Principal	Ratio of Growth	Percent Increase
1	\$108	108/100 = 1.08	8.0
2	120	120/108 = $1.\overline{11}$	$11.\overline{11}$
3	135	135/120 = 1.125	12.5
4	160	160/135 = $1.\overline{185}$	$18.\overline{518}$

Table 1. Stock Calculations

The formula for the geometric mean is:

$$GM = \sqrt[n]{\prod_{i=1}^{n} x_i}$$

The Greek capital P (π), or Pi, means product. The formula says to take the nth root of the product of all of the data values. Returning to our example, you get:

$$GM = \sqrt[4]{8 \cdot 11.1111 \cdot 12.5 \cdot 18.5185}$$
$$= \sqrt[4]{20576.1} = 11.9768\%$$

Now you have two answers: 12.5314 percent using the arithmetic mean, which we've proved to be wrong, and 11.9768 percent using the geometric mean. Test 11.9768 by using the same technique used before:

$$V = 100(1 + 0.119768)^4 = 157.221$$

You're even farther away from the $160 you'd like to see as the result! What's going on here? Maybe we're finding the geometric mean of the wrong set of numbers.

Take the starting principal and the ending principal, put them into a simple financial formula and see if the answer is somewhere in the vicinity of the two you've already calculated, returning a correct $160 when substituted back into the compound interest formula.

$$\sqrt[4]{160/100} - 1 = 0.124682$$

You can easily check to see that this is correct, because:

$$100(1 + 0.124682)^4 = 160$$

You need only realize that we should have been using the following:

$$GM = \sqrt[4]{1.08 \cdot 1.11111 \cdot 1.125 \cdot 1.18519}$$
$$= 1.12468$$

If you subtract 1, you'll have the correct average growth rate of 12.4682 percent.

The geometric mean will be easy to code, but you need to note carefully that we're initializing the product to one and not to zero, as we did in the previous two cases. Sums are always initialized to zero and products to one. Work through this small section of code and see why it must be initialized to one and not to zero.

```
210 P=1
220 FOR I=1 TO N
230 P=P*X(I)
240 NEXT I
250 GM=P↑(1/N)
260 PRINT "GEOMETRIC MEAN    "; GM
```

In line 250 the nth root of a number is found by raising it to the reciprocal of n power.

If you key in the program to this point and run it, you should get the following output:

```
NO. OF DATA VALUES=20
ARITHMETIC MEAN=73.35
HARMONIC MEAN=64.9336
GEOMETRIC MEAN=69.5528
```

Some versions of Basic may have a slightly different precision, more decimal places of accuracy, etc., but if you don't get these answers, or some very close, recheck your program.

Growth-type problems are not the only use for the geometric mean; it is also useful in reducing the effects of "way out" data values. For example, suppose you consider the weekly income of the first 10 people you meet on the street in an effort to describe the typical wage-earner in your town. The first nine people have weekly incomes ranging from $150 to $500 but the last person we encounter is an oil-rich Sheik, making about $500,000,000.00 a week. If you found the arithmetic mean of these 10 values you'd expect it to be about $50,000,315.

Telling the world that the typical person in your town makes $50,000,315.00 a week would certainly create the wrong impression. The geometric mean would produce a figure closer to the weekly earnings of the majority of wage-earners. The same problem might occur in an examination in which most of a class scored high, while a small percentage scored very low. The arithmetic mean would be the average grade, but it could be very misleading; the geometric mean would be a better descriptive statistic to use in this case.

There is an interesting relationship among the arithmetic, harmonic and geometric means: harmonic≤geometric≤arithmet-

✓See List of Advertisers on page 322

ic. The equality relationship holds when all data values are the same.

Measures of Dispersion

These means are called "measures of central tendency" because they describe where the center of a group of data values tends to be. Two other measures of central tendency, the mode and the median, will be discussed after investigating some "measures of dispersion."

There are four popular measures of dispersion, the "range," the "mean deviation" (sometimes called the "average deviation"), the "standard deviation," and the "variance."

Assume a large programming class has just taken an exam, and you want to compare the performance of this class with one taught by another teacher. Assume each class has 100 students. Suppose the first teacher calculates his class average to be 75, and the second teacher replies that this is a real coincidence, since his class also averaged 75.

Can we say that the performance of the two classes was equal? No, we can't—in fact, we can't say that even if both teachers reply that their lowest score was 50, and their highest score was 100.

Consider just two of the many possible arrangements that could produce all of the stated conditions:

- Fifty students score 50, and 50 score 100.
- One student scores 50, 98 score 75, and one scores 100.

In both cases, the arithmetic mean = 75, the range = 50, the high score = 100, and the low score = 50. The arithmetic mean by itself is not enough to describe a set of data values; you need a measure of how spread out the data values are—a measure of dispersion.

The mean deviation, or average deviation, has a fairly formidable-looking formula, but is easy to understand and easy to code into Basic:

$$MD = \frac{\sum_{i=1}^{n} x_i - \bar{X}}{n}$$

The two parallel bars mean "absolute value"; this is a fancy way of saying you should erase the sign of any negative number. The whole formula says: Take the difference between each data value and the mean; if this difference is negative, make it positive. Then add the differences, and divide by the number of data values you have. In other words, you're finding the average distance, or deviation, of the data values from the mean. If the mean deviation is small you'd expect the data values to be clustered near the mean, as in the second case above, where 98 percent of the students made a score exactly at the mean. A large mean deviation would indicate that the data values are spread out.

There are two ways to find the absolute value of the difference between a data value and the mean. You could test this difference and if it was negative multiply it by −1, thereby making it positive, or you could use the ABS arithmetic function found in most versions of Basic. Let's use the second method, since this is nearly a universal intrinsic function. Again, we'll need to add some data values, so we'll use SM and initialize it to zero.

```
270 SM = 0
280 FOR I = 1 TO N
290 SM = SM + ABS(X(I) - AM)
300 NEXT I
310 MD = SM/N
320 PRINT "MEAN DEVIATION    = "; MD
```

A second measure of dispersion is the standard deviation. The standard deviation and its kin, the variance, are used more often than is the mean deviation, because they allow you to make inferential statements about a distribution and a sample drawn from a distribution. For example, you could say, "Based on this sample of integrated circuits we've drawn from that big bin over there, I'm 95 percent sure that the number of bad integrated circuits in the bin won't exceed four percent."

The standard deviation (symbol: small Greek sigma, σ) is simply the square root of the variance (symbol: σ^2), so we'll calculate and print both together. The

SUPER DISCOUNTS ON ALL COMPUTER EQUIPMENT

16K MEMORY .... ONLY $25.95!
For **TRS-80** Keyboard or Expansion interface. KEYBOARD requires jumpers: $2.00 Extra. These are 200 ns tested RAM for the **TRS-80, APPLE** or **EXIDY.**

DISK DRIVES for the TRS-80 OR PMC-80:

All of our drives come complete with power supply and chassis. They may be used with existing Radio Shack drives on the same cable! 40 track drives store 102K bytes single density, and 175K double density. 80 track drives have 175K single density and 345K double density! All drives guaranteed 90 days, one year on power supply.

40 track MPI drives $319.95
40 track TEAC drives $315.95
40 track TANDON drives $319.95
80 track MPI drives $449.95
80 track TEAC drives $429.95
2 drive cable .... $ 25.95
4 drive cable .... $ 39.95
NEWDOS 80 VER. II $139.95
NEWDOS 80 VER. I Special sales price.
Limited Quantity **$89.95**

PERCOM'S DOUBLER II
for double density operation! **$169.95**
The **DOUBLER** works with the **TRS** or **PMC** expansion interfaces to allow you to use your drives in double density! You may still operate your drives as single density also! Comes with **DBLDOS** operating system which allows you to transfer single density files to double and vice versa! *GREAT BUY!*
DISKETTES: VERBATIM DATALIFE! *BOX OF TEN*
SOFT OR HARD SECTORED 5¼" **$32.50**
WE HAVE DRIVES AND CONTROLLERS FOR THE MODEL III. CALL FOR PRICES!!!

MODEMS AND TELE-COMMUNICATIONS

LYNX Telecommunication system for the TRS-80 and PMC-80 .... $279.95
Includes terminal software and connections for instant hookup! Can be connected to the **TRS-80** or **PMC-80** with or without an expansion interface!
LEXICOM MODEM 300 BAUD Requires RS-232 .... $169.95
THE SOURCE: Hook-up to the "SOURCE" .... $ 99.95
'TARICONNECTION: Modem for 400/800, complete with software! $249.00
APPLECONNECTION: Modem for APPLE II $279.95

COMPLETE SYSTEMS:

PMC-80, 16K LEVEL II COMPUTER.................. $739.00
The **PMC-80** is a work alike to the **TRS-80** mod I computer! Comes with Microsoft's **BASIC** in ROM. Built in cassette. 12" video monitor. Expandable to 48K.
Compatible to All **TRS-80 MOD I Programs.**
PMC 80 without monitor $595.00
RF—MOD for PMC to TV hookup $39.95
PMC-80 EXPANDER 100 SYSTEM $499.95
INCLUDES: 32K memory, Parallel printer driver, Disk controller. Fully compatible with **TRSDOS, NEWDOS, VTOS,** and all other **TRS-80 Mod I disk software!**
New New PMC-81 Computer is here......................... **CALL**
APPLE II COMPUTERS 48K $1299.00
ATARI 400 .... $ 495.95
ATARI 800 COMPUTER .... $ 795.00
ZENITH Z-89 48K, 1 DISK ALL IN ONE COMPUTER $2495.00

VIDEO MONITORS

LEEDEX 100 12' B/W MONITOR $159.95
SANYO 9' B/W MONITOR $199.95

PRINTERS

OKIDATA MICROLINE 80 All OKI'S have TRS-80 Graphics!!! $420.00
Comes with friction and pin feed, upper/lower case, Graphics.
EPSON MX-80 PRINTER: Word Processing Quality Printout with Graphics! .... Call
NEC SPINWRITER with Tractor Feed ... $2995.00
DIABLO MODEL 1630 with Tractor $2695.00
UNIVERSAL PRINTER STANDS .... $ 94.50

Other Accessories:
SUP-R-MOD RF Modulator for APPLE ... $29.95
ARCHBOLD SPEED-UP MOD FOR TRS MOD I. Allows up to 300% increase! $45.00
MICROSOFT BASIC Decoded and other mysteries $29.95
TRS-80 DISK and other mysteries $18.95
ZBASIC BASIC COMPILER for MOD I and MOD III TRS-80 and PMC-80 Increase basic program speeds by up to 200 times!!!!!
Tape .... $79.95 Disk $89.95
Both For Only $99.95 Specify MOD I or MOD III

WE HAVE HUNDREDS OF PROGRAMS FOR ALL THE POPULAR COMPUTERS! CALL FOR OUR FREE CATALOG

Personal and Business Checks take 3 weeks to clear. For fastest delivery use Certified Checks, Money Orders, Credit Card, or C O D
We will try and beat any published price on any system! Call!

SIMUTEK COMPUTER PRODUCTS ™ ✓19
4877 E Speedway Blvd
Tucson, Arizona 85712
(602) 323-9391 Technical Questions

CALL TOLL FREE TO ORDER
(800) 528-1149
Please mention this magazine when ordering
ARIZONA RESIDENTS ADD 6% SALES TAX.
WE TAKE VISA, MASTER CARD.

✓See List of Advertisers on page 322

formula for the variance is:

$$V = \frac{\sum_{i=1}^{n} (x_i - X)^2}{n}$$

You could say that we're just getting rid of the possible minus sign by squaring the term, instead of taking the absolute value. Later, when we take the square root of the variance, we're just scaling it back down again to get it in line with what the mean deviation would be. It is important to realize that the mean deviation will not equal the standard deviation because the squaring process tends to more heavily weight the extreme data values.

Since you're confronted with the capital sigma, or summation sign, you're going to have to initialize SM to zero again. The arithmetic mean is still stored as AM, and N contains the number of valid data values.

```
330 SM=0
340 FOR I=1 TO N
350 SM=SM+(X(I)-AM)↑2
360 NEXT I
370 V=SM/N
380 PRINT "VARIANCE        =";V
390 SD=SQR(V)
400 PRINT "STANDARD DEVIATION =";
    SD
```

Adding these three statistics to the existing program, you should get the following results when you run it:

```
NO. OF DATA VALUES = 20
ARITHMETIC MEAN = 73.35
HARMONIC MEAN = 64.9336
GEOMETRIC MEAN = 69.5528
MEAN DEVIATION = 18.515
VARIANCE = 448.628
STANDARD DEVIATION = 21.1808
```

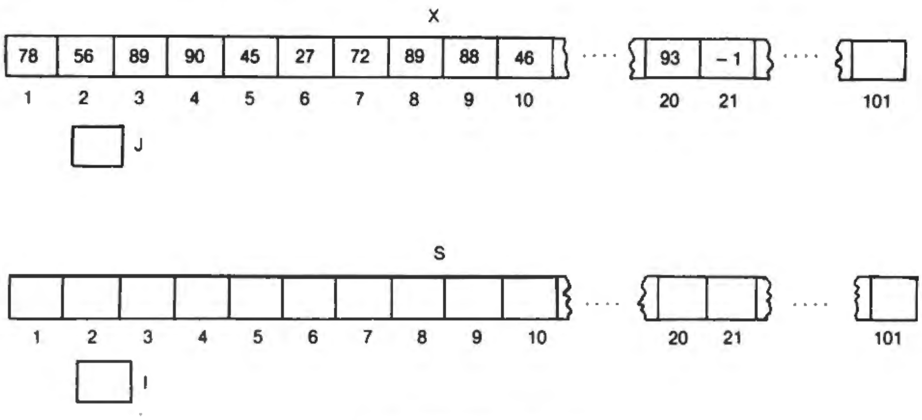


Fig. 2. Memory Diagram

Sort the data values into ascending order, allowing the calculation of the median and the range. There are dozens of sorting routines. The selection is usually somewhat dependent upon the type of data you want to sort, the type computer you have, the size of the vector you want to sort and other things.

We're going to choose a sorting technique that isn't the fastest method, but it is fairly easy to explain and should prove to be easy to modify to suit your particular needs.

Many sorting methods rearrange the data values within the same vector with which you began. In this case, you'd go into the sort routine with your X vector containing the N valid data values. You'd emerge from the sort routine with the same X vector, but this time the values would be sorted.

We're going to take another approach—the creation and filling of a second vector (let's call it S for sorted) which you'll fill

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

with the data values as you sort them. Incidentally, in the process you'll be destroying the X vector. To create this new vector, you'll have to return to the DIM statement and modify it to read:

```
10 DIM X(101), S(101)
```

If you're sorting longer lists and have modified the X dimension, you'll want the S dimension to be the same. With these two vectors, you're going to employ two pointers to indicate the particular storage location in each of the vectors that you're currently working on. The pointer that will point to the location in vector X we'll call J, and the pointer for vector S we'll call I. We can diagram the way we've set up your memory for this segment (see Fig. 2).

One additional piece of information you'll need is the value of the largest possible data value you're likely to encounter. For our example problem, this will be 100, since we're working with examination grades on a 0 to 100 scale. If you're coding a general-purpose, descriptive statistics program, you might want to read this value in using a data statement, or input the value. However you do it, create a variable containing a value larger than the largest value you'll encounter in your valid data values.

I know my largest value will not be over 100, so I'm going to create a variable B (for Big) and put the value 101 into it.

To test this section of the program you might want to look at the X vector before it is sorted and also look at the sorted S vector. This can easily be done by adding the following code:

```
404 FOR I = 1 TO N
406 PRINT X(I);
408 NEXT I
```

```
502 PRINT

503 FOR I = 1 TO N
504 PRINT S(I);
505 NEXT I
```

(Line 502 is necessary so the sorted vector isn't printed starting on the same line that ended the printing of the X vector. The line is necessary since there is a semicolon at the end of the print statement in line 406.)

You might also want to verify that the X vector has been destroyed in the process. This could be done by again inserting a loop to print the X vector:

```
506 PRINT
507 FOR I = 1 TO N
508 PRINT X(I);
509 NEXT I
```

Print in line 506 puts us on a new line, because of the semicolon at the end of line 504. (I'm assuming you'll take these statements out after you see how the vectors look in the example program.)

The sort routine is a pair of nested For...Next loops and might be more easily discussed if we were to indent some of the coding as shown in the sort routine.

The outer loop fills the S vector. When the index I is equal to one, you're looking for the value to put into S(1), etc. Line 410 codes the fact that you want to fill the first N locations in the S vector. Line 420 sets up a storage location B (short for Big) and loads it with a number that is larger than any data value you'll encounter in the X vector.

In lines 430 through 470 you search the X vector for the smallest value. The J pointer points to the particular location in the X vector currently being evaluated. For each of the X lo-

```
410 FOR I = 1 TO N
420    B = 101
430       FOR J = 1 TO N
440          IF X(J)> = B THEN 470
450          B = X(J)
460          RM = J
470       NEXT J
480    S(I) = B
490    X(RM) = 101
500 NEXT I
```

Sort Routine

cations, compare the value with that stored in B. If the value in the X location is less than the value in B, perform the instructions in lines 450 and 460. If X(J) isn't smaller than B, skip down to the bottom of the loop and increment J and consider the next value in X. Line 440 controls this.

Line 450 is where you put the smaller value into location B when one is found. Many values might be placed in B as the X vector is searched, but the one remaining at the end of the loop is going to be the smallest in the vector.

Line 460 remembers where each of the values you put into B came from; location RM (for ReMember) performs this function. When you put the final value into the sorted vector, S, you'll have to go back and arrange things so that that same smallest value won't be found the next time through the loop. If you let this happen, you'd keep finding the same smallest value each time.

Assume that you've been through the 430 to 470 For... Next loop N times. At this point, the B location will contain the value 27, and the RM location will contain a six from the sample data you've been using. Line 480 moves the 27 from B into S(I) and line 490 reloads the sixth X location with a 101, right on top of the 27 that was there. This ensures that we'll never identify that location again as containing the smallest data value in the vector. Line 500 increments the S location you're trying to fill, 420 loads B with 101 again, and you start searching the X vector for the second-smallest value.

This program sorts from low to high; it can be easily changed to sort from high to low. To do so, set B equal to a number smaller than your smallest data value. Use the same number in line 490 and change the logical expression in line 440.

More Statistics

Now that the data values are sorted you can calculate some more descriptive statistics. S(1) is going to contain the smallest data value and S(N) the largest. You can print these values plus calculate and print the range with this code:

```
510 PRINT "SMALLEST DATA VALUE = ";
    S(1)
520 PRINT "LARGEST DATA VALUE = ";
    S(N)
530 R = S(N) - S(1)
540 PRINT "RANGE        = "; R
```

The median is a measure of central tendency which cannot be calculated until the data vector has been sorted. The method used to calculate the median depends upon the number of data values and whether this number is odd or even. This is the next problem—how to determine if a number is odd or even.

To determine if N is odd or even you'll use the integer (INT) function, which removes everything to the right of the decimal point. You'll divide N by two and then "integerize" this value. Next you'll multiply the result by two and compare the result with N. If the two numbers are the same, you can infer that N is an even number. If they are different, N must be odd.

For example, consider a value for N of 15. Dividing by two, you get 7.5 and that becomes seven using the integer function. Multiplying by two, you get 14. Since 14 does not equal the number 15 that you began with, 15 must be an odd number. Try this algorithm with an even number, and you'll see that the result always equals the original number.

If N is an odd number, there is a true middle number in the list. Consider the short list of seven values below. The fourth value is the median, because there are an equal number of numbers above and below the median. The median is such that half of the numbers in the vector are below and half are above the value:

5	
16	
27	
46	Median
58	
98	
99	

In the example below, N is four; and the Median is the arithmetic mean of the two middle values or 80.

34	
79	Median is the Arithmetic Mean of
81	two middle values
92	

✓See List of Advertisers on page 322

In the lines below, C implies center and ME median:

```
550 IF INT(N/2)*2 = N THEN 590
560 C = (N + 1)/2
570 ME = S(C)
580 GO TO 610
590 A = N/2
600 ME = (S(A) + S(A + 1))/2
610 PRINT "MEDIAN          = "; ME
```

Line 550 includes the INT function and transfers control to either 560 or 590 depending upon the outcome.

If N is odd, you calculate C, the Center, by adding one and then dividing by two. If N is even you need the two middle values, to find their arithmetic mean. Line 590 finds a new number, N/2, which it calls A. In the previous example, where N = 4, A would be two. The median is then the arithmetic mean of the second and third values, as delineated in line 600. In our running example, with the 20 data values, you'd calculate the arithmetic mean of the tenth value in the sorted vector (78) and the eleventh value (88). This is 83, and should be the value printed if you add the above coding to your program and execute a Run command.

The method you've used to calculate the median can also be used to calculate other measures of dispersion. For example, you could identify the first and third "quartiles" of the data values and calculate the semi-interquartile range, Q, where: $Q = (Q_3 - Q_1)/2$. The 10-90 percentile range (P_{90}-P_{10}) and the Semi-10-90 percentile range $(P_{90} - P_{10})/2$, together with the semi-interquartile range, are sometimes used to lessen the effects of extreme data values where the simple range could be misleading.

The Mode

Our fifth measure of central tendency, the mode, is the most difficult to program on a computer—and the easiest statistic to find manually. It is the value that you have the most of. In our example, the mode = 89, since there are four input values of 89. Had there been some other score with a frequency of four, there would be two modes, and you would call the distribution "bimodal." If there were more than two modes, you'd call it multi-modal. If there is only one occurrence of each data value, there is no mode. This mode-finding method is going to demand integer (whole number) data with a relatively small range. Our sample program has whole number data in the range 0 to 100 inclusive.

First modify the DIM statement in line 10. You need to add a new vector to keep track of the number of times each of the values appears in the data vector. Begin with a new vector to make things as clear as possible, instead of using the old X vector that was discarded after the sort. Since this vector will keep track of the frequency of occurrence of each data value, call it F and dimension it to 100. You'll be able to store the number of times the value 50 occurs in the data list in F(50), the number of 100's in F(100), the number of zeros in F(0), etc. The DIM statement will now look like this:

```
10 DIM X(101), S(101), F(100)
```

Most versions of Basic fill vectors with zeros automatical-

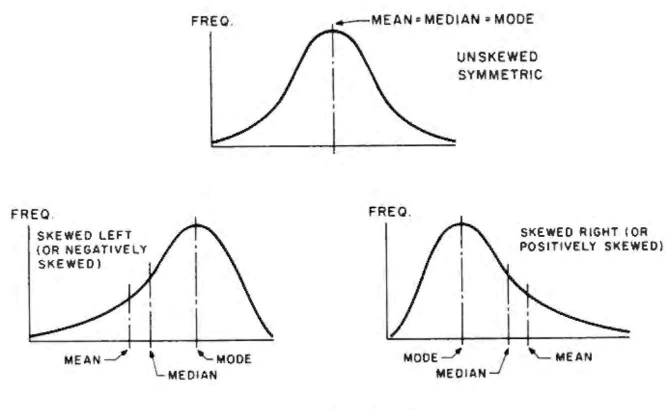


Fig. 3. Mean, Median, Mode Comparison

ly. If yours doesn't you'll have to include a loop to do this. This can be as simple as:

```
FOR I = 0 TO 100
F(I) = 0
NEXT I
```

The coded segment looks like this:

```
620 FOR I = 1 TO N
630 K = S(I)
640 F(K) = F(K) + 1
650 NEXT I
655 REM *** OPTIONAL REMARKS! THE
    F VECTOR IS NOW LOADED
656 REM *** WITH FREQUENCIES. NOW,
    SEARCH F FOR LARGEST
657 REM *** FREQUENCY.
660 BG = 0
670 FOR I = 1 TO 100
680 IF F(I)< = BG THEN 700
690 BG = F(I)
700 NEXT I
705 REM *** MORE OPTIONAL
    REMARKS! THE MODAL
706 REM *** FREQUENCY HAS BEEN
    FOUND. NOW FIND
707 REM *** ALL OCCURRENCES OF
    THIS VALUE.
710 FOR I = 1 TO 100
720 IF F(I)< >BG THEN 740
730 PRINT I; "IS A MODE"
740 NEXT I
```

The loop in 620–650 loads the F vector with the frequency of occurrence of each of the possible data values. S(1) in the example has a value of 27. Line 630 puts this 27 into a location called K. Line 640 then increments F(27) by one. Lines 630 and 640 simply avoid giving a subscript a subscript.

At the completion of this loop, the F vector has been loaded with all of the frequencies of occurrence, and we're ready to search this vector to find out which locations contain the largest value. This is done by setting variable BG equal to zero (BG implies BiGest). Then enter loop 670–700 and look at each frequency in the F vector. If the frequency is less than or equal to the value in BG, skip line 690, which puts larger values into BG. At the conclusion of this loop you've found the largest frequency, and it is stored in location BG. The final loop searches the F vector for all occurrences of the modal frequency stored in BG and prints the corresponding modal value when each is found.

Shape of Distribution

You can tell something about the shape of the distribution of data values by looking at the relationship between the arithmetic mean, median and mode. In a symmetric distribution, the mean, median and mode will all have the same value, while in a skewed, or asymmetrical distribution, the median will have a value between the mode and the mean, with the mode corresponding to the highest frequency of occurrence (Fig. 3).

A more precise statistic of the shape of the distribution may be calculated. Statisticians call this standardized coefficient of skewness the a_3. We'll call it A3.

$$a_3 = \frac{\sum_{i=1}^{n} (x_i - \bar{X})^3}{n} / \sigma^3$$

Since we've named a_3 A3, and have been calling σ (the standard deviation) SD, and calling the arithmetic mean AM, you can rewrite this as:

$$A3 = \frac{\sum_{I=1}^{N} (S(I) - AM)^3}{N} / SD^3$$

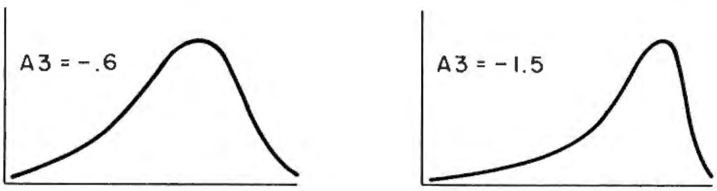


Fig. 4. Distribution Curves

The following code will calculate A3:

```
750 SM = 0
760 FOR I = 1 TO N
770 SM = SM + (S(I) - AM)↑3
780 NEXT I
790 SM = SM/N
800 A3 = SM/SD↑3
810 PRINT "STD. COEF. OF SKEWNESS
    = "; A3
```

For your test data, a value of −.648289 is produced. Since the coefficient is negative, this means the distribution is negatively skewed (skewed to the left). This means the tail of the distribution is to the left; most the data values are closer to 100 than zero.

If A3 was even more negative, say −1.5 or more, the distribution would be even more highly skewed to the left.

The last descriptive statistic is called the "standardized coefficient of kurtosis." Since statisticians use the symbol a_4 to designate it, call it A4 in your program.

The formula is identical to that for A3, except that we calculate the fourth power of the differences and then divide by the standard deviation raised to the fourth power.

$$A4 = \frac{\sum_{I=1}^{N} (S(I) - AM)^4}{N} / SD^4$$

The following code will calculate A4:

```
820 SM = 0
830 FOR I = 1 TO N
840 SM = SM + (S(I) - AM)↑4
850 NEXT I
860 SM = SM/N
870 A4 = SN/SD↑4
880 PRINT "STD. COEF. OF KURTOSIS
    = "; A4
```

When this program segment is added to your existing code, a value of 2.11691 results. Kurtosis reflects the degree of peakedness of a distribution. Usually a comparison is made to a normal (or bell-shaped) distribution. The normal bell-shaped distribution has an A4 value of about 3.0 and is called "Mesokurtic." If A4 is less than 3.0, the distribution is called "Platykurtic" and if it is over 3.0, the distribution is called "Leptokurtic." Our distribution with an A4 of 2.1 is platykurtic.

Statisticians sometimes subtract three from the standardized coefficient of kurtosis. If the result is negative, the distribution is platykurtick; if it is zero, the distribution is mesokurtic; and if it is positive, the distribution is leptokurtic.

There are lots of things you'll probably want to do to this bare-bones program:

- You can add fancy output formatting,
- Add REM statements to make the program easier to modify,
- Add screen-clearing or page-spewing commands,
- Add some instructions to be printed at the beginning, telling the user how to code data statements and how to chose a trailer value.■

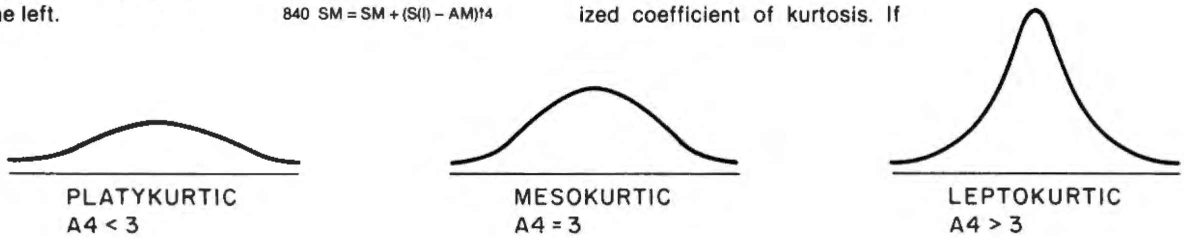


Fig. 5. Comparative Distributions

See List of Advertisers on page 322

Learn the ebb and flow of phonemes.

Programming with The Voice Synthesizer

Craig Werner
Quincy House
Harvard University
Cambridge, MA 02138

Of all the peripherals that can be attached to the TRS-80, none has the power of amazement that the voice synthesizer can provide. Most people have already become accustomed to the computer acting as a calculator, have had experience with computerized billing and record keeping and have been bombarded with computer games. But the ability for the computer to speak is still a novelty.

However, programming with the voice synthesizer can be very painstaking. This is due to two main reasons. The first is the English language, which is not as simple as Basic. For instance, voice synthesizer programs use T for to, too or two. Similar difficulties arise with four, for and fore. It is a problem with all words which have the same pronunciation but different spellings.

Other phonetic problems are seen in the example of the word polish which can be pronounced with a short O (as in shoe polish) or with a long O (as in the Polish Pope). To relearn the English language from a phonetic level is not easy.

The second type of difficulty arises from the synthesizer itself. It is limited to 60 phonemes, some of which are duplicated. Some sounds are missing or can only be achieved by difficult combinations. Other letters, particularly G and K, can be mispronounced when preceded or followed by certain other letters.

Finally, the synthesizer is slower than normal human speech and slower than the computer. This means that normal programming tricks to speed up program execution will overflow the buffer.

These problems should be apparent only to the programmer; the listener enjoys the voice in ignorant bliss.

At first voice programming is a trial and error task. Look at the four-line routine starting at line 10000 of the Program Listing. It prompts an input in the upper left-hand corner of the screen, then displays it and voices it. It has three main advantages: (1) It is short. (2) It displays previously used phonemes, so that a reference can be kept as new combinations are tried. (3) If you didn't catch the sound of the word just entered, hitting Enter again repeats the word.

"I compiled dictionaries for the Declaration of Independence and the Preamble to the Constitution. These provided me with three-quarters of the words I needed for a Woody Allen routine."

A few hints are useful at this stage. Remember the special characters <, >, =, and especially + for NG and / for ER. Also, certain phonemes that sound incorrect when used singly will sound fine when doubled or tripled. Phonemes which are totally wrong alone will be perfect in combination.

Another hint concerns characters such as the comma and colon, which supposedly cannot be input directly from the keyboard. They can be used if the string that contains them is enclosed by a leading quote. Pay attention to your work; many sounds are repeated.

One drawback of this routine is that once the phonemes have been combined to your satisfaction, it is necessary to write the results on paper.

After you have created a dictionary of phonetic words, they can be typed into a program. I prefer to type them as data to be read and alternated with their English equivalents such as W55N, WHEN; I!N, IN; <77,THE; K-OORS, COURSE;77V,OF; etc.

After the data statements have been typed in, the routine beginning at line 20000 is a proofreading aid. It voices the phonetic word and prints the English language word. The most versatile feature of this routine is the ability to skip the number of words you input in order to edit the text.

After you perfect this routine comes the most useful step from the programming point of

A = @*	ABOVE = 7B77VV	ADD = 9DD
ADVANCED = 9DV9NSD	AGO = 20GOU	ALL = „LL
ALTOGETHER = 2LTUG5>/	AND = 99NND	ANY = 43NE
ARE = ;AR	AS = :X	BATTLE = B9T#L
BE = BE	BEFORE = BEFOR	BIRTH = B/= =
BRAVE = BR>•V	BROUGHT = BR12TO	BUT = B87T
BY = B,&	CAN = KK9N	CAN NOT = KK9N2TT
CAUSE = K,8Z	CIVIL = S!V8L	COME = KK77M
CONCEIVED = K6NSEEVD	CONSECRATE = K6NS5KR(•T	CONSECRATED = K6NS5KR>•T45D
CONTINENT = K;NTIN3NT	CREATED = KRE>•T5D	DEAD = D45D
DEDICATE = D44D#K>•T	DEDICATED = D44D#K>•T3D	DETRACT = D!TR9KT
DEVOTION = D!VOU>N	DID = D!DD	DIED = D; !D
DO = D<U	EARTH = 8/ = = =	ENDURE = 3NDOR
ENGAGED = 5NG>□GED	EQUAL = K-W%L	FAR = F;RR
FATHERS = F;8<5RX	FIELD = FELD	FINAL = F;#NL
FITTING = F!T! + +	FOR = FOR	FORGET = FORG45T
FORTH = F%% =	FOUGHT = F12TT	FOUR = FOOR
FREEDOM = FRED8M	FROM = FR7MM	FULL = F%%LLL
GAVE = G>•V	GIVEN = G!V4N	GOD = GAADD
GOVERNMENT = G*V/M45NT	GREAT = GR@•T	GROUND = GR;UNND
HALLOW = H:LLOUW	HAVE = H9VV	HERE = HERR
HIGHLY = ;!LE	HONORED = ;N5RD	IN = I!N
INCREASED = !NKRESD	IS = !ZZ	IT = !TT
LARGER = LARGJ/	LAST = L9SST	LIBERTY = L#B5RTE
LITTLE = L!TLL	Live = L!V	LIVES = L;#VZ
LIVING = L!V! + +	LONG = LO + +	MEASURE = M45X/
MEN = M45N	MET = M34T	MIGHT = M;YT
NATION = N>•>N	NEVER = N45V/	NEW = NUU
NOBLY = NOUBLE	NOR = NOR	NOT = N2TT
NOTE = NOUT	NOW = N;U	*OF* = 77F
ON = 22N	OR = ORR	OUR = AOR
PEOPLE = PEPL	PERISH = P45R!>	PLACE = PL>•
POOR = POR	PORTION = POR>6N	POWER = PAW/
PROPER = PRAP/	PROPOSITION = PR;POZ!>N	RATHER = RA</
REMAINING = REM>•N! +	REMEMBER = REM45MB/	RESOLVE = REZALV
RESTING = R4ST! + +	SAY = S@•	SCORE = SK-OOR-
SENSE = S45NSS	SEVEN = S54V4NN	SHALL = >1LL
SHOULD = >%D	SO = SOU	STRUGGLED = STR6GLD
TAKE = T>•K	TASK = T9SSK	TESTING = T45ST! + +
THAT = <:T	THE = <77	THEIR = <@•R
THESE = <EEZ	THEY = < = @•	THIS = = !! SS
THOSE = <OUZ	THUS = = 7SS	TO = T'
UNDER = 7ND/	UNFINISHED = 7NF!N!>D	UPON = 20P67N
US = 77SS	VAIN = V>••N	WAR = WOR
WE = WE	WHAT = WAT	WHETHER = W55</
WHICH = W!TC	WHO = '	WILL = W!L
WORK = W/K	WORLD = W/LD	YEARS = YE•RX

Table 1. Spelling-phonetic Dictionary

view—the compilation of a spelling-phonetic dictionary. I orginally did this by hand but then came up with the thoroughly original idea of letting the computer do it.

Using a somewhat modified Shell-Metzner sort, the program fills an array with the text. Enough space must be DIMed to allow for the whole program. Rather than count the words yourself, type in 997 RESTORE: N = 1 and 998 PRINT N;: READ A$: READ B$: N = N + 1: GOTO 998.

Run 997. When an out-of-data error message results, N is the number of words in your program. I always try to liberally exceed this number in my DIM statements. One can delete 997 and 998 after doing this.

Next, the program eliminates duplicates and alphabetizes the words. Finally, the words are listed on a line printer.

If you use a quick printer or any printer incapable of handling 100 + characters per line, change lines 420, 430 and 440 to read:

```
420 FOR P = 1TON
430 LPRINT A$(N);" = ";B$(N)
440 NEXT P
```

If you don't have a printer, keep this routine in the program just in case you get one. There is no way to access this routine except via a RUN 200 command, and hence no danger of crashing into it.

Once completed, this dictionary is an immense help. For instance look at the Program Listing. I compiled a dictionary for the Gettysburg Address. Later I compiled dictionaries for the Declaration of Independence and the Preamble to the Constitution. These provided me with three-quarters of the words I needed for a Woody Allen routine. (Man does not live by historical quotes alone.)

Finally, there is the display routine in the program. For this some further additions are made; for instance, punctuation would be considered necessary. Insert the necessary marks in the data statements of the correctly spelled words, making sure to quote any commas or colons. Do it after the dictionary is complete: otherwise the dictionary routine will mistakenly consider punctuated words such as "to," "to", and "to" to be different words. This creates an excess of multiple entries.

Pauses can be entered before or after dictionary compilation, since the dictionary routine ignores them. The space is the longer pause. But leading and trailing spaces are ignored in data. I get around this by beginning and ending the pauses with the short pause, 0. Finally, I add / for a carriage return, /// for a CLS, and //// as an ending symbol to make the video display attractive.

If a word will not fit on a line, the routine will automatically bump it to the next line. This avoids splitting words.

In short, the voice synthesizer technique of the Program Listing is divided as follows:

- Phoneme development: lines 10000-10300
- Editing: lines 20000-20070
- Spelling-phonetic dictionary compilation: lines 200-510
- Display: lines 0-130

Generally, I start the text between 1000 and 9999. I keep a master tape of lines 0-999 and 10000-20070 to be used in the development of new programs for the voice synthesizer. For a sample, I present the first selection in the TRS-80 Historical Series here, Lincoln's Gettysburg Address. ■

Program Listing.

```
0 REM THIS IS THE DOING OF CRAIG WERNER
1 REM                      1824 WATSON RD.
2 REM                      ABINGTON COMPUTING GROUP
3 REM                      ABINGTON PA 19001
5 CLS
10 PRINTCHR$(23);:PRINT"LINCOLN'S GETTYSBURG ADDRESS":F
   ORN=1TO1000:NEXT:CLS
20 READA$
30 POKE16383,63:POKE 16383,32
40 FORV=1TOLEN(A$)
50 POKE 16383,ASC(MID$(A$,V,1)):NEXT
60 POKE16383,32:POKE16383,63:POKE16383,32
70 READA$
80 IFPOS(0)+LEN(A$)>63THENPRINTCHR$(13);
90 IFA$="/"THENPRINTCHR$(13);:PRINTTAB(5);:GOTO130
100 IF A$="///"THEN FOR N=1TO500:NEXT:CLS:PRINTTAB(5);:
```

Program continues

Program continues

```
     GOTO130
110 IF  A$="////"THEN 999
120 PRINTA$;" ";
130 FOR N=1TO220:NEXTN:GOTO20
200 CLEAR5000:DEFINTI-N:DIMA$(300),B$(300):ON ERROR GOT
     O 500:N=1
210 READ B$:IF LEFT$(B$,1)="0"READA$:GOTO210
220 B$(N)=B$:READA$:A$(N)=A$
230 N=N+1:GOTO210
300 N=N-1:M=N:GOSUB310:M=N:GOSUB310:GOTO400
310 M=M/2:IFM=0THENRETURN
320 FORJ=1TON-M:I=J
330 L=I+M:IFA$(I)<A$(L)THENGOTO380
340 IFA$(I)=A$(L)THENA$(L)="ZZZ":GOTO380
350 A$=A$(I):B$=B$(I):A$(I)=A$(L):B$(I)=B$(L):A$(L)=A$:
     B$(L)=B$:I=I-M
360 IFI<1GOTO380
370 PRINTA$(I);"  ";:GOTO330
380 NEXTJ:GOTO310
400 FORE=1TON:IFA$(E)="ZZZ"THENA$(E)="":B$(E)="":N=N-1:
     NEXTELSENEXT
410 LPRINT"DICTIONARY OF WORDS":LPRINT"   "
420 FORP=1TONSTEP3
430 LPRINTA$(P);" = ";B$(P);TAB(30);A$(P+1);" = ";B$(P+
     1);TAB(60);A$(P+2);" = ";B$(P+2)
440 NEXTP
450 END
500 IFERL=210THEN300
510 ON ERROR GOTO 0
999 GOTO999
1000 DATA FOOR,FOUR,SK-OORR-,SCORE,9ND,AND,S54V4NN,SEVE
     N,YE*RX,YEARS,20GOU,AGO
1010 DATA ;UR,OUR,F;8<5RX,FATHERS,BR12T0,BROUGHT,F%%=,F
     ORTH,20P67N,UPON
1020 DATA =!!SS,THIS,K;NTIN3NT,"CONTINENT,",@&,A,NUW,NE
     W,N)*>N,"NATYON,"
1030 DATA K6NSEEVD,CONCEIVED,I!N,IN,L#B5RTE,"LIBERTY,",
     99ND,AND,D44D#K)*T3D,DEDICATED
1040 DATA T',TO,=76,THE,PR;POZ!>N,PROPOSITION,"<:T",THA
     T,",,LL",ALL,M45N,MEN
1050 DATA ;8RR,ARE,KRE)*T5D,CREATED,.K-W%L,EQUAL.,0
          0,"///"
1060 DATA N;U,NOW,WE,WE,;AR,ARE,5NG)*GXD,ENGAGED,I!N,IN
     ,@*,A,GR@*T,GREAT,S!V8L,CIVIL
1070 DATA WOR,"WAR,",T45ST!++,TESTING,W55</,WHETHER,"<:
     T",THAT,N)*>N,NATION
1080 DATA OR,OR,43NE,ANY,N)*>N,NATION,SOU,SO,K6NSEEVD,C
     ONCEIVED,99NND,AND
1090 DATA SOU,SO,D44D#K)*T3D,"DEDICATED,",KK9N,CAN,LO+,
     LONG,3NDOR,ENDURE,0     0,"/"
1100 DATA WE,WE,;AR,ARE,M34T,MET,22N,ON,@&,A,GR@*T,GREA
     T,B9T#L,BATTLE
1110 DATA FELD,FIELD,77V,OF,"<:T",THAT,WOR,WAR.,0     0,
     "  "
1120 DATA WE,WE,H9VV,HAVE,KK77M,COME,T',TO,D44D#K)*T,DE
     DICATE,@*,A
1130 DATA POR>6N,PORTION,77V,OF,"<:T",THAT,FELD,FIELD,"
     :X",AS,@*,A
1140 DATA F;#NL,FINAL,R4ST!++,RESTING,PL)*S,PLACE,FOOR,
     FOR,<OUZ,THOSE,',WHO
1150 DATA H9VV,HAVE,G!V4N,GIVEN,<@*R,THEIR,L;#VZ,LIVES,
     "<:T",THAT,WE,WE
1160 DATA M;YT,MIGHT,L!V,LIVE.,0            0,"/"
1170 DATA !TT,IT,!ZZ,IS,2LTUG5>/,ALTOGETHER,F!T!++,FITT
     ING,99NND,AND
1180 DATA PRAP/,PROPER,"<:T",THAT,WE,WE,>%D,SHOULD,D(U,
     DO,<<!!SS,THIS,0            0,"///"
1190 DATA B87T,"BUT,",I!N,IN,@*,A,LARGJ/,LARGER,S45NSS,
     SENSE,WE,WE,KK9N,CAN
1200 DATA N2TT,NOT,D44D#K)*T,DEDICATE - ,WE,WE,KK9N0N2T
     T,CAN NOT
1210 DATA K6NS5KR)*T,CONSECRATE - ,WE,WE,KK9N2TT,CAN NO
     T,"H:LLOUW","HALLOW - "
1220 DATA=!!SS,THIS,GR;UNND,GROUND.  ,0      0,""
1230 DATA <77,THE,BR)*V,BRAVE,M45N,"MEN,",L!V!++,LIVING
     ,99NND,AND,D45D,"DEAD,",',WHO
1240 DATA STR6GLD,STRUGGLED,HERR,"HERE,",H33V,HAVE,K6NS
     5KR)*T45D,CONSECRATED
1250 DATA !TT,IT,0     0,"",F;R,FAR,7B77VV,ABOVE,AOR,OUR
     ,POR,POOR,PAW/,POWER,T',TO
1260 DATA 9DD,ADD,ORR,OR,D!TR9KT,DETRACT.,0       0,"/"
1270 DATA <77,THE,W/LD,WORLD,W!L,WILL,L!TLL,LITTLE,NOUT
     ,NOTE,NOR,NOR
1280 DATA LO++,LONG,REM45MB/,REMEMBER,WAT,WHAT,WE,WE,S@
     *,SAY,HERR,HERE,0  0,""
1290 DATA B87T,BUT,!TT,IT,KK9N,CAN,N45V/,NEVER,FORG45T,
     FORGET,WAT,WHAT
1300 DATA <=@*,THEY,D!DD,DID,HERR,HERE.,0       0,"/"
1310 DATA !TT,IT,!ZZ,IS,FOR,FOR,<77,THE,L!V!++,"LIVING,
     ",RA</,"RATHER,",0     0,""
1320 DATA T',TO,BE,BE,D44D#K)*T3D,DEDICATED,HERR,HERE,T
     ',TO,<77,THE
1330 DATA 7NF!N!>D,UNFINISHED,W/K,WORK,W!TC,WHICH,<=@*,
     THEY,',WHO,F12TT,FOUGHT
1340 DATA HERR,HERE,HAVV,HAVE,=7SS,THUS,F;RR,FAR,SOU,SO
     ,NOUBLE,NOBLY
1350 DATA 9DV9NSD,ADVANCED.,0            0,"///"
```

Program continues

SOFTWARE UNLIMITED

presenting the LARGEST SELECTION OF SOFTWARE EVER ASSEMBLED...

the best available on the market today for TRS-80® Computers

at SUPER DISCOUNT PRICES!

AVALON HILL
- ☐ MIDWAY 13.50
- ☐ NUKE WAR 13.50
- ☐ PLANET MINERS 13.50
- ☐ CONVOY RAIDER 13.50
- ☐ B1 BOMBER 13.50
- ☐ LORDS OF KARMA 18.00
- ☐ CONFLICT 2500 .... 13.50
- ☐ COMPUTER ACQUIRE 18.00

APPARAT
- ☐ NEWDOS/80 .... 125.00

ADVENTURE INTERNATIONAL
- ☐ ADVENTURE HINT BOOK 7.95
- ☐ ADVENTURE #0 .... 6.25
- ☐ ADVENTURE (1,2,3) [D] 35.95
- ☐ ADVENTURE (4,5,6) [D] 35.95
- ☐ ADVENTURE (7,8,9) [D] 35.95
- ☐ ADVENTURE #10 [D] 18.95
- ☐ ADVENTURE (specify 1-10) 13.55
- ☐ MEAN CHECKERS MACHINE 17.95
- ☐ MEAN CHECKERS MACHINE [D] 22.95
- ☐ DR. CHIPS 13.55
- ☐ DR. CHIPS [D] 17.95
- ☐ INTER FICTION SAMPLER [D] 13.55
- ☐ INTER-LOCAL CALLS [D] 17.95
- ☐ INTER-TWO HEADS [D] 17.95
- ☐ INTER-IMPETUOUS [D] 17.95
- ☐ KID-VENTURE 1 .... 17.95
- ☐ STAR TREK 3.5 .... 13.55
- ☐ STAR TREK 3.5 [D] 17.95
- ☐ ZOSSED IN SPACE 13.55
- ☐ MACES #1, BARLOG 31.00
- ☐ MACES #1, SISYPHUS 31.00
- ☐ LUNAR LANDER 13.55
- ☐ LUNAR LANDER [D] 18.95
- ☐ POKER 13.55
- ☐ GALACTIC TRILOGY [D] 35.95
- ☐ SILVER FLASH 13.55
- ☐ SILVER FLASH [D] 17.95
- ☐ MISSILE ATTACK 13.55
- ☐ STAR SCOUT 13.55
- ☐ SPACE INTRUDERS 18.75
- ☐ COMBAT [D] 17.95

ACORN SOFTWARE
- ☐ ATERM 17.95
- ☐ SYSTEM SAVERS 13.55
- ☐ DISASSEMBLER 13.55
- ☐ DISK/TAPE UTILITY 17.95
- ☐ STAR TREK SIMULATION 8.95
- ☐ GAMMON CHALLENGER 13.55
- ☐ PIGSKIN 13.55
- ☐ ULTRA TREK 13.55
- ☐ SPACE WAR 8.95
- ☐ WARP/LANDER 8.95
- ☐ BASKETBALL [D] 18.95
- ☐ BASKETBALL 13.55
- ☐ DUEL-N-DROIDS [D] 18.95
- ☐ DUEL-N-DROIDS 13.55
- ☐ INVADERS FROM SPACE 13.55
- ☐ INVADERS FROM SPACE [D] 18.95
- ☐ PIGSKIN [D] 18.95
- ☐ PINBALL 13.55
- ☐ PINBALL [D] 18.95
- ☐ EVEREST EXPLORER 13.45
- ☐ EVEREST EXPLORER [D] 18.95
- ☐ SUPERSCRIPT [D] 28.95

MED SYSTEMS
- ☐ DEATHMAZE 5000 .... 11.65
- ☐ RATS REVENGE 11.65
- ☐ REALITY ENDS 8.95

AUTOMATED SIMULATION
- ☐ STAR WARRIOR [CASS. OR DISK] 35.95
- ☐ THREE PACK [D] 45.00
- ☐ STARFLEET [CASS. OR DISK] 22.50
- ☐ INVASION [CASS. OR DISK] 22.50
- ☐ APSHAI [CASS. OR DISK] 35.95
- ☐ RYN (CASS., OR DISK) 17.95
- ☐ MORLOC [CASS. OR DISK] 17.95
- ☐ RIGEL [CASS. OR DISK] 26.95
- ☐ HELLFIRE [CASS. OR DISK] 35.95

LOBO
- ☐ LDOS [D] 125.00

COMPUTER SIMULATIONS COMPANY
- ☐ BATTLE OF BULGE-BASTGONES 17.95
- ☐ D-DAY INVASION OF FRANCE 17.95
- ☐ DARK KINGDOM 11.75
- ☐ DOG RACE, COLOR ONLY [C] 5.95
- ☐ ELECTRIC ALARM 4.95
- ☐ EMPIRE STRIKES BACK 13.55
- ☐ GREAT DICTATOR 7.25
- ☐ JEDI KNIGHT 11.75
- ☐ LUNAR ENCOUNTER 11.75
- ☐ MERCENARY 8.95
- ☐ MICRO ARCADE 13.55
- ☐ SHARK, COLOR ONLY [C] 5.95
- ☐ SLOT MACHINE 7.25
- ☐ BATTLE OF BULGE-ST. VITH 13.55
- ☐ STAR CRUISERS 13.55
- ☐ TYPE WRITERS 17.95
- ☐ U-BOAT, COLOR ONLY [C] 5.95
- ☐ ORION WAR, COLOR ONLY [C] 5.95
- ☐ POLARIS, COLOR ONLY [C] 5.95
- ☐ ASTEROID PIRATES 13.55

INSTANT SOFTWARE
- ☐ AIR FLIGHT SIMULATION 8.95
- ☐ AIRMAIL PILOT 8.95
- ☐ ASTEROIDS [D] 17.95
- ☐ BASIC PROGRAMMING ASSISTANT 13.55
- ☐ BALL TURRET GUNNER 8.95
- ☐ BATTLEGROUND 8.95
- ☐ CHECK MANAGEMENT [D] 36.55
- ☐ CHESSMATE-80 .... 17.95
- ☐ COSMIC PATROL 13.55
- ☐ COSMIC PATROL [D] 17.95
- ☐ DAREDEVIL 8.95
- ☐ DISASSEMBLER 8.95
- ☐ DISK EDITOR [D] 36.55
- ☐ DISK SCOPE [D] 17.95
- ☐ DLDIS [D] 17.95
- ☐ DEVICE DRIVERS 17.95
- ☐ ENHANCED BASIC 22.50
- ☐ FLIGHT PATH 8.95
- ☐ FLYING CIRCUS [D] 36.55
- ☐ I.Q. TEST 8.95
- ☐ INVADERS 8.95
- ☐ INVESTORS PARADISE 8.95
- ☐ IRV 22.50
- ☐ IRV [D] 27.55
- ☐ JET FLIGHTER PILOT 13.55
- ☐ MONEY MADNESS 8.95
- ☐ NIGHT FLIGHT 8.95
- ☐ ONE-D MAILING LIST [D] 22.50
- ☐ OTHELLO 8.95
- ☐ PROGRAMMER'S CONVERTER 8.95
- ☐ SKIRMISH-80 .... 8.95
- ☐ TLDIS 13.55
- ☐ UTILITY I 8.95
- ☐ UTILITY II 8.95
- ☐ WORDSLINGER 26.00

PERSONAL SOFTWARE
- ☐ CCA MGMT-TRS-80 .... 85.00
- ☐ MONTY MONOPOLY [D] 35.95
- ☐ ZORK [D] 35.95

BIG FIVE SOFTWARE
- ☐ ATTACK FORCE (MODEL 1 OR 3) 14.30
- ☐ GALAXY INVASION (MODEL 1 OR 3) 14.30
- ☐ METEOR MISSION II (CASS. OR DISK) 14.30
- ☐ SUPER NOVA [CASS. OR DISK] 14.30
- ☐ COSMIC FIGHTER [CASS. OR DISK] 14.30

HAYDEN
- ☐ SARGON II 25.00
- ☐ SARGIN II [D] 30.00

QUALITY SOFTWARE
- ☐ DEBUG 13.55
- ☐ FASTGAMMON 17.95
- ☐ LOWBALL POKER 10.45
- ☐ BANKSHOT 8.95
- ☐ POKER PETE 10.45
- ☐ RUMMY MASTER 10.45
- ☐ 3D TIC TAC TOE 13.55
- ☐ QS LIGHT PEN 17.95
- ☐ SKETCH 80 .... 13.55

STRATEGIC SIMULATIONS
- ☐ COMPUTER BISMARCK [D] 51.50
- ☐ COMPUTER BISMARCK 42.00

SUB-LOGIC
- ☐ T80-FS1 FLIGHT SIMULATOR 22.00
- ☐ 3D GRAPHICS 26.50

BOTTOM SHELF
- ☐ ANALYSIS PAD [D] 90.00
- ☐ BUSINESS MAIL [D] 100.00
- ☐ CHECKBOOK II [D] 44.50
- ☐ CHECK REGISTER [D] 67.00
- ☐ HEAD CLEANER [D] 17.00
- ☐ INFO SYSTEM [D] 44.50
- ☐ LIBRARY 100 .... 44.50
- ☐ SYSTEM DOCTOR [D] 35.50
- ☐ SYSTEM DOCTOR 26.00

MICROSOFT SOFTWARE
- ☐ ADVENTURE [D] 25.50
- ☐ ASSEMBLY DEVELOPMENT [D] 80.00
- ☐ BASIC COMPILER [D] 175.00
- ☐ EDITOR/ASSEMBLER 25.00
- ☐ FORTRAN COMPILER [D] 80.00
- ☐ LEVEL III BASIC 44.00
- ☐ MuMATH [D] 64.00
- ☐ OLYMPIC DECATHALON (CASS OR DISK) 20.00
- ☐ TYPING TUTOR 13.55

ABC
- ☐ LAZY WRITER [D] 112.50

TPG
- ☐ DEATH DREADNAUGHT 13.45
- ☐ DEATH DREADNAUGHT [D] 17.95
- ☐ DEADLY DUNGEON 13.45
- ☐ OMNI TERM [D] 85.00
- ☐ DRAGON QUEST 14.35
- ☐ DRAGON QUEST [D] 19.75
- ☐ TRIPLE ADVENTURE [D] 31.00

If you don't see it listed, write...we probably have it in stock!

Check program desired.
Complete ordering information
and mail entire ad.
Immediate Shipments from stock.

Ship the above programs as checked to:

Mr./Mrs. ____________

Address ____________

City ____________

State ____________ Zip ______

I have a ____________ (Name of Computer)

with ____________ K memory

80 MICRO-AUG. '81

Number of Programs Ordered .... ______

Amount of order .... ______

N.Y. residents add Sales Tax .... ______

Add shipping anywhere in the U.S. 2.00

Total amount enclosed ______

Charge my: ☐ Master Charge ☐ Visa

Signature ____________

Card No. ______ Expires ______

Personal Checks please allow 3 weeks.

KEY:
C-TRS-80 Color
D-on Disc
If not marked - Cassette

Mail to: **DIGIBYTE SYSTEMS CORP.**

31 East 31st Street, New York, N.Y. 10016
(212) 889-8975 ✓226

TRS-80 is a registered trademark of TANDY CORP.

Prices subject to change without notice.

Program continues

```
1360 DATA !TT,IT,!ZZ,IS,RA</,RATHER,FOR,FOR,77SS,US,T',
     TO,BE,BE,HERR,HERE
1370 DATA D44D#K)*T3D,DEDICATED,T',TO,<77,THE,GR)*T,GRE
     AT
1380 DATA T9SSK,TASK,REM)*N!+,REMAINING,BEFOR,BEFORE,77
     SS,US - ,0      0,""
1390 DATA "<:T",THAT,FR8MM,FROM,<EEX,THESE,;N5RD,HONORE
     D,D45D,DEAD,WE,WE
1400 DATA T)*K,TAKE,<77,THE,!NKRESD,INCREASED,D!VOU>N,D
     EVOTION,T',TO,"<:t",THAT
1410 DATA "K,8Z",CAUSE,FOR,FOR,W!TC,WHICH,<@&,THEY,HERR
     ,HERE,G)*V,GAVE
1420 DATA <77,THE,L9SST,LAST,F%%LLL,FULL,M45X/,MEASURE,
     ",V",OF,D!VOU>N,DEVOTION - ,0       0,""
1430 DATA "<:T",THAT,WE,WE,HERR,HERE,;!LE,HIGHLY,REZALV
     ,RESOLVE,"<:T",THAT
1440 DATA <EEZ,THESE,D45D,DEAD,>1LL-,SHALL,N2TT,NOT,H9V
     V,HAVE,D;!D,DIED
1450 DATA I!N,IN,V)**N,VAIN - ,0       0,""
1460 DATA "<:T",THAT,=!!SS,THIS,N)*>N,"NATION,",7ND/,UN
     DER,GAADD,"GOD,",>!LL,SHALL
1470 DATA H9VV,HAVE,`&,A,NUU,NEW,B/==,BIRTH,77V,OF,FRED
     8M,FREEDOM - ,0       0,""
1480 DATA 9ND,AND,"<:T",THAT,G8V/M45NT,GOVERNMENT,77F,O
     F,<77,THE,PEPL,"PEOPLE,"
1490 DATA "B,&",BY,<77,THE,PEPL,"PEOPLE,",FOR,FOR,<77,T
     HE,PEPL,"PEOPLE,",>1LL,SHALL
1500 DATA N2TT,NOT,P45R!>,PERISH,FR7MM,FROM,<77,THE,8/=
     ==,EARTH.,0       0,"////"
2000 RESTORE
2001 INPUTB
2002 FORN=1TOB
2003 READA$:NEXT:GOTO10
10000 INPUTA$
10010 PRINT@992,"? ";A$;" ?"
10020 PRINT@0,"";
10030 GOTO10000
20000 RESTORE
20010 INPUT B
20020 FORN=1TOB
20030 READB$:READA$:NEXT
20040 READA$:PRINT@992," ?";A$;" ?";
20050 READA$:PRINT`994,A$;"    "
20060 FORN=1TO280:NEXTN
20070 GOTO20040
```

✓See List of Advertisers on page 322

A record-keeping program for teachers.

The Green Register

Glynn Paul LaBorde
Box 63
Angola, LA 70712

Every school teacher has to keep student records. At the beginning of the school year, you are faced with the "green register," as we call it in Louisiana. Whatever name you use makes no difference—every bit of information must be perfect, with no allowance for typographical errors. Every change in a student's status must also be recorded. This record keeping program will not free you from the drudgery of the "green register," but it may make it a little easier to live with.

My students are convicts at the Louisiana State Penitentiary at Angola, Louisiana. Because of the large variation in our school enrollment, a simple "green register" would not serve my needs. This program is not so specialized, however, that you can't use it in a public school or adult education setting. With only two or three changes it can be adapted for use by any teacher.

When powering up your system, answer the memory size question with 31867. This is necessary for the KBeepFix and Babybeep programs to run correctly. Edit line 100 of KBeepFix (*80 Microcomputing*, February, 1980) as follows: 100 DATA 211, 250,197,6,64,16,254,193,16,242, 241,225,193,195,251,3. The change from 255 to 250 will silence the beep but will not affect the debounce and repeat features. When I first wrote the record keeping program, I often wished for an audio cue to let me know when sorting was complete or when the files were loaded. Well, Babybeep came to my rescue with the April 1980 issue of *80 Microcomputing*. Babybeep is used throughout my record keeping programs and adds a certain amount of flair to the operation of the program.

If you don't want audio cues, just remove the appropriate M = USR(0), POKE 16526,nnn, and data statements at the beginning of the program. To increase the number of records, simply reduce the number of data arrays from eight to some lesser number and clear more and more memory until you find the maximum amount that can be cleared while still allowing the program to run. I have a 16K Level II machine and have found 3000 bytes of space sufficient for 100 records, even with 20-character names, five-character student numbers, eight-character dates of birth, four-character reading, math and language scores, nine-character class names, and 20-character dormitory names. Array M$ used for the various messages saves many program lines and makes the program shorter.

The Program

When the program is run, the screen clears for a few seconds while the message array, M$, is being filled. The master menu selections of Load, Save, Search, Add, List, Sort and Lock are printed on the left side of the screen. A message near the top center of the screen reads Master Program Ready.

After pressing 1 for Load, the master menu is erased and a Load Subroutine Engaged message appears near the top center of the screen. In place of the master menu, a single item menu appears as 1 Cancel. Therefore, if you mistakenly entered the load subroutine, you can gracefully return to the master menu without Breaking and therefore losing all data. Pressing any other key turns on the cassette deck and begins the loading.

The first message after loading begins is nn Records To Be Loaded. As each record is loaded, its number in the file is printed along with the person's name. This slows the loading process, but I feel more comfortable knowing I'm loading records and not garbage, which happens from time to time with an '80 cassette system. After all the files are loaded, the program returns to the master menu.

Choose 2 for Save and once again the master menu is erased and replaced by 1 Cancel. Pressing any other key begins data recording onto a cassette tape. This subroutine also displays the number of records to be recorded and the number of each file and the name of each person as it is recorded. After all files have been recorded, the program returns to the master menu.

Choose 3 for Search and the master menu is replaced by a search menu with the choices 1 Cancel, 2 Name, 3 Part Name and 4 Number. When a search by name is requested, the program responds with Name ?, at which time you must enter the full name (last then first) of the person you want. I often use the 3 Part Name command. When a partial name search is requested, you enter whatever part of the person's name you remember. The computer, using the MID$ function, searches the name array until it gets a match. It will stop at Smith, Bob, if you entered Smith as the partial name. If the Smith you wanted was Smith, John, simply step forward through the arrays.

Search by number will hunt the student number array until a match is found. Regardless of the type search being conducted, a No Information On _______________ message is printed near the top center of the screen when the computer fails to find a match. Control then returns to the search menu.

If the file is located, the search menu is replaced by four other menu selections: 1 Cancel, 2 Forward, 3 Reverse and 4 Delete. 1 Cancel returns you to the search menu. The next file in the various arrays will replace the present file on the monitor with 2 Forward. When you reach the end of the file, a File Ends Here message appears near the bottom of the screen and control returns to the search menu. 3 Reverse is just what it sounds like, backing you up to the beginning of the files, one record at a time. When the first file is reached, a File Starts Here message is printed and control returns to the search menu.

There are several safeguards built into 4 Delete. Pressing 4 replaces the menu with 1 Cancel and 4 Delete, thereby keeping you from accidentally wiping out all records on a student. Hit 4 again and the computer begins to delete the record displayed by moving all other records down one at a time. Because of the garbage collection, deleting a file can take as long as three or four minutes, especially if the record being deleted was near the beginning of the file. To reassure myself that the computer was not in some type of endless loop, I added a line or so to the program so that the numbers of the files are displayed as the computer moves them down.

Editing a student's record is also simple. A typical record might look like this:

Name	Smith Bob
PMB Number	12345
Dorm	4 East
Birthdate	2-8-43
Class	Sophomore
Reading	8.5
Math	10.2
Language	7.7
Average	8.8

To edit any of the eight data arrays, hit N for the name, P for the student number, D for the dormitory, B for the birthdate, C for the class, R for the reading score, M for the math score, or L for the language score. A question mark prompt appears, then you type in the new data. If no change was desired, hit Enter without typing in anything and the student's record will not be changed.

After any change, the updated file is displayed. The average of the math, reading and language scores is automatically calculated to the nearest tenth of a grade level. If the student scores 12.0 or better on all three tests and the average score 13.0 or better, the student is eligible, under Louisiana law, to take the G.E.D. test. Those of you in public schools will have no use for the Eligible for GED Test message that appears below the Average line. Those of you in adult education situations in other states will have to make the appropriate changes in line 450.

Pressing 4, while in the master menu, places you in the Add or Build a File mode. The master menu is erased and a Name? prompt appears near the top center of the screen. After the name is entered, a PMB? prompt appears. This continues through each data array. After the language score is entered, another Name? prompt appears and the process repeats itself. To exit the add mode, enter "end" for the name of the next file. This returns you to the master menu.

The eight data arrays are dimensioned for 100 files. There-

A	used in ONAGOTO statements throughout the program
A$	used in sorts as temporary storage and in INKEY$ function
BI$	date of birth array
CL$	class name array
D$	name of class to be listed
DO$	dormitory name array
EOL	total number of records
I	pointer used in sort routines
J	pointer used in sort routines
K	pointer used in sort routines
L	pointer used in sort routines
LA	language score array
M	pointer used in sort routines
M$	message array
MA	math score array
N	used in sorts as temporary storage and as student number being searched
N$	name of student being searched
NU	student number array
P	used in print at statements
Q%	delay timer in line 260
RE	reading score array
S	flag used in sorts and keyboard lock
X	used in calls to machine language subroutines
Y%	used in MID$ function in partial name searches
Z%	timing counter to produce "heartbeat" in line 190

List of Variables

✓See List of Advertisers on page 322

fore, if you reach the end of available array space, a File Full message appears at the top center of the screen and you are returned to the master menu.

The list subroutine, or 5, replaces the master menu with a list menu of 1 Cancel, 2 All, 3 By Class. 1 Cancel works just as all the others; you return to the master menu. 2 All lists only the name, PMB number and dormitory of each student on file, displaying 10 records, pausing a few seconds, and then replacing those 10 with another 10, etc. 3 By Class will respond with Class?, at which time you must enter the class you wish listed. The program then searches the class array, CL$, and prints those students' names, numbers and dormitories. After a listing is complete, control returns to the list menu.

When you choose 6 Sort, the master menu is replaced with 1 Cancel, 2 Alpha, 3 Numeric. To return control the the master menu choose 1 Cancel. To sort by name or number, choose 2 Alpha and 3 Numeric respectively. To prevent accidentally beginning a sort, there are two safeguards. After hitting either 2 or 3, the sort menu is replaced with 1 Cancel. This cancel returns you to the sort menu.

To begin the sort, you must enter a seven-character unlock code. If any character is keyed in incorrectly, you must re-enter the unlock code from the very beginning. In this way, it is impossible to begin a sort accidentally. The sort used in this version of the record keeping program is the Shell-Metzner Sort.

Sorting never takes more than 15 minutes with 100 files in memory. With 70 files in memory, sort time is cut in half. This compares very favorably with eight hours for some of the sort routines suggested by Radio Shack. While sorting, three numbers are displayed in the upper left portion of the screen to reassure you that files are being sorted. There is an occasional pause when the computer takes a deep breath while going through a garbage collection routine, but don't worry about it. The Shell-Metzner sort is still light-years ahead of other Basic sorts.

After hitting 7 for Lock, the master menu is erased and the message Keyboard Lock Subroutine Engaged is displayed. You'll really appreciate this lock command when you have to step out of the room, leaving your system alone with 300 curious fingers just itching to press a key or two. Only the special seven-digit code mentioned earlier will return you to the master menu. (Really mischievous students could always pull the power plug, turn the system off, or hit the Break key. Nothing is really foolproof.)

Non-teachers can use this program for such things as keeping records on business clients, employees, or family members. The most limiting factor concerning the uses of a computer is your imagination. ■

The program is available on cassette from the author.

Program Listing

```
100 '----STUDENTS RECORDS PROGRAM REVISED APRIL 11, 1980 UTILIZI
NGSHELL-METZLER SORT
110 CLS:CLEAR3000:DIMM$(47),NA$(105),NU(105),DO$(105),CL$(105),B
I$(105),RE(105),MA(105),LA(105):EOL=1:X=1:FORQ%=32000TO32303:REA
DA:POKEQ%,A:NEXT:POKE16527,125:FORQ%=1TO44:READM$(Q%):NEXT:RESTO
RE
120 DATA14,8,6,192,205,27,126,197,16,254,193,16,247,13,121,254,0
,32,239,201,14,16,33,32,0,205,27,126,6,160,16,254,43,124,181,32,
244,33,0,10,43,124,181,32,251,13,121,254,0,32,227,201,14,10,6,48
,205,27,126,197,6,255,16,254,193,15,245,6,80,205,27,126,197,6,96
,16
130 DATA254,193,16,245,13,121,254,0,32,224,201,6,64,205,27,126,1
97,6,192,16,254,193,16,245,6,128,205,27,126,197,6,80,16,254,193,
16,245,6,255,205,27,126,197,6,48,16,254,193,16,245,6,176,205,27,
126,197,6,176,16,254,193,16,245,6,255,205,27,126,197,6,32,16,254

140 DATA193,16,245,14,16,6,16,205,27,126,197,6,255,16,254,193,16
,245,13,121,254,0,40,10,33,0,10,43,124,181,32,251,24,227,201,14,
8,6,0,205,27,126,197,4,120,246,0,32,250,193,4,120,246,0,32,239,6
,255,205,27,126,197,16,254,193,16,247,13,121,254,0,32,220,201,6
150 DATA128,205,27,126,197,6,128,16,254,193,16,245,201,6,192,205
```

Program continues

```
,27,126,197,6,64,16,254,193,16,245,201,14,32,33,10,0,205,27,126,
6,255,16,254,43,124,181,32,244,33,0,1,43,124,181,32,251,13,121,2
54,0,32,227,201,58,61,64,230,253,198,2,211,255,214,2,211,255,201
,68
160 DATA32,75,73,84,83,90:M$(45)=STRING$(12," "):M$(46)=STRING$(
40," "):M$(47)=STRING$(44," ")
170 CLS
180 GOSUB230:FORQ%=1TO7:PRINT@128*Q%-61,M$(Q%);:NEXT:PRINT@96,M$
(29);
190 A$=INKEY$:Z%=Z%+1:IFZ%=40THENPOKE16526,223:X=USR(0):Z%=0
200 IFA$<"1"ORA$>"7"THEN190
210 POKE16526,52:X=USR(0)
220 A=VAL(A$):ONAGOTO270,310,350,560,620,690,910:GOTO190
230 PRINT@85,M$(46);:RETURN
240 FORQ%=67TO835STEP128:PRINT@Q%,M$(45);:NEXT:RETURN
250 FORQ%=212TO852STEP64:PRINT@Q%,M$(47);:NEXT:RETURN
260 FORQ%=1TO1000:NEXT:RETURN
270 GOSUB230:PRINT@95,M$(30);:GOSUB240:PRINT@67,M$(8);
280 A$=INKEY$:IFA$=""THEN280ELSEIFA$="1"THENPOKE16526,251:X=USR(
0):GOSUB240:GOTO180
290 GOSUB240:INPUT#-1,EOL:GOSUB230:PRINT@95,EOL;"RECORDS TO BE L
OADED";
300 FORA=1TOEOL:INPUT#-1,NA$(A),NU(A),DO$(A),CL$(A),BI$(A),RE(A)
,MA(A),LA(A):GOSUB230:PRINT@95,"FILE #";A;TAB(5)NA$(A);:NEXT:POK
E16526,52:X=USR(0):GOSUB240:GOTO180
310 GOSUB230:PRINT@95,M$(31);:GOSUB240:PRINT@67,M$(8);
320 A$=INKEY$:IFA$=""THEN320ELSEIFA$="1"THENPOKE16526,251:X=USR(
0):GOSUB240:GOTO180
330 GOSUB240:PRINT#-1,EOL:GOSUB230:PRINT@95,EOL;"RECORDS TO BE S
AVED";
340 FORA=1TOEOL:PRINT#-1,NA$(A),NU(A),DO$(A),CL$(A),BI$(A),RE(A)
,MA(A),LA(A):GOSUB230:PRINT@95,"FILE #";A;TAB(5)NA$(A);:NEXT:POK
E16526,52:X=USR(0):GOSUB240:GOTO180
350 GOSUB230:PRINT@94,M$(32);:GOSUB240:PRINT@67,M$(8);:PRINT@195
,M$(9);:PRINT@323,M$(10);:PRINT@451,M$(11);
360 A$=INKEY$:IFA$<>"1"ANDA$<>"2"ANDA$<>"3"ANDA$<>"4"THEN360ELSE
A=VAL(A$):ONAGOTO370,380,550,390
370 POKE16526,251:X=USR(0):GOSUB250:GOSUB240:GOTO180
380 GOSUB230:PRINT@94,M$(15);:INPUTN$:GOSUB230:PRINT@94,M$(32);:
FORA=0TOEOL:IFN$=NA$(A)THEN400ELSENEXT:GOSUB230:PRINT@94,M$(28);
:POKE16526,251:X=USR(0):GOTO350
390 GOSUB230:PRINT@94,M$(16);:INPUTN:GOSUB230:PRINT@94,M$(32);:F
ORA=0TOEOL:IFN=NU(A)THEN400ELSENEXT:GOSUB230:PRINT@94,M$(28);:PO
KE16526,251:X=USR(0):GOTO350
400 POKE16526,0:X=USR(0)
410 AV=(RE(A)+MA(A)+LA(A))/3:AV=AV+.05:AV=INT(AV*10)/10:PRINT@21
2,M$(46);:PRINT@212,M$(15);:PRINT@224,NA$(A);:PRINT@276,M$(46);:
PRINT@276,M$(16);:PRINT@287,NU(A);:PRINT@340,M$(46);:PRINT@340,M
$(17);
420 PRINT@352,DO$(A);:PRINT@404,M$(46);:PRINT@404,M$(18);
430 PRINT@416,CL$(A);:PRINT@468,M$(46);:PRINT@468,M$(19);:PRINT@
480,BI$(A);:PRINT@532,M$(46);:PRINT@532,M$(46);:PRINT@532,M$(20)
;:PRINT@543,RE(A);:PRINT@596,M$(46);:PRINT@596,M$(21);:PRINT@607
,MA(A);:PRINT@660,M$(46);:PRINT@660,M$(22);:PRINT@671,LA(A);
440 PRINT@724,M$(46);:PRINT@724,M$(23);:PRINT@735,AV;
450 IFRE(A)>=12ANDMA(A)>=12ANDLA(A)>=12ANDAV>=13THENPRINT@852,M$
(25);:ELSEPRINT@852,M$(46);
460 PRINT@67,M$(8);:PRINT@195,M$(12);:PRINT@323,M$(13);:PRINT@45
1,M$(14);
470 A$=INKEY$:IFA$=""THEN470:ELSEIFA$="1"THENGOSUB250:GOTO350:EL
SEIFA$="2"THEN510:ELSEIFA$="3"THEN500:ELSEIFA$="4"THEN520
480 PRINT@852,M$(46);:PRINT@852,"";:IFA$="N"THENINPUTNA$(A):GOTO
410:ELSEIFA$="P"THENINPUTNU(A):GOTO410:ELSEIFA$="D"THENINPUTDO$(
A):GOTO410:ELSEIFA$="C"THENINPUTCL$(A):GOTO410
490 IFA$="B"THENINPUTBI$(A):GOTO410:ELSEIFA$="R"THENINPUTRE(A):G
OTO410:ELSEIFA$="M"THENINPUTMA(A):GOTO410:ELSEIFA$="L"THENINPUTL
A(A):GOTO410:ELSE470
500 A=A-1:IFA<1THENPRINT@852,M$(46);:PRINT@852,M$(26);:GOSUB260:
GOSUB250:GOTO350:ELSE410
510 A=A+1:IFA>EOLTHENPRINT@852,M$(46);:PRINT@852,M$(27);:GOSUB26
0:GOSUB250:GOTO350:ELSE410
520 GOSUB240:PRINT@67,M$(8);:PRINT@195,M$(14);
530 A$=INKEY$:IFA$=""THEN530ELSEIFA$="1"THENGOSUB240:GOTO460:ELS
EIFA$="4"THEN540:ELSE530
540 GOSUB240:IFNU(A)=0THEN350:ELSEFORA=ATOEOL:PRINT@323,A;:B=A+1
:NA$(A)=NA$(B):NU(A)=NU(B):DO$(A)=DO$(B):CL$(A)=CL$(B):BI$(A)=BI
$(B):RE(A)=RE(B):MA(A)=MA(B):LA(A)=LA(B):PRINT@451,A;:NEXT:EOL=E
OL-1:GOSUB240:GOSUB250:GOTO350
550 GOSUB230:PRINT@94,M$(15);:INPUTN$:GOSUB230:PRINT@94,M$(32);:
FORA=0TOEOL:FORY%=1TOLEN(NA$(A)):IFMID$(NA$(A),Y%,LEN(N$))=N$THE
N400:ELSENEXT:NEXT:GOSUB230:PRINT@94,M$(28);:POKE16526,251:X=USR
(0):GOTO350
560 GOSUB240:GOSUB230:PRINT@95,M$(33);:FORQ%=1TO500:NEXT:GOSUB23
0:FORA=EOL+1TO100:PRINT@94,M$(15);:INPUTNA$(A):POKE16526,223:X=U
SR(0)
570 IFNA$(A)="END"THENEOL=A-1:POKE16526,251:X=USR(0):GOTO170
580 GOSUB230:PRINT@94,M$(16);:INPUTNU(A):GOSUB230:PRINT@94,M$(17
);:INPUTDO$(A):GOSUB230:PRINT@94,M$(18);:INPUTCL$(A):GOSUB230
590 PRINT@94,M$(19);:INPUTBI$(A)
600 GOSUB230:PRINT@94,M$(20);:INPUTRE(A):GOSUB230:PRINT@94,M$(21
);:INPUTMA(A):GOSUB230:PRINT@94,M$(22);:INPUTLA(A):GOSUB230:NEXT

610 GOSUB230:PRINT@94,"FILE FULL";:EOL=100:GOSUB260:GOTO180
620 GOSUB230:PRINT@94,M$(34);:GOSUB240:PRINT@67,M$(8);:PRINT@195
,M$(37);:PRINT@323,M$(38);
630 P=0
640 A$=INKEY$:IFA$<>"1"ANDA$<>"2"ANDA$<>"3"THEN640:ELSEA=VAL(A$)
:ONAGOTO370,650,670:GOTO640
650 GOSUB240:FORA=1TOEOL:P=P+1:PRINT@P*64+148,NA$(A);:PRINT@P*64
+167,NU(A);:PRINT@P*64+174,DO$(A);:PRINT@P*64+179,CL$(A);:IFP/10
=INT(P/10)THENGOSUB260:P=0:GOSUB250
660 NEXT:GOSUB260:GOSUB260:GOSUB250:GOTO620
670 GOSUB230:PRINT@94,M$(18);:INPUTD$:GOSUB230:PRINT@94,M$(34);:
FORA=1TOEOL:IFD$=CL$(A)THENP=P+1:PRINT@P*64+148,NA$(A);:PRINT@P*
64+167,NU(A);:PRINT@P*64+174,DO$(A);:PRINT@P*64+179,CL$(A);:IFP/
10=INT(P/10)THENGOSUB260:P=0:GOSUB250
680 NEXT:GOSUB260:GOSUB260:GOSUB250:GOTO620
690 GOSUB230:PRINT@94,M$(35);:GOSUB240:PRINT@67,M$(8);:PRINT@195
,M$(41);:PRINT@323,M$(42);
700 A$=INKEY$:IFA$<"1"ORA$>"3"THEN700ELSEA=VAL(A$):ONAGOTO370,71
0,810
710 GOSUB230:PRINT@94,M$(24);:PRINT@195,M$(45);:PRINT@323,M$(45)
```

Program continues

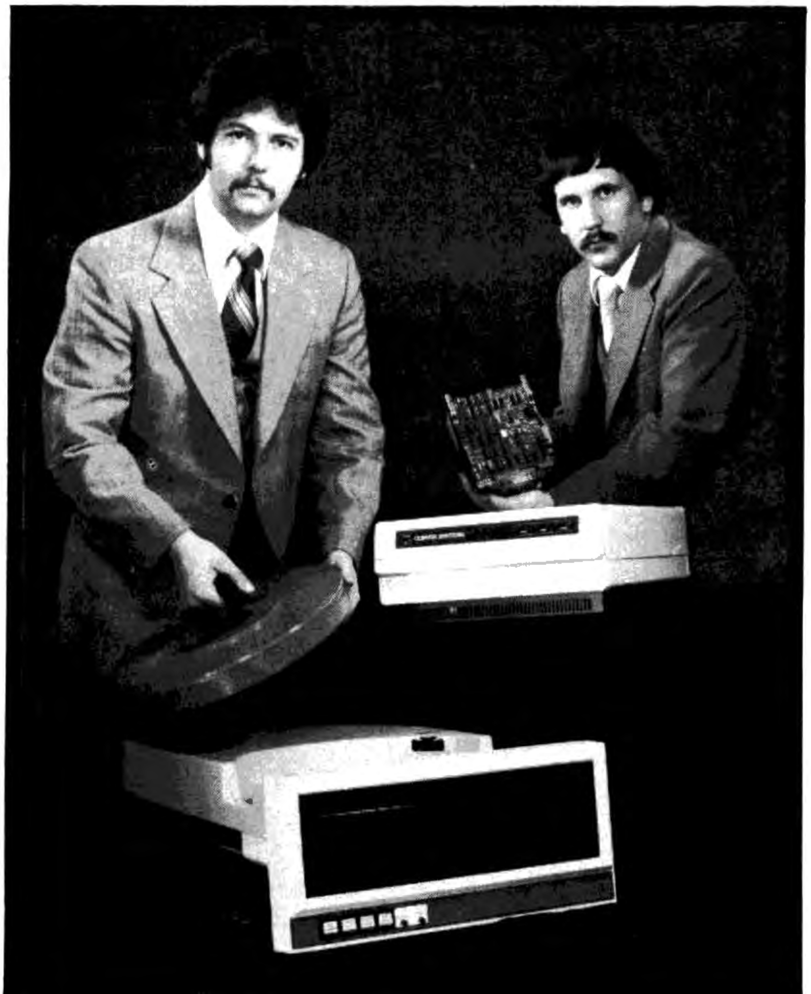
✓See List of Advertisers on page 322

```
;:S=2:GOSUB920:IFS=1THENS=2:GOTO690:ELSEGOSUB230:PRINT@94,M$(43)
;:M=EOL:GOSUB240
720 M=INT(M/2):IFM=0THEN800ELSEJ=1:K=EOL-M
730 I=J
740 L=I+M
750 IFNA$(I)<=NA$(L)THEN780
760 A$=NA$(I):NA$(I)=NA$(L):NA$(L)=A$:N=NU(I):NU(I)=NU(L):NU(L)=
N:A$=DO$(I):DO$(I)=DO$(L):DO$(L)=A$:A$=CL$(I):CL$(I)=CL$(L):CL$(
L)=A$:A$=BI$(I):BI$(I)=BI$(L):BI$(L)=A$
770 N=RE(I):RE(I)=RE(L):RE(L)=N:N=MA(I):MA(I)=MA(L):MA(L)=N:N=LA
(I):LA(I)=LA(L):LA(L)=N
780 I=I-M:PRINT@69,I;:PRINT@133,L;:PRINT@197,M;:IFI<1THEN790ELSE
740
790 J=J+1:IFJ>KTHEN720ELSE730
800 POKE16526,87:X=USR(0):GOTO170
810 GOSUB230:PRINT@94,M$(24);:PRINT@195,M$(45);:PRINT@323,M$(45)
;:S=2:GOSUB920:IFS=1THENS=2:GOTO690:ELSEGOSUB230:PRINT@94,M$(44)
;:M=EOL:GOSUB240
820 M=INT(M/2):IFM=0THEN900ELSEJ=1:K=EOL-M
830 I=J
840 L=I+M
850 IFNU(I)<=NU(L)THEN880
860 A$=NA$(I):NA$(I)=NA$(L):NA$(L)=A$:N=NU(I):NU(I)=NU(L):NU(L)=
N:A$=DO$(I):DO$(I)=DO$(L):DO$(L)=A$:A$=CL$(I):CL$(I)=CL$(L):CL$(
L)=A$:A$=BI$(I):BI$(I)=BI$(L):BI$(L)=A$
870 N=RE(I):RE(I)=RE(L):RE(L)=N:N=MA(I):MA(I)=MA(L):MA(L)=N:N=LA
(I):LA(I)=LA(L):LA(L)=N
880 I=I-M:PRINT@69,I;:PRINT@133,L;:PRINT@197,M;:IFI<1THEN890ELSE
840
890 J=J+1:IFJ>KTHEN820ELSE830
900 GOTO800
910 S=0:GOSUB230:GOSUB240:PRINT@94,M$(36);:GOSUB240:GOSUB920:GOT
O170
920 A$=INKEY$:IFA$=""THEN920ELSEIFA$="1"ANDS=2THENS=1:RETURN:ELS
EIFA$="G"THEN930ELSE920
930 A$=INKEY$:IFA$=""THEN930ELSEIFA$="P"THEN940ELSE920
940 A$=INKEY$:IFA$=""THEN940ELSEIFA$="L"THEN950:ELSE920
950 A$=INKEY$:IFA$=""THEN950ELSEIFA$="1"THEN960:ELSE920
960 A$=INKEY$:IFA$=""THEN960ELSEIFA$="1"THEN970:ELSE920
970 A$=INKEY$:IFA$=""THEN970ELSEIFA$="2"THEN980:ELSE920
980 A$=INKEY$:IFA$=""THEN980ELSEIFA$="9"THEN990:ELSE920
990 RETURN
1000 DATA1-LOAD,2-SAVE,3-SEARCH,4-ADD,5-LIST,6-SORT,7-LOCK,1-CAN
CEL,2-NAME,3-PART,4-NUMBER,2-FORWARD,3-REVERSE,4-DELETE,NAME,P.M
.B.,DORM,CLASS,BIRTHDATE,READING,MATH,LANGUAGE,AVERAGE,SAFETY IN
TERLOCK,ELIGIBLE FOR GED TEST
1010 DATAFILE STARTS HERE,FILE ENDS HERE,NOTHING ON FILE,MASTER
PROGRAM READY,LOAD SUBROUTINE ENGAGED,SAVE SUBROUTINE ENGAGED,SE
ARCH SUBROUTINE ENGAGED,INPUT SUBROUTINE ENGAGED,LIST SUBROUTINE
 ENGAGED,SORT SUBROUTINE ENGAGED,SAFETY INTERLOCK
1020 DATA2-ALL,3-BY CLASS,4-BY DORM,DORM,2-ALPHA,3-NUMERIC,ALPHA
BETIC SORT,NUMERIC SORT
```

NOBODY KNOWS MORE ABOUT ADDING MEGABYTES TO YOUR TRS-80 MODEL II* THAN AMERICAN SMALL BUSINESS COMPUTERS

THE CAMEO CONNECTION

The Cameo 10 Million Character Disk Drive Subsystem (pictured in foreground) Is A 5 Megabyte Fixed, 5 Megabyte Removable Cartridge-Type Disk Drive. Advantages To This Unit Are Ease of Backup For Data and Program Files and Low Cost. By Attaching A Multiplexor Unit Several Model II's Can Be Attached To The Cameo Disk Drive. For Further Pricing, Availability, and Operating Systems Options, Call or Write to American Small Business Computers and ask for the CAMEO CONNECTION.



THE CORVUS CONNECTION

The Corvus Disk Drive Subsystem (center of picture) Is Available in 5, 10, and 20 Million Character Capacities. The Corvus System Is The Ultimate In The Sealed "Winchester, All-Fixed" Type Drive. The Corvus Is Designed To Operate In The Harshest Environments Such As Factories, Oil Rigs, Etc. Backup Of Data Is Accomplished Via Optional MIRROR Backup Device Whereby Data Is Written Onto A Standard Video Tape Recorder. A Great Number of Model II's May Be Attached To The Corvus Via An Optional CONSTELLATION Multiplexor. For Further Pricing, Availability, and Operating Systems, Call Or Write to American Small Business Computers and Ask for the CORVUS CONNECTION.

THE AMERICAN SMALL BUSINESS COMPUTERS MINI-WINCHESTER SPECIAL

The MINI-5 (being held by Corporate President Robert Webster) Is American Small Business Computer's Very Own Disk Subsystem. The MINI-5 Offers Approximately 5 Million Characters of Information Storage and is Based On the Famous Seagale Technology ST-506 Micro Winchester Drive. Now You Can Own State-Of-The-Art Technology At An Affordable Price — Only $3500 — Which Includes Drive, Controller Electronics, Power Supply, Cabinet, Cables, Documentation and Operating System Software.

DEALER INQUIRIES INVITED

AMERICAN SMALL BUSINESS COMPUTERS ✓529

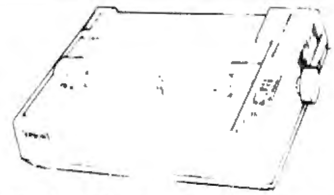
*TM TANDY CORPORATION

118 SOUTH MILL ST. PRYOR, OK. 74361 918-825-4844

BUSINESS COMPUTERS

*TM TANDY CORP.

THE MX-80 NOT ONLY DOES EVERYTHING, IT DOES EVERYTHING WELL.



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-round 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 9x9 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.

...Call For Unbelievably 'Low Price

525

OKIDATE MICROLINE 80 PRINTER - $479

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. 524

AW...WHAT THE HECK
RAM Memory Chips for the TRS-80

It is the policy of American Business Computers to offer merchandise at the lowest price possible. Serveral 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.

$22

PER 16K SET

These chips are brand new "4116's". These 200 nanosecond chips are fully compatible with all TRS-80 produces. 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.

526

★ Check It Out! ★

EPSON MX-80FT

That's right — **MX-80FT.**
The **FT** on the end means Friction and Tractor. The Fantastic **MX-80** Printer is now available in a version which will accept letterhead-type paper **And** tractor-type paper. Call or write for our (as always) unbelievable low price.

527

Epson.

This printer costs less than $450. just how much less we can't tell you - But if you will give us a call we think you will be pleasantly suprised.

This is the Epson MX-70. The lowest priced dot matrix printer you can buy. Now, that in itself should make it very attractive to a lot of people. But you ain't heard the half of it.

To begin with, the MX-70 has a lot more in common with our now-famous MX-80 than just the name. Like unequalled Epson reliability. And technological breakthroughs like the world's first disposable print head. But frankly, the MX-80 packs a lot more power than some people need. So Epson built the MX-70 to be a no-frills printer. At a no-frills price.

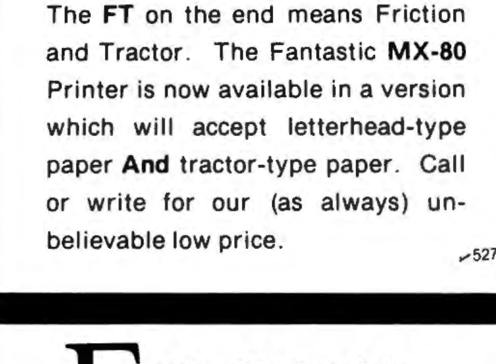
But the MX-70 is still a great little printer. They give you 80 CPS unidirectional printing. Top-of-form recognition. Programmable line feed and form lengths. Plain paper printing. An easy-to-read 5x7 matrix. Self test. And an adjustable tractor feed.

That's what you'd expect from a basic little printer. But here's something you wouldn't expect: the finest graphics package on the market today. Free.

They call it GRAFTRAX II. And it means 480 dots across the page, resolution to 60 dots per inch, and a graphic image free of the jitter and overlap that plagues other printers. You get cleaner grays and finer point resolution.

So now you've got a choice. You want more power and extra functions, you buy the MX-80. You want a basic little printer that prints, and keeps on printing, you buy the MX-70. They're both at American Business Computers.

Call for unbelievably low price.



528

What you need to know.

Choosing a Data Base

Aids III with Calcs III
Meta Technologies
Euclid, OH
$95

CCA Data Manager
Personal Software
Sunnyvale, CA
$75

Maxi Micro Manager
Exador Inc.
Adventure International
Longwood, FL
$100

Wynne Keller
RD1, Box 507
Solon, ME 04979

The concept of a data base program can be confusing to a computer novice. It is a program that holds data, but how, and why? How depends on the program, but usually disks are used because cassette output is too slow. The purpose is up to the user. The most typical data base applications are mailing lists and inventories. Other feasible applications include recording and updating data on book or stamp collections, tracking financial transactions and indexing magazine articles.

File, record and field are terms that must be defined. A file is the total collection of data, such as the whole mailing list. A record is all the information about an item, such as one person's name and address. A field is one section of all the records; the zip codes, for example. On a page of columnar paper, a field may be thought of as a column of information and a record as one line. The file would be the entire page.

Obviously there is a limit to how many records can be held in one file. This is partially determined by the type of data base used. There are two types of data base programs.

Meta Technologies' Aids III is a sequential data base. Sequential data bases contain all the records in the computer's memory at one time. How many will fit depends on the size of the machine and how many fields are used. The more fields there are, the fewer records will fit.

A random data base brings the records into a computer's memory from a disk one at a time. The limit is not machine capacity but disk size. Personal Software's CCA Data Manager is a random data base in which no one file can be larger than a single disk.

Maxi Micro Manager System (MMS) from Exador Inc. is also a random data base. However, MMS can span disk drives, allowing one file to carry over to more than one disk. Thus it has the largest capacity of the three programs.

Choosing the Right Program

Which program to choose, however, depends on more than size of files. Even if your planned application seems to require a huge capacity, a little thought could make it possible to break the data into smaller, more manageable files. An inventory, for example, could be divided by department, so each department is one file.

No data base program is perfect for all applications. A business may need to buy different programs for specific problems. The trick is to know the assets of each program, so that the purchase is not guesswork.

The three programs reviewed here are:

- MTC-Aids III with Calcs III by Meta Technologies, Euclid, OH, $95.
- CCA Data Manager by Personal Software, Sunnyvale, CA, $75.
- Maxi Micro Manager by Exador, Inc., from Adventure International, Longwood, FL, $100.

Documentation

In respect to documentation, CCA gives the best first impression. The format is an 8-inch-by-10-inch three-ring notebook. The pages are heavy quality paper, and able to withstand the constant use to which most manuals are subjected. The author is meticulous in covering details for the novice, even including sketches of the proper way to insert a disk in a drive.

Messages that appear on the screen are numbered to simplify looking them up in the manual. For reference, the organization of the manual into a "getting started" section and a "more about" section is inconvenient. To look up any function requires a reference to both sections. Yet, this minor point may be offset by the manual's clarity, which is an advantage for the novice.

Unlike CCA, the Aids manual assumes the user has some computer knowledge. The inexperienced user is referred to the DOS manual. The Aids manual does a good job of explaining how to use the program and includes many examples and flow charts. The format is 8½-inch-by-11-inch, spiral-bound.

The MMS manual is a paperbound booklet. There is no index: a serious weakness. In general, the manual is stronger on the theory of operation than specific details. Guesswork and experimentation win in the end, but there may be some very confusing moments in the meantime. The manual is particularly difficult for a beginner. The first version had some errors in the instructions for merging the disk provided with the various operating systems. A new version, just completed, has corrected those errors and greatly expanded and clarified the instructions for setting up the system. MMS

"It takes time [to format disks for program use]; go have a cup of coffee, or lunch if you have several drives."

has been on the market only six months. It appears the manual was not given enough attention initially, but the author is working to improve it.

One thing the MMS manual does cover thoroughly is file structure and the new Basic words created for the program. Thirty pages are spent on this section, which will be helpful to any programmer trying to access the files. An additional plus is the inclusion of sample data on the disk which helps the user decipher the manual.

Initializing

Organizing a data base can be very complicated, particularly for the novice. It takes experience to become adept at choosing which fields should be established, whether to make them string or numeric and how long they must be. The beginner is almost certain to make mistakes. It is best to try out possible fields on a small scale first. In most data base programs, fields cannot be changed once they are established and records are entered—except by beginning all over.

Aids is a remarkable exception to this rule. It allows old records to be retrieved even after a field has been added. Of course, some work is necessary to update old records.

Recognizing the difficulty of initialization, the CCA manual includes a sample inventory and address file initialization. It is helpful as a starting point, although more fields will probably be needed.

CCA initialization is a choice on the main menu and is straightforward. The user enters each field name, code name, length, and type; then the program dimensions the file accordingly. Calculation fields are available, using add, subtract, multiply, and divide functions.

CCA capacity on a single-drive system is quite small—unless the program and data are kept on separate disks. I had only about 100 financial transactions on a disk, but it was impossible to sort the data due to lack of disk space. Removing the program helps considerably. Single-drive owners can only half-fill a disk if sorting is desired.

With Aids III the program is initialized by editing data lines within the program. This is a simple procedure and should present no problems even for someone who has never edited a Basic program. If any errors are made, they are easy to correct: just edit the line again. For numeric fields, the number of digits to the right and left of the decimal point may be specified. After modification, the program is saved on a separate disk and each time it is run, file layout is read from the data lines. Field calculations are not available instantly, as in CCA and MMS, but may be obtained from running a separate program, Calcs III.

MMS has a unique system which allows more than one drive to hold a single file. Thus it holds far more records than either CCA or Aids. Unlike these two, however, MMS limits the length of any one field to 40 characters. Some fields might exceed this limit. If so, they must be broken into smaller categories.

Numeric fields may be rounded to a specific number of decimal places. Calculation fields may use any valid mathematic expression except exponents. A date field that allows dates to be sorted by year, month and day may be specified.

Following initialization, the MMS user specifies the number of drives and number of tracks; then the disks are formatted for program use. This takes time; go have a cup of coffee, or lunch if you have several drives. Like Aids, but unlike CCA, MMS informs the user how many records will fit based on present field definitions—very useful information.

Input

In overall display effectiveness and user interaction techniques, CCA rates the lowest; Aids and MMS are about equal and much better than CCA.

CCA screen displays are very poor and of a standard which is not acceptable in today's software. However, this is perhaps better understood when one realizes that CCA is the oldest of the three programs. It was first marketed in 1979, which was a very long time ago in the microcomputer world. The input screens actually scroll from one record to the next, and there is no graphic indication of field length as a guide to correct entry. If the correct length is exceeded, an error message is printed and entry must begin again.

CCA has two field identification names. One, the full name, is displayed on screen. The other is a user selected abbreviation of the full name and is used for all keyboard input but is not displayed on screen. The idea, of course, is to save time for the typist. The difficulty is that the operator may not remember the abbreviation. Is the field "Code Number" abbreviated as "code," "no," or "#"? The program doesn't always

"Adding records can be a very tedious process, and the CCA method only aggravates the situation."

tell and the memory may not either. One solution would be to define each abbreviated field name as the first three or four characters of the full name.

After a CCA record is added, it is written to disk immediately, before the accuracy is verified. Thus if a mistake is made, there is a delay while the record goes to disk and further time lost by calling it back so the error can be corrected. Adding records can be a very tedious process, and the CCA method only aggravates the situation.

In contrast, the input screens for MMS are nicely designed and functional. Dots indicate the size of each field and graphic blocks show the field being typed. A helpful feature is the special clear key function. When clear is pressed the contents of the display are duplicated from the previous record displayed at the bottom of the screen. Data base entries are often repetitive in one or more fields, and this feature can save a lot of time.

If there are more than 10 fields, arrow keys are used to page back and forth to view the whole record. Another important feature allows errors to be corrected before the record is saved. Disk access time seems to be quicker than CCA.

The Aids input screen, like MMS, clearly shows the length of each field, using graphic blocks. Input is very flexible. It is not necessary to finish the record before correcting an error. The up and down arrows move the cursor to any field for data correction. Since Aids is a sequential data base, records are not written to disk after each entry. Therefore, adding records is a fast process.

The Aids program is especially easy to use because of its consistency. One key returns the user to menu from any part of the program at any time. This makes Aids the only one of these programs which allows the user to quit easily. CCA allows quitting in some portions but the commands vary. MMS almost never allows it.

Aids and MMS are both a little slow during input. Typing must be done at moderate pace, with frequent glances at the screen to be sure all data is accepted. The flashing cursor causes this slow pace.

CCA, with its more primitive style, allows faster typing but is less clear. However fast the typing, the job of adding records is slow and is certainly not something you would want to do twice. Remember: *Never* fail to make that backup copy!

Select

One of the most important functions of a data base is the search. Users want to find specific records as quickly as possible. Sometimes it is important to find records falling within a certain range. Speed as well as versatility is important. MMS has the most capable search functions, followed closely by Aids, with CCA a distant third.

CCA's only type of search is instring, seeking only for equals conditions. The operator specifies a group of letters or numbers and the field to search. An exact match must be found but not necessarily the first letters. If a search for a specific street were desired, it would be found even if a street number preceeded it. The CCA search is not machine code assisted so it is the slowest of the three programs.

Aids searches according to six relational choices: equal, not equal, greater than, less than, less than or equal and greater than or equal. Up to four criteria may be selected,

✓See List of Advertisers on page 322

"Aids finds records so quickly there is no need for record numbers."

and the record must meet them all. In other words, the selection is by a logical AND function, not a logical OR. After the search relationships are chosen, the computer can count how many records meet the criteria before proceeding. The Aids III menu choices include search with almost all functions. However, if the search function is not desired, it can be bypassed with the shift/down arrow. This indicates all records are to be handled.

The Aids III search is done by machine language and is very fast. Keep in mind that Aids has the fastest search partially because records are in memory. If you count the time needed to load all those records, Aids is the slowest program.

MMS searches by five criteria: equal, not equal, less than, greater than and instring. As in Aids, a logical AND is assumed between fields. But within a field, it is possible to request a logical OR. For example, a search could be made of the name field for carrots OR beets. If either one is found AND the price field datum is less than $1, that record can be selected.

MMS also has a speed search for use on sorted fields only, which seeks an exact match beginning at the leftmost character. MMS assumes a lowercase capability and will probably be most often operated in this mode. Copy the upper/lower case letters exactly as entered or no match will be found in any search operation.

MMS is the only one of these programs with masked search capability. It can be told to ignore certain characters during its search and seek a match only at a specified section of the field. For example, a search could specify that the two rightmost characters must equal RS. Unlike instring searches, which would return the first occurrence of RS in the field, the masked search only returns a match if the RS occurs where the operator specified. The masked search will not work on numeric fields; a situation the manual fails to emphasize.

Update

MMS and Aids assume the operator won't always know the record number when he requests an update. They provide the search function as a subcommand for any update. Aids finds records so quickly there is no need for record numbers. MMS requests a number if it is known simply to save operator time. Both these programs provide nice screens for updates with field length indicators. MMS requests the number of the field to change. Aids uses arrow keys to move the cursor to the desired spot (a little slow if the field you want to change is near the end). Aids uses shift/up or down arrow to page forward or back to the next record in the search. MMS allows paging back or forward to the next record in the file, rather than the next record in the search.

CCA again suffers from sloppy screen display in this mode. The request for editing a record requires a record number. If you don't know it, you must search for it first. Search is not a choice after an edit is requested. When the selected record is displayed, CCA prompts, "Is this the record?," to which the user, incredibly, must reply either yes or no typed in full. A simply Y or N defaults to menu and the search must be started over.

Sort

A sort routine should handle more than one field at a time. This is vital for some uses. For example, it may be necessary to sort an inventory by vendor, then by item number. Similarly, address files may need to be sorted by zip code, then alphabetically within each zip. If a data base cannot sort at least two fields at once it is, in my opinion, seriously flawed. (The data base Profile, sold by Radio Shack, was omitted from this article largely because it can sort only one field at a time. It cannot perform calculations either.) Although MMS also sorts on only one field at a time, it is possible to circumvent the problem by specifying a range of records to be sorted, and a multi-field sort is to be available at a later date.

The CCA sort can easily handle multiple fields. This feature alone may make it the best data base to choose. There are some drawbacks, however. One is that the sort is the slowest of the three programs: It is entirely in Basic. Another is the disk space limitation. Sorting creates another full file of the data, as well as a working file. For this reason, a disk must be less than half full of data before the sort is attempted, on a one disk system. Multi-disk system users can route the sorted file to another drive by specifying a drive number after the sort file name. Failure to specify a drive sends the file to drive zero, and a disk-full error occurs. Then you must start the sort over.

The Aids program has the fastest sort: It is in machine language, and all the records are in memory. Like CCA, Aids can sort more than one field at a time. It cannot sort within a stated range but since printouts are available on a range basis, this feature is not vital.

The MMS program sorts only one field at a time. The only solution is to sort the most important field, for example, zip codes. When that sort is complete, note the record numbers for major zip code divisions. Ask for an alphabetic sort of the last name field, and answer the Range question with the

"THIS IS DIGITALKER*" OUT 127,0

This is all it takes to say this or 143 other expressions with MICROMOUTH*. Absolutely no software drivers or subroutines to load. MICROMOUTH*, the latest Circuit Cellar project can be used as an emergency annunciator, as an aid for the handicapped, for process control and automatic monitoring, and to add new dimensions to computer games.

Sample phrases that can be programmed are:
"THE TIME IS 4 HOURS 23 MINUTES . . .(BEEP)"
"NUMBER 4 IS 3.47 VOLTS"
"THE SPEED IS 100 METERS A SECOND . . SLOW DOWN"
Thousands of expressions can be added by changing the ROM chips.

MICROMOUTH* is plug compatible with APPLE II and TRS-80* computers. Directions are included for S100, H8 and parallel port operation. *DIGITALKER is a trademark of National Semiconductor Corp

Complete Kit (Model I) $150.00
Assembled and tested:
Apple II....................................$150.00
TRS-80 Model I w/power supply, case, cable...$175.00
TRS-80 Model III.......................$200.00

To order call:
1-800-645-3479
In N.Y. 516-374-6793

*MICROMOUTH is a trademark of Micromint Inc

The MicroMint Inc. ✓310
917 Midway
Woodmere, NY 11598
Dealer inquiries invited.

MICRO MINT

No. 2 UNBELIEVABLE OPPORTUNITY!

If You've Written an Outstanding Program-- We'd Like to Publish It!

We're looking for EDUCATION Programs:

- SPECIAL EDUCATION/AID TO THE HANDICAPPED
- MULTI-MEDIA COMPUTER INSTRUCTION
- TEACHER-AUTHORING LANGUAGES
- MANAGEMENT TRAINING
- SIMULATIONS

Earn money while helping others. Write for our free Programmer's Kit today!

INSTANT SOFTWARE, INC. ✓2
Submissions Dept.
Peterborough, NH 03458

INTERNAL MEMORY™
48K NOW AVAILABLE!

TRS-80
MODEL I
LEVEL II

Memory expansion need not be expensive or difficult. INTERNAL MEMORY™ boards expand RAM capacity to 32K(IM-1) or 48K(IM-2— INSIDE the keyboard unit with no soldering, cutting or other modifications. (Remove TRS-80™ RAM chip & plug IM into RAM sockets.) NO "piggyback" chips; NO heat, noise or power problems! Compatible with high-speed tape systems, peripherals and hardware modifications (describe what you have). Assembled & tested, one year guarantee, postpaid UPS in Cont. U.S. SEND S.A.S.E. FOR QUICK INFORMATION.

PRICES:

IM-1(32K)	$52.50 (less RAM)
IM-2(48K)	$79.50 (less RAM)
4116-3(200ns)RAM	$26.00 per set of 8 (16K)

Holmes Engineering ✓401
6246 West 3705 South
Salt Lake City, UT 84120
(801) 967-2324

THE GRAPHIC SOURCE

The lowest price
Epson MX 80 FT $599
Epson MX 80 $469
Okidata Microline 83 $949
Model III drives and controllers
16K RAM NEC 200 as Prime $19 set of 8

We will supply any computer, software, or peripheral or related product at the lowest possible price. Call 24 hours a day, 7 days a week or drop us a line. **We will meet or beat anyone's price.**

We Appreciate Your Business!

THE GRAPHIC SOURCE

April Lorenzen
Route Two, Box Forty-Four
Canton, Kansas 67428
(316) 628-4935
Quantity discounts available ✓73

5,000,000/10,000,000 BYTES

SERIES III H from Micro-Mainframe. Hard disk drive(s) in a Model III ($7,995), or add to your Model I/III ($3,795).

SERIES III F. Model III with a controller board (available separately) and operating system which allows you to start with or move up to, dual-headed or eight-inch floppys.

GL80 — $150 — Radio Shack General Ledger 1.1 with over 30 added features: A general ledger, classified balance sheet, check register and option to use an "automatic" account number or re-do an entry or document.

AR80 — $200 — Radio Shack Accounts Receivable 1.2 with automatic posting of standard monthly amounts such as installment contracts.

GL/M1 — $50 — Modifications to convert Radio Shack General Ledger 1.1 to GL80.

AR/M1 — $50 — Modifications to convert Radio Shack Accounts Receivable 1.2 to AR80.

GL/M1 and AR/M1 are supplied as program lines which must be MERGED into the original programs, or, send a disk copy of the original programs for installation at no additional charge. Documentation (apply to purchase) — $5.

FTDEMO80 — $12 — Displays and Executes the NEWDOS/80 Appendix A programs/keyboard entries. Requires 32K.

NEWDOS/80 - $135 LDOS - $135 EPSON MX-70/80/FT - $Call

AT-80 3827 Dismount Dallas, Texas 75211 (214) 339-0498 ✓445

"The [MMS] manual is extremely vague... [but] if one can figure it out, the reward is an incredibly sophisticated printing system..."

record numbers of one zip code division. The names within that zip code will emerge in sorted order. This is feasible if only a few sorts are needed. Otherwise, it takes too much time.

While sorting, MMS is much more efficient with disk space than CCA. The records themselves are not sorted; a key file is created which stores the record numbers in the proper order. The number of key files is specified during initialization. If the file will be sorted by item name and by item number, then two key files should be created. The results of each sort go into each key file. There is no problem with records suddenly becoming too large for the disk when sorting is attempted. This is a machine language sort, but reading the records from the disk is time consuming. Operators of one-drive systems must exchange the program disk and the data disk four times per key sort.

Print

It is axiomatic that the more sophisticated the printout, the more complicated it will be to create. Producing a report with any of these programs takes some diligent study of the manual. It is extremely helpful to have a shortcut for simple listings, such as item names. Aids provides this feature, MMS has a screen print, and CCA has nothing.

The quick printout for Aids goes to screen if the printer is shut off. As in all Aids commands, any range of records may be selected. User-selected fields are printed across the page.

For a full report, the Aids package includes a separate program, called Maps. After print criteria are chosen, the file to be printed must be reloaded. A document title may be specified. Field names are used for column headings. Column totals and subtotals are possible via the Calcs program (discussed under Calculations). The Aids printout is the simplest of the three programs to operate.

In CCA, there is no on-screen help for using the print function. You must refer to the manual and type in a long list of parameters. Usually you must start over when an error is made. Considering how many lines are required to define a print file, this is a real annoyance. Once correct, print file specifications can be saved on disk.

One complication to printing a CCA file is that a sorted file cannot be used until it is renamed—by exiting the program and using DOS. If a printout of a new sort is desired, both the new file and the original file must be specified. The print file cannot be saved after the sorted file is renamed. CCA's printout section seems unnecessarily difficult to use.

CCA printouts may be complicated to generate, but MMS is even worse. The manual is extremely vague at crucial points. Nonetheless, if one can figure it out, the reward is an incredibly sophisticated printing system which will merge database files with ASCII files (such as those generated by Scripsit or Electric Pencil). It is even possible to print documents with pauses for input from the keyboard. And it overcomes Scripsit's inability to handle form letters. Also, any document (monthly statements, for example) can be designed with a word processor and printed for specified data base files which meet selection criteria. The MMS data base can then be used for

"You won't need to hire a programmer to write access programs, although you may need one to understand MMS!"

many business applications. You won't need to hire a programmer to write access programs, although you may need one to understand MMS!

Calculate

All three programs have the ability to calculate new fields based on the entries in other fields. This is a very useful function. Standard markups, sales tax and discounts can be calculated by the computer automatically. MMS has the most sophisticated calculation abilities. If your needs go beyond the four basic mathematical functions, it could be your best choice.

In CCA and MMS, the calculation fields are defined during initialization. As records are added in MMS, the calculated fields are skipped until all fields are entered. Then computations are displayed. In CCA, calculations are not visible at the time the record is added. In Aids, the calculations are done by a separate program and are not as easily accessible. Also Aids has only two fields for calculated fields, but there is a "balance forward" field in addition to these. The Aids program is really geared to producing calculation for a printout, whereas MMS and CCA produce calculations at any time.

Special Features

There are a few features of these programs which deserve special mention. For instance, MMS has a built-in driver for lowercase. There is also the ability to output to RS232-C and TRS-232 printer interfaces, which eliminates the need to load special software first.

CCA allows permanent fields to be specified and the data in them can never be changed. It allows recovery of deleted records until the file is compacted.

The method Aids uses to combine files is one of its most important special feature. Flies larger than machine capacity (about 200 records in 48K) should be divided at a logical point. Perhaps half the alphabet will be file A, and the other half file Z. To add a record, determine in which file it belongs, load that file and add the record.

If the A file becomes too large, divide it. Use Select and Save to put records from A to B on disk as a new file AB. The rest can be selectively saved as new file C.

Since records can be added to the end of a new file, it is possible to sort a file which exceeds machine capacity by loading it a little at a time, sorting each segment and chaining them to a new disk file.

Summary

Choosing the right data base depends on the application. For any use involving relatively few records, Aids is my preference because of speed and ease of use.

For larger applications, the choice between CCA and MMS is more difficult. There are important differences between MMS and CCA which the buyer needs to consider.

CCA would benefit from some modernization. But if the operator is inexperienced and multiple sorts are needed, the ease of running printouts and its clear documentation would make it a good choice.

If handshaking with a word processor is needed for form letters, if calculations of more complexity than + – * / are needed or if files are longer than one-disk capacity, MMS would be the choice. MMS is clearly the more sophisticated program overall. ■

Space saving techniques.

RAM Squeeze

John D. Adams
13126 Tripoli Ave.
Sylmar, CA 91342

I learned to program in 4K of RAM. At that time just buying a computer had inflicted severe trauma on my budget and there was no way I could put out another third of the computer's price for 12K more of memory. I remember vividly the frustration of trying to squeeze my developing programming efforts into that space. Like many other trying experiences, some good came of it. It taught me to out-Scrooge Scrooge in penny-pinching RAM space.

The price of memory has come down considerably, but the jump from 16K upwards is still tough because the expansion interface is now necessary.

Actually, 16K allows considerable programming latitude. Sooner or later we are all going to write a program that leaves us tight for RAM. By being frugal I have shoehorned in some programs that I thought would never fit. The following suggestions may help you. Please understand they are given with absolutely no regard for the "beauty" or "elegance" of programming style. They are nitty-gritty last resorts for getting a size 10 foot into a size nine shoe.

Line Density

Listings 1 and 2 are routines that draw a border around your monitor. Listing 1 requires 219 bytes of storage, while Listing 2 needs only 99, a savings of over half the storage space. To get this kind of economy, here are some of the things we can do.

1. **Write multiple statement lines.** Each program line we write uses five bytes in "overhead," two for the line number, two for the pointer and one for a carriage return. Level II allows us to put more than one step on a single line by inserting a colon between them. Listing 2 reduces the number of lines by 16 saving 80 bytes in one fell swoop. Since 255 characters can be entered in any line, we can get a lot of instruction on that single line.

Granted, Listing 2 is not as pretty or as easy to read as Listing 1. When push comes to shove in a program, elegance is expendable. Pack those lines good and tight.

When using multiple statement lines be careful using the If. . .Then statements. Should such a test fail, execution proceeds to the next *line* and not the next statement. Using the Else instruction can get around this problem. An example follows:

XXX IF B = C THEN 500 ELSE ZZ = 0:STATEMENT: STATEMENT

```
10 CLS
20 Y = 0
30 FOR X = 1 TO 127
40 SET (X,Y)
50 NEXT X
60 X = 127
70 FOR Y = 1 TO 47
80 SET (X,Y)
90 NEXT Y
100 Y = 47
110 FOR X = 127 TO 0 STEP -1
120 SET (X,Y)
130 NEXT X
140 X = 0
150 FOR Y = 47 TO 0 STEP -1
160 SET (X,Y)
170 NEXT Y
180 GOTO 180
```

Program Listing 1.

ZZ is a dummy, it is not used in the program. If the test fails, it allows the remaining statements on the line to be read.

2. **Eliminate spaces.** The computer sees FORX = 1TO10, SET(X,Y) and GOTO20 exactly as it does FOR X = 1 TO 10, SET(X,Y) and GOTO 20. Long lines may have 20 or so spaces that can be saved.

3. **Eliminate unnecessary lines.** Note that line 20 in Listing 1 is gone in Listing 2. All variables are set to zero at Run, so it is not needed. Look for such lines and weed them out.

4. **Omit variable names after Next.** Level II does not usually require a variable name after Next as did Level I. This is true even in nested loops. Each variable you eliminate saves byte space.

5. **Write lines economically.** Look at lines 110 and 150 in Listing 1 and compare them with their counterparts in Listing 2. We have saved the space used by "STEP-1" in both cases. While the operation is perhaps not quite as aesthetic, the job gets done. Design lines to do maximum work with minimum coding.

Clearing String Space

The 80 stores string data from the top of memory (less reserved space) downwards. The stack is built below this area. The computer must know where to start the stack so that string data does not write over it. On power-up, 50 bytes are set aside for strings. If you need more, you must reserve it with the Clear nnn instruction. If you need less than 50 (or none at all), this space can be used. After loading KBFIX the Print Memory query returns 15515. After running the line 10 CLEAR 0 the same query returns 15557. We have reclaimed 42 bytes. When RAM gets short be miserly with string space, even if it means counting string characters by hand.

"When push comes to shove in a program, elegance is expendable. Pack those lines good and tight."

Dimensioning Arrays

Don't over-dimension arrays. We usually know how many elements are going into one and it should be set at that exact number. Once it is dimensioned, that space is lost whether we use it or not. Zero can be used as a subscript. As an example, assume that we want to store 15 elements in an array. We can run the line 10 DIM A(15) that allows us to store the elements in A(1) through A(15) and uses 84 bytes. If, instead, we run the line 10 DIM A(14), the elements can be stored in A(0) through A(14) and only 80 bytes are required.

The more dimensions an array has the more overhead we incur. Here are some representative numbers:

Running 10 DIM A(250) uses 1025 bytes
Running 10 DIM A(10,25) uses 1169 bytes
Running 10 DIM A(5,5,10) uses 1612 bytes

Even though each of these arrays will hold 250 elements, the third line uses almost 600 bytes more than the first. Keep arrays simple.

Consider number precision when designing arrays. Use a type declaration symbol if only integers are going into the array. Look at these numbers:

Running 10 DIM A%(250) uses 524 bytes
Running 10 DIM A%(10,25) uses 598 bytes
Running 10 DIM A%(5,5,10) uses 821 bytes

Again each array holds 250 elements, but compare these figures with those given above.

REM Statements

Remarks help us to understand and operate complex programs, but they use up a lot of room. We can have our cake and eat it too. Programs will often fit into memory but leave insufficient room for storing operating data. Such programs may contain remark lines. The following is an example.

```
10  REM * THIS IS A PROGRAM TO COUNT TO 100 BY
    USING A FOR-NEXT LOOP.
20  REM * EACH TIME THE LOOP CYCLES X IS IN-
    CREMENTED BY 1 AND THAT
30  REM * VALUE IS PRINTED. THIS CONTINUES UN-
    TIL THE VALUE OF X
40  REM * HAS REACHED 100 AT WHICH TIME THE
    LOOP IS EXITED AND
50  REM * THE PROGRAM TERMINATES.
100 FOR X = TO 100
110 PRINT X;
120 NEXT
```

As written, the routine takes 305 bytes. It can be CSAVEd intact. When it is CLOADed, the REM statements can be inspected by listing either to the monitor or the printer. Before running the program (while still in command mode) enter Delete 10-50. The byte expense is now 31 and we have picked up 274 bytes for data. Grouping the REM statements together makes them easier to LLIST or Delete.

Number Precision

The 80 stores integers from −32768 to +32767 in two bytes, whereas it takes four for single precision numbers and eight for double precision. Use as little precision as you need. Take advantage of the DEFINT instruction. Should the location A contain an integer and appear 25 times during the course of a program it saves space to use the line "10 DEFINT A" at the beginning of the program, rather than to use the % symbol 25 times. If (For...Next) loop counters are to run as integers, define them as such, and this will cut their storage space in half.

Use Subroutines

Write any operation that has to be repeated frequently as a subroutine, after which it can be called from any location in the program with automatic return. Should the subroutine always be called from the same location, a GOTO is more economical as it carries no byte overhead. A GOSUB requires six bytes. If, for instance, a subroutine at 500 is always called from line 100, line 100 could contain GOTO 500 and use the instruction GOTO 110 at the subroutine's termination.

"Shortcut" Instructions

Some of Level II's instructions are really routines that do a great amount of work without hogging RAM space. DEFINT has

```
10 CLS:FORX=1TO127:SET(X,Y):NEXT:X=127:FORY=1TO47:SET(X,Y):NEXT:
Y=47:FORX=0TO127:SET(X,Y):NEXT:X=0:FORY=0TO47:SET(X,Y):NEXT
20 GOTO20
```

Program Listing 2.

```
10 INPUT"ENTER ANY RATIONAL NUMBER BETWEEN 0 AND 12";X
20 GOSUB 500
30 CLS:PRINT"LETTER GRADE FOR";X;"IS ";G$:GOTO 10
500 IF X>=11.5 THEN G$="A+":RETURN
510 IF X>=10.5 THEN G$="A":RETURN
520 IF X>=9.5 THEN G$="A-":RETURN
530 IF X>=8.5 THEN G$="B+":RETURN
540 IF X>=7.5 THEN G$="B":RETURN
550 IF X>=6.5 THEN G$="B-":RETURN
560 IF X>=5.5 THE G$="C+":RETURN
570 IF X>=4.5 THEN G$="C":RETURN
580 IF X>=3.5 THEN G$="C-":RETURN
590 IF X>=2.5 THEN G$="D+":RETURN
600 IF X>=1.5 THEN G$="D":RETURN
610 IF X>=.5 THEN G$="D-":RETURN
620 G$="F":RETURN
```

Program Listing 3.

```
10 INPUT"ENTER ANY RATIONAL NUMBER BETWEEN 0 AND 12";X
20 ON INT(X+1.5) GOSUB 500,510,520,530,540,550,560,570,580,590,6
00,610,620
30 CLS:PRINT"LETTER GRADE FOR";X;"IS ";G$:GOTO 10
500 G$="F":RETURN
510 G$="D-":RETURN
520 G$="D":RETURN
530 G$="D+":RETURN
540 G$="C-":RETURN
550 G$="C":RETURN
560 G$="C+":RETURN
570 G$="B-":RETURN
580 G$="B":RETURN
590 G$="B+":RETURN
600 G$="A-":RETURN
610 G$="A":RETURN
620 G$="A+":RETURN
```

Program Listing 4.

"If all of this editing does not appeal to you, there is yet another solution."

already been mentioned. Two others are the ON X GOTO and ON X GOSUB statements. They are particularly economical in menu-driven programs. Check the two routines given below.

```
10 REM * ROUTINE #1
20 CLS:PRINT"1. ENTER AMOUNT"
30 PRINT"2. ADD AN AMOUNT"
40 PRINT"3. PRINT TOTALS"
50 INPUT"ENTER YOUR CHOICE";X:IF X=1 THEN 300
60 IF X=2 THEN 400
70 IF X=3 THEN 500

10 REM * ROUTINE #2
20 CLS:PRINT"1. ENTER AMOUNT"
30 PRINT"2. ADD AN AMOUNT"
40 PRINT"3. PRINT TOTALS"
50 INPUT"ENTER YOUR CHOICE";X:ON X GOTO
   300,400,500
```

Both routines do the same thing, but the second stores in 26 bytes less. These instructions are also useful when values are assigned on the basis of another variable's value. I use a grading program that contains a subroutine to convert number grade averages to letter grades. It is based on A+ being equal to 12, A to 11, A- to 10, B+ to 9, etc. Listing 3 shows the original routine. Listing 4 is the revision I made when I ran out of RAM. It does the same job and saves 76 bytes in the process.

Parentheses

Parentheses are invaluable for evaluating expressions in a specific order, but also take up byte space. Each level takes four bytes plus 12 more for each temporary value. Too often we tend to use parentheses to accomplish something that would be done by the order of operations. The algebraic hierarchy that the 80 follows is listed on page 1/6 of your manual. There is no point to writing 10 E= (((A↑2)*B)/C)+D when the line would be evaluated in the indicated order with no parentheses. Familiarize yourself with the order of operations and design expressions so that they take advantage of the computer's logic.

The suggestions made above can be effective if used *while* the program is being written. But let's say you have completed that pet project and spent hours typing it into your computer. You have 10 or 12 lines left and the RAM poops out. By doing a little surgery you may be able to fit the rest of the program into RAM.

Get some idea of how much space you will need to finish. Count five bytes per line and one for each character in the line. Remember to allow for instructions such as For. . . Next loops, arrays and GOSUBs that carry overhead. Now look for areas in which large savings can be made. A printed listing helps by letting you see the entire program. Delete REM statements that are not *absolutely* necessary. Check array dimensions and precison. Inserting a % after the array name could save hundreds of bytes. Look for areas where several short lines occur consecutively and combine them into one line. Apportion string space sparingly. Throw out lines that are really not necessary. Look for expressions which can be shortened. Start deleting spaces until you gain enough space to finish. If you keep at it diligently, you can pick up the space you need.

If all of this editing does not appeal to you, there is yet another solution. Several software utilities that reduce programs are now on the market. (You will probably have to delete some lines to get the program and the utility into the computer.) The program is then dumped to tape or disk. After clearing the RAM, the utility is loaded and then the program. Running the program then puts the utility into motion. These utilities can reclaim an impressive amount of space, but they do have their idiosyncracies, so follow instructions carefully. After its reduction, the program can be dumped again, the RAM cleared of the utility, the program reloaded and the additional space utilized. ■

Word processing software from Micro Architect.

Word

Word
Micro Architect
Arlington, MA
$49

D. Wilson Cooke
Box 203
Tigerville, SC 29688

Bargain hunting is a favorite pastime when I'm looking for software. I saw a need for a word processing program at the school where I teach, but did not feel a large cash outlay was justified.

Micro Architect offers a low-cost group of programs for word processing under the group heading Word.

Word2 is a 16K cassette-based program designed for the TRS-80 Level II. Word3 and Word4 are two similar programs for a disk-based Model I. Word-M2 is a program for the Model II. I ended up ordering Word4.

An Overview

Word arrived on a disk containing four files and enough TRSDOS programming to enable it to run on a one-disk system. Two of the files were Word3 and Word4. The other two were text files that either could be listed or formatted with Word, consisting of instructions on how to use Word. A 12-page manual was shipped with the disk. The manual is adequate only for someone fairly familiar with computers and the basic concept of word processing. The rank beginner might have difficulty.

After copying the four files on another disk and storing the original in our safe area, I ran both Word3 and Word4 to print the instructions in the text files INST and INSTW4. It worked!

There are slight variations in different versions of Word, but all are based on the same philosophy. Unlike machine language programs, such as Electric Pencil, it is written in Basic. This makes the program relatively easy to modify and use. It is quite easy, for example, to insert a line to select a particular type style on a printer. It also runs slower than a program written in machine language.

Word takes a text file prepared independently of the processing program and formats it into a printed document. The text file is prepared using Basic. It is written as if it were a Basic program, line numbers and all, using roughly 20 special format instructions appropriately interspersed within the text. The text can be listed or printed on paper using LLIST at any stage of its preparation. The Edit function allows text editing at any time.

Once the text is prepared and edited, it can be saved on a System disk using the ASCII format (the A option). Since Word is not resident in memory while preparing the text file, it is possible to prepare a single text as large as the available RAM. Multiple text files for long documents such as books would be limited only to the number of disks available. When the formatted document is desired, Word is called into memory and run. Word3 takes about 4K bytes of RAM and Word4 takes a few hundred bytes more.

Different Versions

Word4 is a slightly enhanced version of Word3. Both programs are for a standard disk-based machine using TRSDOS or NEWDOS. Word4 has the ability to generate lowercase characters that are sent to the printer even though they are not displayed on the video monitor. This feature alone may be well worth the slightly extra cost ($10) over Word3. Because of the software needed to generate the lowercase characters, Word4 runs slower than Word3. Also, files prepared for one version of Word will not run properly on the other version. A second difference in the versions of Word is that Word3 can accept only one text file and process it, while Word4 can accept several text files and chain them together for printing longer documents. Up to ten files can be entered at one time. A third difference is that Word4 can also number lines in the formatted document so that they correspond with those in the text file. This makes editing much easier.

Modifications

Radio Shack's lowercase modification works with both versions of Word, but has one problem that must be overcome before using Word. Typed lowercase letters will automatically be converted to uppercase by Basic unless the character is enclosed in quotes or is part of a Remark statement.

```
5 .SP 3
10 .LL 65
20 .PL 50
30 .IN 8
40 .CE 24
50 EXAMPLE OF 'WORD' USAGE
60 .NC
70 .SP
80 .FI
90 '      Line 5 skips 3 lines at the beginning.  Line 10 specifies
 that the length of each line is to be 65 characters.
100 'Line 20 specifies that the page will contain
110 'a total of 50 lines maximum.  Line 30 specifies
120 'that the text will be indented 8 spaces from the margin.
130 'Line 40 specifies that the next line will be centered.
140 'Line 50 is a text line.
150 'Line 60 defeats the centering function.  Line 70 causes
160 'a blank line to be output.  Line 80 specifies that the
170 'text is be filled and justified on both margins.
180 'Lines 90 through 190 are text lines.  Line 200 is the
190 'end statement.
200 .EN
```

Program Listing 1.

"It is possible to get Word hung in an infinite loop if the commands are not used properly. A little experience will solve problems like this."

Rather than modify the machine language lowercase driver program to defeat this problem, I chose to modify Word.

As text files are prepared, the text lines begin with the apostrophe symbol (Radio Shack's Basic abbreviation for REM). Using the Auto function to generate the line numbers, it is rather easy and nearly automatic to start out each line with an apostrophe. Word was then modified to ignore the extra apostrophe at the beginning of a line but nowhere else. (See Program Listing 1.) Only one line must be added to Word3 to cause it to reject the extra apostrophe: 9033 IF MID$(A$,I+1,1)="'" THEN I=I+1.

The modifications to Word4 are more complex, but quite easy. Lines 503 and 9035 through 9050 must be deleted. These are replaced by the lines:

```
9032 LN=VAL(LEFT$(A$,I))
9033 IF MID$(A$,I+1,1)="'"THEN I=I+1
9035 A$=RIGHT$(A$,LEN(A$)-I):RETURN
```

A Multiple Copy Routine

When multiple copies of a document are desired, it is simple to run Word more than once. The speed can be improved if the formatting is done only once and dumped to the disk in formatted form. The formatted disk file is then used to print the second and later copies. I modified both versions of Word to print an image on the disk that is exactly like the one printed on paper. The new file generated is automatically named the same as the old file with the letter F prefixed to the old file name. Thus, the old file name should be chosen so that it is one letter less than the maximum length the system allows. The necessary Basic code to modify both versions of Word is shown in Program Listing 2. A short program to read the disk file and print it on the line printer is also given in this Listing.

The only problem likely to be encountered with this modification is lack of disk space for long text files. It is necessary for both the formatted and the unformatted version of the text to reside on the disk at the same time. This could present problems with long texts on one-disk systems. I got around this problem, to some extent, by having a master disk with all the necessary software from TRSDOS along with the various versions of Word that I have created. Another disk is made using Backup and everything is deleted from it that is not absolutely essential. The master disk is used to load Basic, Word, or anything else needed. This leaves maximum disk space on the second disk for text files. It is a simple matter to switch disks in the drive.

After using Word for a few months to write letters, memos, and several longer documents, I feel that it does what the supplier says it will do. The only serious problem I have encountered is in not using the format instructions correctly in preparing the text file. It is possible to get Word hung in an infinite loop if the commands are not used properly. A little experience will solve problems like this. For the price, Word does a good job.■

```
50 REM ****************************************************
60 REM ** MODIFICATION TO WORD3 TO PRINT FORMATTED IMAGE **
61 REM ** OF DOCUMENT ON DISK.                          **
62 REM ** WRITTEN BY D. W. COOKE, OCTOBER, 1980          **
70 REM ****************************************************
506 B$="F"+A$:OPEN "O",2,B$
9980 LPRINT TAB(TI)O$:PRINT #2,TI;O$:IF MV=1 THEN RETURN
9990 FOR I=2 TO MV: PRINT #2,SP$:LPRINT SP$: NEXT I: RETURN


50 REM ****************************************************
60 REM ** MODIFICATION TO WORD4 TO PRINT FORMATTED IMAGE **
61 REM ** OF DOCUMENT ON DISK.                          **
62 REM ** WRITTEN BY D. W. COOKE, OCTOBER, 1980          **
70 REM ****************************************************
509 GOSUB 530: OPEN"I",1,FI$(XU): LN=0: B$="F"+FI$(XU):
    OPEN"O",2,B$
9980 IF FL=0 THEN LPRINT TAB(TI)O$:PRINT #2,TI;O$ ELSE LPRINT
    LN,TAB(TI)O$:PRINT #2,TI;O$
9990 FOR I=2 TO MV: LPRINT SP$: PRINT #2,SP$: NEXT I: RETURN


10 REM ***************************************************
20 REM ** PROGRAM TO PRINT FILE FROM DISK WHEN FILE IS **
30 REM ** CREATED BY WORD3 OR WORD4.                    **
40 REM ** WRITTEN BY D.W. COOKE, OCTOBER, 1980          **
50 REM ***************************************************
60 CLEAR 2000:CLS:PRINT:PRINT
70 INPUT "NAME OF FILE TO BE PRINTED FROM DISK"; A$
80 OPEN "I",1,A$
90 IF EOF(1) THEN 160  ELSE LINE INPUT #1,B$
100 A = LEN(B$)
110 D$ = LEFT$(B$,3): TI = VAL(D$)
120 IF A<4 THEN LPRINT " ": GOTO 90
130 C$ = RIGHT$(B$,A-3)
140 LPRINT TAB(TI) C$
150 GOTO 90
160 CLOSE:END
```

Program Listing 2.

SSM SOFT SECTOR MARKETING, INCORPORATED

Our One Year Anniversary!

New Toll-Free Order Line for Buyers Outside Michigan.

800-521-6504

For The Game Players

Larry Ashmun Is Back . . . and Brings You . . .

SUPER VADERS

With Sound - Mod I & Mod III
©1981 Soft Sector Marketing, Inc.

The third in the evolution of the most popular space invaders games for Tandy machines. Now for 1 or 2 players, 10 levels of play.

Cassette (goes to disk) **$19.95**

Ask for upgrade information for TRS-Super Invaders or Invaders-Plus.

CONFLICT

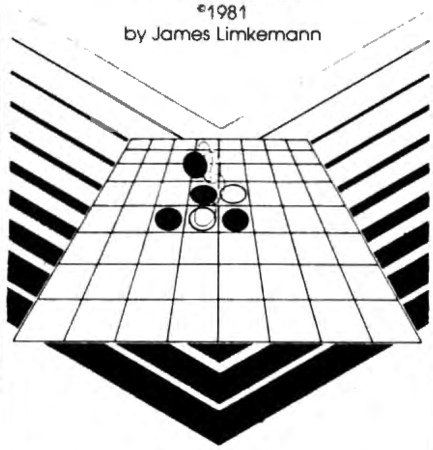
A strategy game involving land conquest.
2-8 players

Cassette **$14.95**

MOD I & MOD III, Color.
Please state machine when ordering.

Flippy

©1981
by James Limkemann



A Machine Language, Othello Game.
MOD I & MOD III
– 1 or 2 Players–

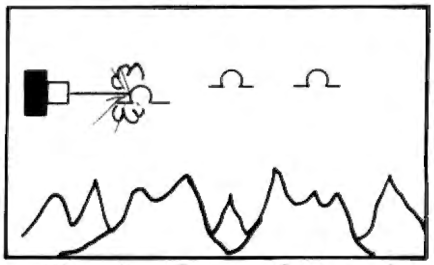
Cassette **$14.95**

Complete with Good Game Instructions.

Coming Soon!

"It's more than just another machine language arcade game."

ALIEN DEFENCE



It's An Experience!

Super Fast Action!
– 1 or 2 Players –

Lander	50 Points
Mutant	100 Points
Cruiser	150 Points
Bomber	200 Points
Mother Ship	1000 Points
Swarmers	150 Points

Call for Availability
MOD III Only

MASTER CONTROL

For the Color Machine
©1981 Soft Sector Marketing, Inc.
Requires 16K

Ask about our other Color Machine programs!

FEATURES:

1. 50 preprogrammed command keys, standard and extended commands.
2. Direct control of motor, trace, and audio.
3. Relocatable Machine code.
4. Automatic line numbering, starting point and increment alterable.
5. Programmable Custom key.
6. Direct Run button.
7. Keyboard overlay for easy program use.
8. Easy entry of entire commands into the computer and display.
9. Full instruction manual.

Cassette **$24.95**

Outside Michigan Order Line Only
800-521-6504

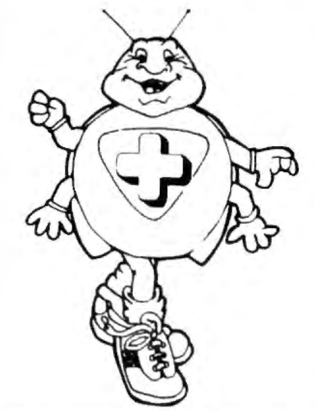
SSM SOFT SECTOR MARKETING, INCORPORATED ✓515
6250 Middlebelt
Garden City, Michigan 48135

Questions & Michigan Orders.
313-425-4020

Utilities

BUG +

©1981 by James Limkemann



A Self-relocating Machine Language Monitor for MOD I & MOD III

BUG+

©1981 by J. Limkemann. Bug+ is a powerful machine language monitor. The one point most improved over other monitors, is the tape write. Bug+ has the ability to write a "clean" tape (at 500 baud), this tape will read into the TRS-80 under the system command, without the problems previously associated with the volume setting. Regardless what version basic you have or whether or not you have a Radio Shack cassette fix, this monitor will improve the reliability of your cassette by 100%. There is also a verify command that works the same as a "CLOAD?", except when an error is found; the memory address and what is found on the tape is displayed.

Finally a breāk point that works! When a break point is reached, there is a blinking astrisk in the bottom right hand corner, you are able to see what is on the screen before the monitor takes control. Press the enter key the screen clears and the monitor comes to life. When you continue from a break point, the monitor will restore the screen first then load the CPU registers and return to your program. You do not lose your program or display, and it does work!

Bug+ also has all the commands of T-Bug, they just work better. Bug+ loads into low memory, then relocates itself.

Mod III has all the commands of the Mod I version plus it gives you the ability of reading or writing 1500 baud or 500 baud or 500 baud tapes. You can read at one rate and write at another. MOD I or III, 4K, both on same tape.

Cassette **$14.95**

BOSS III

The **B**ASIC **O**PERATED **S**INGLE **S**TEPPER

©1981 Soft Sector Marketing, Inc

for Mod III

This Machine Language utility is designed to aid you in creating and debugging programs written in BASIC. The utility allows you to trace the program flow, to single step the BASIC program, to observe the conditions of variables during program execution, and to push your BASIC program on the stack during program development. The utility is known to operate with Mod III, TRS-DOS or Mod III Rom BASIC.

Cassette (goes to disk) **$18.95**

– New Lower Price –

Also **Boss 2.1 version 2.2.** ©1980 V.B Hester for Mod I Cassette (goes to disk) **$18.95**

TAPE COPY 2

©1981 Soft Sector Marketing, Inc.

This program will load most any TRS-80 500 Baud system tape (standard) Mod I speed) and load it into memory and save it at either 500 or 1500 Baud on the Mod III. NO KNOWLEDGE OF MACHINE LANGUAGE NEEDED. Now it gives you a way to back up a machine language program that loads at the lower speed and makes cassette loading into your new Mod III a much faster and more reliable process. Works with Mod I* & Mod III.

Only **$14.95**

*Mod I reads and saves at 500 Baud only

SUPER UTILITY

©1980 by Kim Watt of Breeze Computing, Inc. Sold on protected media only.

Main Program list:

Zap; display sector (disk, file), display memory, compare disk sectors, copy disk sectors, verify disk sectors, zero disk sectors, string search, sector search, single or double density diskettes (if your machine works normally with double density).

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, remove passwords.

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 many protected disks) - purchaser use - only for his own personal disks.

Tape Copy; this program is to make backup of many TRS-80 tapes, no matter how it is recorded (note agin this program is for the use of the original purchaser for his own programs only).

Disk Repair; repair gat table, repair hit 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.

Disk Repair; repair gat table, repair hit table, repair boot, read protect directory track, recover killed files, check directory.

MOD I, protected disk only **$49.95**

Back up copy available!

MOD III ROM COMMENTED

©1981 Soft Sector Marketing, Inc.

Only $22.50
+$2.50 Shipping & Handling

Not just a rehash of old information, but detailed comments on the ROMS in the latest machine from Tandy.

Now in its 3rd printing!
Over 150 Pages!

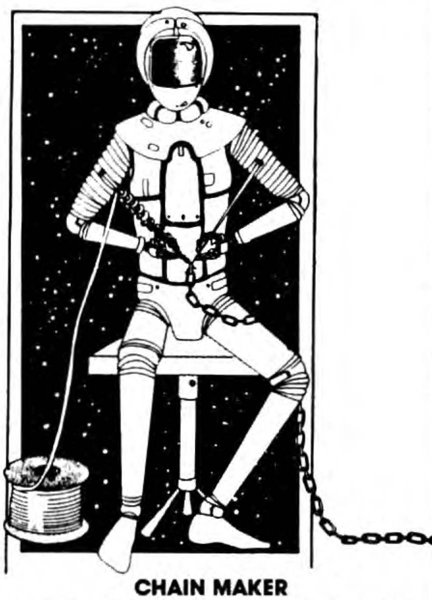
SPECIAL

If you purchase both Postman and Lazy Writer at the same time, at list price, (or if you already own one of the packages and now want ot purchase the other), we will at no extra cost include the Postwriter package FREE. To purchase this package say "I want the best package on the market so send me the full system at one time". This includes the Postman, Convert 1 and Convert 2. The package works on all quality operating systems based on the Tandy disk format (TRSDOS, LDOS, NEWDOS, NEWDOS-80, etc.)

MOD I Only

$250.00

See page 3 of this ad for details on these programs.



CHAIN MAKER

©1981 Steve Skindell. This program makes writing chain files on NEWDOS-80 a breeze. This is not only a program that creates chain files but it also is an editor of the file. A file can be removed from the disk and edited and saved back to the disk quick and easy. An outstanding program and a must for every NEWDOS-80 owner. Comes with complete, easy to read manual. Purchaser must own NEWDOS-80 .... **$14.95**

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. All hardware must be prepaid. No hardware shipped collect. Out of the country orders add $10.00 extra shipping.

VISA

*TRS-80 is a product of Radio Shack, division of the Tandy Corporation ✓515

The Serious Side of Computing . . .

Business Programs

CRITPATH*

©PM Programming 1980

CRITPATH* will help

- Determine the earliest possible completion date for your project.
- Identify the critical activities in your project; the jobs that will delay completion of the entire project if they are not completed on time.
- Provide you with the scheduled start and finish date for all project activities so you can plan your labor and material needs to support the project completion date.
- Develop the information needed to permit you to shift resources and funds to critical activities from those activities identified by the program as being capable of delay without delaying the completion of the project.

CRITPATH* provides you with

- Both screen and printer outputs of your schedule in work days, tabulating the earliest start/finish and latest start/finish days as well as the total float for each project activity.
- The length of the critical path in work days.
- A listing of all activities that are on the critical path; jobs that cannot be delayed without delaying the entire project.
- A printer output of the actual calendar dates for the erliest start/finish and latest start/finish along with both free and total float for each activity.
- Your choice of printing out data sorted consecutively by:
  a. Job sequence
  b. Earliest start dates
  c. Earliest finish dates
  d. Latest start dates
  e. Latest finish dates.

Specially designed for builders, by a builder. Also works well with many other businesses.

MOD I **$600.00**
MOD III **$750.00**
Manual **$50.00**
Overview .... **Free**

Requires 48K, 2 disk drives and 132 column printer.

COMPLEAT IDIOTS BOOKKEEPER

©1981 S.F.I. This will probably not be the only bookkeeping program you'll ever need (but you can never tell) but it should be your first.

Why?

- Easy to use
- Easy to understand instructions
- Has Purge feature
- Advanced Edit features
- Batch input
- Build new fiels from parts of your existing IDIOT files
-Flexable Report Generator

Mod I or III, 48K, 1 drive **$49.95**

NEW

MAKE VC

©1981 S.F.I. Changes IDIOT files to VISI CALC interchange format or take VC files and makes it into IDIOT readable file.

Now Only **$24.95**

Make VC requires Compleat Idiots Bookkeeper.

POSTMAN

By Alger Software. ©1980 S.S.M. Inc. A machine language mailing list program that will do the following:

- 645 labels on a 35 track disk drive
- 745 labels on a 40 track disk drive
- 1534 labels on an 80 track disk drive
- 10 fields (2 user defined)
- Fast sorts (500 racords in 30 seconds)
- Written for (1) drive system (capacities shown are for a single drive system)
- Most any lable stock that is one up can be used (you save no money buying labels that are more than one up anyway)
- Print one label at a time or a sequence of labels
- Purge duplicates with or without user assistance
- 9 digit zip code
- Super fast search on any field - 3 second average
- Easy screen editing

Now Postman has been upgraded with many new features. Now this very popular mailing package is not just the best way for most people and small businesses to do their mailing lists, but now we give you a way to uncramp, convert and formletter your mailing list. You need this package if one of the following is true: If your mailing package is memory dependent. This means that you must have more memory to handle more names in your machine; If you need a way to get rid of duplicate names in your mailing list; If you want to sort on more than just name or zip (our package can sort on any or all 10 fields at once.); If you are waiting more than one minute for your sort to finish. (our package is in machine language so it runs very fast); If your present program will not handle the 9 digit zip code; If your program doesn't have full screen editing. This package is a machine language program; this is the reason for the super fast speed of all functions!

This is random access disk based program and any name can be called to read, write, print or update in 3 seconds or less. Now along with it you get utilities that permit you to do the following: CONVERT 1; takes all the files from most other mailing list and converts them to our system. (why should you change to our system if we made the change hard?) CONVERT 2; convert from our package back to ASCII files if you want to do something with them (like send them to another computer over the phone). This program runs on all quality operating systems. Requires I disk drive and 32K memory. .... Only **$125.00**

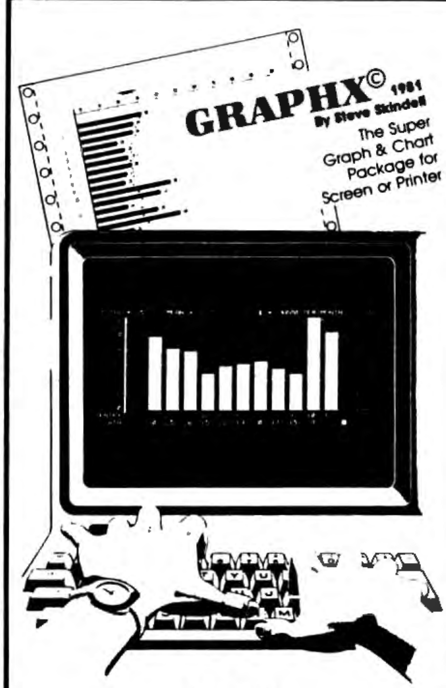
GUARANTEE

All programs described as **Business Programs** come with a 30 day, money back guarantee, less 10% restocking charge. We feel the best way to see if a program works in your business is to use it.

POSTWRITER

©1981 S.S.M., Inc. Now there is, at extra cost, a formletter package that permits inserting any of the 10 fields of information from "Postman" into any part of a letter, (yes even in the body of the letter), and right and left justify the letter. This program is made to be used only with the Postman progam and one of the following word processing packages: LazyWriter or Electric Pencil. **$49.95**

Requires Postman



GRAPHX

By Steve Skindell ©1981. This is a program that is for the person who does reports or requires some sort of plotted output to show gains or losses, or any type of output that needs graphs. This program puts to the screen, or to a printer, the plotted points in bar program for accountants, CPAs and the average businessman to evaluate, at a moments glance where he is, was or where he is going. Files saved to disk can be recalled at any time to be reexamined, modified, or just reprinted. An extra feature: if you have the Microline printer, by Okidata or Epson MX-80, your output is in true graphics. Information is supplied for the user so he can modify this program for other printers. Comes complete and ready to run. Requires MOD I or MOD III, 48K disk. Printer optional (132 col.) .... Only **$49.95**

5,000 MAILING LABELS
$14.95
Plus $2.00 Shipping

Outside Michigan
Order Line Only
800-521-6504

SSM SOFT SECTOR MARKETING, INCORPORATED ✓515
6250 Middlebelt
Garden City, Michigan 48135

Questions & Michigan Orders.
313-425-4020

In the June, 1981 Micro-computing, Page 20 . . .

Lazy Writer® was compared to a lesser costing word processing program and some interesting information was brought out. We would like to clarify the facts:

Reviewers Comments	Clarification
"Lazy Writer is the easiest to use"	– That is what is ment by - The Innovative Word Processing 'System'.
"Features a unique cancel edit feature"	
"Allows loading any ASCII files"	– In true fact, Lazy Writer will load any program or sequencial file.
"Built-in 'Help' command that can be called to explain most used features"	– This file can be added to, by the user, to include your most used features.

To Error Is Human.

In the same article it was said that Lazy Writer COULD NOT DO the following: WRONG!

Features reviewer thought not included in in Lazy Writer	Does Lazy Writer Do This	Clarification
Underlining, Page Numbering, Headers/Footers	YES	Lazy Writer has always had this ability. June 1, 1981 these features were improved to support almost every printer on the market - standard!
Append Modules	YES	Called Extentions
Keep track of free space	YES	Hit clear/break in program
Pull directory from program	YES	
Repeating keyboard	YES	
Support for MX80 printer	YES	All of the tricks the MX-80 can do.
Superscript/Subscript	YES	Added June 1, 1981. You can embed any printer control codes any place in the text.

As a matter of fact in June, 1981 Lazy Writer was improved to such a degree that a new manual (over 100 pages was issued) along with the new program. (by the way, there was NO charge for this update to registered owners).

Important Things Not Mentioned In The Article:

A) **Chain Printing.** Gives you the ability to print files longer than can be loaded into memory. (e.i. our 157 page MOD III ROM Commented Book).

B) **User Defined Special Character.** For use with printers that have printable characters that the TRS-80 keyboard does not directly support. (e.i. Daisy Wheel)

C) **Customize Your Printer Default Commands** with PRINTGEN.

E) **Types of Printer Outputs Supported,** RS232, TRS 232, and parallel printers.

F) Nowhere did they mention the **Communication portion** of Lazy Writer.

& MUCH, MUCH MORE!
Send for Overview.

Comments in the July 1981 Creative Computing

Pg. 34-36 - Summary: **"Very easy to use"**

"All the functions of Lazy Writer are covered well in its documentation, which ranks among the best I have seen. it is on a par with the construction manuals for Heathkits and Dynakits"

"This being written on a TRS-80 Model I computer with ease and flexibility that previously had been available only on larger computers with much more expensive word processing programs. What has moved the Model I up into at least the Triple-A League, if not the true Big Leagues, of writing is a modestly priced word processer called Lazy Writer."

"The best thing about Lazy Writer, from the viewpoint of a professional writer and editor, is not that it does exotic things, undreamt of by users of Electric Pencil and Scripsit. It is that it does the same thing in a much simpler way."

Lazy Writer®

an innovative word processing system

434

Even though many advantages were overlooked, one over shadowing fact is . . .

There's nothing even close.®

Mod I **$125.00**
Mod III **$175.00**

Purchase direct, or from your local retailer.

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. All hardware must be prepaid. No hardware shipped collect. Out of the country orders add $10.00 extra shipping.

*TRS-80 is a product of Radio Shack, division of the Tandy Corporation.

VISA

An exercise in deduction.

TRS-80 Disassembler

Carl Wuebker
505 Duke Lane
Fort Collins, CO 80525

Like many people, I enjoy a good puzzle. Working puzzles helps you to exercise your deductive powers and logic abilities. When you bought your computer you probably weren't aware you bought a challenging puzzle at the same time. Trying to solve this puzzle will help you understand the mysteries of your computer and write clearer programs.

The puzzle I am referring to is your operating system. For instance, how does your TRS-80 write a byte on tape? Why does it skip a line after printing a 64-character line? What does your TRS-80 program look like in memory? On tape? How can you fix a tape that won't CLOAD?

The TRS-80 Disassembler program, written in TRS-80 Level II Basic, allows you to read the bits in the ROM (via the PEEK command) and convert them into Z-80 standard mnemonic instructions. This program performs most of the routine tasks of solving the puzzle, and leaves the thinking part up to you.

Lines	Module	Variable	Description
1-99	Main Program		Sets the starting location, page number, calls proper output routine, stops on each page if requested.
		HX$	String to allow easy hex conversion
		L	Current disassembler location
		P	Current page number
		P$	L if line printer, S if screen
		S$	S means stop after current page
100-199	Lineprinter		Line prints a disassembled page
		N	Line counter
		L$	Complete disassembled line
200-299	Screenprinter		Screen prints a disassembled page
1000-1799	Dis-A-Line		Disassembles one instruction line at L and bumps L
		I0	First byte of instruction
		I1	Second byte of instruction
		I2	Third byte of instruction
		I3	Fourth byte of instruction
		RN$	# is replaced by RN$ for instruction
		RS$	* is replace by RS$ for instruction
		M$	Instruction text w/out hex data
		DL	Number of bytes in instruction
		FM	Found a # in instruction text
		FS	Found a * in instruction text
1800-1999	Replace-#-OR-*		Replaces a # or * in instruction text with RN$ or RS$
		I	Character counter
2000-2199	Access-I-Text		Reads instruction text from data array and formats instruction
		OPC$	Opcode = >alpha array
		FLD$	Register = >alpha array
		I0$	String used to hold opcode only
		I1$	String used to hold opcode argument
		T	Format type (0 is first instruction character is not numeric)
		I	General character pointer
2200-2299	Load-I-Text		Moves data at 2300-2999 into FLD$ and OPC$ arrays
3X00-3X99	Formatter		Formats specific instruction type X
		C	Pass this number of a GOSUB 5000
		C$	Get this hex string back...
4000-4199	Prelude Formatter		This routine formats the address, ASCII, and hex interpretations of the instruction
		LT	Temporary for the location counter
5000-5019	One-Byte-Hex		Converts C into a one byte hex C$
5020-5069	Two-Byte-Hex		Converts C into a two byte hex C$

Fig. 1.

Unraveling

To use the program, type it into your machine, save it on your cassette or disk and set L (in line 40) to the first location you want disassembled. Run the program, answer the "PRINTER (L,S), STOP (S,N)" question (S,N means Screen, Non-stop, for example) and the program will start disassembling instructions at a rate of about three to five lines per second. Take the disassembled listing and start unraveling!

The Operating System

The operating system (also known as the Basic interpreter) consists of about 6000 simple steps which act in concert to follow your high level instructions (given in Basic). When you type: PRINT "HELLO" (enter), the system obligingly displays HELLO on the screen. During the time between pressing enter and the appearance of the message, the operating system has used thousands of simple steps (known as machine instructions). These instructions tell the Z-80 CPU what you typed and then (once it had "understood" the Print statement) to move the data (HELLO) into another area of memory (which is written to the display). The operating system is the interface between you, the user, who is speaking Basic and the Z-80 which is speaking machine instruction.

The operating system is written in assembly language (which is a form of machine instruction). It's much easier for a programmer to remember how to write the statement:

```
HERE: JP HERE ; LOOP FOREVER (in BASIC, 10 GOTO 10)
```

than it is for him or her to remember: 0010: C3 10 00, which is Z-80 "GOTO $0010" in hexadecimal. (I use a $ preceding a number to indicate a hexadecimal number; I feel it's clearer and easier for an assembler to syntax). The TRS-80 operating system involves about 150 pages filled with assembly language instructions. These instructions were assembled and the bits (analogous to line 0010: C3 10 00) where made into a set of ROMs. These ROMs are shipped with every TRS-80 so when it is powered up, the Z-80 will act on the bits to interpret your Basic language input.

While writing the operating system, the programmer named the routines, subroutines and data. For example, when a keyboard character was needed, he or she probably wrote CALL GETC instead of CALL $0013. Not only was this easier than looking up the current ROM location for the subroutine, it made life easier for other programmers who attempted to fix the bugs that creep into every large program. When the operating system was finally bug free, it was assembled by an assembler (that is, the names were converted into addresses which the Z-80 understands). In this process, all the names were left out of the TRS-80 ROM; only the addresses which were referenced remained. The puzzle is, therefore, to find out what the programmer asked the Z-80 to do without relying on names and other comments.

A Look Inside

The TRS-80 Disassembler is written in a modular fashion; a description of the modules, their functions, and their variables appears in Fig. 1.

A Sample Disassembly—The TWRITE Subroutine

While reading through my disassembled TRS-80 listing, I noticed a routine which called a subroutine and then delayed short-

```
0261: CD 64 02    TWRIT2: CALL   $0264       ; CALL TWRITE, then fall into it

0264: E5          TWRITE: PUSH   HL          ; Write byte in A on tape
0265: C5                  PUSH   BC          ; Save all registers on stack
0266: D5                  PUSH   DE
0267: F5                  PUSH   AF

0268: 0E 08               LD     C,$08       ; Bit counter = eight bits to write
026A: 57                  LD     D,A         ; Save byte to be written in D

026B: CD D9 01     TLOOP: CALL   $01D9       ; Write a pulse on the tape

026E: 7A                  LD     A,D         ; Rotate D left into carry
026F: 07                  RLCA               ; (Get current bit in carry)
0270: 57                  LD     D,A

0271: 30 0B               JR     NC,$027E    ; If bit to be written is 0,
                                             ; wait 2ms silently
0273: CD D9 01            CALL   $01D9       ; If bit to be written is 1,
                                             ; write a second pulse 1 ms out
0276: 0D                  DEC    C           ; Decrement bit counter
0277: 20 F2               JR     NZ,$026B    ; If bit counter>0 then loop
0279: F1                  POP    AF          ; Restore all registers
027A: D1                  POP    DE
027B: C1                  POP    BC
027C: E1                  POP    HL
027D: C9                  RET                Return

027E: 06 87        TZERO: LD     B,$87       ; Wait 1ms silently
0280: 10 FE               DJNZ   B,$0280     ; to signify a zero bit...
0282: 18 F2               JR     $0276       ; Continue writing

01D9: 21 01 FC     PULSE: LD     HL,$FC01    ; Write pulse on tape
01DC: CD 21 02            CALL   $0221       ; Set port $FF to pppppp01
01DF: 06 0B               LD     B,$0B       ; Wait...
01E1: 10 FE               DJNZ   B,$01E1
01E3: 21 02 FC            LD     HL,$FC02    ; Set port $FF to pppppp10
01E6: CD 21 02            CALL   $0221
01E9: 06 0B               LD     B,$0B       ; Wait...
01EB: 10 FE               DJNZ   B,$01EB
01ED: 21 00 FC            LD     HL,$FC00    ; Set port $FF to pppppp00
01F0: CD 21 02            CALL   $0221
01F3: 06 5C               LD     B,$5C       ; Wait to finish off 1ms
01F5: 10 FE               DJNZ   B,          01F5
01F7: C9                  RET                ; Pulse written on tape

021E: 21 00 FF     WRPFF: LD     HL,$FF00    ; Write ($403D) to port $FF
0221: 3A 3D 40     CHPFF: LD     A,($403D)   ; Change port $FF
                                             ; Send port $FF
0224: A4                  AND    H           ; (Old port $FF AND H) OR L
022F: B5                  OR     L
0226: D3 FF               OUT    ($FF),A
0228: 32 3D 40            LD     ($403D),A   ; Save current port $FF
022B: C9                  RET                ; Port $FF written...
```

Fig. 2.

✓See List of Advertisers on page 322

ly. Usually a computer won't use a delay unless the operation it's doing is time-dependent. After thinking through some of the operations which could be time-dependent, I decided it probably had something to do with the cassette tape unit. On digging further, I discovered the subroutine was used by another subroutine to write a byte on the tape. When entered at TWRITE ($0264), the subroutine writes the byte in A to the tape (provided that the tape is running and connected). When entered at TWRIT2 ($0261), it writes the byte in A to the tape twice. Bytes are written to tape starting with bit 7 and continuing until bit 0 is written. Bits are written to tape as follows:

0 bit: Pulse, 2ms silence
1 bit: Pulse, 1ms silence, pulse, 1 ms silence

("A Look Inside The TRS-80," *Kilobaud*, April, 1979, is an excellent article for those interested in the tape format.)

Below are the disassembled routines—TWRIT2 and TWRITE—with my own labels and comments added.■

Program Listing for Disassembling the TRS-80

```
10 CLEAR 3000
20 GOSUB 2200
30 HX$="0123456789ABCDEF"
40 L=0
50 P=0
60 INPUT "PRINTER (L,S) , STOP (S,N)";P$,S$
70 IF P$="L" THEN GOSUB 100 ELSE GOSUB 200
80 IF S$="S" THEN 60 ELSE 70
90 END
100 P=P+1
110 FOR N=1 TO 45
120  GOSUB 1000
130  LPRINT L$
140 NEXT N
150 LPRINT " "
160 LPRINT TAB(24);"-";P;"-"
170 RETURN
200 FOR N=1 TO 15
```

Program continues

Program continued

```
210  GOSUB 1000
220  PRINT L$
230 NEXT N
240 RETURN
1000 I0=PEEK(L)
1010 IF I0=203 THEN 1200
1020 IF I0=237 THEN 1300
1030 IF I0=221 THEN 1500
1040 IF I0=253 THEN 1520
1050 I1=PEEK(L+1):I2=PEEK(L+2)
1060 GOSUB 2000
1070 RN$="HL":RS$="(HL)"
1080 GOSUB 1800
1090 GOSUB 4000
1100 L$=L$+M$
1110 RETURN
1200 I0=PEEK(L+1)+256
1210 GOSUB 2000
1220 IF M$="???" THEN 1090
1230 DL=2
1240 RN$="HL":RS$="(HL)"
1250 GOSUB 1800
1260 GOSUB 4000
1270 L$=L$+M$
1280 RETURN
1300 I0=PEEK(L+1)
1310 IF I0<64 OR (I0>127 AND I0<160) OR I0>191 THEN I0=
     191
1320 IF I0<128 THEN I0=I0+448 ELSE I0=I0+416
1330 I1=PEEK(L+2):I2=PEEK(L+3)
1340 GOSUB 2000
1350 IF M$="???" THEN 1390
1360 DL=DL+1
1370 RN$="HL":RS$="(HL)"
1380 GOSUB 1800
1390 GOSUB 4000
1400 L$=L$+M$
1410 RETURN
1500 RN$="IX"
1510 GOTO 1530
1520 RN$="IY"
1530 C=PEEK(L+2)
1540 GOSUB 5000
1550 RS$="("+RN$+"+$"+C$+")"
1560 IF PEEK(L+1)=203 THEN 1660
1570 I0=PEEK(L+1):I1=PEEK(L+2):I2=PEEK(L+3)
1580 IF I0=54 THEN I1=PEEK(L+3):I2=0
1590 GOSUB 2000
1600 FM=0:FS=0
1610 IF M$<>"???" THEN GOSUB 1800
1620 DL=DL+(FM OR FS)+FS
1630 GOSUB 4000
1640 L$=L$+M$
1650 RETURN
1660 I0=PEEK(L+3)+256
1670 GOSUB 2000
1680 FM=0:FS=0
1690 IF M$<>"???" THEN GOSUB 1800
1700 DL=DL+3*FS
1710 GOSUB 4000
1720 L$=L$+M$
1730 RETURN
1800 FM=0:FS=0:I=5
1810 I=I+1:IF I>LEN(M$) THEN RETURN
1820 R$=MID$(M$,I,1):IF R$<>"#" AND R$<>"*" THEN 1810
1830 IF R$="*" THEN 1880
1840 FM=1
1850 M$=LEFT$(M$,I-1)+RN$+RIGHT$(M$,LEN(M$)-I)
1860 I=I+LEN(RN$)
1870 GOTO 1810
1880 FS=1
1890 M$=LEFT$(M$,I-1)+RS$+RIGHT$(M$,LEN(M$)-I)
1900 RETURN
2000 IN$=OPC$(I0)
2005 T=ASC(IN$)-ASC("0")
2010 IF T<1 OR T>9 THEN T=0 ELSE IN$=RIGHT$(IN$,LEN(IN$
     )-1)
2015 FOR I=1 TO LEN(IN$)
2020  IF MID$(IN$,I,1)=" " THEN 2045
2025 NEXT I
2030 I0$=IN$+STRING$(5-LEN(IN$)," ")
2035 I1$=""
2040 GOTO 2055
2045 I0$=LEFT$(IN$,I)+STRING$(5-I," ")
2050 I1$=RIGHT$(IN$,LEN(IN$)-I)
2055 ON T+1 GOTO 3000,3100,3200,3300,3400,3500,3600,370
     0,3800,3900
2200 DIM OPC$(607),FLD$(7)
2205 RESTORE
2210 FOR I=0 TO 7:READ FLD$(I):NEXT I
2215 I=0
2220 READ OPC$(I)
2225 IF LEFT$(OPC$(I),1)<>"1" THEN 2240
2230 FOR J=1 TO 7:OPC$(I+J)=OPC$(I):NEXT J
2235 I=I+7
2240 I=I+1:IF I<=607 THEN 2220
2250 RETURN
2300 DATA "B","C","D","E","H","L","*","A"
```

Program continues

Program continued

```
2400 DATA "NOP","3LD BC","LD (BC),A","INC BC","INC B"
2405 DATA "DEC B","2LD B","RLCA","EX AF,AF'","ADD #,BC"
2410 DATA "LD A,(BC)","DEC BC","INC C","DEC C","2LD C"
2415 DATA "RRCA","4DJNZ B","3LD DE","LD (DE),A","INC DE
"
2420 DATA "INC D","DEC D","2LD D","RLA","4JR","ADD #,DE
"
2425 DATA "LD A,(DE)","DEC DE","INC E","DEC E","2LD E"
2430 DATA "RRA","4JR NZ","3LD #","8LD #","INC #"
2435 DATA "INC H","DEC H","2LD H","DAA","4JR Z","ADD #,
#"
2440 DATA "6LD #","DEC #","INC L","DEC L","2LD L","CPL"
2445 DATA "4JR NC","3LD SP","8LD A","INC SP","INC *"
2450 DATA "DEC *","2LD *","SCF","4JR C","ADD #,SP"
2455 DATA "6LD A","DEC SP","INC A","DEC A","2LD A","CCF
"
2460 DATA "1LD B","1LD C","1LD D","1LD E","1LD H","1LD
L"
2465 DATA "1LD *","1LD A","1ADD A","1ADC A","1SUB A"
2470 DATA "1SBC A","1AND","1XOR","1OR","1CP"
2475 DATA "RET NZ","POP BC","3JP NZ","3JP","3CALL NZ"
2480 DATA "PUSH BC","2ADD A","RST 0","RET Z","RET"
2485 DATA "3JP Z","9","3CALL Z","3CALL","2ADC A"
2480 DATA "PUSH BC","2ADD A","RST 0","RET Z","RET"
2485 DATA "3JP Z","9","3CALL Z","3CALL","2ADC A"
2490 DATA "RST $8","RET NC","POP DE","3JP NC"
2495 DATA "7OUT A","3CALL NC","PUSH DE","2SUB A"
2500 DATA "RST $10","RET C","EXX","3JP C","5IN A"
2505 DATA "3CALL C","9","2SBC A","RST $18"
2510 DATA "RET PO","POP #","3JP PO","EX (SP),#"
2515 DATA "3CALL PO","PUSH #","2AND","RST $20"
2520 DATA "RET PE","JP (#)","3JP PE","EX DE,HL"
2525 DATA "3CALL PE","9","2XOR","RST $28"
2530 DATA "RET P","POP AF","3JP P","DI","3CALL P"
2535 DATA "PUSH AF","2OR","RST $30","RET M"
2540 DATA "LD SP,#","3JP M","EI","3CALL M"
2545 DATA "9","2CP","RST $38"
2600 REM CBXX
2605 DATA "1RLC","1RRC","1RL","1RR"
2610 DATA "1SLA","1SRA","9","9","9","9","9","9","9","9"
,"1SRL"
2615 DATA "1BIT 0","1BIT 1","1BIT 2","1BIT 3"
2620 DATA "1BIT 4","1BIT 5","1BIT 6","1BIT 7"
2625 DATA "1RES 0","1RES 1","1RES 2","1RES 3"
2630 DATA "1RES 4","1RES 5","1RES 6","1RES 7"
2635 DATA "1SET 0","1SET 1","1SET 2","1SET 3"
2640 DATA "1SET 4","1SET 5","1SET 6","1SET 7"
2700 REM ED40 - ED7F
2705 DATA "IN B,(C)","OUT (C),B","SBC HL,BC"
2710 DATA "8LD BC","NEG","RETN","IM 0","LD I,A"
2715 DATA "IN C,(C)","OUT (C),C","ADC HL,BC"
2720 DATA "6LD BC","9","RETI","9","LD R,A"
2725 DATA "IN D,(C)","OUT (C),D","SBC HL,DE"
2730 DATA "8LD DE","9","9","IM 1","LD A,I"
2735 DATA "IN E,(C)","OUT (C),E","ADC HL,DE"
2740 DATA "6LD DE","9","9","IM 2","LD A,R"
2745 DATA "IN H,(C)","OUT (C),H","SBC HL,HL"
2750 DATA "8LD HL","9","9","9","RRD"
2755 DATA "IN L,(C)","OUT (C),L","ADC HL,HL"
2760 DATA "6LD HL","9","9","9","RLD"
2765 DATA "9","9","SBC HL,SP","8LD SP"
2770 DATA "9","9","9","9","IN A,(C)","OUT (C),A"
2775 DATA "ADC HL,SP","6LD SP","9","9","9","9"
2800 REM EDA0 - EDBF
2805 DATA "LDI","CPI","INI","OUTI","9","9","9","9"
2810 DATA "LDD","CPD","IND","OUTD","9","9","9","9"
2815 DATA "LDIR","CPIR","INIR","OTIR","9","9","9","9"
2820 DATA "LDDR","CPDR","INDR","OTDR","9","9","9","9"
3000 DL=1
3010 M$=I0$+I1$
3020 RETURN
3100 DL=1
3110 IF LEN(I1$)<>0 THEN I1$=I1$+","
3120 M$=I0$+I1$+FLD$(I0 AND 7)
3130 IF I0=118 THEN M$="HALT"
3140 RETURN
3200 DL=2
3210 IF LEN(I1$)<>0 THEN I1$=I1$+","
3220 C=I1
3230 GOSUB 5000
3240 M$=I0$+I1$+"$"+C$
3250 RETURN
3300 DL=3
3310 IF LEN(I1$)<>0 THEN I1$=I1$+","
3320 C=256*I2+I1
3330 GOSUB 5020
3340 M$=I0$+I1$+"$"+C$
3350 RETURN
3400 DL=2
3410 IF LEN(I1$)<>0 THEN I1$=I1$+","
3420 IF I1<128 THEN C=L+2+I1 ELSE C=L+2+I1-256
3430 GOSUB 5020
3440 M$=I0$+I1$+"$"+C$
3450 RETURN
3500 DL=2
3510 C=I1
3520 GOSUB 5000
3530 GOTO 3630
3600 DL=3
3610 C=256*I2+I1
```

Program continues

Program continued

```
3620 GOSUB 5020
3630 IF LEN(I1$)<>0 THEN I1$=I1$+","
3640 M$=I0$+I1$+"($"+C$+")"
3650 RETURN
3700 DL=2
3710 C=I1
3720 GOSUB 5000
3730 GOTO 3830
3800 DL=3
3810 C=256*I2+I1
3820 GOSUB 5020
3830 IF LEN(I1$)<>0 THEN I1$=","+I1$
3840 M$=I0$+"($"+C$+")"+I1$
3850 RETURN
3900 DL=1
3910 M$="???"
3920 RETURN
4000 C=L
4010 GOSUB 5020
4020 L$=C$+"  "
4025 D$=""
4030 FOR LT=L TO L+DL-1
4040  CT=(PEEK(LT) AND 127)
4050  IF CT<32 THEN CT=46
4060  L$=L$+CHR$(CT)
4070  C=PEEK(LT)
4080  GOSUB 5000
4090  D$=D$+C$+" "
4100 NEXT LT
4110 L$=L$+STRING$(6-DL," ")+D$+STRING$(3*(5-DL)," ")+"
     "
4120 L=L+DL
4130 RETURN
5000 C$=MID$(HX$,INT(C/16)+1,1)+MID$(HX$,(C AND 15)+1,1
     )
5010 RETURN
5020 CT=INT(C/256)
5030 C$=MID$(HX$,INT(CT/16)+1,1)+MID$(HX$,(CT AND 15)+1
     ,1)
5040 CT=C-256*CT
5050 C$=C$+MID$(HX$,INT(CT/16)+1,1)+MID$(HX$,(CT AND 15
     )+1,1)
5060 RETURN
```

WE WILL NOT BE UNDERSOLD

DISK DRIVES

FOR TRS-80* Model I

CCI-100	5¼", 40 Track (102K)	**$314**

ADD-ON DRIVES FOR ZENITH Z-89

CCI-189	5¼", 40 Track (102K)	**$394**
Z-87	Dual 5¼" system	**$995**

External card edge and power supply included. 90 day warranty/one year on power supply.

RAW DRIVES
5¼" TEAC or TANDON **$ CALL** POWER SUPPLIES **$ CALL**

DISKETTES — Box of 10

5¼"	**Maxell $40**	**BASF/Verbatim**	**$26.95**
8"	**Maxell $45**	**BASF/Verbatim**	**$36.00**
PLASTIC FILE BOX–Holds 50 5¼" diskettes			**$19.00**
PLASTIC LIBRARY CASE	5¼" **$3.00**	8"	**$ 4.00**
HEAD CLEANING DISKETTE			**$25.00**
FLOPPY SAVER	**$11.95**	RINGS	**$ 6.95**

16K RAM KITS 2 for $37 $19

200 ns for TRS-80,* Apple II, (specify): **Jumpers** **$2.50**

SYSTEM SPECIAL

Apple II Plus 48K w/drive and controller. Epson MX-80 printer and interface. SUP-R Mod RF Modulator: List $2965 You Pay **$2299**

COMPUTERS/TERMINALS

ARCHIVES	64K, 2-Drives, 77 Track			**$ CALL**
ALTOS	ACS8000 Series			**$ CALL**
TRS-80*	II-64K	**$3499**	III-16K	**$ 899**
ZENITH	48K, all-in-one computer			**$2200**
ZENITH	Z-19			**$ 725**
TELEVIDEO	920C	**$ 729**	950	**$1039**
IBM	3101 Display Terminal			**$1189**
ATARI	400	**$ 479**	800	**$ 795**
MATTEL	INTELLIVISION			**$ 229**
APPLE PERIPHERALS				**$ CALL**

S-100 CALIFORNIA COMPUTER SYSTEMS

MAINFRAME	Model 2200A	**$349**
Z80 CPU	Model 2810	**$259**
MOTHER BOARD	Model 2501	**$106**
16K STATIC RAM, 200ns	Model 2116C	**$309**
32K STATIC RAM, 200ns	Model 2032C	**$619**
64K DYNAMIC RAM	Model 2065C	**$580**
FLOPPY DISC CONTROLLER	Model 2422A	**$345**
EXTENDER BOARD	Model 2520K	**$ 52**
2P + 2S I/O	Model 2718A	**$309**

PRINTERS

NEC SPINWRITER
Letter Quality High Speed Printer

R.O.	**$2395**
R.O. with tractor feed	**$2555**
KSR with tractor feed	**$2795**

NEC SPINWRITER	3500 Series, 5 new models				**$CALL**
EPSON	MX-70	MX-80	MX-80FT	MX100	**$CALL**
PAPER TIGER					
IDS 445	Graphics & 2K buffer				**$ 695**
IDS 460	Graphics & 2k buffer				**$1050**
IDS 560	Graphics				**$1450**
ANADEX	DP-8000	**$ 849**		DP-9500/01	**$1295**
OKIDATA					
Microline 80	Friction & pin feed				**$ 415**
Microline 80	Friction, and pin & tractor feed				**$ 500**
Microline 82	Friction & pin feed feed				**$ 615**
Microline 83	120 cps, uses up to 15" paper				**$ 849**
CENTRONICS	739, new model with graphics				**$ 825**

PRINTER SPECIAL

SEIKOSHA GP-80M List $399 You Pay **$319**

MONITORS

BELL & HOWELL	9" B & W BHD911			**$195**
LEEDEX	12" B & W	**$ 129**	13" Color	**$369**
SANYO	9" B & W VM4509			**$149**
SANYO	12" B & W DM5012			**$219**
SANYO	12" Green Screen DM5112			**$238**
SANYO	13" Color DMC6013			**$406**
ZENITH	13" Color			**$349**

TELECOMMUNICATIONS

LIVERMORE STAR MODEM 2-year guarantee	**$125**
UNIVERSAL DATA SYSTEMS UDS-103	**$179**
D-CAT HARD WIRED DIRECT MODEM	**$189**
AUTO-CAT Auto Answer, Direct Connect Modem	**$249**
D.C. HAYES MICRO-MODEM	**$295**
CCI Telnet Communications Package	**$135**

APPLE ACCESSORIES AND SOFTWARE

VISICALC	**$155.00**
DB MASTER	**$159.00**
Z-80 SOFTCARD	**$259.00**
VIDEX BOARD	**$259.00**
KEYBOARD ENHANCER	**$110.00**
16K CARD	**$169.00**
APPLE JOYSTICK	**$ 49.00**
SUP-R MOD	**$ 25.00**
CCS CARDS	**$ CALL**
GALAXIAN	**$ 22.95**
SPACE ALBUM	**$ 35.00**
ASTEROIDS	**$ 17.95**
FLIGHT SIMULATOR	**$ 29.00**
WIZARD & PRINCESS	**$ 28.00**
SARGON 2	**$ 29.00**
HI-RES FOOTBALL	**$ 35.00**
MYSTERY HOUSE	**$ 24.00**

For fast delivery, send certified checks, money orders or call to arrange direct bank wire transfers. Personal or company checks require two to three weeks to clear. All prices are **mail order only** and are subject to change without notice. **Call for shipping charges.**

DEALER (NATIONAL/INTERNATIONAL) INQUIRIES INVITED **Send for FREE Catalogue**

The CPU SHOP

✓298

TO ORDER CALL TOLL FREE 1-800-343-6522
TWX: 710-348-1796 Massachusetts Residents call 617/242-3361

5 Dexter Row, Dept. MC08M
Charlestown, Massachusetts 02129
Hours 10AM-6PM (EST) Mon.-Fri. (Sat. till 5)

Technical Information call 617/242-3361
Massachusetts Residents add 5% Sales Tax
Tandy Corporation Trademark/®Digital Research

master charge THE INTERBANK CARD
VISA

✓See List of Advertisers on page 322

A slide and negative indexing program.

Hold That Pose

Doug and Pam Ebert
301 N. Clement
Aviston, IL 62216

Are you plagued with boxes of jumbled photos or slides? Have you spent hours looking for a picture you just took last week? For the serious amateur or professional photographer, keeping track of your creative efforts can be time consuming and confusing.

Here's the program that's a perfect accessory for your camera. You can index your negatives, photos and slides with a simple numbering system. Not only will each photo be cross-referenced under three subjects, but exposure data (i.e. date taken, f/stop, shutter speed) can be recalled on each entry when needed.

Listing Options

This program allows you to see all the photos you have in the index or list photos by subject area or a particular special effect. No longer will you sort through stacks of slides looking for pictures of a certain subject—the computer will search the index for you.

Need to know how that picture of an orchid was taken? Enter the file number of the photo in list mode #4 (Table 2) and the listing will include the f/stop, shutter speed, lenses and special effects used.

If you have a printer, use the photo index option #6 (Table 1) and you won't need to keep a written copy of the entries in case of an I/O error.

The Data Tape

After setting up memory areas, the program will ask if you want a data tape read in; respond with a no if this is your first run. On

Program Listing.

```
10 'PHOTO INDEX PROGRAM
20 'BY DOUG & PAM EBERT
30 '119 E MAIN, LEBANON, IL  62254
40 CLS:PRINT@410,"STAND-BY"
50 CLEAR4500
60 DEFINTN,P,S,D,F,L,R,I,J,C
70 DIMN(600),PS(600),S1(600),S2(600),D1(600),D2(600),OS(600),SH$
(600),L1(600),L2(600),F1(600),F2(600)
80 DIMDA$(32),LN$(14),FL$(9)
90 FA$=" ###           %    %        %    %        %    %
  ##/##"
100 FB$=" ###       %    %       F/##.#      %         %      %
      %"
110 FC$=" DATE      %    %       %    %      %         %      %
      %"
120 FD$="##/##      %    %"
130 FE$="    ##. %    %    "
140 FF$="##. %              % "
150 'READ DATA STRINGS
160 FORI=1TO32
170 READDA$(I):NEXTI
180 FORI=1TO14
190 READLN$(I):NEXTI
200 FORI=1TO9
210 READFL$(I):NEXTI
220 'READ DATA TAPE
230 R=0
240 CLS:INPUT"DO YOU WISH TO LOAD A DATA TAPE";YN$
250 IFLEFT$(YN$,1)="Y"GOSUB3500
260 CLS:PRINT@155,"PHOTO INDEX":PRINT:PRINT
270 PRINT"1  - INSTRUCTIONS":PRINT"2  - LIST ENTRIES":PRINT"3  -
 ADD AN ENTRY"
280 PRINT"4  - DELETE AN ENTRY":PRINT"5  - SAVE INDEX ON TAPE":P
RINT"6  - PRINT OUT INDEX":PRINT:PRINT
290 INPUT"ENTER MODE DESIRED";CH
300 IFCH<1ORCH>6GOTO290
310 ONCHGOSUB500,1000,2000,2500,3000
320 IFCH=6THENST%=PEEK(14312)AND240:IFST%=48THENGOSUB3400ELSEPRI
NT"PRINTER UNAVAILABLE":FORI=1TO1000:NEXTI
330 GOTO260
500 'INSTRUCTIONS
510 CLS:PRINT@27,"INSTRUCTIONS":PRINT
```

Program continues

1-Instructions
2-List Entries
3-Add An Entry
4-Delete An Entry
5-Save Index on Tape
6-Print Out Index

Table 1. Photo Index

1) All Entries
2) By Subject
3) By Special Effect
4) By File Number
5) Return to Menu

Table 2. List Modes

N	=	File number
PS	=	Primary Subject
S1	=	Secondary subject
S2	=	Secondary subject
D1	=	Month photo was taken
D2	=	Year photo was taken
OS	=	F/stop
SH$	=	Shutter speed
L1	=	Lens used
L2	=	Secondary lens
F1	=	Filter/special effect used
F2	=	Secondary filter/special effect
DA$	=	Subject names
LN$	=	Lens names
FL$	=	Filter/special effect names

Table 3. Variables

"This program allows for easy modification to fit your needs. As it appears here, the program needs 32K. . . and holds a total of 600 entries."

subsequent runs the computer will read in the data you saved on tape at the end of a previous run and will compute the number of entries you now have on file. (The computer automatically checks the tape to insure it is photo data and will respond with an incorrect data tape message if it is the wrong tape or not fully rewound.) Add or delete entries as needed and before ending the run, save the index on tape by using the photo index option #5.

Preparing Photos

In order to use this program effectively, go through the photos, negatives and slides you already have. As you enter each photo in the index, give any data asked for that you have available. Perhaps you don't know the f/stop or lens used—just enter 00 or NONE as applicable. Be sure to record the file number on each photo and its negative as you enter it in the computer. Write numbers lightly on the backs of photos and corners of slides. Negatives can be stored in numbered envelopes or specially-made sleeves.

The computer generates each new file number for you—just record the number on the photo and store it in numerical order. To update the index, sit down for a few minutes when you get a new set of photos and enter them. Then you'll be able to find any picture with ease when you want it.

Modifications

This program allows for easy modification to fit your needs. As it appears here, the program needs 32K of memory and holds a total of 600 entries. For a 16K system the number of entries would decrease to 150. You must then edit lines 50 and 70 to fit your computer's memory. Line 50 would be CLEAR 2200 and line 70 would look like this: DIMN(150),PS(150).... Also change line 3550 to FORI = 1TO150.

All of the subject areas, lenses and special effect choices are in data statements at the end of the program, lines 6001-6006. If you take more photographs of football than of mountains, just change subject names to fit your needs. The types of lenses and special effects can be modified in the same way. Please note that there is a six-character limit on subject names, eleven on lenses, and ten on special effects. To add names to any of these classifications, you must increase the respective dimension statement in line 80. Also, change the lines which print the information tables: lines 160 and 4020 for subject types, 180 and 4120 for lens types, and lines 200 and 4220 for special effect types.

So put a new roll of film in your camera. You've got time to take more pictures and know you can find them when you're through.■

Program continued

```
520 PRINT"THIS PROGRAM CREATES AN INDEX TO CATALOG & SORT PHOTOS
."
530 PRINT"THE 'LIST ENTRIES' MODE WILL DISPLAY A LIST OF ALL PHO
TOS:
540 PRINTTAB(10)"A) IN CATALOG":PRINTTAB(10)"B) OF A PARTICULAR
SUBJECT"
550 PRINTTAB(10)"C) USING A SPECIFIC SPECIAL EFFECT":PRINTTAB(10
)"D) BY ENTRY NUMBER"
560 PRINT"THE 'ADD ENTRY' MODE ALLOWS ADDITION OF NEW PHOTOS TO
THE"
570 PRINT"CATALOG IN MEMORY."
580 PRINTTAB(10)"FOR EACH ADDITION, THE PROGRAM WILL SUPPLY THE
PHOTO"
590 PRINTTAB(10)"NUMBER.  USER MUST INPUT EXPOSURE DATA AND PRIM
ARY"
600 PRINTTAB(10)"SUBJECT.  USER MAY ALSO CROSS-REFERENCE EACH PH
OTO"
610 PRINTTAB(10)"IN TWO ADDITIONAL SUBJECT CATEGORIES."
620 PRINTTAB(5)"PRESS ANY KEY TO CONTINUE"
630 I$=INKEY$:IFI$=""GOTO630
640 CLS:PRINT"THE 'DELETE ENTRY' MODE REMOVES PHOTO FROM CATALOG
 IN MEMORY.":PRINT
650 PRINT"THE 'SAVE INDEX ON TAPE' MODE PLACES THE CATALOG IN ME
MORY"
660 PRINT"ONTO A CASSETTE TAPE FOR FUTURE USE.":PRINT
670 PRINT"THE 'PRINT OUT INDEX' MODE CREATES A HARD COPY OF THE
CATALOG"
680 PRINT"(WITH AN IN-LINE PRINTER)."
690 PRINT:PRINTTAB(5)"PRESS ANY KEY TO RETURN TO MENU"
700 I$=INKEY$:IFI$=""GOTO700
710 RETURN
1000 'LIST ENTRIES MODE
1010 CLS:PRINT@150,"CHOOSE LIST MODE DESIRED":PRINT:PRINT
1020 PRINTTAB(10)"1) ALL ENTRIES":PRINTTAB(10)"2) BY SUBJECT":PR
INTTAB(10)"3) BY SPECIAL EFFECT"
1030 PRINTTAB(10)"4) BY FILE NUMBER":PRINTTAB(10)"5) RETURN TO M
ENU"
1040 PRINT:INPUT"ENTER LIST MODE";CH
1050 IFCH<1ORCH>5GOTO1040
1060 IFCH=5RETURN
1070 ONCHGOSUB1250,1500,1700,1800
1080 GOTO1010
1250 'LIST ALL ENTRIES
1260 C=0:GOSUB4300
1270 FORI=1TOR
1280 IFPS(I)=0GOTO1310
1290 C=C+1:PRINTUSINGFA$;N(I),DA$(PS(I)),DA$(S1(I)),DA$(S2(I)),D
1(I),D2(I)
1300 IFC=10THENGOSUB1360
1310 NEXTI
1320 C=10-C:FORI=1TOC:PRINT:NEXTI
1330 PRINT@896,"END OF DATA   PRESS ANY KEY TO RETURN TO LIST ME
NU      "
1340 I$=INKEY$:IFI$=""GOTO1340
1350 RETURN
1360 PRINT:INPUT"ENTER '1' TO SEE NEXT PAGE    '2' TO SEE PREVIO
US PAGE";CH
1370 C=0:IFCH=2THENI=I-20:IFI<1THENI=0
1380 PRINT@128," ":RETURN
1500 'LIST BY SUBJECT
1510 GOSUB4000
1520 PRINT@704,"CHOOSE SUBJECT #    ENTER '0' TO RETURN TO LIST
MENU"
1530 INPUTCH:ST=1:IFCH=0RETURN
1540 IFCH>32GOTO1530
1550 GOSUB5000
1560 RETURN
1700 'LIST BY EFFECT
1710 GOSUB4200
1720 PRINT"CHOOSE EFFECT #   ENTER '0' TO RETURN TO LIST MENU"
1730 INPUTCH:ST=2:IFCH=0RETURN
1740 IFCH>9GOTO1720
1750 GOSUB5000
1760 RETURN
```

Program continues

Program continued

```
1800 'LIST BY PIC #
1810 CLS:INPUT"ENTER # OF PHOTOS YOU WANT TO SEE(1 TO 3)";J1
1820 IFJ1<1ORJ1>3GOTO1810
1830 FORI=1TOJ1
1840 INPUT"ENTER PHOTO # DESIRED";J2(I):NEXTI
1850 GOSUB3200
1860 FORJ=1TOJ1
1870 FORI=1TOR
1880 IFPS(I)=0GOTO1900
1890 IFN(I)=J2(J)GOTO1920
1900 NEXTI
1910 IFI>RPRINT"ENTRY #";J2(J);" NOT FOUND":PRINT:GOTO1930
1920 GOSUB3300
1930 NEXTJ
1940 PRINT@896,"ENTER '1' TO LIST MORE PHOTOS    '2' TO RETURN T
O LIST MENU"
1950 INPUTCH:IFCH=1GOTO1810ELSERETURN
2000 'ADD ENTRY
2010 CLS:R=R+1:N(R)=N(R-1)+1
2020 GOSUB4000
2030 PRINT@704,"CHOOSE PRIMARY SUBJECT # FOR NEW ENTRY - #";N(R)
:INPUTPS(R)
2040 IFPS(R)>32GOTO2020
2050 INPUT"CHOOSE TWO SECONDARY SUBJECTS (#,#)";S1(R),S2(R)
2060 IFS1(R)>32ORS2(R)>32GOTO2050
2070 CLS:INPUT"ENTER DATE PHOTO WAS TAKEN (MM,YY)";D1(R),D2(R)
2080 IFD1(R)>12ORD2(R)>99GOTO2070
2090 INPUT"ENTER F/STOP USED";OS(R)
2100 INPUT"ENTER SHUTTER SPEED (1/1000, 1 MIN) UP TO 6 CHARACTER
S";SH$(R)
2110 GOSUB4100
2120 PRINT:INPUT"CHOOSE BASIC LENS # USED";L1(R)
2130 IFL1(R)>14GOTO2120
2140 INPUT"CHOOSE ATTACHMENT # (EX-TUBE,NONE)";L2(R)
2150 IFL2(R)>14GOTO2140
2160 CLS:GOSUB4200
2170 INPUT"CHOOSE BASIC EFFECT # USED";F1(R)
2180 IFF1(R)>9GOTO2170
2190 INPUT"CHOOSE SECOND EFFECT #";F2(R)
2200 IFF2(R)>9GOTO2190
2210 GOSUB3200
2220 I=R:GOSUB3300
2230 PRINT:PRINT:INPUT"IS THIS DATA CORRECT";YN$
2240 IFLEFT$(YN$,1)="N"GOTO2020
2250 INPUT"ADD ANOTHER ENTRY";YN$
2260 IFLEFT$(YN$,1)="N"RETURN
2270 GOTO2010
2500 'DELETE ENTRY
2510 CLS:INPUT"ENTER PIC # TO BE DELETED ('0' TO RETURN TO MENU)
";J
2520 IFJ=0RETURN
2530 PRINT"IS PIC # ";J;:INPUT"TO BE DELETED";YN$
2540 IFLEFT$(YN$,1)="N"GOTO2510
2550 FORI=1TOR
2560 IFN(I)<>JNEXTI
2570 IFI>RPRINT"PIC # ";J;" NOT FOUND.":FORI=1TO1000:NEXTI:GOTO2
510
2580 PS(I)=0
2590 PRINT"ENTRY DELETED":FORI=1TO1000:NEXTI:GOTO2510
3000 'SAVE ENTRIES
3010 CLS:I$="":PRINT"PREPARE CASSETTE FOR RECORDING DATA TAPE.":
INPUT"PRESS <ENTER> WHEN READY";I$
3020 IFI$<>""GOTO3010
3030 R=R+1:N(R)=-1:PS(R)=-1:S1(R)=-1:S2(R)=-1:D1(R)=-1:D2(R)=-1:
OS(R)=-1:SH$(R)="-1":L1(R)=-1:L2(R)=-1:F1(R)=-1:F2(R)=-1
3040 PRINT#-1,"PHOTO INDEX"
3050 FORI=1TOR
3060 IFPS(I)=0GOTO3080
3070 PRINT#-1,N(I),PS(I),S1(I),S2(I),D1(I),D2(I),OS(I),SH$(I),L1
(I),L2(I),F1(I),F2(I)
3080 NEXTI
3090 INPUT"DO YOU WISH TO MAKE ANOTHER COPY";YN$
3100 IFLEFT$(YN$,1)="Y"GOTO3010
3110 R=R-1:RETURN
3200 'DATA HEADINGS
3210 CLS:PRINT@24,"EXPOSURE";:PRINT@55,"FILTER &"
3220 PRINT@64,"PIC #";:PRINT@72,"SUBJECTS";:PRINT@90,"DATA";:PRI
NT@105,"LENS";:PRINT@120,"EFFECTS"
3230 PRINTSTRING$(63,"-"):RETURN
3300 'PRINT ALL DATA
3310 PRINTUSINGFB$;N(I),DA$(PS(I)),OS(I),LN$(L1(I)),FL$(F1(I))
3320 PRINTUSINGFC$;DA$(S1(I)),SH$(I),LN$(L2(I)),FL$(F2(I))
3330 PRINTUSINGFD$;D1(I),D2(I),DA$(S2(I)):PRINT:RETURN
```

Program continues

DISK DRIVE SALE

MDX-2 System Expansion for TRS 80's

The new MDX-2 system expansion board from COMPUTEX provides more capabilities than any other system expansion currently available.

FEATURES INCLUDE

- Capability to build only what you need.
- Silk screened and solder masked P.C.B.
- Dip switches are used instead of hardware jumpers.
- No messy vertically mounted termination resistors.

• On board direct connect modem kit	$39.95
• 2K/4K Eprom kit	$17.95
• RS232/20m.A. 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
• MDX-2 printed circuit board	$74.95
• Hardware and socket kit	$19.95
• Centronics line printer port	$8.95

Buy just the system expansion board and those kits you want, or save even more and buy the complete kit.

MDX-2 KIT COMPLETE
$269.00

TANDON TM100-1

Bare Drive $235.00 — **With case/supply $285.00**
Single sided 40 track • 102KB Single Density, 180KB Double Density

TANDON TM100-2

Bare Drive $335.00 — **With case/supply $385.00**
Double sided 40 track • 204KB Single Density, 360KB Double Density

TANDON TM100-3

Bare Drive $335.00 — **With case/supply $385**
Single sided 80 track • 204KB Single Density, 360KB Double Density

TANDON TM100-4

Bare Drive $455 — **With case/supply $505.00**
Double sided 80 track • 408KB Single Density, 735KB Double Density

All TANDON drives from **COMPUTEX** *come with comprehensive users manual, TANDON maintenance manual and schematics. In addition, all drives are electrically tested before, during and after their 48 hour burn in.*

DISK DRIVE ACCESSORIES
Two Drive Cable $24.95
Four Drive Cable $34.95
Drive Extender Cable $14.95
Model III Second Drive Cable $4.95

Model III Disk Drive Kits & Drives

Now available from **COMPUTEX**, a Radio Shack Disk Drive Kit for your Model III for **only $429.95.**
Compare these standard features on our list to the others.

1. Drive Controller Board
2. All necessary interconnecting cables
3. Switching Power Supply
4. Disk Drive Mounting Brackets
5. Internal Disk Drive Cable
6. Comprehensive installation and theory manual
7. 16K RAM

COMPLETE MODEL III DRIVE PACKAGES

These packages all include, the Model III Disk Drive Kit and the Micro Systems Software Model III DOS.

M3DK-1	(2-TM100-1 Drives) provides over 350K Bytes storage	919.95
M3DK-2	(2-TM100-2 double headed drives) provides over 700K Bytes storage	1,149.95
M3DK-3	(2-TM100-3 80 track drives) provides over 700K Bytes storage	1,199.95
M3DK-4	(2-TM100-4 double headed 80 track drives) provides over 1.4 M Bytes storage	1,399.95

COMPUTEX Comprehensive Test Program

As one of the leading suppliers of disk drives and TRS80 systems, we had to have a systems test program that we knew worked. Since no complete systems test program already existed, we wrote our own. The *COMPUTEX COMPREHENSIVE TEST* (**CCT**) program was written exclusively for **COMPUTEX** and is now available to everyone.

Features

DISK DRIVE SPEED TEST • DISK DRIVE ALIGNMENT PROGRAM*
EXPANSION INTERFACE RAM TEST • KEYBOARD RAM TEST • VIDEO RAM TEST
KEYBOARD BOUNCE TEST • ROM CHECKSUM PROGRAM
FULL SCREEN RAM DISPLAY PROGRAM
*The Drive Alignment Program requires an Alignment Disk.

The CCT program is a must for disk drive users and should be a part of everyones library.

CCT Program $21.95
Dysan Alignment Disk $39.95

MEMORY/SPECIAL FUNCTION

FD1771B-01	23 95	FD1793B-01	44 95
TR1602	3 95	AY5 1013A	4 95
MC1372	6 48	MC6674	13 00
2716-450	12 95	2114 450	5 95
Z80A	14 95	2102	1 75
CC74S188	Programmed for LNW Color CPU		9 95
CC2716-6	Level II Roms for Color CPU		99 95

LINEAR

LM1488	1 20	TL084	1 75
LM1489	1 20	MC1458	58
UA7805	1 10	MEM311P1	90
UA7812	1 10	MC14412	17 00
UA7905	1 10	8N805	6 00
[illegible]	[illegible]	75452	35

DIODES TRANSISTORS / SEMICONDUCTORS

1N914/1N4148	05	KBP02	79
1N4001	09	2N3904	25
1N5231	25	2N3906	25
1N5244	25	MPU131	89
1N5244	25	J175	89

CRYSTALS

1 0 MHZ	4 00	3 579 MHZ	3 95
4 0 MHZ	4 00	16 0 MHZ	4 95

1% PRECISION RESISTORS

267 OHMS	324 OHMS	619 OHMS
845 OHMS	1 24 K OHMS	2 55 K OHMS
8 66 K OHMS	8 87 K OHMS	18 7 K OHMS
21 5 K OHMS	95 3 K OHMS	97 6 K OHMS
165 K OHMS	210 K OHMS	237 K OHMS

All 1% Resistors are $ 18 each

I C SOCKETS

8 PIN Low Profile	12	20 PIN Low Profile	23
14 PIN Low Profile	14	22 PIN Low Profile	27
16 PIN Low Profile	16	24 PIN Low Profile	30
18 PIN Low Profile	21	40 PIN Low Profile	47

7400 SERIES TTL

7438	20	74157	60
7442	45	74161	83
7451	56	74164	83
7474	32	74165	83
7486	32	74166	1 15
7490	42	74173	1 15
7492	42	74175	75
7493	42	74176	75
7495	58	74194	83
74121	32	74195	64
74123	45	74151	54
74125	45	74366	64
74132	41	74367	68

74LS00 SERIES - LOW POWER SCHOTTKY

74LS00	27	74LS139	75
74LS02	27	74LS153	75
74LS04	29	74LS155	90
74LS05	39	74LS157	75
74LS08	27	74LS161	1 10
74LS09	32	74LS164	1 18
74LS10	27	74LS166	1 85
74LS11	71	74LS174	1 69
74LS13	56	4LS175	95
74LS14	1 15	74LS193	1 10
74LS15	31	74LS240	1 85
74LS20	27	74LS241	1 85
74LS21	31	74LS244	1 85
74LS30	26	74LS245	2 25
74LS32	31	74LS257	85
74LS74	44	74LS367	69
74LS86	41	74LS373	1 89
74LS93	70	74LS374	1 85
74LS123	1 18	74LS393	2 35
74LS124	1 69	74LS138	86
74LS132	95		

74S00 SCHOTTKY

74S04	55	74S05	47
74S22	47	74S32	47
74S64	47	74S74	61
74S112	71	74S161	1 75
74S174	1 59	74S175	1 50
74S280	2 45	74S387	5 95

LNW COLOR COMPUTER KIT
$499.95
(less P.C.B.)

Includes all components to assemble the LNW color computer: high resolution video RAM and 16K user RAM, keyboard, and Level II compatible ROMS. Printed circuit board not included.

LNW EXPANSION INTERFACE KIT
$219.95
(less P.C.B.)

Includes all components to assemble the LNW Expansion Interface plus 32K of RAM. It does not include the printed circuit board, dual cassette relay or cabinet.

•

A beautiful custom all wood cabinet is available for housing the LNW Expansion Interface. **$99.95**

ONE YEAR GUARANTEE
on our 200 N.S. 16K Memory Kits

16K - $23.95 — 32K - $45.95 — 48K - $74.95

These are top quality, new memory chips at unbelievably low prices.

(713) 488-8022

COMPUTEX ✓392
ELECTRONICS COMPUTERS

321 El Dorado Blvd. • Webster, Tx. 77598
Visa and Mastercard accepted
Personal and company checks require two weeks to clear

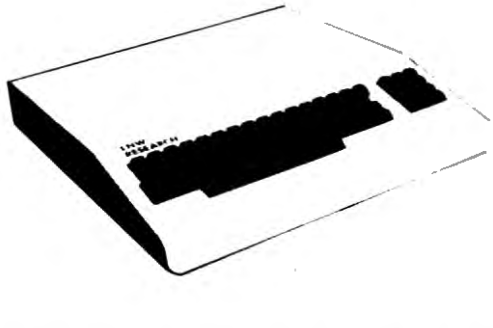
Please visit our new showroom just off the Gulf Freeway. We have expanded, changed our name back to COMPUTEX, added more personnel in order to offer you complete service and reliability.

✓See List of Advertisers on page 322

Program continued

```
3400 'PRINT OUT INDEX
3410 LPRINTTAB(24)"EXPOSURE"TAB(55)"FILTER &"
3420 LPRINT"PIC #"TAB(9)"SUBJECTS"TAB(27)"DATA"TAB(42)"LENS"TAB(
57)"EFFECTS"
3430 LPRINTSTRING$(63,"-")
3440 FORI=1TOR
3450 IFPS(I)=0GOTO3490
3460 LPRINTUSINGFB$;N(I),DA$(PS(I)),OS(I),LN$(L1(I)),FL$(F1(I))
3470 LPRINTUSINGFC$;DA$(S1(I)),SH$(I),LN$(L2(I)),FL$(F2(I))
3480 LPRINTUSINGFD$;D1(I),D2(I),DA$(S2(I)):LPRINT
3490 NEXTI
3495 RETURN
3500 'READ DATA TAPE
3510 CLS:INPUT"PREPARE CASSETTE WITH DATA TAPE.  PRESS <ENTER> W
HEN READY";I$
3520 IFI$<>""GOTO3510
3530 INPUT#-1,A$
3540 IFA$<>"PHOTO INDEX"PRINT"INCORRECT DATA TAPE":FORI=1TO1000:
NEXTI:GOTO3510
3550 FORI=1TO600
3560 INPUT#-1,N(I),PS(I),S1(I),S2(I),D1(I),D2(I),OS(I),SH$(I),L1
(I),L2(I),F1(I),F2(I)
3570 IFN(I)=-1GOTO3590
3580 NEXTI
3590 R=I-1:PRINT:PRINTR;" ENTRIES NOW IN MEMORY"
3600 INPUT"PRESS <ENTER> TO CONTINUE";I$
3610 IFI$=""THENRETURNELSEGOTO3610
4000 'PRINT SUBJECTS
4010 CLS:PRINT@24,"LIST OF SUBJECTS"
4020 FORI=1TO32
4030 PRINTUSINGFE$;I,DA$(I);:NEXTI
4040 PRINT" ":PRINTSTRING$(64,"-"):RETURN
4100 'PRINT LENSES
4110 CLS:PRINT@30,"LENSES"
4120 FORI=1TO14
4130 PRINTUSINGFF$;I,LN$(I);:NEXTI
4140 PRINT" ":PRINTSTRING$(64,"-"):RETURN
4200 'PRINT EFFECTS
4210 CLS:PRINT@24,"FILTERS AND EFFECTS"
4220 FORI=1TO9
4230 PRINTUSINGFF$;I,FL$(I);:NEXTI
4240 PRINT" ":PRINTSTRING$(64,"-"):RETURN
4300 'HEADING
4310 CLS:PRINT@14,"PRIMARY":PRINT@64,"PIC #";:PRINT@78,"SUBJECT"
;:PRINT@93,"SECONDARY SUBJECT(S)";:PRINT@123,"DATE"
4320 PRINTSTRING$(63,"-"):RETURN
5000 'SEARCH ROUTINE
5010 GOSUB4300
5020 C=0
5030 FORI=1TOR
5040 IFPS(I)=0GOTO5060
5050 ONSTGOSUB5100,5200
5060 NEXTI
5070 C=11-C:FORI=1TOC:PRINT:NEXTI
5080 PRINT"END OF LIST.  PRESS ANY KEY TO RETURN TO LIST MENU"
5090 I$=INKEY$:IFI$=""GOTO5090
5100 IFPS(I)=CHORS1(I)=CHORS2(I)=CHGOSUB5300
5110 RETURN
5200 IFF1(I)=CHORF2(I)=CHGOSUB5300
5210 RETURN
5300 C=C+1:PRINTUSINGFA$;N(I),DA$(PS(I)),DA$(S1(I)),DA$(S2(I)),D
1(I),D2(I)
5310 IFC<10RETURN
5320 C=0:PRINT:PRINTTAB(5)"PRESS ANY KEY TO CONTINUE LISTING"
5330 I$=INKEY$:IFI$=""GOTO5330
5340 PRINT@128," ":RETURN
6000 'DATA
6001 DATA "ANIMAL","AUTUMN","BIRDS","CATS","CELEB","DOGS","FLOWE
R","FOOD","FOREST","HOLDAY","HORSES","HOUSE"
6002 DATA "INSECT","KIDS","MACRO","MOOD","MTNS","PEOPLE","PLANTS
","RELIG","WATER","SCENIC","SEA","SILOET","SPORTS"
6003 DATA "SPRING","STILIF","SUMMER","SUNSET","TREE","WINTER","N
ONE"
6004 DATA "50mm","80-200 ZOOM","WIDE-ANGLE","MACRO","2X-TELECONV
","3X-TELECONV","EX-TUBE12mm","EX-TUBE20mm"
6005 DATA "EX-TUBE32mm","EX-TUBE36mm","EX-TUBE48mm","EX-TUBE56mm
","EX-TUBE68mm","NONE"
6006 DATA "COLORBURST","DIFFUSION","DOUBLE-EXP","MULT-IMAGE","PO
LARIZER","HALO SPOT","6 PT. STAR","82B","NONE"
3040 PRINT#-1,"PHOTO INDEX"
3050 FORI=1TOR
3060 IFPS(I)=0GOTO3080
3070 PRINT#-1,N(I),PS(I),S1(I),S2(I),D1(I),D2(I),OS(I),SH$(I),L1
(I),L2(I),F1(I),F2(I)
```

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

LNW RESEARCH
CORPORATION

2620 WALNUT
TUSTIN CA. 92680 53

ORDERS & INFO. NO. 714-544-5744
SERVICE NO. 714-641-8850

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, LWN 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

VISA MASTER CHARGE ACCEPTED

UNLESS NOTED ADD $3 FOR SHIPPING

When man meets machine code, it's important to remember the basics.

Be a Super USR!

Roger C. Alford
406 Packard #104
Ann Arbor, MI 48104

As most programmers know, machine language programs and routines run much faster than equivalent Basic programs and allow greater control over the computer. Also, machine language is used almost exclusively when real-time programming is necessary. However, when Level II Basic was developed it appears that linking Basic and machine language routines was not a major concern. Yet, there are ways to make your computer fulfill your needs in spite of these limitations.

For Basic to know the location of the USR routine, the address must be POKEd into locations 16526 (low address byte) and 16527 (high address byte). One method of implementing several different USR function calls is to POKE a particular function's address into these locations immediately prior to the use of the USR call. This method is fairly straightforward and is illustrated in Program

```
10 CALL 1000
.
.
50 CALL 2000
.
.
.
1000 POKE 16526,N1:POKE 16527,N2:
PRINT USR(A)
1010 RETURN
.
.
.
2000 POKE 16526,N3:POKE 16527,N4:
Z% = USR(4)
2010 RETURN
```

Program Listing 1. This is one method that allows the single USR function to make various function calls.

```
10 POKE 16526,1:POKE 16527,125
20 CLS
30 INPUT "XX$";XX$
40 INPUT "X%";X%
50 Y=USR(X%)
60 PRINT
70 GOTO 40
```

Program Listing 2(a).

Program Listing 2(b).

```
7D01         00100        ORG     32001       ;STARTING ADDRESS = 7D01H
28A7         00110 STROUT EQU     28A7H       ;ROM STRING-OUT ROUTINE
0A7F         00120 GETVAL EQU     0A7FH       ;ROM GET-USR PARAM. VALUE ROUT.
260D         00130 VARPTR EQU     260DH       ;ROM VARPTR ROUTINE
033A         00140 COUT   EQU     033AH       ;ROM CHARACTER-OUT ROUT.
0A9A         00150 RETURN EQU     0A9AH       ;RETURN VALUE ROUTINE
             00160 ;
             00170 ;PROGRAM STARTS HERE
             00180 ;
7D01 CD7F0A  00190 START  CALL    GETVAL              ;GET PARAMETER VALUE
7D04 7C      00200        LD      A,H                 ;CHECK FOR 0 IN HIGH BYTE
7D05 FE00    00210        CP      0
7D07 2015    00220        JR      NZ,ERR              ;IF NOT, ERROR
7D09 7D      00230        LD      A,L                 ;FIND DESIRED ROUTINE
7D0A FE01    00240        CP      1                   ;L=1?
7D0C 2815    00250        JR      Z,ROUT1             ;IF YES, EXEC. ROUTINE 1
7D0E FE02    00260        CP      2                   ;L=2?
7D10 2820    00270        JR      Z,ROUT2             ;IF YES, ROUTINE 2
7D12 FE03    00280        CP      3                   ;L=3?
7D14 2821    00290        JR      Z,ROUT3             ;IF YES, ROUTINE 3
7D16 FE04    00300        CP      4                   ;L=4?
7D18 2822    00310        JR      Z,ROUT4             ;IF YES, ROUTINE 4
7D1A FE05    00320        CP      5                   ;L=5?
7D1C 2833    00330        JR      Z,ROUT5             ;IF YES, ROUTINE 5
7D1E 217C7D  00340 ERR    LD      HL,ERRMSG           ;IF NONE, PRINT ERR MESSG
7D21 181C    00350        JR      PRTRET              ; AND RETURN
7D23 21267E  00352 ROUT1  LD      HL,VARY             ;LOAD ADDR OF "Y%"
7D26 CD0D26  00354        CALL    VARPTR              ;FIND ADDR OF VALUE
7D29 EB      00356        EX      DE,HL               ;PLACE ADDR IN HL PR.
7D2A CB26    00358        SLA     (HL)                ;SHIFT LOW BYTE-PUT HIGH
             00360                                    ; BIT INTO CARRY
7D2C 23      00362        INC     HL                  ;POINT TO HIGH BYTE
7D2D CB16    00364        RL      (HL)                ;ROTATE LEFT W/CARRY IN
7D2F C39A0A  00366        JP      RETURN              ;RETURN TO BASIC
7D32 21B07D  00380 ROUT2  LD      HL,MSG2
7D35 1808    00390        JR      PRTRET
7D37 21CC7D  00400 ROUT3  LD      HL,MSG3
7D3A 1803    00410        JR      PRTRET
7D3C 21E87D  00420 ROUT4  LD      HL,MSG4
7D3F E5      00430 PRTRET PUSH    HL                  ;SAVE HL VALUES
7D40 3E0D    00440        LD      A,0DH               ;SEND A RETURN
7D42 CD3A03  00450        CALL    COUT
7D45 E1      00460        POP     HL                  ;RESTORE HL VALUES
7D46 CDA728  00470        CALL    STROUT              ;PRINT STRING
7D49 3E0D    00480        LD      A,0DH               ;SEND ANOTHER RETURN
7D4B CD3A03  00490        CALL    COUT
7D4E C39A0A  00500        JP      RETURN              ;RETURN TO BASIC
7D51 21047E  00510 ROUT5  LD      HL,MSG5             ;PRINT MESSAGE 5
7D54 CDA728  00520        CALL    STROUT
7D57 21227E  00530        LD      HL,VARBLE           ;PUT VAR. NAME ADDR IN HL
7D5A CD0D26  00540        CALL    VARPTR              ;GET DESCRIPT BLOCK IN DE
             00550 ;
             00560 ;PREPARE TO PRINT VALUE OF XX$
             00570 ;
7D5D EB      00580        EX      DE,HL               ;PLACE PTR VALUE IN HL
7D5E 46      00590        LD      B,(HL)              ;USE B REG AS CHAR CNTR
7D5F 23      00600        INC     HL                  ;PT TO LOW ADDR. BYTE
7D60 5E      00610        LD      E,(HL)              ;GET LOW ADDR. BYTE
7D61 23      00620        INC     HL
7D62 56      00630        LD      D,(HL)              ;GET HIGH ADDR. BYTE
7D63 EB      00640        EX      DE,HL               ;PT. TO STRING W/ HL
7D64 3E00    00650        LD      A,0                 ;CHECK FOR 0 LENGTH STR
7D66 B8      00660        CP      B
7D67 280B    00670        JR      Z,DUN               ;IF YES, DONE W/ PRINT
```

Program continues

✓See List of Advertisers on page 322

Program continued

```
7D69 7E        00680 LOOP    LD      A,(HL)          ;GET CHARACTER
7D6A E5        00690         PUSH    HL              ;SAVE HL PTR
7D6B C5        00700         PUSH    BC              ;SAVE B CNTR
7D6C CD3A03    00710         CALL    COUT            ;PRINT CHARACTER
7D6F C1        00720         POP     BC              ;RESTORE VALUES
7D70 E1        00730         POP     HL
7D71 23        00740         INC     HL              ;PT. TO NEXT CHAR.
7D72 10F5      00750         DJNZ    LOOP            ;LOOP UNTIL B=0
7D74 3E0D      00760 DUN     LD      A,0DH           ;SEND RETURN
7D76 CD3A03    00770         CALL    COUT
7D79 C39A0A    00780         JP      RETURN          ;RETURN TO BASIC
               00790 ;
               00800 ;THESE ARE THE CONSTANTS AND MESSAGES
               00810 ;
7D7C 2A        00820 ERRMSG  DEFM    '**** IMPROPER CODE ****'
7D93 00        00830         DEFB    0
7D94 2A        00840 MSG1    DEFM    '**** THIS IS ROUTINE 1 ****'
7DAF 00        00850         DEFB    0
7DB0 2A        00860 MSG2    DEFM    '**** THIS IS ROUTINE 2 ****'
7DCB 00        00870         DEFB    0
7DCC 2A        00880 MSG3    DEFM    '**** THIS IS ROUTINE 3 ****'
7DE7 00        00890         DEFB    0
7DE8 2A        00900 MSG4    DEFM    '**** THIS IS ROUTINE 4 ****'
7E03 00        00910         DEFB    0
7E04 0D        00920 MSG5    DEFB    0DH             ;CARRIAGE RETURN
7E05 54        00930         DEFM    'THE VARIABLE XX$ CONTAINS: '
7E20 0D        00940         DEFB    0DH
7E21 00        00950         DEFB    0
7E22 58        00960 VARBLE  DEFM    'XX$'
7E25 00        00970         DEFB    0
7E26 59        00982 VARY    DEFM    'Y%'
7E28 00        00984         DEFB    0
0000           00986         END
```

```
RUN
XX$? THIS IS A TEST
X%? 2
**** THIS IS ROUTINE 2 ****
X%? 1
**** THIS IS ROUTINE 1 ****
X%? 5
THE VARIABLE XX$ CONTAINS:
THIS IS A TEST
X%? 89
**** IMPROPER CODE ****
X%? 4
**** THIS IS ROUTINE 4 ****
X%?
```

Figure 2. This is a sample run of the Basic program of listing 2(a) with the program of listing 2(b) loaded in memory. All characters that are underlined are printed by the computer.

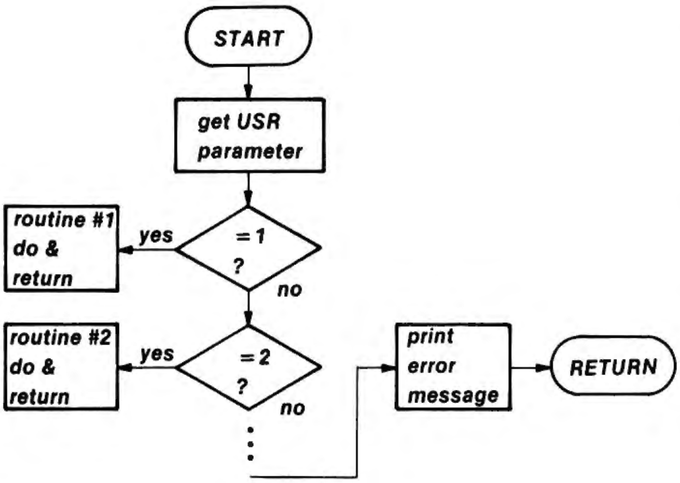


Fig. 1. Flowchart of the method for machine language routine to decide which USR routine is desired and branch to proper routine.

```
10 N=32001
20 READ A$
30 IFA$="-1"THEN END
40 B$=LEFT$(A$,1):GOSUB200
50 B1=16*B
60 B$=RIGHT$(A$,1):GOSUB200
70 B1=B1+B
80 POKE N,B1
90 N=N+1
100 GOTO 20
200 IFB$="A"THENB$="10"ELSEIFB$="B"THENB$="11"ELSEIFB$="C"THENB$
="12"ELSEIFB$="D"THENB$="13"ELSEIFB$="E"THENB$="14"ELSEIFB$="F"T
HENB$="15"
210 B=VAL(B$):RETURN
1000 DATA CD,7F,0A,7C,FE,00,20,15,7D,FE,01,28,15,FE,02
1010 DATA 28,16,FE,03,28,17,FE,04,28,18,FE,05,28,29,21,72
1020 DATA 7D,18,12,21,8A,7D,18,0D,21,A6,7D,18,08,21,C2,7D
1030 DATA 18,03,21,DE,7D,E5,3E,0D,CD,3A,03,E1,CD,A7,28,3E
1040 DATA 0D,CD,3A,03,C3,9A,0A,21,FA,7D,CD,A7,28,21,18,7E
1050 DATA CD,0D,26,EB,46,23,5E,23,56,EB,3E,00,B8,28,0B,7E
1060 DATA E5,C5,CD,3A,03,C1,E1,23,10,F5,3E,0D,CD,3A,03,C3
1070 DATA 9A,0A,2A,2A,2A,2A,20,49,4D,50,52,4F,50,45,52,20
1080 DATA 43,4F,44,45,20,2A,2A,2A,2A,00,2A,2A,2A,2A,20,54
1090 DATA 48,49,53,20,49,53,20,52,4F,55,54,49,4E,45,20,31
1100 DATA 20,2A,2A,2A,2A,00,2A,2A,2A,2A,20,54,48,49,53,20
1110 DATA 49,53,20,52,4F,55,54,49,4E,45,20,32,20,2A,2A,2A
1120 DATA 2A,00,2A,2A,2A,2A,20,54,48,49,53,20,49,53,20,52
1130 DATA 4F,55,54,49,4E,45,20,33,20,2A,2A,2A,2A,00,2A,2A
1140 DATA 2A,2A,20,54,48,49,53,20,49,53,20,52,4F,55,54,49
1150 DATA 4E,45,20,34,20,2A,2A,2A,2A,00,0D,54,48,45,20,56
1160 DATA 41,52,49,41,42,4C,45,20,58,58,24,20,43,4F,4E,54
1170 DATA 41,49,4E,53,3A,20,0D,00,58,58,24,00,-1
```

Program Listing 2(c).

Listing 1.

Another method, one that requires less code, is faster, and allows several different routines to be called from the same place in a Basic program at different times, is illustrated in the flowchart of Fig. 1. This method uses the USR parameter (the value represented in parenthesis after the USR) to decide which routine is desired. It then branches to the appropriate routine and each routine is responsible for its own return to the main program.

VARPTR Function

The question arises: What if the routines require a parameter? If this is the case, we use the TRS-80's VARPTR function which is accessible from machine language. To use the VARPTR function routine, place in memory the variable name in ASCII, followed by a zero byte; load the address of the beginning of the variable name into the HL register pair and call location 260DH. The desired address is returned in the DE register pair in the format described in the *Level II Reference Manual.*

The programs in Listings 2(a) and 2(b) demonstrate this second USR call method and also illustrate the use of the VARPTR function. In the Basic program of Listing 2(a), the address of the beginning of the machine language routine is POKEd into locations 16526–16527 to establish the address of the routine when the USR function is encountered. When the USR function is called, it is the responsibility of the machine language routine to decide which routine is desired and branch accordingly. The machine language

```
RUN
XX$? THIS IS A TEST
Y%? 3
X%? 2
**** THIS IS ROUTINE 2 ****
Y% = 3
X%? 1
Y% = 6
X%? 1
Y% = 12
X%? 4
**** THIS IS ROUTINE 4 ****
Y% = 12
X%? 5
THE VARIABLE XX$ CONTAINS:
THIS IS A TEST
Y% = 12
X%? 1
Y% = 24
X%? -3
**** IMPROPER CODE ****
Y% = 24
X%?
```

Figure 3. This is a sample run of the Basic program of listing 3(a) with the programs of listings 2(b) and 3(b) (the modification) loaded in memory. All characters that are underlined are printed by the computer.

TRS-80® DISCOUNT $ SAVE $

MODEL III

26-1061 4K I	$599.00
26-1062 16K III	859.00
26-1063 32K III W/2 Drives, RS 232	2,225.00
26-1162 1st Drive	765.00

Color Computer 4K $353

26-3001 4K	$353.00
26-3002 16K Ext. Basic	533.00
26-3009 Joysticks	22.50
26-3010 Color Video	353.00
26-1206 Recorder	54.00

Proto Board For Color Computer —
Fits inside 8-Track Cartridge **$24.95**

Model I

26-1140 Expansion Interface	$249.00
26-1141 16K Exp. Interface	359.00
26-1142 32K Exp. Interface	469.00
26-1145 RS 232C Board	84.00
26-1160/1 Mini Disk Drive	419.00
26-1563 Scripsit-Disk	79.00
26-1566 Visicalc	83.00

MODEL II

Model II 64K $3395

26-4160 1 Drive Exp	$1014.00
26-4161 2 Drive Exp	1559.00
26-4162 3 Drive Exp	2099.00
26-4530 Scripsit II	259.00
26-4512 Profile II	159.00
26-4511 Visicalc II	259.00
26-4501 Gen Ledger	179.00
26-4506 Mail List	72.00

PRINTERS

26-1167 9½ Line Printer VII	$ 360.00
26-1166 Line Printer VI	1080.00
26-1158 Daisy Wheel II	1798.00
26-1165 Line Printer V	1710.00
26-1401 Cable	36.00

EPSON MX80/MX70

Low-Priced Professional Print Quality

EPSON MX80	(List $645)	$499.00
EPSON MX70 Dot Graphics, 5x7 Matrix	(List $450)	$425.00

IDS PAPER TIGERS

Dot Resolution Graphics quality print

IDS 445G 7 wire printhead, graphics	(List $985)	$ 795.00
IDS 460G 9 wire printhead, graphics	(List $1394)	1195.00
IDS 560G 9 wire, wide carriage, graphics	(List $1794)	1529.00

All prices are subject to change without notice. Freight, handling, and insurance charges are extra. Most items are shipped United Parcel Service. Model II and other hardware weighting over 50 lbs. is shipped freight collect by truck. Certified Check for immediate shipment from stock. Master Charge, Visa or Bank Card add 3% surcharge.

No Taxes on Out Of State Shipments

Okla Include 2% State Tax

DOES YOUR SMART PRINTER SUDDENLY BECOME DUMB WHEN YOU PRINT FROM SCRIPSIT?

Is your printer capable of underlining but not from Scripsit?

Is your printer capable of **Bold Printing** but not from Scripsit?

Can your printer super script and sub script but not from Scripsit?

Can your printer change pitches but not from Scripsit?

The answer is SCRIPMOD

SCRIPMOD does not require a separate printer driver. With SCRIPMOD control codes can be embedded in the test of your document. You use the same format line syntax you're used to now. SCRIPMOD adds one format instruction to Scripsit & two control codes.

SCRIPMOD is supplied on disk with full documentation for **$39.95.**

PENCIL-FIX Save your warranty. Use PENCIL-FIX to avoid custom control key on your keyboard. Redefines the control key for EP to be the (« key. Use RS lower case mod or the EP mod without the control key. Disk based EP only . . . **$14.95**

PRINT-CENTRAL Send any control code directly to your smart printer from the BASIC command mode or from DOS. Avoid having to type such things as "LPRINT CHR$(31) just to change pitch. Use CLEAR right arrow (two key strokes) instead. Any code from 1 to 31 may be sent. Mod I 32K disk . . . **$24.95**

SPOOL-REL An in-memory print buffer that runs in Mod I 32K or 48K disk systems. Fully relocatable code and buffer. A true background spooler at an unbelievably low price . . . **$24.95**

TIGGER-GRAF Create engineering, scientific, business, or just plain fun on your IDS 440G or 460G printer. Resolution is 495 x 575. Easy BASIC programs provided for data entry and machine language module for speed. Includes setting individual points, drawing lines, shading shapes. Several graphs may be catenated along the Y-axis for larger graphs. Requires Mod I 32 or 48K 1 disk . . . **$149.95**

ALL POCKET AND COLOR COMPUTER SOFTWARE SOLD AT DISCOUNT

WRITE US FOR A FREE CATALOG

1-800-331-9128 Toll Free Order Entry

Immediate Shipment From Stock on Most Items

Vern Street Products ✓111
The Computer Store, Inc.

114 West Taft
Sapulpa, Oklahoma 74066
918-747-9333

4949 South Peoria
Tulsa, Oklahoma 74105

WE TAKE TRS-80 MODEL I TRADE-INS

TRS-80 is a registered trademark of the Tandy Corp.

routine of Program Listing 2(b) first calls location 0A7FH to get the parameter value, which is returned in the HL register pair (as described in the Level II reference manual). It then checks the most significant byte (in the H register) to make sure that it is zero; if not, an error message is printed. If the H register is zero it proceeds to check the least significant byte (in the L register) and branch to the routine corresponding to the value found. If it finds a value between one and four, inclusive, the corresponding routines merely print messages indicating which routine was called. If the value is five, the corresponding routine prints a message and the current contents of the variable XX$. If none of those values are encountered, an error message is printed and control is returned to the Basic program.

```
10 POKE 16526,1:POKE 16527,125
20 CLS
30 INPUT"XX$";XX$
40 INPUT"Y%";Y%
50 INPUT"X%";X%
60 Y=USR(X%)
70 PRINT"Y%=";Y%
80 GOTO 50
```

Program Listing 3(a).

Routine 5 is an example of the use of the VARPTR function, as described earlier. It uses the ASCII characters XX$ in memory (from the DEFM 'XX$' statement) followed by a zero byte to find the descriptor block for the variable XX$. As shown in the *Level II Reference Manual*, the address returned (in the DE register pair in this case) points to a block of three bytes that has the size and location of the string; the first byte is the length of the string, and the second and third bytes contain the location of the beginning of the string in memory.

It is important to note that even though no values are being passed back to the main program, I used a JP 0A9AH instead of the usual return. This is because the string print routine (location 28A7H in the Level II ROM) changes the value in location 40AFH from two to three, which causes a TM error upon return to Basic, unless either a two is replaced in location 40AFH or a jump to 0A9AH is made. If a jump to 0A9AH is chosen the value in the HL register pair is returned as a value, which may be garbage but does not affect the operation of the Basic program unless it uses that value. It is suggested that this method of return be used when using Level II ROM subroutines, because of the uncertainty of the effects of the routines.

```
7D23 21267E    00352 ROUT1    LD      HL,VARY          ;LOAD ADDR OF "Y%"
7D26 CD0D26    00354          CALL    VARPTR           ;FIND ADDR OF VALUE
7D29 EB        00356          EX      DE,HL            ;PLACE ADDR IN HL PR.
7D2A CB26      00358          SLA     (HL)             ;SHIFT LOW BYTE-PUT HIGH
               00360                                   ; BIT INTO CARRY
7D2C 23        00362          INC     HL               ;POINT TO HIGH BYTE
7D2D CB16      00364          RL      (HL)             ;ROTATE LEFT W/CARRY IN
7D2F C39A0A    00366          JP      RETURN           ;RETURN TO BASIC
7E26 59        00982 VARY     DEFM    'Y%'
7E28 00        00984          DEFB    0
```

Program Listing 3(b).

If any of the routines (or all of them) need one or more parameters, the user may use any variable or variables he desires (as many as are needed by each of the routines) and access those variables in machine language with the VARPTR function. Note that the user is not limited to only one parameter, or to only integer values, but he may access as many variables as is necessary and these may be any type, including integers, single precision values, double precision values, strings, and arrays. This would allow, for instance, a very fast machine language array sort. New values may be returned by either changing the values of the variables in memory or returning a value in the HL register pair with a jump to 0A9AH as described in the *Level II Reference Manual.*

Program Listings 3(a-c) show modifications to those in Listings 2(a-c), which are used to show how to access a value as a parameter and return a new value in the same variable. Listing 3(a) is a test driver and Listing 3(c) is a Basic program that loads in the program of Listing 3(b). The routine of 3(b) multiplies the value in Y% by two and returns the new value in the same variable (the multiplication is done by merely shifting the whole value one bit position to the left). See Fig. 2 for a test run of Listing 2 and Fig. 3 for test run of Listing 3.

As can be seen, there is a large amount of flexibility open to the assembly language programmer, with the combined efforts of the USR function and the Level II VARPTR function. With a little thought and a little imagination one can enhance his programs beyond the limitations of Level II Basic.■

```
10 POKE 32035,195:POKE 32036,28:POKE 32037,126
20 N=32284
30 READ A
40 IF A=-1THEN END
50 POKE N,A:N=N+1:GOTO30
100 DATA 33,43,126,205,13,38,235,203,38,35,203,22,195,154,10,89,
37,0,-1
```

Program Listing 3(c).

Get Inkey$ working.

UC/LC Fix

Dennis Thurlow
Instant Software
Peterborough, NH 03458

If you have a TRS-80 disk system with Radio Shack's lowercase modification, you have probably noticed that INKEY$ does not work properly. PEEKing at the video RAM area produces strange values when the driver is not loaded; when the driver is loaded, many Basic programs still do not work. The problem stems from the fact that, for some reason, the tables are backwards in the routine. Here is an easy way to fix this problem.

The routine should first be put onto disk as a CMD file (example: if you want to call the file LCD save it as LCD/CMD) with either Dump or Tapedisk. Both are supplied with your TRSDOS. *Do not use NEWDOS*! It will not work properly with the lowercase or driver.

Boot the system up, type Debug and hit Enter.

Call the driver by typing only the name part of the file name (Example: if you saved it as LCD/CMD, type only LCD) and hitting Enter.

The routine will now load but instead of executing, it will drop you into Debug. Your DOS manual will explain what you see on the screen, but you don't really need to know to follow these directions. Type D7000 and hit the space bar. Then type S. The screen should be identical to Fig. 1. Type ; (semicolon) to see Fig. 2. Type — (dash) to return to Fig. 1. If your screen doesn't look like this, start over.

Type M70BA and hit the space bar. This is the modify mode and brackets should appear around the byte being modified. The address and contents of the byte also appear in the lower left of the screen. Note the value here is 61 and the next 26 bytes count up the hex number system to 7A. These need to be changed so the table goes from 41 to 5A.

To change the first byte, type 41 and hit the space bar. The byte changes, the brackets move to the next byte and the display in the lower left indicates the new address and value. Continue typing in numbers sequentially (42, 43, 44...5A) and hitting the space bar until you reach 5A. Hit X to end this session.

Type M70FA and hit the space bar. Here and for the next 26 bytes, you find 41 through 5A exactly the way you just typed it. This table needs to be changed to 61 through 7A, using the same technique. (Again, type X to end the session.) Note: The screen will not move to the next page when you get to 47, but the display in the lower left is still accurate. Hit ; (semicolon) to see the next page and the changes you just made.

Type M713A and hit the space bar. The table here should be changed to 41 through 5A also. Fig. 3 shows what these two pages should look like when done.

To save this repaired driver,

```
7000 =>  2A 49 40 01 0D 02 B7 ED  42 E5 2B 22 49 40 11 CE
7010 =>  FF 19 22 A0 40 D1 21 59  70 C5 D5 E5 EB B7 ED 52
7020 =>  E5 D1 2A A0 70 19 22 A0  70 2A D7 71 19 22 D7 71
7030 =>  E1 D1 C1 ED B0 EB B7 ED  52 E5 D1 21 64 70 19 22
7040 =>  16 40 21 B9 71 19 22 1E  40 21 63 72 19 22 26 40
7050 =>  21 00 00 22 19 40 C3 2D  40 C5 01 CB 04 CD 60 00
7060 =>  C1 0A 18 37 01 01 38 21  36 40 16 00 0A 5F AE 73
7070 =>  A3 20 07 14 2C CB 01 F8  18 F2 5F 7A 17 17 17 57
7080 =>  7B 0F 38 03 14 18 FA 21  80 38 CB 46 28 02 CB F2
7090 =>  21 19 40 CB 46 28 02 CB  FA 18 BE A3 20 01 C9 21
70A0 =>  B9 70 5A 16 00 19 7E FE  80 20 09 3E 01 21 19 40
70B0 =>  AE 77 AF C9 FE 01 C0 EF  C9 40 61 62 63 64 65 66
70C0 =>  67 68 69 6A 6B 6C 6D 6E  6F 70 71 72 73 74 75 76
70D0 =>  77 78 79 7A 00 00 00 00  00 30 31 32 33 34 35 36
70E0 =>  37 38 39 3A 3B 2C 2D 2E  2F 0D 1F 01 5B 0A 08 09
70F0 =>  20 FF FF FF FF FF FF FF  00 60 41 42 43 44 45 46
7100 =>  47 48 49 4A 4B 4C 4D 4E  4F 50 51 52 53 54 55 56
7110 =>  57 58 59 5A FF FF FF FF  00 80 21 22 23 24 25 26
7120 =>  27 28 29 2A 2B 3C 3D 3E  3F 0D 1F 01 1B 1A 18 19
7130 =>  20 FF FF FF FF FF FF FF  00 40 41 42 43 44 45 46
7140 =>  47 48 49 4A 4B 4C 4D 4E  4F 50 51 52 53 54 55 56
7150 =>  57 58 59 5A 00 00 00 00  00 30 31 32 33 34 35 36
7160 =>  37 38 39 3A 3B 2C 2D 2E  2F 0D 1F 01 5B 0A 08 09
7170 =>  20 FF FF FF FF FF FF FF  00 60 41 42 43 44 45 46
7180 =>  47 48 49 4A 4B 4C 4D 4E  4F 50 51 52 53 54 55 56
7190 =>  57 58 59 5A FF FF FF FF  00 80 21 22 23 24 25 26
71A0 =>  27 28 29 2A 2B 3C 3D 3E  3F 0D 1F 01 1B 1A 18 19
71B0 =>  20 FF FF FF FF FF FF FF  00 DD 6E 03 DD 66 04 DA
71C0 =>  9A 04 DD 7E 05 B7 28 01  77 79 FE 20 DA 06 05 FE
71D0 =>  80 D2 A6 04 E5 D5 21 E3  71 5F 16 00 19 7E D1 E1
71E0 =>  C3 7D 04 20 20 20 20 20  20 20 20 20 20 20 20 20
71F0 =>  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
```

Fig. 1.

```
7000 =>  2A 49 40 01 0D 02 B7 ED  42 E5 2B 22 49 40 11 CE
7010 =>  FF 19 22 A0 40 D1 21 59  70 C5 D5 E5 EB B7 ED 52
7020 =>  E5 D1 2A A0 70 19 22 A0  70 2A D7 71 19 22 D7 71
7030 =>  E1 D1 C1 ED B0 EB B7 ED  52 E5 D1 21 64 70 19 22
7040 =>  16 40 21 B9 71 19 22 1E  40 21 63 72 19 22 26 40
7050 =>  21 00 00 22 19 40 C3 2D  40 C5 01 CB 04 CD 60 00
7060 =>  C1 0A 18 37 01 01 38 21  36 40 16 00 0A 5F AE 73
7070 =>  A3 20 07 14 2C CB 01 F8  18 F2 5F 7A 17 17 17 57
7080 =>  7B 0F 38 03 14 18 FA 21  80 38 CB 46 28 02 CB F2
7090 =>  21 19 40 CB 46 28 02 CB  FA 18 BE A3 20 01 C9 21
70A0 =>  B9 70 5A 16 00 19 7E FE  80 20 09 3E 01 21 19 40
70B0 =>  AE 77 AF C9 FE 01 C0 EF  C9 40 41 42 43 44 45 46
70C0 =>  47 48 49 4A 4B 4C 4D 4E  4F 50 51 52 53 54 55 56
70D0 =>  57 58 59 5A 00 00 00 00  00 30 31 32 33 34 35 36
70E0 =>  37 38 39 3A 3B 2C 2D 2E  2F 0D 1F 01 5B 0A 08 09
70F0 =>  20 FF FF FF FF FF FF FF  00 60 61 62 63 64 65 66
7100 =>  67 68 69 6A 6B 6C 6D 6E  6F 70 71 72 73 74 75 76
7110 =>  77 78 79 7A FF FF FF FF  00 80 21 22 23 24 25 26
7120 =>  27 28 29 2A 2B 3C 3D 3E  3F 0D 1F 01 1B 1A 18 19
7130 =>  20 FF FF FF FF FF FF FF  00 40 61 62 63 64 65 66
7140 =>  67 68 69 6A 6B 6C 6D 6E  6F 70 71 72 73 74 75 76
7150 =>  77 78 79 7A 00 00 00 00  00 30 31 32 33 34 35 36
7160 =>  37 38 39 3A 3B 2C 2D 2E  2F 0D 1F 01 5B 0A 08 09
7170 =>  20 FF FF FF FF FF FF FF  00 60 41 42 43 44 45 46
7180 =>  47 48 49 4A 4B 4C 4D 4E  4F 50 51 52 53 54 55 56
7190 =>  57 58 59 5A FF FF FF FF  00 80 21 22 23 24 25 26
71A0 =>  27 28 29 2A 2B 3C 3D 3E  3F 0D 1F 01 1B 1A 18 19
71B0 =>  20 FF FF FF FF FF FF FF  00 DD 6E 03 DD 66 04 DA
71C0 =>  9A 04 DD 7E 05 B7 28 01  77 79 FE 20 DA 06 05 FE
71D0 =>  80 D2 A6 04 E5 D5 21 E3  71 5F 16 00 19 7E D1 E1
71E0 =>  C3 7D 04 20 20 20 20 20  20 20 20 20 20 20 20 20
71F0 =>  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
```

Fig. 2.

```
7000 =>  2A 49 40 01 0D 02 B7 ED  42 E5 2B 22 49 40 11 CE
7010 =>  FF 19 22 A0 40 D1 21 59  70 C5 D5 E5 EB B7 ED 52
7020 =>  E5 D1 2A A0 70 19 22 A0  70 2A D7 71 19 22 D7 71
7030 =>  E1 D1 C1 ED B0 EB B7 ED  52 E5 D1 21 64 70 19 22
7040 =>  16 40 21 B9 71 19 22 1E  40 21 63 72 19 22 26 40
7050 =>  21 00 00 22 19 40 C3 2D  40 C5 01 CB 04 CD 60 00
7060 =>  C1 0A 18 37 01 01 38 21  36 40 16 00 0A 5F AE 73
7070 =>  A3 20 07 14 2C CB 01 F8  18 F2 5F 7A 17 17 17 57
7080 =>  7B 0F 38 03 14 18 FA 21  80 38 CB 46 28 02 CB F2
7090 =>  21 19 40 CB 46 28 02 CB  FA 18 BE A3 20 01 C9 21
70A0 =>  B9 70 5A 16 00 19 7E FE  80 20 09 3E 01 21 19 40
70B0 =>  AE 77 AF C9 FE 01 C0 EF  C9 40 41 42 43 44 45 46
70C0 =>  47 48 49 4A 4B 4C 4D 4E  4F 50 51 52 53 54 55 56
70D0 =>  57 58 59 5A 00 00 00 00  00 30 31 32 33 34 35 36
70E0 =>  37 38 39 3A 3B 2C 2D 2E  2F 0D 1F 01 5B 0A 08 09
70F0 =>  20 FF FF FF FF FF FF FF  00 60 61 62 63 64 65 66
7100 =>  67 68 69 6A 6B 6C 6D 6E  6F 70 71 72 73 74 75 76
7110 =>  77 78 79 7A FF FF FF FF  00 80 21 22 23 24 25 26
7120 =>  27 28 29 2A 2B 3C 3D 3E  3F 0D 1F 01 1B 1A 18 19
7130 =>  20 FF FF FF FF FF FF FF  00 40 61 62 63 64 65 66
7140 =>  67 68 69 6A 6B 6C 6D 6E  6F 70 71 72 73 74 75 76
7150 =>  77 78 79 7A 00 00 00 00  00 30 31 32 33 34 35 36
7160 =>  37 38 39 3A 3B 2C 2D 2E  2F 0D 1F 01 5B 0A 08 09
7170 =>  20 FF FF FF FF FF FF FF  00 60 41 42 43 44 45 46
7180 =>  47 48 49 4A 4B 4C 4D 4E  4F 50 51 52 53 54 55 56
7190 =>  57 58 59 5A FF FF FF FF  00 80 21 22 23 24 25 26
71A0 =>  27 28 29 2A 2B 3C 3D 3E  3F 0D 1F 01 1B 1A 18 19
71B0 =>  20 FF FF FF FF FF FF FF  00 DD 6E 03 DD 66 04 DA
71C0 =>  9A 04 DD 7E 05 B7 28 01  77 79 FE 20 DA 06 05 FE
71D0 =>  80 D2 A6 04 E5 D5 21 E3  71 5F 16 00 19 7E D1 E1
71E0 =>  C3 7D 04 20 20 20 20 20  20 20 20 20 20 20 20 20
71F0 =>  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
```

Fig. 3

type G402D and hit Enter. When DOS is ready, use Dump to save the driver as a CMD file. Start is 700, End is 7270. TRA is 700. Don't forget to specify the drive. For example:

DUMP LCD/CMD:0 (START = X'7000', END = X'7270', TRA = X'7000)

Use this driver the same way you used the old one. The only difference is that this one works right! ■

✓See List of Advertisers on page 322

For the day when you'll want to communicate.

Datacom Decisions

Frank J. Derfler Jr.
P.O. Box 691
Herndon, VA 22070

Certainly you can make a TRS-80 into a sophisticated and flexible system, capable of exchanging multiple files with the fastest IBM mainframes, but do you need to pay for all of that capability?

Someday you may want to exchange programs with a friend or business associate across town or across the nation, or perhaps access an electronic message system such as a Forum 80 or a computer bulletin board. You may want to use a time-shared computer utility or a multipurpose utility, such as the Source. For all of the above your 80 will need to communicate by telephone.

There are a bewildering number of paths to follow. I'll try to describe some common ways to make your TRS-80 Model I into a data communications device.

You have to start with a statement of need. What is it you need to do? Do you want to only act as a remote off of a large computer system? Do you want to engage in a heavy electronic mail flow on several different systems? Do you need to save what you send and receive locally on your TRS-80, or are you interested only in reading and leaving information on some other system? Do you want to exchange programs by telephone with other TRS-80 users and with electronic message systems and program exchanges?

After we sketch what you want to do, we need to know what you already can do. Do you have a disk system? Do you have an expansion interface and RS232 card? How much system power do you have available? The most basic 4K Level I machine can be a very effective terminal.

Next we have to know what hardware and software is on the market and what it costs. My list will discuss some of the most popular hardware and software.

Communications are usually done over a telephone line. If you're a ham, you might also use an amateur radio rig. Telephone and radio systems only pass sound, so your computer's signals have to be converted to sound waves. This is done by a device called a modem (modulator/demodulator). Perhaps the most popular modem is the CAT by Novation. Radio Shack sold this modem under its own label for several years. Their newest device is called Modem I and it is a completely Radio Shack designed and manufactured product. The modem connects between the computer and phone line and exchanges dc voltage signals with the computer.

The most common form is the RS232 ASCII signal coming from a serial port. In the TRS-80 Model I, the circuit board that makes RS232 signals is an option that plugs into the expansion interface. The Radio Shack method of communications means you must buy an expansion interface, RS232 card and modem.

The Modem I does have the capability to operate with the Model I keyboard computer without the expansion interface. A special cable and software is required. The only mode of operation available is one in which your transmissions are not echoed back from the distant end system (half-duplex); however, in many applications this is not a problem.

All of this hardware must be used with the proper software. Radio Shack sells a communications program package called Videotex which allows your system to act like a terminal. It will send and receive a program or

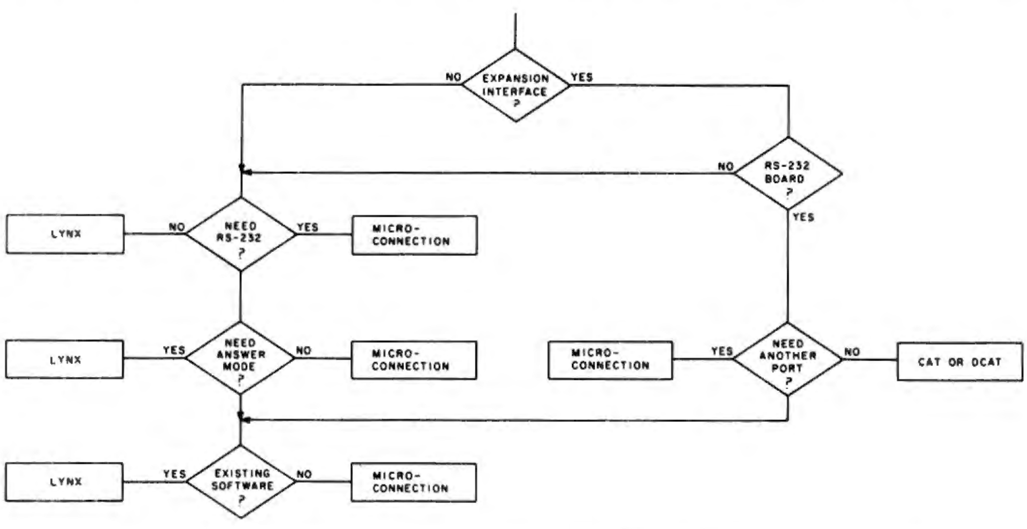


Fig. 1. This flowchart will help you decide on the best modem to suit your needs.

data and display what it receives on the screen. This communications program sells for about $30. Including the software, it costs about $330 to give your TRS-80 an elementary communications capability in the standard configuration, *if* you already have the expansion interface. There are other ways to do it.

Different Strokes

Two manufacturers make TRS-80 modem devices that don't require the RS232 board or the expansion interface. Both the Microconnection and the Lynx modems plug directly into the TRS-80 keyboard and both are FCC accepted so they can attach directly to telephone lines. There are some important differences between the devices, so you should consider your needs and available equipment. Here are some other differences and similarities:

- The Microconnection has an extra output to give RS232C signals to drive a printer or other RS232C devices. The Microconnection provides a port separate from the standard TRS-80 RS232C output but it requires special software to do it. The Lynx cannot be used if the standard RS232 card is in place because it decodes the same signals. The Lynx has no RS232C output.
- The Lynx can use any software communications package written for the TRS-80 in its standard communications configuration. Because the Microconnection uses a different address, it requires customized software.
- Both devices plug into either the keyboard or expansion interface.
- The Lynx comes with a very adequate communications program on tape. The Microconnection provides a listing of an elementary program. Communications programs are available at prices starting about $20.
- Both devices work with any memory size or language level of the TRS-80 Model I.
- The Lynx sells for about $280. The Microconnection sells for about $250, but it does need communications software.

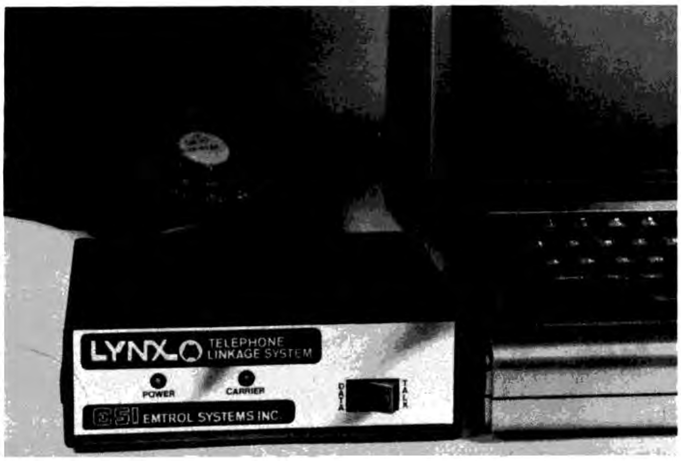


Photo 1. The Lynx modem features both originate and answer signalling and operates with standard TRS-80 communications software.

Photo 2. This is the DCAT. It is the direct connection version of the popular Cat modem. Direct connection means the modem plugs into the phone jack and does not use acoustical coupling. The Cat and DCAT require an RS-232 interface to the computer.

Now let's talk about software. As I pointed out, the Lynx comes with an adequate communications program that is ready to load and run. Slightly different programs are provided for Level I and Level II machines. The Level II program will receive and transmit Basic programs and messages and serve as a reasonably flexible terminal. This would be adequate for most bulletin board and hobby system users.

TCP

The Bottom Shelf is a company known to many TRS-80 users. Among their many program packages they have a Terminal Control Program. TCP is a utility package for TRS-80s equipped with an RS232 board or Lynx. It will send and receive Basic programs, data and memory blocks. The RS232 parameters, as well as the video and printer output, may be controlled directly from the keyboard.

TCP has several different operating options. One option allows the TRS-80 to function as a dumb terminal without memory. A second menu selection allows you to transmit and receive Basic programs in the compressed format used internally by the TRS-80. Another very flexible option allows the computer to transmit any designated block of memory within the RAM. This allows you to dump the contents of any file you can load into a computer out the RS232 port.

TCP can be entered and left easily, allowing you to use other utility programs, the operating system, or Basic programs while still retaining TCP in memory. This program sells for about $20 on tape and $30 on disk.

ST80

The ST80 series is made up of three programs for the Model I and one program for the Model II. The programs differ in the number of features they give your Tandy terminal and, of course, the cost.

The most elementary program is called the ST80-UC. It makes your TRS-80 think it is a terminal with most of the standard terminal features. It will run on a Level II system with 4K or more memory.

A TRS-80 running under ST80-UC has full cursor control and scrolling. It transmits control codes, escape, left and right brackets, repeat, rub out and an extended null break. The video driver program is compatible with all popular lowercase modifications. The program also will drive the TBEEP 2 audio device manufactured by Web Associates. This provides the bell function found on video terminals.

The ST80-UC is available in versions to run with the TRS-80 serial board, the Lynx direct connection modem and the Microconnection. The price is $25.

ST80-III is a full capability smart terminal program with professional features. You can customize your TRS-80 to serve as a terminal for practically any computer system. Since the customizing is all done in software, you need only change disks to be perfectly meshed with the different host systems you may use.

ST80-III is available in versions for both the Model I and Model II TRS-80s. The Model I version costs $150 and Model II costs $200. The most current version should be 2.2. A version for Model II CP/M should be available by the time this is published for $300.

There is another program between the $25 UC and the $150 ST80-III. The middle program is called the ST80-D which is sometimes marketed as the ST80-II. The D version does not have some of the ST80-III's security, special transmission, and large system transmission features, but it is adequate for most hobby uses. ST80-D is available only for the Model I TRS-80 with disk. It costs $80.

ST80 programs are available from Small Business Systems Group, Westford, MA.

From ISI

There is one terminal program which I personally consider to be "the best" disk-based program available for Model I (and Model III) systems. It is called OMNITERM and is written by David Lindbergh. OMNITERM is available from the Programmers Guild and under the name of SUPERTERM from Instant Software for $99. OMNITERM is a powerful but easy to use program. It is menu driven, so the user doesn't have to remember any commands or key sequences. The instruction manual is very complete and includes an extensive index and glossary. Included in the program is a complete customization and translation capability so you can make your computer interface with almost any ASCII and EBCDIC speaking system. Files can be easily transmitted or saved, electronic mail can be read and re-read using a unique ability to scroll through the buffer while staying on-line, and many helpful operating features are provided. OMNITERM is a great value at the price.

Decisions, Decisions

If you want more information on the software and equipment I mentioned here, you can find it in the *Microcomputing* feature, "Dial-up Directory." The Lynx was reviewed in depth in the January, 1981 issue. The ST80 series of software was described more fully in the December, 1980 issue. We took a closer look at the Microconnection in September, 1980 and reviewed TCP in August, 1980.

One of the prime appeals of the TRS-80 is its versatility. A system can be configured to meet whatever your data communications needs might be. Research the options to find the system that serves you best. ■

THE PATCH

WE'VE MOVED!

Now we can offer faster and more reliable service -- Including UPS upon request!

DUALCASE*

UPPER/lowercase, full time from power-up; **NO** software; Standard typewriter keyboard operation (shift to UPPERCASE); Control characters can be displayed; 128 Total character set plus full graphics.

SWITCHABLE*

No switches. No driver. Operates from keyboard. Totally disables "PATCH" functions.

KEYBOARD DEBOUNCE*

Extra keyboard debounce, full time from power-up; **NO** software. If dirty keys are a problem, this is for you.

BLOCK CURSOR*

Replaces the underline style cursor directly. Easier to locate on a full screen. No distracting blinking.

SHORT CASSETTE LEADER*

For tape based systems. Does **NOT** change baud rate. Only shortens recorded leader. Saves four seconds of waiting time. Great for data files! No conflict with high baud rate cassette systems.

ELECTRONIC SHIFT-LOCK*

No extra keys or switches. Simply tap either shift key, UPPERCASE lock, normal shift unlocks.

"THE PATCH" is not only a lowercase modification, but a complete system offering many of the enhancements you would like available on your TRS-80. How is this possible? "THE PATCH" is a "smart chip" which monitors the same address lines as the ROM in your TRS-80 Model I Level II. "THE PATCH" recognizes certain addresses, turns off the LII ROM, then substitutes its' own byte of data. This "Patching" is what allows ALL these enhancements with **NO** software overhead, thus **NO** software conflict. In fact "THE PATCH" is the ONLY lowercase modification that is compatible with both Basic and machine language programs AND the ONLY all hardware lowercase modification.

You install "THE PATCH", no need to send your TRS-80 away so there's no hidden installation cost. Detailed instructions guide even the most inexperienced owner to complete installation in about 30 minutes. Works with the two or three chip ROM set. "THE PATCH" unit plugs into the ROM sockets (does not replace existing ROM). Lowercase does require installation of the extra video RAM (supplied with "THE PATCH").

"THE PATCH" is compatible with any word processor, any DOS and also other languages which use ROM sub-routines. "THE PATCH" is never obsolete. As new features become available they can be installed in your "PATCH".

"THE PATCH" is covered with a one year limited warranty on materials and workmanship. (Does void Radio Shack's 90 day warranty). Ask for details . . . we can install for a small fee.

✓62

CALL NOW (208) 772-9571

CECDAT, INC.
P. O. Box 497
Hayden Lake, ID 83835

TRS-80 is a Trademark of Tandy Corp. — ORDER INFORMATION — "THE PATCH" is a trademark of CECDAT, INC.

Name ....
Street ....
City .... State .... ZIP ....
Signature ....
☐ Check, Money Order, Bank Draft
☐ VISA, MASTERCHARGE, Purchase Orders (add 3%)
Card/PO No. ....
Expiration Date .... Today's Date ....
You must check one:
☐ "MEM SIZE" ☐ "MEMORY SIZE"
Price valid through December 31, 1981

"THE PATCH 2" Switchable @ $94.97 ea. ....

OPTIONS: (please check)
☐ "Electronic Shift-Lock" Add $30.00 ....
☐ "Short Cassette Leader" Add $10.00 ....
☐ "Block Cursor" No Charge
☐ "Debounce" No Charge
Ship. & Hand. @ 2.50 ea. ....
COD ADD 5% . . . .
ID Sales Tax 3% (Id Res) ....
TOTAL ORDER . . . .

✓See List of Advertisers on page 322

Keep track of those directories.

Disk Index

Robert A. Schilling
725 SE Vance Circle
Palm Bay, FL 32905

Have you ever wanted to demonstrate your TRS-80 to an admiring audience, only to spend the next 15 minutes rummaging through disk directories in search of those great game programs? If it's too much trouble to routinely update your disk labels to reflect the current contents, then read on. This program will make it much easier to keep an up-to-date list of each disk directory.

One drawback of TRSDOS is its inability to print or display a disk directory from DOS while in Basic. If you have a one-drive system the problem is aggravated by the system overlays required on each disk, leaving about 50 grans of space for your programs at best. As a result, the number of disks you have to keep track of grows constantly.

Program Listing 1 allows me to print or display all my disk directories, neatly formatted to use a minimum amount of paper.

Program Use

The program first prompts to find out if you want system and invisible files included in the output. Since I have minimum system information on every disk, I rarely use this option. It then asks for the drive number to be used for this listing session: in order to change drive selection it is necessary to rerun the program. It then asks if you want a hard copy of the listings; don't answer yes if you have no printer. The program then asks for a label for each disk, followed by a prompt to insert the disk into the proper drive and hit (enter). Since I have systems information on all disks, the disk name stored in the directory bears no relationship to the disk contents. For those of you who use descriptive names on formatted disks, the remarks section at the end of the program gives you two lines to be added to allow these disk names to be printed as well.

Subroutine 5000 POKEs a small machine language routine into high RAM that will read disk sectors, regardless of their protection status (such as the directory on track 17), into a 256-byte RAM buffer just below the program itself. This version is designed for a 16K system, however, the machine language program is relocatable if the ninth and tenth elements of the data statement are changed to reflect the desired start address of the 256-byte buffer, along with the PEEK statements in lines 110 and 120 that look through this buffer, and the

```
10 CLS:CLEAR200:DEFINTA-C,F,I,J:DEFSTRR,W
20 PRINT"BASIC DISK INDEX LABEL READER"
30 INPUT"DO YOU WANT SYSTEM AND INVISIBLE FILES PRINTED ";W
40 IFLEFT$(W,1)="Y"THENFL=1ELSEFL=0
50 GOSUB5000:DEFUSR0=&H7FC0
55 INPUT"WHICH DRIVE IS TO BE USED (0-3) ";JD:IFJD<0ORJD>3THEN55

57 POKE32706,JD
60 INPUT"DO YOU WANT HARDCOPY ";R:IFLEFT$(R,1)="Y"THENF1=1ELSEF1
=0
70 INPUT"DISK NAME ";W
80 PRINT"PUT DISK CONTAINING ";W;" IN DRIVE ";JD;" AND HIT ENTER
 ";:INPUTR:PRINT
90 IFF1=1THENLPRINTWELSEPRINTW:IFF1=1THENLPRINTELSEPRINT
100 R="* ":CT=0
105 FORJS=2TO9:POKE32709,JS:B=USR0(1)
110 FORJL=0TO7:IFPEEK(32448+JL*32)=0THEN162
120 IFFL=0ANDPEEK(32448+JL*32)<>16THEN162
130 CT=CT+1:FORIN=5TO12:R=R+CHR$(PEEK(32448+JL*32+IN)):NEXTIN
140 R=R+"/":FORIN=13TO15:R=R+CHR$(PEEK(32448+JL*32+IN)):NEXTIN
150 R=R+" * ":IFCT<4ANDJL<8THEN162
160 GOSUB4000:CT=0:R="* "
162 NEXTJL
164 NEXTJS
166 IFR="* "THENR=""
170 GOSUB4000:CT=0:R="":GOSUB4000:GOTO70
4000 IFF1=1THENLPRINTRELSEPRINTR:RETURN
4010 RETURN
5000 FORD=32704TO32718:READE:POKED,E:NEXT:RETURN
5010 DATA243,1,0,0,17,0,17,33,192,126,205,221,70,251,201
6000 ;TO INCLUDE THE DISK LABEL IN THE OUTPUT, ADD THE
6010 ;FOLLOWING TWO LINES
6020 ;82 POKE32709,0:B=USR0(1)
6030 ;83 W=W+"    ":FORJL=32656TO32663:W=W+CHR$(PEEK(JL)):NEXT
6040 ;TO INCLUDE FREE GRANS ADD THE FOLLOWING
6050 ;84 FR=0
6060 ;85 FORJL=32448TO32489
6070 ;86 IFPEEK(JL)=252THENFR=FR+2
6080 ;87 IFPEEK(JL)=254ORPEEK(JL)=253THENFR=FR+1
6090 ;88 NEXT
6100 ;89 W=W+"    "+STR$(FR)+" FREE GRANS"
```

Program Listing

POKE statements in lines 57, 105 and 5000 that put the program in place and modify it while running. To protect this area you should answer the Memory Size question with 32447 upon power up.

The machine language routine was developed after picking through TRSDOS with my disassembler. The DOS routine at 46DDH, when called with B = byte count (00 = 256), C = drive #, D = track # (11H for directory), E = sector, and HL = starting RAM address will load the specified sector into the buffer addressed by HL, regardless of its protection status.

The For...Next loop in line 105 loads each sector into the RAM buffer in turn. Only sectors two through nine contain file labels, sector zero holds the disk name and format data as a 16-character string starting with the 208th character from the start of the sector.

Each sector of the directory is formatted as eight 32-byte blocks that can contain information on one program located on the disk. If the first byte of a particular block is zero that block contains no valid information (line 110). For all files other than system files, this first byte seems to be 10 hex. System and invisible files have some value other than 10 hex in this location. Line 120 checks for this depending on the option selected at the beginning of the program.

For all blocks containing valid file information, the file name and extension is an 11-byte string starting with the fifth byte of the block. Lines 130 and 140 PEEK this information into a string R. Lines 150 to 170 format the output into print lines of four labels each and print or display them as desired.

For those with multi-drive systems this routine can easily be modified to produce a disk file of all directories that could be searched by disk name or program name, and that could update itself in a few minutes.■

Organize your software by the disk.

Dirprog

Jack Egbert
801 Welch Road
Palo Alto, CA 94304

DIRPROG is a disk directory program that provides a ready and flexible directory of computer programs that are spread over many disks and are frequently modified.

The basic requirement for best utilization of this program is a TRS-80 with 48K memory, one or two disk drives, NEWDOS and GSF sort. It can accommodate up to 950 separate programs distributed over 60 disks before it becomes limited by CPU memory. To handle even more files, the program could simply call up additional directories of equal size as required.

The program employs the DIR (DOS directory command) and scans the video memory for the information presented. It stores this data in string arrays for sorting and recall. Disks can be entered quickly, and when loaded into the disk drive, a record number and disk title are assigned to each.

Printout or display of the directory is presented as a tabulation of all disks in numerical sequence with the programs listed alphabetically and spacing provided between each grouping. Or, program titles can be listed in alphabetical sequence with the disk identifier shown after each title. If a disk is formatted or just has a DOS with no visible programs, the disk name will be listed with a "#" preceding the title.

Disk program data can be entered and deleted; a title can be searched for without knowing where it is or even how to spell the entire name; and the entire directory may be sorted by several parameters within 10 seconds for the maximum directory size. Sorting employs a machine language subroutine sold by Racet that is called GSF which, by comparison, seems to eclipse all other sorting subroutines.

This program has proven to be extremely helpful among our TRS-80 users group, so I decided to make it widely available through publication.

Instructions For Using DIRPROG (A Disk Directory Program)

Before this Basic program can be used, it is necessary to first load in the machine language "GSF—Sort Program." To do this, perform the following procedure:

1. From NEWDOS, "LOAD GSF48/OBJ"
2. Then load "BASIC 1,62700" (This protects high memory)
3. Next, RUN"DIRPROG"

Entering Disk Data

If you want to number the disks, the program will sort on the first four places. Therefore, start with 01 or 01 A rather than 1 because in sorting, 2 would follow 19 whereas 02 would put it where you expect it to be.

In this same reply, the string response can be 255 characters long so a completely descriptive title can be used for each disk heading, e.g. 33 B (tab)NEWDOS (tab)(tab) GENERAL UTILITY PROGRAMS.

After adding disks to the directory or editing by deletion of disk numbers, you should then sort so that storage of the file will not include the deleted records.

If you elect to modify DIRPROG for your own purposes, be advised that the DIR command requires about 8,300 bytes of free memory to execute. Should your free memory be less than this, you will get an error message, Out of Memory IN 380. You must then either reduce the length of the program or take away from the string space that has been reserved in line 80. With NEWDOS 80 this limitation does not apply. ■

Program Listing.

```
20 CLS:PRINT,"####   D I R P R O G   ####":PRINT
25 PRINT"BY BRYAN DEVENDORF -- 9/15/79"
30 PRINT"REVISED BY WILLIAM RAMSEY AND JACK EGBERT":PRINT
35 PRINT"####   A PROGRAM TO INDEX YOUR DISK FILES   ####"
40 PRINT"        ==================================":PRINT
45 PRINT:PRINT"TO RETRIEVE EXISTING FILES -- PRESS  ( R )":PRINT

50 PRINT"TO BUILD A NEW FILE  --  PRESS     ( B )",
55 DEFUSR=&HFE80:CLEAR:POKE-1,0
60 POKE18007,26
62 A$="":C$="":DN$="":E$="":I=0:J=0:L$="":L5$="":MP=0:MS$="":N=0
:PN$="":Q=0:SP=0:SW$="":TI$="":TP$="":X=0:XX=0:Z=0
65 CLEAR 16600:DIM PN$(952),DN$(952)
70 DEFINT J,X
75 A$=INKEY$:IFA$=""THEN75
80 IF A$="R" CLEAR 16600:DIM PN$(950),DN$(950):MP=-1:DEFINT J,X:
GOTO 100
85 CLS:PRINT@330,"PLEASE WAIT WHILE I INITIALIZE ARRAYS . . . ."

90 PRINT,"( ABOUT  15  SECONDS )"
95 FORX=0TO950:PN$(X)="          "+STR$(X+100): NEXT X:XX=1
100 CLS:PRINT:PRINT:PRINT,"     MENU":PRINT : MS$=""
110 PRINT,"1 -- ENTER DISKS"
120 PRINT,"2 -- SORT AND LIST"
130 PRINT,"3 -- SAVE DATA ON DISK"
140 PRINT,"4 -- GET DATA FROM DISK"
150 PRINT,"5 -- SEARCH FOR A TITLE"
160 PRINT,"6 -- DELETE A DISK"
170 PRINT,"7 -- LIST DISK TITLES ONLY"
180 PRINT,"8 -- REVIEW DIRECTORY LISTINGS FOR A DISK"
190 PRINT:PRINT"INPUT YOUR SELECTION . . . . .";
200 MS$=INKEY$:IFMS$=""THEN200
210 MS=VAL(MS$):IF MS<1 OR MS>8 THEN100
220 ON MS GOTO 230,980,710,800,600,500,1350,1400
230 CLS:PRINT:PRINTTAB(10)" INSERT DISK":PRINT
240 PRINT"ENTER FILE NO. AND NAME OF DISK . . . .";:INPUT DN$(XX
)
250 CMD"DIR :1"
260 DN$=LEFT$(DN$(XX),4)+" Z":PN$(XX)="!"
270 IF MP <> -1 THEN XX=XX+1:GOTO 290
280 XX=XX+1:DN$(XX)=DN$
290 SP=15488 :Z=0 :PN$=""
300 GOSUB 390
```

Program continues

```
310 IF PEEK(15488)=32 THEN DN$(XX)=DN$:PN$(XX)="#"+MID$(DN$(XX-1
),7,12):XX=XX+1:GOTO 350
320 IF Z=-1 THEN GOTO 350
330 IF Z/3 = INT(Z/3) THEN SP=SP+24 ELSE SP=SP+20
340 GOTO 300
350 PRINT:PRINT"TO ENTER ANOTHER DISK , HIT 'ENTER' . . . ."
360 PRINT:PRINT"TO RETURN TO MENU , ENTER  '1'"
370 SW$=INKEY$:IFSW$=""THEN370
380  IF SW$="1" THEN GOTO 100 ELSE GOTO 230
390 IF CHR$(PEEK(SP))=" " THEN Z=-1 :RETURN
400 IF MP=-1 GOTO 470
410 PN=256*(PEEK(VARPTR(PN$(XX))+2))+PEEK(VARPTR(PN$(XX))+1)
420 IF PN>32767 THEN PN=PN-65536
430 FOR X=0 TO 11
440 POKE PN+X,PEEK(SP+X)
450 NEXT
460 DN$(XX)=DN$:GOTO 490

470 FOR X=0TO11:PN$=PN$+CHR$(PEEK(SP+X)):IF CHR$(PEEK(SP+X))=" "
 THEN X=11
480 NEXT X:DN$(XX)=DN$:PN$(XX)=PN$:PN$=""
490 XX=XX+1:Z=Z+1:RETURN
500 CLS:PRINT    '         ####    TO DELETE A DISK    ####
510 INPUT"WHAT IS THE NO. OF THE DISK YOU WANT TO DELETE . . . .
";DN$
520 CLS:PRINT@340,"WORKING . . . ."
530 FOR J=1 TO XX-1
540 IF LEFT$(DN$(J),4)=LEFT$(DN$,4) THEN DN$(J)="ZZ":PN$(J)="ZZ"

550 NEXT J
560 CLS:PRINT@330,"DISK NO. -- ";DN$;" -- IS DELETED"
570 PRINT:LINEINPUT"HIT 'ENTER' TO RETURN TO MENU . . . .";A$
580 GOTO 100
600 CLS:PRINT"WHICH TITLE WOULD YOU LIKE TO SEE . . . "
610 INPUT"( TYPE TITLE THEN PRESS 'ENTER' )  ";TI$
620 CLS:PRINT@335,"SEARCHING FOR :  ";TI$:PRINT:PRINT
630 PRINT"TITLE";TAB(20)"DISK":PRINTSTRING$(34,"=")
640 LPRINT"TITLE";TAB(20)"DISK":LPRINTSTRING$(34,"=")
650 FOR J=1 TO XX-1
660 Q=INSTR(PN$(J),TI$) : IF Q=0 THEN 690
670 IF TI$=MID$(PN$(J),Q,LEN(TI$)) THEN PRINT PN$(J);TAB(20)LEFT
$(DN$(J),5)
680 LPRINT PN$(J);TAB(20)LEFT$(DN$(J),5)
690 NEXT J:PRINT:PRINT
700 LINEINPUT"THAT IS ALL !!!  ---  HIT 'ENTER' FOR MENU . . . .
";C$:GOTO 100
710 CLS:PRINT:PRINT"MAKE PREPARATIONS FOR OUTPUT TO DISK . . . .


        PRESS 'A' FOR COPYING DISK NO.'S  1  TO  60
                        OR
        PRESS 'B' FOR  NO.'S  61  ON UP":PRINT:PRINT:INPUT". . .
 . ";E$
715 CLS:PRINT@340,"COPYING . . . ."
720 IF E$="A" GOTO 730 ELSE 735
730 OPEN"O",1,"DISKINDX":GOTO 740
735 OPEN"O",1,"DISKINDX/PT2"
740 PRINT #1,XX
750 FORX=1 TO XX
760 PRINT#1,DN$(X);",";PN$(X)
770 NEXT X
780 CLOSE:CLS:PRINT@335,"TRANSFER TO DISK COMPLETED":PRINT
790 LINEINPUT"PRESS 'ENTER' TO RETURN TO MENU . . . .";E$:GOTO 1
00
800 CLS:PRINT:PRINT"MAKE PREPARATIONS FOR INPUT FROM DISK . . .
.

        PRESS  'A'  FOR DISKS NO. 1 TO 60
                       OR
        PRESS  'B'  FOR NO. 61 ON UP":PRINT:PRINT:INPUT". . . .
";E$
805 CLS:PRINT@340,"INPUTTING . . . ."
810 IF E$="A" GOTO 820 ELSE 825
820 OPEN"I",1,"DISKINDX":GOTO 830
825 OPEN"I",1,"DISKINDX/PT2"
830 INPUT #1,XX
840 FOR X=1 TO XX
850 INPUT#1,DN$(X),PN$(X)
860 IF VAL(PN$(X)) > 99 PN$(X)="   "+PN$(X)
870 NEXT X:CLOSE
880 LINEINPUT"PRESS ENTER TO CONTINUE";E$:GOTO100
890 SP$="+DN$,+PN$"
900 GOTO 920
910 SP$="+PN$,+DN$
920 CLS:PRINT@335,"SORTING ---- ONE MOMENT PLEASE"
930 I=USR(17) OR USR(VARPTR(SP$)) OR USR(0) OR USR(XX-1)
940 IF I <> 0 THEN PRINT"SORT ERROR":GOTO 940
950 FOR X=1TO XX-1
960 IF DN$(X)="ZZ" THEN XX=X:RETURN
970 NEXT X :RETURN
980 CLS:PRINT@330,"PRINT OUT SHALL BE BY :":PRINT
990 PRINTTAB(10)"( 1 )   DISK NO."
1000 PRINTTAB(10)"( 2 )   PROGRAM TITLE"
1010 PRINT:PRINT"PLEASE MAKE YOUR SELECTION . . . . ";
1020 PS$=INKEY$:IFPS$=""THEN1020
1030 PS=VAL(PS$)
1040 ON PS GOSUB 890,9X+1),PN$(X+2):LPRINTPN$(X+1),PN$(X+2):X=X+
2:GOTO 1210
1200 IF DN$(X+1)=DN$(X) PRINT PN$(X+1):LPRINT PN$(X+1):X=X+1
1210 TP$=DN$(X):NEXT
1220 PRINT:LINEINPUT"PRESS 'ENTER' TO CONTINUE";A$:GOTO 100
1230 CLS:LPRINT CHR$(30):N=0
1240 FOR X=1 TO XX-1
1250 L$=LEFT$(PN$(X),1):IF PS=1 GOTO 1270
1260 IF L$=" " OR L$="!" GOTO 1340
1270 IF L$=" " OR L$="!" PRINT" ";ELSE PRINTPN$(X);TAB(13)STRING
$(6,"-");"  ";LEFT$(DN$(X),5);
1280 L5$=LEFT$(PN$(X+54),1):IF L5$=" " OR L5$="!" PRINT" ":GOTO
 1300
1290 IF X+54 < XX PRINTTAB(36)PN$(X+54);TAB(49)STRING$(6,"-");"
 ";LEFT$(DN$(X+54),5) ELSE PRINT" "
1300 IF L$=" " OR L$="!" LPRINT" ";ELSE LPRINTPN$(X);TAB(13)STRI
NG$(6,"-");"  ";LEFT$(D:PRINT
1410 FOR J=1 TO XX-1
1420 IF LEFT$(DN$(J),4)=DR$ AND PN$(J)<>"!" THENPRINT PN$(J),
1430 NEXT:PRINT:GOTO 700
```

Special Grand Opening values on complete TRS-80™ Model III systems and disk drives, now through Sept. 15, 1981!

A.M. Electronics, Inc. has created the most powerful and reliable TRS-80™ hardware and software systems available to get *your* system up and running. Take advantage of special Grand Opening values like these:

New TRS-80™ Model III now with disk drives

Model III with dual 80-track disk drives—704K of reliable disk storage on only two drives. System features two double-density 80-track drives, 48K RAM, A.M. Electronics controller board. Complete, ready-to-run! **$2,295**

Model III with dual 40-track double-density disk drives, 48K RAM and A.M. Electronics controller board. Complete, ready-to-run! **$1,895**

DOS-PLUS operating system for TRS-80 Model III. **$69.95**

New! MAKE 80© now available for Model III

Converts Model I single-density 40-track or Model III double-density 40-track standard diskettes into Model III double-density 80-track-readable diskettes. **$29.95**

New Teac disk drives, complete with power supply, case and one-year warranty

Teac 40-track disk drive. **$265**
Teac 80-track disk drive with 10ms access time. **$375**
Teac 50F dual-headed 80-track disk drive with 10ms access time. **$499**

Disk drive cases & power supplies

5½-inch single drive unit with case and power supply. **$85**
5½-inch dual drive unit with case and power supply. **$120**
8-inch single drive unit with case and power supply. **$150**

More Grand Opening Specials!

16K 200 nanosecond RAM memory chip. **$2.00** each
Disk drive extender cable. **$8.95** each

Free Disk Drive Guide

Get your concise and fact-filled guide from A.M. Electronics, written in engaging Question/Answer style. Call or write us to order your Guide today! ✓452

Also, we specialize in complete, powerful software packages for business. Put your TRS-80™ to work for you!

Dealer/Service Center Record Package

This package, built in several modular packages, can be used separately, or in combination as a complete system. The modular units are self-contained. When combined, they will make the most complete and comprehensive service center bookkeeping/record-keeping package on the market.

A complete history of customer repair files is maintained on-line. Customer files can be accessed by account number or customer name. The information file includes customer name, address and telephone number, make of unit, service dates, name of servicing technician, services performed, and total billed. This information can be called to the screen or printer as a total list, or by the last service date.

The customer records module can post to accounts receivable module, payroll module and our *Inventory Control System*.

A Model I computer, with four 80-track drives, or a Model III computer, with two 80-track drives will hold approximately 1,800 customers and records. By adding additional drives to the Model III, more than twice this amount of information can be maintained. By switching diskettes in drives, the capacity is unlimited.

Versions of *Dealer Service Center Record Package* are available for Auto Dealers, Mechanics, TV Repair Centers, Electronic Technicians and Appliance Repair Centers. **$199.95**

Order Entry (With Invoicing & Inventory)

Orders are composed of name and address information (with optional company name), method of shipment, method of payment, items on order, quantities, prices, description, and stock number. The system also prints invoices.

At point of shipment, inventory is automatically updated and dollar sales recorded. Shipping charges are logged and shipping date recorded. All entries may be edited and partial shipping is permitted. The inventory module allows checking and updating stock, inventory reports, sales reports, shipping charges, etc.

Order Entry is a complete order entry system, designed for almost any retail and/or wholesale business. Its capacity is upward-expandable from a dual 40-track Model I system: 1,750 inventory items, 385 orders per diskette. **$199.95**

See your nearest A.M. Electronics, Inc. dealer

Here is a partial listing of authorized A.M. Electronics, Inc. dealers:

ARIZONA

Simutek Computer Products
PO Box 13687
Tucson, Arizona 85710

Microcomputers, Inc.
4322 E Fairmont
Phoenix, Arizona 85018

CALIFORNIA

RAC Products
3200 Knightswood Way
San Jose, California 95148

CONNECTICUT

Mountain View Software
Mirybrook Road
Danbury, Connecticut 06810

Computer Services of Danbury
14 Summit Street
Danbury, Connecticut 06810

DIST. OF COLUMBIA

Program Store
4200 Wisconsin Ave., N W
Washington, D C 20016

FLORIDA

Computer Worlds
2232 East Bay Drive
Clearwater, Florida 33516

Adventure International
507 East Street
Longwood, Florida 32750

All Systems Go
105 West Plant Street
Winter Garden, Florida 32787

ILLINOIS

3H Computer
1950 Biller Road
Aurora, Illinois 60504

AAA Chicago Computer Center
120 Chestnut Lane
Wheeling, Illinois 60090

Midwest Computer Peripherals
1467 S. Michigan Ave
Chicago, Illinois 60605

MASSACHUSETTS

Small Business Systems Group
6 Carlisle Road
Westford, Mass 01886

Omnitek Systems
1899 Main Street
Tewksbury, Mass 01876

Computer Plus
245A Great Road
Littleton, Mass 01460

MICHIGAN

Adapts —
600 28th Street
Grand Rapids, Michigan 49509

Remarkable Software
1508A Defense
Muskegon, Michigan 49441

James Butler
438 East Lake Street
Petoskey, Michigan 49770

Eight Bit Corner
722 Evanston Avenue
Muskegon, Michigan 49442

Soft Sector Marketing, Inc.
6250 Middlebelt Road
Garden City, Michigan 48135

Matrix Software
315 Marion Avenue
Big Rapids, Michigan 49307

Breeze Computing
PO Box 1013
Berkley, Michigan 48072

NEW HAMPSHIRE

Hardside
6 South Street
Milford, New Hampshire 03055

NEW JERSEY

Floppy Disk Services
40 Misty Morn Lane North
Trenton, New Jersey 08638

NEW MEXICO

Rocky Mountain Engineering
4749 Southern, S E
Albuquerque, New Mexico 87108

NEW YORK

John D. Owens Associates, Inc.
12 Schubert Street
Staten Island, New York 10305

B.T. Enterprises
171 Hawkins
Centereach, New York 11720

NORTH CAROLINA

Alpha Technology
1201 Wicker Drive
Raleigh, North Carolina 27604

OHIO

Electronics Unlimited
824 East 14th Street
Ashtabula, Ohio 44004

OREGON

1300 Centre Electronics
1300 6th Street
Umatilla, Oregon 97882

PENNSYLVANIA

Stevens Radio Shack
562 Nutt Road
Phoenixville, Pennsylvania 19460

TEXAS

Quality Software
11500 Slemmons Expressway
Dallas, Texas 75229

Inventory Control System

This system gives the user complete access to and control over an unlimited number of inventory items. (Note: It has been found that systems containing more than 13,000 items are difficult to handle, from a maintenance time standpoint. Operations, such as Block File reoganization can take more than 3 hours.)

As presently configured, a Model I computer with one 40-track drive will hold 1,925 items, or for a four-drive system a total of 6,625 items. Four 80-track drives will hold in excess of 13,000 items. A Model III computer, with two 40-track drives, will hold about 13,000 parts. If required, A.M. Electronics can modify both the program and hardware configuration to allow on-line storage of an unlimited number of items. As the file becomes larger, maintenance operations take longer. Average search time is six seconds, with 12 seconds the longest time.

Upon program initialization, the user specifies the item number and description digit length. This allows for item numbers up to 23 alpha-numeric characters. (As item number digits increase, digits for description usage are decreased.)

The program is completely menu-driven. Items can be added, edited or deleted from the file. Items can be placed on order, received to stock, or sold from inventory. Complete printout capabilities are available through the printout menu.

Item information includes: item number, description, supplier, re-order point, cost, *wholesale* and retail selling price, quantity, on-order, and total sold. The re-order point is calculated by the program, based on number of items sold over a period of time. The time period is established at program initialization. **$199.95**

Get a sample software printout—Free!

Call or write us to get sample program printouts for these programs, or if you need more information. We'll be happy to send it to you.

The power behind the drives®

A.M. ELECTRONICS, INC. ✓452

Ann Arbor: 3366 Washtenaw Ave., Ann Arbor, MI 48104 (313) 973-2312 Store Hours: Monday-Friday 10-6, Sat. 10-5
Grand Opening of our Radio Shack Authorized Dealership at: 111 North Grand, Box 1071, Fowlerville, MI 48836 (517) 223-7281
Attention Dealers, OEM's & Distributors: Call us for details on our attractive pricing.

™TRS-80 is a trademark of Tandy Corp.

Prices subject to change without notice

✓ See List of Advertisers on page 322

Julian to Gregorian converter.

The Date Maker

Garry H. Grosse
Box 3455
Seymour Johnson AFB
Goldsboro, NC 27531

```
1000 ' THIS ROUTINE DETERMINES THE VALIDITY OF A GREGORIAN
1001 ' DATE STORED IN THE VARIABLES:
1002 '                    MM - MONTH
1003 '                    DD - DAY
1004 '                    YY - YEAR
1005 ' RETURNED IS A FLAG (FL):
1006 '                    FL=0  VALID DATE
1007 '                    FL=1  INVALID DATE
1008 '
1010 FL=0
1020 IFDD<1ORMM<1THEN1080
1030 ONMMGOTO1050,1060,1050,1070,1050,1070,1050,1050,1070,1050,1
070,1050
1040 GOTO1080
1050 IFDD>31THEN1080ELSERETURN
1060 IFYY=(INT(YY/4))*4THEN1065
1063 IFDD>28THEN1080ELSERETURN
1065 IFDD>29THEN1080ELSERETURN
1070 IFDD<=30THENRETURN
1080 FL=1
1090 RETURN
```

Program Listing 1.

The task of date keeping has always been a complexing one. With the advice of an astronomer, Julius Caesar produced the first modern calendar. This calendar adopted 365.25 days as a year, but the actual time span of the earth's orbit is 0.0078 days less than the 365.25 days of the Julian calendar. Noting this inconsistancy, Pope Gregory XIII instituted the Gregorian calendar in 1582 making the length of the year 365.2425 days. Through the following years, this calendar was accepted by most countries.

Due to the fact that most computer systems use the Julian date, and also that the Julian date uses one byte less for storage, such as in a file record, it's convenient to have some type of routine to flip-flop between Julian for computers and Gregorian for humans. This article deals with three subroutines that can be incorporated within

Program Listing 2.

```
2000 ' THIS ROUTINE CHANGES A GREGORIAN DATE TO A JULIAN DATE.
2001 ' THE GREGORIAN DATE IS STORED IN THE VARIABLES:
2002 '                    MM - MONTH
2003 '                    DD - DAY
2004 '                    YY - YEAR
2005 ' RETURNED ARE:
2006 '                    N2 - DAY OF THE YEAR
2007 '                    YY - YEAR
2008 '
2010 N2=MM
2020 N1=YY
2030 IFN2<=3THENN1=N1-1
2040 N1=INT(N1*1461)
2050 N1=INT(N1/4)
2060 N2=N2-3
2070 IFN2<=0 N2=N2+12
2080 N2=INT(N2*153)
2090 N2=N2+2
2100 N2=INT(N2/5)
2110 N2=N2+N1
```

Program continues

Listing 2 continued

```
2120 N2=N2+DD
2130 N1=YY
2140 N1=N1-1
2150 N1=INT(N1*1461)
2160 N1=INT(N1/4)
2170 N1=N1+306
2180 N2=N2-N1
2190 IFMM=3N2=N2-2
2200 RETURN
```

your programs to provide such features and other added extras.

In Program Listing 1, the first subroutine is called Editdate, and it performs just as its name implies. You enter a Gregorian date at some point in your program, then execute a GOSUB 1000. After Return, to determine the validity of the date, you perform a comparison on the variable FL. If FL = 0 the date is correct, but if FL = 1 then the date is invalid. If the date was invalid, you would have the operator reenter the correct date. Notice that the routine does take into account the month of February during leap years.

The second subroutine, in Program Listing 2, converts a Gregorian date, stored in the variables MM, DD, YY. The Julian date is returned in the variables N2 and YY. N2 is the actual day number of the year, and YY remains unchanged.

This subroutine is presented in one-line, simple equations. This was done to present the date conversion routine in an understandable way. Many of these could be combined into one equation if you were in need of conserving program length.

The Julian converter, Program Listing 3, converts a Julian date into a Gregorian date. The Julian date is stored in the variables DD and YY—day of the

Program Listing 3.

```
3000 ' THIS ROUTINE CHANGES A JULIAN DATE TO A GREGORIAN DATE.
3001 ' THE JULIAN DATE IS STORED IN THE VARIABLES:
3002 '                    YY - YEAR
3003 '                    DD - DAY
3004 ' RETURNED ARE:
3005 '                    N2 - 1 THRU 7 (SUNDAY THRU SATURDAY)
3006 '                    N3 - 1 THRU 12 (JAN THRU DEC)
3007 '                    DD - DAY OF MONTH
3008 '                    YY - YEAR
3009 '                   DA$ - DATE STRING
3010 '
3020 N1=YY-1
3030 N1=INT(N1*1461)
3040 N1=INT(N1/4)
3050 N1=N1+306
3060 N1=N1+DD
3070 N2=INT(N1/7)
3080 N2=INT(N2*-7)
```

Program continues

Listing 3 continued

```
3090 N2=N2+N1-3
3100 IFN2<=0 N2=N2+7
3110 N1=N1*4
3120 N1=N1-1
3130 DD=N1
3140 DD=INT(DD/1461)
3150 DD=INT(DD*-1461)
3160 DD=DD+N1
3170 DD=DD+4
3180 DD=INT(DD/4)
3190 N1=DD
3200 N1=INT(N1*5)
3210 N1=N1-3
3220 N3=N1
3230 N3=INT(N3/153)
3240 DD=N3
3250 N3=N3-9
3260 IFN3<=0 N3=N3+12
3270 DD=INT(DD*-153)
3280 DD=DD+N1
3290 DD=DD+5
3300 DD=INT(DD/5)
3310 ONN2GOTO3311,3312,3313,3314,3315,3316,3317
3311 DB$="SUNDAY":GOTO3320
3312 DB$="MONDAY":GOTO3320
3313 DB$="TUESDAY":GOTO3320
3314 DB$="WEDNESDAY":GOTO3320
3315 DB$="THURSDAY":GOTO3320
3316 DB$="FRIDAY":GOTO3320
3317 DB$="SATURDAY"
3320 ONN3GOTO3321,3322,3323,3324,3325,3326,3327,3328,3329,3330,3
331,3332
3321 DC$="JANUARY":GOTO3340
3322 DC$="FEBRUARY":GOTO3340
```

Program continues

year and the year. The Gregorian date is returned in the variables N3, DD and YY which are the month, day, and year respectively.

This routine also provides some added features. It returns the variable N2 which contains numbers one through seven representing Sunday through Saturday. The subroutine also returns a string, DA$, which contains an entire date string already formatted. For example, say you GOSUBed with a Julian date of 58064. DA$ would contain Wednesday, March 5, 1958, upon return from the subroutine.

Again, this subroutine is written in simple equations which could be combined in a more complex form if need be. Also, the DA$ routine at lines 3310 through 3350 could be removed if you did not desire this feature.

Program Listing 4 provides a short program which uses all three of the subroutines for your own testing. ■

Listing 3 continued

```
3323 DC$="MARCH":GOTO3340
3324 DC$="APRIL":GOTO3340
3325 DC$="MAY":GOTO3340
3326 DC$="JUNE":GOTO3340
3327 DC$="JULY":GOTO3340
3328 DC$="AUGUST":GOTO3340
3329 DC$="SEPTEMBER":GOTO3340
3330 DC$="OCTOBER":GOTO3340
3331 DC$="NOVEMBER":GOTO3340
3332 DC$="DECEMBER"
3340 Y1=YY+1900
3345 DA$=""
3350 DA$=DB$+", "+DC$+STR$(DD)+","+STR$(Y1)
3360 RETURN
```

```
5 CLEAR1000
10 CLS:PRINT@20,"TOTAL DATE CONVERSIONS":PRINT@85,"*** TEST PROG
RAM ***":PRINT
15 INPUT"ENTER DATE (MM,DD,YY): ";MM,DD,YY
20 GOSUB1000:IFFL=1PRINT"INVALID DATE - PLEASE REENTER":FORX=1TO
1000:NEXT:GOTO10
30 GOSUB2000
40 PRINT"JULIAN DATE:";YY;N2
50 DD=N2:GOSUB3000
60 PRINT"CONVERTED BACK:";N3;DD;YY
70 PRINT"STRING RETURNED: ";DA$
80 INPUT"TRY ANOTHER DATE (Y/N)";A$
90 IFA$="Y"THEN10
100 STOP
```

Program Listing 4.

Happiness is a warm boot.

Command File Modifier

Donald G. Crawford
4220 W. Carol Ave.
Phoenix, AZ 85021

One of the problems I encountered early in my use of the disk-based TRS-80 was the need to enter several commands before my application would run.

A friend solved my problem. He has a short machine language program that "spoon feeds" commands to the TRS-80. When executed, it overlays the keyboard driver address with one of its own. Then, each request for input is retrieved from the program's internal buffer. When the last character is retrieved, the program then restores the ROM keyboard driver address.

You can enter commands into the program with Apparat's Superzap. But there is one feature of this machine language program that I don't like. It does not save the keyboard driver address on entry so as to return to the keyboard debounce code after executing a command file.

A Solution

My solution is the BASIC program shown in Program Listing 1. I call it CMDFILE, and it will create a new command file or modify an existing one. Instructions are within the program and describe most of the operational details. A few points, however, may clarify this program for you.

Machine language files for the TRS-80 disk system can contain up to three record types. The type is given in byte one of each variable length record. A type 01 record contains machine language data to be loaded by the system.

Byte two of a type 01 record is the count of data bytes +2 in the record (it includes bytes three and four). Bytes three and four are the address at which to start loading the record (in the normal Z 80 reversed manner).

A type 02 record is a transfer address record at which to begin execution of the program. Byte two is always 02 (A type 02 record is always four bytes long). Bytes three and four are the entry address for the program.

A type 03 record is a comment record and is not used in this program.

Program Commentary

As you can see in lines 1180 through 1240, responding ? to the FILESPEC request results in displaying the instructions. A null response (enter only) terminates the program. In DOS, the default extent for a system loadable file is CMD. Line 1250 through 1270 ensure, if the user has not supplied an extent, that "/CMD" is inserted in the proper place.

Lines 1290 through 1380 notify the user if the file specified exists. He may then modify this file if he wishes, but the default is *no* modification. If the file does not exist, error processing will send control to line 1370. Any error other than "non-existent file" will also be displayed and the program will terminate.

Lines 1520 through 1550 check an existing file to make certain it is a command file written by this program. We wouldn't want to destroy another file! Lines 1650 through 1670 list its current contents. Line 1630 sets a flag to say that no changes have been made to the file.

Line 1970 keeps the user from entering too many characters.

In order to keep the machine language program simple, the

Program Listing 1

```
1000 '
************************** CMDFILE ******************* 80-07-27

1010 '
1020 '         BUILD A COMMAND FILE
1030 '
1040 '
****************************************************************
        INITIALIZATION

1050 CLEAR 1000 ' MAKE SPACE TO WORK WITH THE STRINGS
1060 '
1070 ' BUILD THE TYPE 01 RECORD
1080 '
1090 FOR I=1 TO 52
1100 READ D
1110 P1$=P1$+CHR$(D)
1120 NEXT
1130 '
1140 ' BUILD THE TYPE 02 RECORD
1150 '
```

program continues

```
1160 P2$=CHR$(2)+CHR$(2)+CHR$(0)+CHR$(107)
1170 '
1180 CLS
1190 PRINT TAB(27)"CMDFILE"
1200 PRINT "TYPE A QUESTION MARK (?) IF YOU NEED INSTRUCTIONS."
1210 PRINT
1220 LINEINPUT "FILESPEC: ";F$
1230 IF F$="?" GOSUB 2160:GOTO 1180
1240 IF F$="" THEN 2360
1250 J=INSTR(F$,"/") ' SUPPLY "/CMD" IF NO EXTENT PRESENT
1260 IF J<>0 THEN 1280
      ELSE J=INSTR(F$,"."):
      IF J=0 J=INSTR(F$,":")
1270 IF J=0 F$=F$+"/CMD"
      ELSE F$=LEFT$(F$,J-1)+"/CMD"+MID$(F$,J)
1280 '
1290 ON ERROR GOTO 1370 ' SEE IF THE FILE ALREADY EXISTS
1300 OPEN "I",1,F$
1310 ON ERROR GOTO 0
1320 PRINT "ERROR:  ";F$;" ALREADY EXISTS."
1330 GOSUB 2100 ' BEEP
1340 INPUT "DO YOU WANT TO MODIFY IT";A$
1350 A$=CHR$(ASC(LEFT$(A$,1)) OR 32) ' FORCE LOWER CASE
1360 IF A$="y" THEN 1520 ELSE CLOSE:GOTO 1180
1370 IF ERR/2 = 53 THEN RESUME 1380
       ELSE ON ERROR GOTO 0:RESUME 2360
1380 ON ERROR GOTO 0
1390 '
****************************************************************
       BUILD A NEW FILE

1400 GOSUB 1920   ' OPEN THE FILE
1410 LSET R1$=P1$ ' ESTABLISH TYPE 01 RECORD
1420 LSET R2$=P2$ ' ESTABLISH TYPE 02 RECORD
1430 PUT 1,1      ' MAKE SURE THERE'S ROOM ON THE DISK
1440 CLOSE                ' LEAVE THE FILE CLOSED
1450 P$=""
1460 PRINT "READY"
1470 LINEINPUT "*";A$
1480 IF A$<>"" GOSUB 1970:GOTO 1470
1490 GOSUB 2010 ' CLOSE THE FILE
1500 GOTO 1180  ' END
1510 '
****************************************************************
       MODIFY AN EXISTING FILE

1520 GOSUB 1920 ' OPEN THE FILE
1530 GET 1,1    ' GET THE OLD DATA
1540 '            SEE IF IT'S ONE OF OURS
1550 IF P1$<>R1$
       CLOSE:
       PRINT "ERROR:  INVALID FILE FORMAT":
       GOSUB 2100:
       PRINT:
       LINEINPUT "HIT ENTER.";A1$:
       GOTO 1180
1560 '
1570 '          IT'S OK, LET'S HAVE A LOOK AT IT
1580 '
1590 D$=DD$     ' PUT THE DATA IN D$
1600 CLOSE      ' AND LEAVE THE FILE CLOSED
1610 P$=""
1620 J=0
1630 CHANGED=0     ' SET "NO CHANGES MADE" FLAG FALSE
1640 K=INSTR(1,D$,CHR$(10)) ' FIND THE END
1650 PRINT
1660 PRINT LEFT$(D$,K-1)
1670 PRINT
1680 PRINT "ADD TO THE FRONT OF THE FILE OR <ENTER> TO CONTINUE.
"
1690 LINEINPUT "*";A$
1700 IF A$<>"" GOSUB 1970:CHANGED=-1:GOTO 1690
1710 L=J+1
1720 J=INSTR(L,D$,CHR$(13))  ' FIND THE END OF THE FIRST LINE
1730 IF J=0 J=K              ' THERE'S ONLY ONE LINE
1740 A$=MID$(D$,L,J-L)
1750 PRINT A$;
1760 A1$=""
1770 PRINT TAB(35)" (ADD, CHANGE, DELETE)";
1780 INPUT A1$
1790 IF A1$="" GOSUB 1970:GOTO 1870
1800 A1$=LEFT$(A1$,1)
1810 IF A1$="D" CHANGED=-1:GOTO 1870
1820 IF A1$="A"
       GOSUB 1970:
       A1$="C"
1830 IF A1$<>"C"
       PRINT "INPUT ERROR":
       GOSUB 2100:
       GOTO 1750
1840 CHANGED=-1
1850 LINEINPUT "*";A$
1860 IF A$<>"" GOSUB 1970:GOTO 1850
1870 IF J < K THEN 1710
1880 IF CHANGED GOSUB 2010 ' IF IT'S BEEN CHANGED, RE-WRITE IT.
1890 GOTO 1180
1900 '
****************************************************************
       << SUBROUTINES >>
****************************************************************
1910 '
****************************************************************
       SUBROUTINE TO OPEN AND FIELD FILE # 1

1920 CLOSE
1930 OPEN "R",1,F$
1940 FIELD 1,52 AS R1$,200 AS DD$,4 AS R2$
1950 RETURN
1960 '
****************************************************************
       SUBROUTINE TO ADD A COMMAND TO P$

1970 IF LEN(P$)+LEN(A$) > 199
```

program continues

command file is restricted to a length of one sector (256 bytes). I have not had any trouble with this restriction. Lines 2100 through 2140 produce an error tone. Lines 2320 through 2350 contain the type "01" record. The assembled machine language code is shown in Program Listing 2.

This program will work under TRSDOS as well as NEWDOS 2.1. It is unnecessary under NEWDOS80 because of the Chain feature.

The command file can be used as the Auto command to load machine language routines and run a program. If you want to supply a null response to the TRSDOS request for Number Of Files? or Memory Size?, use a single blank space in the command file.

One word of caution: the command file uses the upper portion of the DOS area (6B00H - 6BFFH). Some of the DOS commands and BASIC will eventually overlay it. If you run into problems (for example, when the system "hangs" and you can't do anything through the keyboard) your error may be overlaying the command file. Reboot the system and try another way. I hope you will find this program as useful as I have.■

```
        PRINT "ERROR:  TOO MANY COMMANDS.":
        GOSUB 2100:
        PRINT "LENGTH =";LEN(P$)+LEN(A$)+1;"CHARACTERS.":
        LINEINPUT "HIT ENTER.";F$:
        GOTO 1180
1980 P$=P$+A$+CHR$(13)
1990 RETURN
2000 '
*******************************************************************
        SUBROUTINE TO WRITE THE FILE AND CLOSE IT

2010 MID$(P$,LEN(P$))=CHR$(10) ' SET THE END FLAG
2020 GOSUB 1920       ' OPEN THE FILE
2030 LSET DD$=P$      ' PUT THE COMMANDS IN THE BUFFER
2040 LSET R1$=P1$     ' PUT THE TYPE 01 RECORD IN THE BUFFER
2050 LSET R2$=P2$     ' PUT THE TYPE 02 RECORD IN THE BUFFER
2060 PUT 1,1          ' WRITE THE COMMAND FILE
2070 CLOSE
2080 RETURN
2090 '
*******************************************************************
        SUBROUTINE TO BEEP AN ERROR

2100 IF INP(255)=63 OUT 255,9 ELSE OUT 255,1
2110 FOR I=1 TO 50
2120 NEXT
2130 IF INP(255)=127 OUT 255,0 ELSE OUT 255,8
2140 RETURN
2150 '
*******************************************************************
        PRINT THE INSTRUCTIONS

2160 CLS
2170 PRINT TAB(27)"CMDFILE"
2180 PRINT "THIS PROGRAM WILL BUILD A FILE OF COMMANDS EXECUTABL
E UNDER"
2190 PRINT "DOS.  THE FILE MAY BE USED AS THE <AUTO> FILE OR MAY
"
2200 PRINT "BE ISSUED AS A COMMAND.  YOU MAY ENTER UP TO 200 CHA
RACTERS"
2210 PRINT "OF COMMANDS (INCLUDING <ENTER> CHARACTERS).  THE FIL
ESPEC"
2220 PRINT "SHOULD HAVE THE EXTENT ";CHR$(34);"CMD";CHR$(34);
        " SO THAT THE SYSTEM WILL LOAD IT."
2230 PRINT "(I WILL SUPPLY /CMD AS A DEFAULT.)  I WILL PROMPT YO
U WITH AN"
2240 PRINT "ASTERISK (*) AND YOU RESPOND WITH THE COMMANDS YOU W
ANT."
2250 PRINT "FOR NULL ENTRIES (SUCH AS RESPONSE TO NUMBER OF FILE
S OR"
2260 PRINT "MEMORY SIZE) TRY A SPACE FOLLOWED BY <ENTER>."
2270 PRINT "TERMINATE YOUR ENTRIES WITH A NULL LINE ( <ENTER> ON
LY )."
2280 PRINT
2290 LINEINPUT"HIT ENTER.";F$
2300 RETURN
2310 '
*******************************************************************
        DATA FOR MACHINE LANGUAGE ROUTINE

2320 DATA 1,250,0,107,42,22,64,34,37,107,33,21,107,34,22,64
2330 DATA 33,48,107,34,46,107,195,45,64,229,42,46,107,126,254
2340 DATA 10,40,6,35,34,46,107,225,201,33,227,3,34,22,64,62
2350 DATA 13,24,244,48,107
2360 END
```

Program Listing 2

```
                 00100 ;
                 00110 ;          MACHINE LANGUAGE COMMAND FILE
                 00120 ;
6B00             00130            ORG     6B00H
402D             00140 DOS        EQU     402DH
4016             00150 KEYDVR     EQU     4016H
000D             00160 CR         EQU     0DH
000A             00170 LF         EQU     0AH
                 00180 ;
6B00 2A1640      00190 INIT       LD      HL,(KEYDVR)      ;GET THE DRIVER ADDRESS
6B03 22256B      00200            LD      (QUIT+1),HL      ;SAVE IT
6B06 21156B      00210            LD      HL,GETCHR        ;GET ROUTINE ADDRESS
6B09 221640      00220            LD      (KEYDVR),HL      ;PUT IT IN THE DRIVER
6B0C 21306B      00230            LD      HL,BUFFER        ;POINT TO BUFFER START
6B0F 222E6B      00240            LD      (BUFPNT),HL      ;SAVE POINTER
6B12 C32D40      00250            JP      DOS              ;AND RETURN TO DOS
                 00260 ;
6B15 E5          00270 GETCHR     PUSH    HL               ;SAVE HL
6B16 2A2E6B      00280            LD      HL,(BUFPNT)      ;POINT TO NEXT CHARACTER
6B19 7E          00290            LD      A,(HL)           ;GET THE CHARACTER
6B1A FE0A        00300            CP      LF               ;IS IT END OF BUFFER?
6B1C 2806        00310            JR      Z,QUIT           ;YES, QUIT
6B1E 23          00320            INC     HL               ;NO, BUMP POINTER
6B1F 222E6B      00330            LD      (BUFPNT),HL      ;SAVE IT
6B22 E1          00340 EXIT       POP     HL               ;RESTORE HL
6B23 C9          00350            RET                      ;RETURN TO CALLER
                 00360 ;
6B24 21E303      00370 QUIT       LD      HL,03E3H         ;GET SAVED DRIVER ADDRESS
6B27 221640      00380            LD      (KEYDVR),HL      ;RESTORE IT
6B2A 3E0D        00390            LD      A,CR             ;RETURN A CR
6B2C 18F4        00400            JR      EXIT             ;AND GET OUT
                 00410 ;
6B2E 306B        00420 BUFPNT     DEFW    BUFFER           ;POINT TO BUFFER
00C8             00430 BUFFER     DEFS    200              ;RESERVE STORAGE SPACE
6B00             00440            END     INIT
00000 TOTAL ERRORS
```

COMING NEXT MONTH

Have you ever wanted a low-cost servant who never goes on strike, never takes a vacation and can work 24 hours a day? Have you ever wanted a particular book and found the library has loaned out its only copy to an arctic explorer who wanted it for light reading over a winter at the North Pole? Then you'll want to read our September issue.

Next month the *80 Microcomputing* staff will bravely explore the uncharted forests of robots. Our own Chris Brown will provide you with an overview of robotics as it is today and its connection with microcomputers. The article will explain the two kinds of robots—show and industrial—and what they can and cannot do. It will include a look at a couple of robot manufacturers and comments by the editor of *Robotics Age Magazine*.

For those who want to get involved personally, freelance writer Don McAllister will tell you how to interface R2D2 with your TRS-80 for a total cost (exclusive of the microcomputer) of $63. The resulting robot will respond to voice and infra-red light commands to execute any of a number of preprogrammed sets of behavior.

The news section will deal with another important issue, information distribution. Bert Latamore has looked behind the scenes of the electronic information revolution. He discovered that the networks that bring all those databases to you at the cost of a local telephone call make just as fascinating and important a story as the services themselves. His stories explain how the international packet network system came to be and how it works. He also lists some of the more interesting databases available on them and gives you a look at Ethernet, a new local network system for businesses, developed by Xerox and supported by Digital Equipment Co. (DEC) and Itel.

Chris Brown will also give us a look at a new use for the TRS-80 as a coin-operated machine in a public library.

Dennis Kitsz will bring us another Exclusive Oracle column of questions and answers as well as his regular 80 Applications column.

✓See List of Advertisers on page 322

Enter common Basic words in a single keystroke for programming efficiency.

Basic Shorthand

Jared Radin
1458 Sheffield Dr.
Atlanta, GA 30329

This program will let you type 25 of the most commonly used Basic words in a single keystroke. After the program is executed, hold down the shift down-arrow key and a letter from A to X, to produce a Basic word—for example, pressing the F key will produce the word LPRINT; the G key will produce the word GOTO; the P key will produce the word PEEK, and so on. The program can be modified for other Basic words by modifying entries in a table. The program uses only 141 bytes in high memory and does not disable any other keyboard drivers present. Most Basic words are matched to a key with the same starting letter so the abbreviations are easy to memorize.

The first part of the program allows the user to gain control of the keyboard. The second part disables the shift down-arrow key, and when the shift down-arrow and a letter are hit simultaneously, it initializes the third part of the program for a Basic word, and passes control of the keyword to the third routine. The third routine then prints each letter of the Basic word and then returns control to the second part of the program.

The first part determines the current keyboard routine, and does one of two things. If the program running is the keystroke program, it returns control to DOS, or Basic II, if modified. If the current keyboard routine is any other routine, the keystroke program modifies the second part of this program to call the previous keyboard routine. For example, if a keyboard debounce routine was loaded prior to the execution of the keystroke program, the keyboard debounce routine will still be active. That routine is activated by putting the address of the second part of the program in the keyboard device control block.

The second part of the program is called whenever Basic or DOS, is seeking a character from the keyboard. At this point the previous keyboard routine is called. This routine loads the ASCII value of the key hit to the accumulator. If no key has been hit, the value will be zero and the program will return to the calling program. The program will also return to the calling program if the shift down-arrow was just hit, is not down, or a letter key was not hit. At this point, to disable the break key check the accumulator to see if it equals one; if so, set it equal to zero. We are now ready to substitute a word for a single key. Each Basic word is assigned a number, and listed in a Basic keyword table located at the end of the program. The value of the key hit on the keyboard and its address in the table are added, and the result points to the number of the keyword.

In the Basic keyword table, the first letter of each word has the high order bit turned on. The words are counted by counting the high bits encountered until the number of counted words in the table equals the number of the desired word, pointing to the first letter of the desired Basic word. The pointer is saved, the third part of the program is activated by putting its address in the keyboard control device block, and then jumping to the third part of the program for the first letter of the Basic word.

The third part of the program is called to get the next letter of the word. The pointed-to letter is sent to the calling program (exactly as if that letter was hit on the keyboard), the pointer is advanced to the next letter and saved. If the pointer is now pointing to the first letter of the next word, the address in the keyboard control block is set to point to the second part of the program.

The last part of the program is a table of 25 numbers, one for each of the alphabet keys. Each number is the number of a Basic word in the Basic keyword table, and can be modified to represent any Basic word.

Two modifications are necessary if you do not have a 48K disk system. First, change the return address to Level II Basic instead of DOS, as written in the program. Second, change the origin of the program to fit your computer.

If you do not have disk, I would recommend changing some of the DOS Basic keywords to words that you will find useful, such as CLOAD or CSAVE. If you do not have NEWDOS (this program is compatible with NEWDOS) you may wish to use the words Edit, Delete and List as part of the table.

When using the program remember to set the memory size to an address below the program origin. ■

```
                00100 ;PROGRAM TO ALLOW SINGLE KEY ENTRY OF BASIC WORDS
                00110 ;EXAMPLE HITTING THE DOWN ARROW AND SHIFT 'D'
                00120 ;IS THE SAME AS TYPING THE WORD 'DATA'
                00130 ;PROGRAM COPYRIGHT APRIL, 1980 BY JARED RADIN
                00140 ;1458 SHEFFIELD DRIVE NE, ATLANTA, GA. 30329
                00150 ;PERMISSION IS GRANTED TO MAKE COPIES SOLELY FOR
                00160 ;PERSONAL USE OR TO GIVE COPIES TO YOUR FRIENDS
FF00            00170         ORG     0FF00H          ;MODIFY IF LESS MEMORY
4016            00180 KEYDRV  EQU     4016H
402D            00190 DOS     EQU     402DH   ;1A19H FOR RETURN TO LEVEL II
1650            00200 WORDS   EQU     1650H   ;POINTER TO LIST OF BASIC WORDS
FF00 ED5B1640   00220 BEGIN   LD      DE,(KEYDRV)     ;GET OLD DRIVER ADDRESS
FF04 211AFF     00230         LD      HL,DRIVER       ;ADDRESS OF NEW DRIVER
FF07 AF         00240         XOR     A               ;CLEAR CARRY BIT
FF08 ED52       00250         SBC     HL,DE           ;SUBTRACT DRIVERS
FF0A CA2D40     00260         JP      Z,DOS           ;DRIVER ALREADY INSTALLED
FF0D ED531BFF   00270         LD      (DRIVER+1),DE   ;AND USE IN NEW DRIVER
FF11 211AFF     00280         LD      HL,DRIVER       ;ADDRESS OF NEW DRIVER
FF14 221640     00290         LD      (KEYDRV),HL     ;AND SAVE AS NEW DRIVER
FF17 C32D40     00310         JP      DOS             ;BACK TO DOS
                00320 ;THIS IS THE NEW KEYBOARD DRIVER
FF1A CD         00330 DRIVER  DEFB    0CDH    ;CALL INSTRUCTION WITH NO ADDRESS
                00340 ;THE NEXT TWO BYTES ARE FOR THE ADDRESS - WHEN LOADING
                00350 ;THE PROGRAM THE PREVIOUS ADDRESS MUST BE UNMODIFIED
                00360 ;AS THE PROGRAM MAY HAVE BEEN USED ALREADY AND IN THIS
                00370 ;CASE THE NEXT TWO BYTES ARE THE ONLY PLACE WHERE THE
                00380 ;ADDRESS OF THE PREVIOUS DRIVER IS PRESENT - THAT IS THE
                00390 ;KEYBOARD DCB IS ALREADY POINTING TO DRIVER - AND NOT TO
                00400 ;THE ORIGINAL DRIVER
FF1D            00410         ORG     $+2     ;LEAVE ROOM FOR ADDRESS
FF1D B7         00420         OR      A       ;WAS A KEY HIT ?
FF1E C8         00430         RET     Z       ;RETURN IF NO KEY HIT
FF1F E5         00440         PUSH    HL
FF20 214038     00450         LD      HL,3840H        ;PNTR TO DOWN ARROW ROW
FF23 CB66       00460         BIT     4,(HL)          ;TEST DOWN ARROW BIT
FF25 2805       00470         JR      Z,NODWN   ;IF NO ARROW DON'T TEST SHIFT
FF27 218038     00480         LD      HL,3880H        ;POINT TO SHIFT ROW
FF2A CB46       00490         BIT     0,(HL)          ;SEE IF SHIFT IS ON
FF2C E1         00500 NODWN   POP     HL
FF2D C8         00510         RET     Z       ;RETURN IF NO SHIFT OR DOWN ARROW
FF2E FE1A       00520         CP      26              ;SHIFT DOWN ARROW HIT ?
FF30 2002       00530         JR      NZ,LETTER       ;NOT DOWN ARROW
FF32 AF         00540         XOR     A               ;GET RID OF DOWN ARROW
FF33 C9         00550         RET
FF34 FE1A       00560 LETTER  CP      26      ;COMPARE TO CONTROL 'Z'
FF36 D0         00570         RET     NC      ;RETURN IF GREATER THAN CONTROL X
                00580 ;SHIFT DOWN ARROW IS ON AND ALSO HAVE LETTER
FF37 E5         00590         PUSH    HL
FF38 C5         00600         PUSH    BC
FF39 2172FF     00610         LD      HL,TABLE-1      ;POINTER TO TABLE
FF3C 4F         00620         LD      C,A
FF3D 0600       00630         LD      B,0     ;BC=VALUE OF LETTER
FF3F 09         00640         ADD     HL,BC           ;HL=>TABLE OF #
FF40 46         00650         LD      B,(HL)          ;B=WORD NUMBER
FF41 215016     00660         LD      HL,WORDS        ;HL=>START OF BASIC WORDS
FF44 05         00670 NWORD   DEC     B               ;COUNT OFF WORDS
FF45 2807       00680         JR      Z,FOUND         ;HL => WORD
FF47 23         00690 NLET    INC     HL              ;POINT TO NEXT LETTER
FF48 CB7E       00700         BIT     7,(HL)          ;HIGH BIT ON (STRT. WORD)
FF4A 28FB       00710         JR      Z,NLET          ;IF NOT KEEP LOOKING
FF4C 18F6       00720         JR      NWORD           RIGHT WORD ?
                00730 ;NOW HL POINTS TO THE FIRST LETTER IN THE WORD
FF4E 2271FF     00740 FOUND   LD      (POINT),HL      ;SAVE HL
FF51 2159FF     00750         LD      HL,WRDDRV       ;DRIVER TO GET WORD
FF54 221640     00760         LD      (KEYDRV),HL     ;AND CHANGE DRIVER
FF57 C1         00770         POP     BC
FF58 E1         00780         POP     HL
                00790 ;DRIVER TO GET LETTERS IN WORD
FF59 E5         00800 WRDDRV  PUSH    HL
FF5A 2A71FF     00810         LD      HL,(POINT)      ;GET POINTER
FF5D 7E         00820         LD      A,(HL)          ;GET LETTER POINTED TO
FF5E E67F       00830         AND     7FH             ;GET RID OF HIGH BIT
FF60 23         00840         INC     HL              ;POINT TO NEXT LETTER
FF61 2271FF     00850         LD      (POINT),HL      ;SAVE POINTER
FF64 CB7E       00860         BIT     7,(HL)          ;SEE IF HIGH BIT ON
FF66 E1         00870         POP     HL
FF67 C8         00880         RET     Z               ;RETURN IF STILL A LETTER
FF68 E5         00890         PUSH    HL
                00900 ;RESTORE ORIGINAL DRIVER
FF69 211AFF     00910         LD      HL,DRIVER
FF6C 221640     00920         LD      (KEYDRV),HL
FF6F E1         00930         POP     HL
FF70 C9         00940         RET
FF71 0000       00950 POINT   DEFW    0
FF73 77         00960 TABLE   DEFB    119     ;ASC
FF74 14         00970         DEFB    20      ;REM
FF75 06         00980         DEFB    6       ;CMD
FF76 09         00990         DEFB    9       ;DATA
FF77 16         01000         DEFB    22      ;ELSE
FF78 30         01010         DEFB    48      ;LPRINT
FF79 0E         01020         DEFB    14      ;GOTO
FF7A 12         01030         DEFB    18      ;GOSUB
FF7B 0A         01040         DEFB    10      ;INPUT
FF7C 4A         01050         DEFB    74      ;INKEY$
FF7D 2B         01060         DEFB    43      ;KILL
FF7E 28         01070         DEFB    40      ;LOAD
FF7F 7B         01080         DEFB    123     ;MID$
FF80 08         01090         DEFB    8       ;NEXT
FF81 23         01100         DEFB    35      ;OPEN
FF82 66         01110         DEFB    102     ;PEEK
FF83 32         01120         DEFB    50      ;POKE
FF84 13         01130         DEFB    19      ;RETURN
FF85 2E         01140         DEFB    46      ;SAVE
FF86 4B         01150         DEFB    75      ;THEN
FF87 3D         01160         DEFB    61      ;TAB(
FF88 41         01170         DEFB    65      ;VARPNT
FF89 78         01180         DEFB    120     ;CHR$
FF8A 79         01190         DEFB    121     ;LEFT$
FF8B 7A         01200         DEFB    122     ;RIGHT$
FF00            01210         END     BEGIN
00000 TOTAL ERRORS
```

Program Listing

Model I owners take note.

Color Computer First Impressions

Richard L. Kilmon
106 East Green Valley Circle
Green Valley
Newark, DE 19711

Gadgets. Those who know me know I love gadgets. The neat new wristwatch that is also a calculator and that new computer with color. (Sometimes you have to write articles to afford those neat new gadgets.)

The Color Computer caught my eye in a local Radio Shack while I was helping a friend evaluate a Model III. As a Model I owner, I must admit my first impression of the TRS-80 Color Computer was unfavorable. This was partly due to store conditions.

The video display was not adjusted for color or fine tuning. The system was sitting on top of another (Radio Shack stores are always cramped), and of course the salesman was friendly, but not very knowledgeable about his product.

I must admit that he did have one selling point: As an independent dealer, he could sell for a discount. That, and the prospect of playing with sound and color graphics, tweaked me just right to make the plunge.

The unit comes in a shipping box with several precut foam pieces designed to fit, protect and allow easy removal of the main unit. Cords and instruction manuals are located below the unit. These are less convenient to get at. I had to be careful here since I didn't want to destroy the packing—just in case.

The complete contents of the box are as follows: one computer with U-ground power cord; one tv switching box; one interface cable; one packet of instruction materials—operator's manual, instruction manual (similar to the Level I manual), a Basic reference card, warranty cards, a notice to programmers about the Let statement, and an additional information request card.

Getting Hooked

Setting up the computer is very simple. The operator's manual covers just about every possible combination for hooking it up. The tv switching box attaches to the VHF terminals on the back of the set. I am using a Sony KV1513 tv. An RCA phone plug cable attaches between the computer and the tv. The antenna connects to the switching box. Plug in the computer and the tv, switch the box to the computer setting, and you're ready to go. Well, almost.

A word of caution here. Sony tv sets, as well as many others, are often AC/DC units. This means there is no transformer between the line and the set's electronics. My set has one large blade on the plug which is meant to aid plugging it in properly, but I know many of you folks probably have older homes and don't know if the holes in your receptacles are polarized correctly. Polarization is important for the sake of the computer—don't destroy your new toy turning it on.

I bought joysticks and connected them, also. These plugs are DIN plugs, and are well marked on the back of the computer. The cartridge game plugs into the right side of the computer. Make sure it is right side up. The computer should be turned off while you are making these connections; the cartridge will seize control of the computer.

Learning the Lingo

When you turn it on, the computer initializes and the screen reads: "Color Basic 1.0 ; (C) 1980 Tandy ; OK." That's the last you'll see of the title until the next time it's turned on.

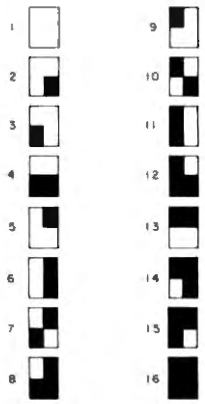
I entered the command: PRINT MEM, even though this is not mentioned in the literature. The answer came back 2343 bytes, which left me a bit perplexed. A 4K computer that only lets me play with 2.3K bytes? It looks like a lot of memory is used in housekeeping functions and graphics. To me this means that more memory is a must. I verified this number at the local RS store on one of their demos.

The Basic commands are straightforward and appear to syntactically match Level II Basic in most ways, with additional graphics and sound commands to use with this computer's new features. Color Basic is more

AUDIO on or off	Allows you to hear a program listing being loaded from the cassette recorder.
CLEAR n,h	The h allows you to sets high memory. CLEAR 150,10000 clears 150 bytes and sets memory protect at 10,000.
CLOADM	Will load a machine language program and an offset address if you desire.
CLS(c)	C equals 0 to 8. This clears the screen and sets the background to one of eight colors.
CSAVE,A	Will save a cassette load in ASCII format.
EXECa	This is similar to System, but allows you to put the beginning address in the command statement.
MOTOR on or off	This allows program control of the cassette recorder motor.
OPENm,#d,f	Allows files to be opened on screen, keyboard, cassette or printer.
SET(H,V,C)	Is the same as the Level II command, except the computer needs to be told what color the block should be.
SOUND(F,D)	This gives a sound controlled with F being the frequency and D being the duration.
EOF(f)	Is used to find out if all the file data has been used or not.
JOYSTIK(j)	Senses the joystick position.
PRINT# – 2	Color Basic's LPRINT.

Table 1. Color Basic Commands

● 1 Through 32 are unknown except for:
CHR$ 8 = Backspace and erase
CHR$ 13 = Linefeed and carriage return
CHR$ 32 = Space
● 33 Through 95 follow in order:
! " # $ % & ' () * + , — . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [/] ↑ ←
● 96 Through 127 are shown on the screen as uppercase but are sent out to printers as lowercase.
@ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [/] ↑ ←
You can tell lowercase because they are in reverse video.
● 128 Through 255 are graphics blocks for the eight colors.



	CHR$ fm	CHR$ to
Green	128	143
Yellow	144	159
Blue	160	175
Red	176	191
Buff	192	207
Cyan	208	223
Magenta	224	239
Orange	240	255

1/4 of the graphics block is six scan lines high

Low Resolution Mode
512 graphics blocks fill the screen,
2048 with the Set command (32 × 64)

Table 2. CHR$ Codes for the Color Computer

powerful than Level I, but not up to the Level II package. The first feature I missed was an edit command. This and some powerful graphics commands are promised in the Extended Basic package.

Table 1 lists some of the commands peculiar to Color BASIC.

Unfortunately, the manual does not discuss some of the new commands, like EOF and Open. DOS users will be able to decipher them, but most will have to wait for Tandy's promised additional information. Don't forget to send in the request card.

Features

The printing on the screen is similar to the 32 character mode on the Model I. I'm not crazy about this but it appears to be standard in the industry.

There is a full set of upper and lowercase characters in the Color Computer. The video can be switched to lowercase mode, but the video display will show lowercase as reverse video uppercase. The printer will get the proper ASCII code to print lowercase. Graphics remind me of the Model I.

The basic graphic block is broken down into four sub-blocks, not six as in the Model I. The four sub-blocks can be configured 16 different ways. Therefore, there are 16 codes for the variations of each color. It is not as bad as it sounds (see Table 2).

If you have a color computer, you can print out the character set by running the following program:

```
10 CLS (0)
20 FOR X = 0 TO 255
30 PRINT 225,"CHR$ CODE NO. = ";X
40 PRINT 370,CHR$(X);
50 FOR Y = 0 TO 500: NEXT Y
60 NEXT X
```

Graphics in a 32 × 64 format are not particularly impressive, even in color, but denser graphics are promised for the Extended Basic version. Tandy claims that access to 192 × 256 graphics is possible through their Extended Basic, machine language or their program packs. It may even be possible in the 4K machine, but it would not be practical with only 2.3K bytes available for storage of the graphics program.

Colors can vary drastically from one tv to another. It is important for everyone that has one of these computers to run the program on page 24 of the operator's manual. This makes the computer act like a color bar generator, and will allow you to adjust your set for the best combination of colors possible. Mixed color graphics can really get messed up if you don't do this.

I have a Quickprinter II that I interfaced with this computer. I wanted to take advantage of the RS-232 output on the back. Unfortunately, Tandy uses a four pin DIN plug which was not available in any Tandy store. I wound up butchering another DIN plug to get what I needed. The next problem was that it didn't work. I finally realized that RD and CD were reversed on the RS-232 pinout in the operator's manual. Beware of this if you plan to connect a serial printer to the port—I don't know if it is a hardware bug, instruction manual bug or if it is unique to my unit.

"I wound up butchering another DIN plug to get what I needed. The next problem was that it didn't work."

How it Feels

The keyboard does not have the feel or look of that of the Model I and II, but it still performs nicely. Each key has a slight amount of tactile feedback and is set in regular typewriter format. A touch-typist will have no trouble with this keyboard. The Model I keybounce problem does not exist on this unit.

I had trouble connecting my recorder to the machine. I tried to use the cable that came with the Model I machine, but the outer shell is too thick. This unit requires the five pin DIN plugs that have the thin metal shield around the outside. I bought a set at Radio Shack, and this wouldn't fit either! Taking a closer look, I noticed that the plastic molding on this plug was getting in the way. I finally shaved back the plastic on the cable plug. I wonder what Tandy used to connect their recorder to these units.

No, your TRS-80 programs won't load into this machine. the speed is wrong and the syntax is different. Machine language is worse, since this unit uses a 6809 processor, which means the machine code is completely different for this machine. Some enterprising person will have to come up with a translation program (cross-compiler).

About Sound

You can play music or sound tones anywhere in a program using the Sound command. After typing in Sound, you need to tell the computer what frequency and what duration you wish to use. SOUND 89,8 represents about one quarter note of middle C.

The Color Basic machine allows you only one voice (one note at a time). I like to put sound at user entry points, at error points and at the end of programs. This allows you to concentrate on something else while the computer is running, knowing the tone will indicate when the computer is ready for input.

I think this computer is worth the price. It has moderate expandability and a decent Basic language. It can be used as a game computer, since the ROM paks seize control and allow an inexperienced user to play games. Buy another pak and a modem and you can communicate over the phone lines to The Source or CompuServe for big computer power. This unit is a good compromise for those of you who have been wanting to get into the microcomputing area but didn't have the big dollars or the family go-ahead. Sell them on the games—you know it will do a lot more. ■

BUYER **SELLER**

ARE YOU PUZZLED BY THE USED COMPUTER MARKET?

Our unique, nationwide listing service puts buyers and sellers of used micro-equipment together.

Listings from $300 — $25,000
(Apples, TRS 80, IBM 5100's included)

YOU PAY ONLY FOR RESULTS!

We also offer for $6.75 the
USED MICRO-TRENDS REPORT
Data on:
Manufacturers • Models • Prices
Trends • Maintenance
call toll-free

USED COMPUTER EXCHANGE ✓494
(800) 327-9191 ext. 61

Professional software

TRS-80® MODEL II

NOW AVAILABLE!

BASIC CROSS REFERENCE

- FIND WHERE NAMES ARE USED FAST!
- CAN YOU DELETE THAT LINE? FIND OUT!
- DO YOU HAVE DEAD CODING?
- WANT A NICE PROGRAM LISTING WITH DATE & TIME IN HEADING
- SAVE HOURS!
- PARALLEL/SERIAL PRINTERS
- SEVERAL OPTIONS
- 6/8 LPI VARIABLE LINE WIDTH & PAGE DEPTH

$59.99 + POSTAGE & HANDLING

DOCUMENTATION ONLY $10.00 DEDUCTIBLE ON PURCHASE

DISK SORT

- YOU DON'T HAVE TO BE A PROGRAMMER TO USE IT!
- MENU DRIVEN
- RANDOM FILES
- CHAINS TO SYSTEM OR BASIC PROGRAMS
- EASY TO USE
- FAST!
- SPECS SAVED ON DISK
- EASY TO INSERT INTO JOB STREAM FOR NON-STOP RUNNING!

$69.99 + POSTAGE & HANDLING

DOCUMENTATION ONLY $10.00 DEDUCTIBLE ON PURCHASE

BASIC COMPILER

- WRITTEN BY MICROSOFT
- TRSDOS®/BASIC COMPATIBLE!
- FASTER THAN BASIC BY UP TO 30 TIMES
- EASY TO USE
- AUGMENTED WITH OUR DOCUMENTATION

$350.00 + POSTAGE & HANDLING

'TRS-80®' & 'TRSDOS®' ARE REGISTERED TRADEMARKS OF TANDY CORP

GOOD-LYDDON DATA SYSTEMS ✓218
5486 RIVERSIDE DR., CHINO, CA. 91710
MASTER CHARGE or VISA accepted.

THE BOOKKEEPERS

FOR INFO CALL (603)-447-2745

Full Charge Bookkeeper—48K, 3 DRIVE, w/ALPHA	**$129.95**
Intermediate Bookkeeper—48K, 2DRIVE & Printer	**$109.95**
Cheap Bookkeeper—32K, 2DRIVE & Printer	**$ 89.95**

All Above Are Daily Journal—G/L Systems

Ann Rose, our Accounts Receivable Clerk— 48K, 2 DRIVE & PRINTER **$150.00**

STURDIVANT & DUNN, INC. ✓82
BOX 277, 124 WASHINGTON ST., CONWAY, NH, 03818

COMPUTE COMPUTED COMPUTING COMPUTES COMPUTER COMPUTERS COMRADE COMRADES COMTE COMTES CON CONNED CONNING CONS CONATION CONATIONS CONATIVE CONATUS CONCAVE CONCAVED CONCAVING CONCAVES CONCEAL CONCEALED CONCEALING CONCEALS CONCEDE CONCEDED CONCEDING CONCEDES CONCEDED CONCEDER CONCEDERS CONCEIT CONCEITED CONCEITING CONCEITS CONCEIVE CONCEIVED CONCEIVING CONCEIVES CONCENT CONCENTS CONCEPT CONCEPTS (... NCERTO CONCERTOS CONCH CONCHS CONCHES CONCHA ... CONCLAVE CONCLAVES CONCLUDE CONCLUDED CONCLU ... TE CONCRETING CONCRETES CONCUR CONCURRED CONCL ... NING CONDEMNS CONDENSE CONDENSED CONDENSING C(... M CONDOMS CONDONE CONDONED CONDONING CONDON ... CES CONDUCER CONDUCERS CONDUCT CONDUCTED CONC ... NES CONELRAD CONELRADS CONENOSE CONENOSES CON ... IFABS CONFECT CONFECTED CONFECTS CONFECTING CONFER CONFERRED CONFERRING CONFERS CONFEREE CONFEREES CONFERVA CONFERVAE CONFERVAS CONFESS CONFESSED CONFESSING CONFESSES CONFETTO CONFETTI CONFIDE CONFIDED CONFIDING CONFIDES CONFIDER CONFIDERS CONFINE CONFINED CONFINING CONFINES CON-

JOE COMPUTER PRESENTS **WORD GRINDER** A DATA BASE CONTAINING OVER

80,000 WORDS!

WORD GRINDER is a new and unique product. Over 700,000 bytes, this data base has applications in many aspects of data processing. **WORD GRINDER** is intended to be the definitive data base of words for computers.

WORD GRINDER is over 80,000 English words in alphabetical order with a space delimiting each word. The TRS-80† and APPLE versions include a BASIC program for displaying and editing the data files using random access.

Applications are limited only by a Programmer's imagination. Put a subset of **WORD GRINDER** on your system. Write programs for:

- Word processing spell correction
- Auto-poetry
- Anagrams
- Word games
- Crossword puzzles
- Computerized palindromes:
  ("Naomi, sex at noon taxes," I moan.)

These words have been proofread once by computer and twice by humans. The editor allows you to add your own words, delete words, or make modifications. A software license is available for selling your programs containing **WORD GRINDER** or a subset of **WORD GRINDER**. Quantity dealer prices available.

WORD GRINDER IS AVAILABLE FOR:

TRS - 80 MODEL I — Contained on nine 35 track 5¼'' floppy disks including system disk with editor **$259.00**
Or seven 40 track floppy disks **249.00**

TRS - 80 MODEL II — Contained on two double density 8'' data floppy disks with editor **295.00**

TRS - 80 MODEL III — Contained on five 40 track double density 5¼'' floppy disks including system disk and editor ... **249.00**

APPLE II — Contained on six 35 track 5¼'' 3.3 DOS system floppy disks including Applesoft editor **249.00**
Or seven 3.2 DOS 5¼'' floppies **254.00**

CP/M* — Contained on 3 single density 8'' data floppy disks .. **295.00**

RT-11/HT-11 — Contained on 3 single density 8'' data floppy disks **295.00**

DEC/ANSI — Contained on standard nine track tape produced on a PDP-11 .... **295.00**

✓193

Make checks payable to JOE COMPUTER — Phone orders and information: (213) 992-0514
Send to: JOE COMPUTER, 22713 Ventura Blvd., Suite F, Woodland Hills, CA 91364
California residents add 6% sales tax.

** CP/M is a trademark of Digital Research* † TRS-80 is a trademark of Tandy Corp.

✓See List of Advertisers on page 322

Handle some text.

Take a Letter

Arthur T. Mullin, Jr.
2390 Tyler
Beaumont, TX 77703

This program is dedicated to the task of writing business letters. The text is input via line input, and any editing must be performed in real time, using the backspace key. While not a sophisticated program, it does have some features which should be of interest to persons considering a text handling routine.

The program was written on a Level II TRS-80 which drives an Anadex DP8000 printer. The user is prompted to enter his name and address and the name and address of the business being written to. It automatically starts off with "Dear Sir," and from there on the special effects come into play.

The program features keyboard inversion, to make it conform to the familiar typewriter standard of upper/lowercase method of input. The exciting feature is that it automatically capitalizes those characters which obviously need capitalization, such as the personal pronoun I, letters which occur after the termination of a sentence, and characters at the beginning of a paragraph.

It is not flawless: If the letter I occurs at the end of a line, it must be followed by one space in order for the program to detect it as a "stand alone character." Except for this one idiosyncrasy, the program can be a viable tool.

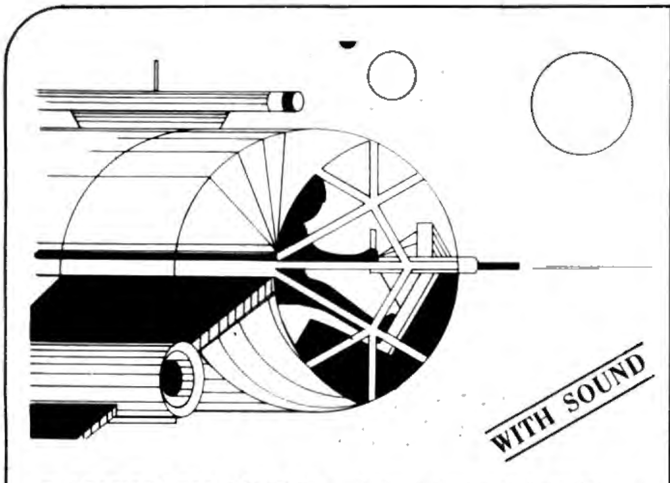
Except for capitalizations in the middle of sentences, it is not necessary to use the shift key.

I believe the user will find adequate REM statements in the Program Listing that will assist in any efforts to understand or modify the program.■

```
10 CLEAR5000:' * RESERVE STRING SPACE
20 DIMA$(100):' * ALLOW UP TO 100 LINES OF TEXT
30 CLS
40 LINEINPUT"YOUR NAME ";E$:' * ACCEPT ADDRESSOR INFORMATION
50 LINEINPUT"YOUR STREET ADDRESS ";A$
60 LINEINPUT"YOUR CITY, STATE, ZIP ";C$
70 LINEINPUT"THE DATE ";D$
80 LPRINTSTRING$(45," ");E$:LPRINT
90 LPRINTSTRING$(45," ");A$:LPRINT
100 LPRINTSTRING$(45," ");C$:LPRINT
110 LPRINTSTRING$(45," ");D$:LPRINT:LPRINT:LPRINT:LPRINT
120 LINEINPUT"BUSINESS NAME ";N$:' * ACCEPT ADDRESSEE INFORMATIO
N
130 LINEINPUT"STREET ADDRESS ";A$
140 LINEINPUT"CITY, STATE,ADDRESS ";C$
150 LPRINTSTRING$(10," ");N$:LPRINT
160 LPRINTSTRING$(10," ");A$:LPRINT
170 LPRINTSTRING$(10," ");C$:LPRINT:LPRINT
180 LPRINTSTRING$(10," ");"Dear Sir:":LPRINT:LPRINT
190 CLS
200 P=P+1:' * NOTE LINE NUMBER
210 PRINT@X," ";
220 PRINTSTRING$(60,"-");:' * PROVIDE VIDEO REFERENCE FOR LINE L
ENGTH
230 PRINT@X," ";:X=X+64:IFX>960THENCLS:X=0:GOTO220:' * CLEAR A F
ULL SCREEN
240 LINEINPUTA$(P):' * TEXT INPUT
250 IFA$(P)="["THENP=P-1:GOTO260ELSE200:' * ENTER UP ARROW TO PR
INT TEXT
260 FORN=1TOP:' * LOOK AT EACH LINE (A$(P))
270 C$=RIGHT$(A$(N-1),1):' * LOOK AT LAST CHARACTER OF PREVIOUS
LINE
280 IFC$="."ORC$="?"ORC$="!"ORA$(N-1)=""THENZ=1:' * CHECK PREVIO
US LINE FOR TERMINATION
290 Y=1:' * THIS IS THE FIRST TIME THROUGH
300 FORX=1TOLEN(A$(N)):' * LOOK AT EACH CHARACTER IN STRING
310 A=ASC(MID$(A$(N),X,1)):' * NOTE NUMERICAL VALUE OF EACH CHAR
ACTER
320 A$=MID$(A$(N),X,1):' * NOTE EACH CHARACTER
330 IFZ=1ANDX=1THENZ=0:GOTO390:' * IF THIS IS THE FIRST CHARACTE
R AND IF THE PREVIOUS STRING WAS TERMINATED, SKIP INVERSION
340 IFY=1ANDA$<>" "ANDLEFT$(A$(N),2)="  "THENY=0:GOTO390:' * IF
THIS IS THE FIRST TIME THROUGH AND TWO (OR MORE) SPACES START TH
E STRING, THEN THIS IS A PARAGRAPH: SKIP INVERSION
350 N$=RIGHT$(B$,2):IFN$=". "ORN$="! "ORN$="? "THEN390:' * IF A
SENTENCE HAS BEEN TERMINATED, SKIP INVERSION
360 I$=MID$(A$(N),X+1,1):' * LOOK AHEAD ONE CHARACTER
370 IFA$="I"THENIFI$="'"ORI$=" "ORI$="?"ORI$="!"ORI$="."ORI$=","
ORI$=":"THEN390:' * IF CHARACTER "I" DEDUCED TO BE PERSONAL PRON
OUN, SKIP INVERSION
380 IFA>=65ANDA<=90THENA$=CHR$(A+32)ELSEIFA>=97ANDA<=122THENA$=C
HR$(A-32):' * INVERT KEYBOARD
390 B$=B$+A$:' * ACCUMULATE LAUNDERED LINE
400 NEXTX:' * GO BACK FOR NEXT CHARACTER IN THIS STRING
410 LPRINTSTRING$(10," ");B$:' * SEND LAUNDERED LINE TO PRINTER
420 LPRINT:' * DOUBLE SPACE
430 B$="":' * CLEAR ACCUMULATION STRING
440 NEXTN:' * GO BACK FOR NEXT STRING
450 LPRINT:LPRINT:LPRINT
460 LPRINTSTRING$(40," ");"Your's truly"
470 LPRINT:LPRINT:' * PROVIDE SPACE FOR SIGNATURE
480 LPRINTSTRING$(40," ");E$:' * PRINT ADDRESSOR'S NAME
490 RUN
```

Program Listing

GO BOLDLY . . .
Where No TRS-80* Program Has Gone Before!



DANGER IN ORBIT

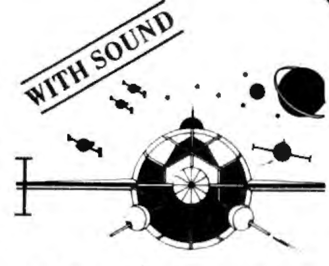
DATE: 28.02.2047
LOCATION: 270 million miles from Terra
MISSION: Maintaining Terra's Space Lanes

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 score keeping. (T1 or T2)

Order No. 0237R
$14.95 Tape.
Order No. 0247RD
$19.95 Disk.

BALL TURRET GUNNER

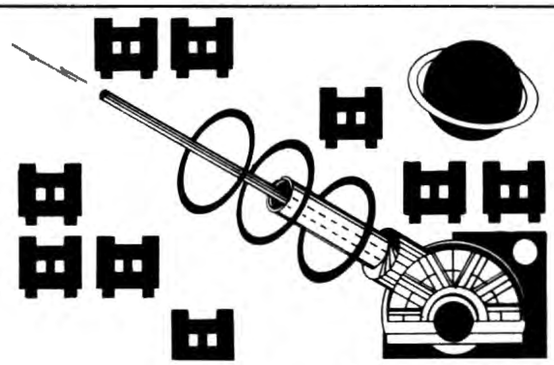


For years the Petro Resource Conglomerate has attacked our photon collection stations and strangled our deep-space trade routes. The PRC Exxonerator Class light fighters (code name: Gnat) have been their main weapon. Now you can strike back, by joining the Ball Turret Gunner Service.

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! (T1)

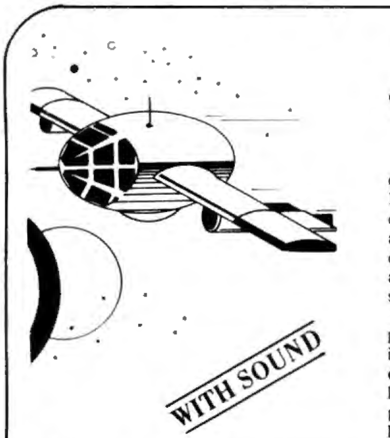
Order No. 0051R $9.95.



ALIEN ATTACK FORCE

The INVADERS are coming! Earth's defenses are dead except for your Laser base. Your assignment is to destroy the approaching INVADERS before they destroy 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 short profile craft which is the most difficult to destroy. If you cannot stop these space attackers they will stop Earth..... for good. (T1)

Order No. 0240R $9.95.



COSMIC PATROL

WARNING: PLAYERS OF THIS GAME SHOULD BE PREPARED FOR A STATE OF REALISM HITHERTO UNAVAILABLE ON THE TRS-80

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! (T1 or T2)

Order No. 0223R
$14.95 Tape.
Order No. 0224RD
$19.95 Disk.

TO ORDER
See Your Local Instant Software Dealer or Call Toll-Free 1-800-258-5473
(Orders Only)
In NH dial 1-603-924-7296

Instant Software™
PETERBOROUGH, N.H. 03458 530

(T1) = TRS-80 Model I, Level II, 16K RAM.
(T2) = TRS-80 Model I, Level II 16K, expansion interface 16K + 1 disk drive.

See List of Advertisers on page 322

Try this method of programming.

Three's Not a Crowd

David W. Mawdsley
8 New Light Dr.
Danbury, CT 06810

When I write a computer program, I now do two things I never did before. First, I always keep two other utility programs in memory to help me when needed. Second, with a minimum of paper scratch notes, I compose most of the program while I'm seated at the computer keyboard.

This gives me full use of the edit features of Level II Basic, the protected memory for system programs, the regular work spaces for a Basic program, and the efficiency of a line printer. My printer is also required to simulate a regular computer terminal upon command!

Program Loading

This is the procedure I use when programming on my 16K microcomputer system:

- To the memory size question, I respond 31792.
- After I type System, I put Radio Shack's RENUM line renumbering program into my tape recorder and type RENUM. When the program has finished loading, I push the Break key. Designate this System program as program A, which resides at memory locations 31820 through 32767.
- Next I put my videoscreen copier program into the tape recorder, followed by a CLOAD (Program Listing 1). After this program loads, I type Run. When it has finished, this Basic program has loaded a *second* System program into the protected memory locations 31793 through 31819. I refer to this system program as program B. (Listing 2 is a detailed assembled Z-80 list of program B. The decimal list of its object code appears in lines 60 and 65 of Listing 1.)
- Finally, I load my project tape into the tape recorder and again CLOAD. (If no lines have been coded yet I type New.) I call this Basic program C. It resides—or will reside—in the usual memory locations starting at 17129.

Program A helps me format programs by neatly renumbering the list of program C. Program B keeps a printed record of any runs, lists, trials and errors of program C which I feel should be saved. Program B copies the

```
READY
>SYSTEM
•? RENUM
•?
READY
>CLOAD
READY
>RUN
POWER DOWN IF MEMORY SIZE
WASN'T SET < = 31792. OKAY ? YES
READY
>CLOAD "C"
READY
```

Fig. 1.

```
10 ' ** VIDEO TO LINE PRINTER USR(0)-CALLED PROGRAM **
     BY
15 ' DAVID W. MAWDSLEY, 8 NEW LIGHT DR., DANBURY, CT 06
     810
20 PRINT "POWER DOWN IF MEMORY SIZE WASN'T SET < = 3179
     2 .";
25 INPUT "  OKAY ";A$
30 ' ** SET THE USR WINDOW. **
35 POKE 16526,49: POKE 16527,124
40 ' ** UNSTICK THE DATA POINTER. **
45 POKE 16553,255
50 ' ** LOAD THE SYSTEM PROGRAM. **
55 FOR I = 31793 TO 31819: READ X: POKE I,X: NEXT I
60 DATA 1,64,13,33,0,60,126,205,59,0,13,40,3,35
65 DATA 24,246,62,13,205,59,0,14,64,5,248,24,242
70 END
```

Program Listing 1.

```
7C31             00100      ORG     7C31H        ;ORIGIN 31793 DECIMAL
7C31 01400D      00110      LD      BC,0D40H     ;B=13, C=64 DECIMAL
7C34 21003C      00120      LD      HL,3C00H     ;VIDEO ORGIN
7C37 7E          00130      LD      A,(HL)       ;LOAD CHAR. OF VIDEO LOC.
7C38 CD3B00      00140      CALL    003BH        ;SEND TO LINE PRINTER
7C3B 0D          00150      DEC     C            ;DEC CHAR. COUNT OF LINE
7C3C 2803        00160      JR      Z,$+5        ;GO IF 0 (E.O.L. - VIDEO)
7C3E 23          00170      INC     HL           ;BUMP VIDEO LOCATION
7C3F 18F6        00180      JR      $-8          ;GET AND LOAD NEXT CHAR.
7C41 3E0D        00190      LD      A,0DH        ;CARRIAGE RET. IN A REG.
7C43 CD3B00      00200      CALL    003BH        ;SEND TO LINE PRINTER
7C46 0E40        00210      LD      C,40H        ;RESTART LINE CHAR. COUNT
7C48 05          00220      DEC     B            ;NEXT LINE OF VIDEO
7C49 F8          00230      RET     M            ;RETURN THRU BASIC WINDOW
7C4A 18F2        00240      JR      $-12         ;GET NEXT VIDEO LOC.
7C31             00250      END     7C31H
00000 TOTAL ERRORS
```

Program Listing 2.

top 14 lines of the video screen onto the line printer. Experimentation with program B will show why I coded it to skip the last two lines of the video screen.

Operation of the Programs

Runs of program A and B can be made without interfering with program C. I run program A by typing System followed by /31820 after the prompt appears. Typing X = USR(0) runs program B. (Fig. 1 shows a run of program B after I loaded the three programs A, B and C.)

When program A or B finishes, control is fully returned to the keyboard. Unless program C is complicated by its use of extra POKEd memory locations or by its own System program, it will continue to list and operate without disturbing programs A and B.

My decision to employ the USR(0) command rather than a System command to run program B was motivated by two concerns. First, the command X = USR(0) causes less upward line scrolling on the video screen than the System command. Second, the two types of command formats help me remember the correct command to start each program.

Program C and Protected Memory

If program C is complicated by a System program it calls or loads, there are two programming options I can use. First, if program C requires memory where program A or B resides, I let it use the portion of memory it requires, keeping track of the locations used, to decide if either program A or B can still be used. Second, if I haven't yet worked out the object code of the System program operated by program C, I plan to code it so the System program will reside and operate below memory location 7C31 hex (31793 decimal)—so program B is still useable.

Under these circumstances, when I turn the computer system on I set the memory size to one less than the origin of the System program operated by program C. This ensures that I can still use programs A and B. (Unbelievably, four or more programs are now neighbors in the computer's memory. Consider the possibilities!)

USR Calls

Another important concern must be addressed when program C makes USR calls to its own System programs. The USR call window, which is set and used by program C when it runs its own System programs, must be reset if I choose to run program B. Carefully enter one line: POKE 16526,49 : POKE 16527,124 : X = USR(0). This multiple command will reset the USR window for 7C31 hex and then correctly call program B.

Relocating Program B

I sometimes use programs B and C without the line renumbering program A. Since program B is only 27 bytes long, I first set the memory size to 32740—or less if I need to protect memory for another system program. Next I CLOAD the program which loads program B. Before running it, however, I change lines 35 and 55 to read:

```
35 POKE 16526,229 : POKE 16527,127
55 FOR I = 32741 TO 32767 : READ
X : POKE I,X : NEXT I
```

When the revised program is run, it loads the relocatable object code of program B into memory starting at hex location 7FE5 (32741 decimal) after it has set the USR window to 7FE5 hex. Typing X = USR(0) calls program B. If program C has set and used the USR window since the last time I ran program B, the one-line command I use to run program B is: POKE 16526,229 : POKE 16527,127 : X = USR(0).

If you try my method of programming, you will find it a natural way to do your work. The advanced versions of Basic available on the modern microcomputer systems invite a variety of innovative programming techniques for programmers to match to their own writing styles.■

Here comes the sun. Track its position with your 80.

Solar Altitude Plotter

Edward H. Rea III
633 Dunedin Rd. #34B
Portsmouth, VA 23701

Graphic representation of solar altitude is a Basic language program written for the Level II TRS-80. The program occupies 2595 bytes of RAM, so it will easily run with only 4K. Single date run-time is on the order of 130 seconds.

The program calculates the altitude of the sun above the horizon for any latitude and longitude on any date from 1900 to 2099 A.D. It covers a period of 24 hours, with positions displayed at intervals of one hour of Universal time.

The user is required to supply only the latitude and longitude of the site as decimal numbers (no minutes or seconds, please), and the Gregorian calendar date in numerical format as month/day/year (i.e., 12/25/1980).

While the mathematical method used to predict solar longitude is low precision, it usually yields accurate results within ± 1 minute of arc.

The method used to determine solar coordinates outlined here is described in the *Almanac for Computers 1980*, published by the Nautical Almanac Office.

Basically, time calculations generating the Julian date, Greenwich mean, sidereal time, and Greenwich as well as local hour angles are needed along with a bit of trigonometric figuring to produce the solar altitude. The algorithm by which the solar coordinates are determined is usable (though with decreasing accuracy) over the next several centuries, while the relationship used to generate the Julian date will remain accurate until 2099 A.D.

The Julian calendar, which was introduced in Rome in 46 B.C., established the 12-month, 365-day year, with each fourth year having 366 days. The Gregorian calendar is the revised Julian calendar, the only difference being the restriction that only those years with dates divisible by 400 should be leap years. Greenwich is located in England where the meridian of Greenwich is used as the prime basis of standard time throughout the world.

The Julian date is a count of days and fractions of days beginning at Greenwich Mean Noon, January 1, 4713 B.C. In the program, the Julian date for zero hours Universal Time (or Greenwich Time) is calculated from the Gregorian calendar month/day/year format with the formula:

$$JD = 367 \cdot K - INT(7 \cdot (K + INT((M + 9)/12))/4) + INT(275 \cdot M/9) + I + 1721013.5 - 0.5 \cdot SGN(100 \cdot K + M - 190002.5) + 0.5$$

JD is the Julian date at zero hours, M is the numerical month (1–12), I is the calendar day (1–31) and K is the year (1900–2099). Julian date at zero hours Universal time is usually abbreviated JD0, but since this is not an acceptable variable name in Level II Basic, I have shortened the form. I have distinguished the Julian date at zero hours Universal time (JD), from the Julian date at any specific time of day, the latter of which I have elected to call JT in the variable assignment. Simply, $JT = JD + UT/24$ where UT is Universal time in the range 0-24 hours. I have defined both JD and JT as double numbers, since current Julian dates are quite large and force Level II single number representation into exponential notation and truncation.

Once the Julian date has been established, time correlation factor T is computed from $T = (JT - 2415020)/36525$. T represents Julian centuries, and each of 36525 Julian days, that have elapsed since 1900.

You can then compute the true geocentric longitude of the sun (LS) from the mean anomaly of the sun (MS), and the motion in the lunar orbit (L) from the relationships:

$$MS = 358.476 + 35999.05 \cdot T$$
$$L = 279.691 + 36000.769 \cdot T$$
$$LS = L + (1.919 - 0.0048 \cdot T) \cdot SIN(MS) + 0.02 \cdot SIN(2MS)$$

The units of each calculation are degrees and decimal parts of a degree, and the magnitude of LS can be reduced to the range

0-360 degrees by simple modulo arithmetic.

If one assumes the latitude of the sun to be essentially zero, the right ascension (RA) and the declination of the sun (DE) can be calculated as:

TAN(RA) = COS(EP)•TAN(LS)
SIN(DE) = SIN(EP)•SIN(LS)

EP is the obliquity of the ecliptic computed from EP = 32.452 – 0.013•T and again the units are degrees. Because Level II Basic performs trigonometric functions in units of radian measure, it is necessary to convert degrees-to-radians using trigonometry which is in the program KC: (KC = 57.29578).

In using the arc tangent function, it is necessary to provide quadrantial correction as the ATN(TAN(RA)) operation provides correct values only over the range – 90 to + 90 degrees. Correcting for quadrants is simplified by the fact that the right ascension of the sun (RA) is always in the same quadrant as

Greenwich sidereal time (GA) may be computed from Greenwich mean sidereal time (GM). The mean longitude of the ascending lunar orbit (OM), and the equation of the equinoxes (E) are figured by:

OM = 372.1133 – 0.0529539•(JT – 2433282.5)
E = – 0.00029•SIN(OM)
GA = GM + E

with Greenwich apparent sidereal time (GA) generated in hours. The local hour angle (LH) is then: LH = 15•(GA – RA/15) – LO. RA is the sun's right ascension and LO is the longitude of the observer, both in degrees.

Solar altitude (AL) can be computed trigonometrically as: SIN(AL) = SIN(LA)•SIN(DE) + COS(LA)•COS(DE)•COS(LH). LA is the latitude of the observer in degrees. The sun's altitude (AL) is then determined by use of the arc sine function, or in the case of Level II Basic: AL = ATN(SA/SQR(– 1•SA•SA + 1)). SA is the sine of the sun's altitude (SA = SIN(AL)).

"Greenwich sidereal time (GA) may be computed from Greenwich mean sidereal time (GM)."

the sun's true longitude. The arc sine function is derived from ATN to produce the sun's declination (DE), but no correction of quadrants is required, as solar declination must be in the – 90 to + 90 degree range.

In order to figure the ascension and declination for local altitudes, sidereal time relationships and hour angles must be used. Greenwhich mean sidereal time (GM) may be computed from the formula: GM = 6.58852667 + 0.0657098232•(JD – 2433282.5) + 1.0027379093•UT.

JD is the Julian date at zero hours Universal time and UT is the Universal time in hours and decimal parts of an hour. The hour is the unit of GM and its numerical range should be 0-24.

The program uses this method to perform calculations for the desired date in the range 0-24 hours universal time in increments of one hour within a For...Next loop configuration A single altitude calculation is generated for each loop operation. If the calculated altitude is greater than the minimum altitude required for visibility, program execution goes to the display subroutine and the altitude is plotted using the Set instruction.

The graphic display routine plots altitude on the y axis and universal time on the x axis. Since the range of Y in the Set(X,Y) format is 0 to 47, scaling by division is necessary to compress a 0 to 90 degree range of altitude onto the screen. Scale division on two would provide vertical resolution, but as a y range of 0 to 45 would then be needed for the plot, no print lines would remain for axis margin and scaling notation.

I selected a scaling divisor of three, leaving adequate room below the x axis at Set(X,30) to facilitate printed identification of graphic dimension.

X axis scaling is a multiplicative process. Since the range of universal time to be displayed is 0 to 24 hours, a y axis index of 24 and an x axis scaling multiplier of four gives an offset but wide display.

Alternative display techniques might include scaling y at a 1/4 or 1/6 division ratio and displaying a y axis range of – 90 to + 90 degrees altitude. If this approach is taken, the conditional test for visible altitude at line 420 should be removed and the axis indexing and calibration (lines 170-196) are modified or omitted. I have run the program with several different display subroutines

```
50 CLS
60 PRINT"      *************************************************"
61 PRINT"      *                                               *"
62 PRINT"      *          GRAPHIC REPRESENTATION               *"
63 PRINT"      *                    OF                         *"
64 PRINT"      *              SOLAR ALTITUDE                   *"
65 PRINT"      *                                               *"
66 PRINT"      *************************************************"
67 PRINT
100 REM***** OBTAIN LATITUDE AND LONGITUDE VALUES
105 DEFDBL J
110 PRINT" > INPUT LATITUDE AND LONGITUDE IN DEGREES";:INPUT LA,
LO
120 KC=57.29578 : SL=SIN(LA/KC) : CL=COS(LA/KC)
130 REM***** CALCULATE JULIAN DATE
135 PRINT
140 PRINT" > INPUT DESIRED DAY AS MONTH,DAY,YEAR";:INPUT M,I,K
150 JD=367*K-INT(7*(K+INT((M+9)/12))/4)+INT(275*M/9)+I+1721013.5
-0.5*SGN(100*K+M-190002.5)+0.5
160 PRINT" > JULIAN DATE IS";JD
165 CLS
170 FOR S=0 TO 51 : PRINT@650+S,"-"; : NEXT S
171 FOR S=125 TO 701 STEP 64 : PRINT@S,"!"; : NEXT S
172 FOR S=10 TO 61 : PRINT@S,"-"; : NEXT S
175 PRINT@839,"DATE ";M;I;K;"    LATITUDE";LA;"  LONGITUDE";LO;
180 PRINT@714,"0";:PRINT@714+25,"12";:PRINT@714+50,"24";
185 PRINT@789,"TIME IN HOURS UNIVERSAL TIME";
186 PRINT@256," SOLAR";
187 PRINT@320,"ALTITUDE";
190 FOR S=10 TO 650 STEP 64 : PRINT@S,"!"; : NEXT S
195 PRINT@7,"90 -";:PRINT@199,"60 -";:PRINT@391,"30 -";
196 PRINT@584,"1 -";
200 REM**** OK THE LOOP
210 FOR UT=0 TO 24
215 ES=1900+UT*100
216 IF ES>=2400 THEN ES=ES-2400
220 JT=JD+UT/24
230 T=(JT-2415020)/36525
240 MS=358.476+35999.05*T
250 L=279.691+36000.769*T
260 LS=L+(1.919-0.0048*T)*SIN(MS/KC)+.02*SIN(2*MS/KC)
265 IF LS>360 THEN LS=LS-360
266 IF LS>360 THEN 265
270 EP=23.452-.013*T
280 TR=COS(EP/KC)*TAN(LS/KC)
290 SD=SIN(EP/KC)*SIN(LS/KC)
300 RA=ATN(TR)*KC
310 IF RA<0 THEN RA=RA+180
320 IF ABS(LS-RA)>90 THEN RA=RA+180
335 PRINT@924,"          ";
337 IF RA>360 THEN RA=RA-360
340 DE=ATN(SD/SQR(-1*SD*SD+1)) : CD=COS(DE)
350 GM=6.67170278+0.0657098232*(JD-2433282.5)+1.0027379093*UT
360 OM=372.1133-.0529539*(JT-2433282.5)
370 E=0.00029*SIN(OM/KC)
380 GA=GM+E
390 LH=15*(GA-RA/15)-LO : SH=SIN(LH/KC) : CH=COS(LH/KC)
400 SA=SL*SD+CL*CD*CH
410 AL=ATN(SA/SQR(-1*SA*SA+1))*KC
420 IF AL<-.25 THEN 435
430 GOSUB 500
435 PRINT@924,"WORKING";
437 PRINT@960,"  TIME";UT*100;"UT","  SOLAR ALTITUDE ";AL,"  EST
";ES;
440 NEXT UT
445 PRINT@924,"COMPLETED";
446 FOR Q=0 TO 62 STEP1 : PRINT@1022-Q," "; : NEXT Q
450 PRINT@960," > SELECT ANOTHER DATE (Y/N)";:INPUT Z$
460 IF Z$="Y" THEN 135
470 PRINT" > SELECT A NEW POSITION (Y/N)";:INPUT Z$
480 IF Z$="Y" THEN 110
490 END
500 REM ***** DISPLAY SUBROUTINE
510 X=24+4*UT
520 Y=30-AL/3
530 IF X<0 OR X>127 THEN 560
540 IF Y<0 OR Y>47 THEN 560
550 SET(X,Y)
560 RETURN
```

Program Listing

and encourage the user to experiment with routines of his or her own devising.

I also elected to display the numerical result of each loop operation as well as the Eastern Standard Time equivalent of each universal time increment.

To execute the program, type Run, input the latitude and longitude of the observation site and the calendar date, and the graph will be generated, though the display may not plot the first altitude position for several loop iterations. After you are accustomed to the program's operation, try some different latitudes, longitudes and dates. Examine, for instance, the difference in maximum altitude attained at your coordinate location on March 22, June 22, September 22 and December 22 of a particular year; or, examine the same dates in equatorial latitudes and in latitudes greater than 60 degrees.

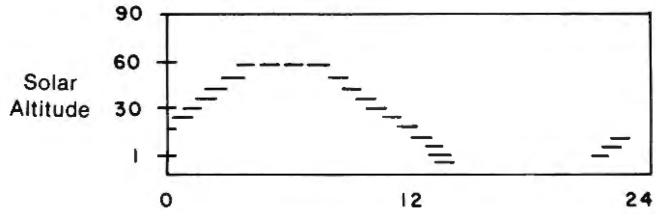
The user may wish to include output for his or her own time zone instead of Eastern Standard Time. Such a modification can be accomplished by substituting the appropriate time zone correlation constant for 1900 in line 215 of the program.

While I did not choose to include a computation of the sun's azimuth (AZ) in the program, users might add this calculation, as all necessary elements have already been supplied:

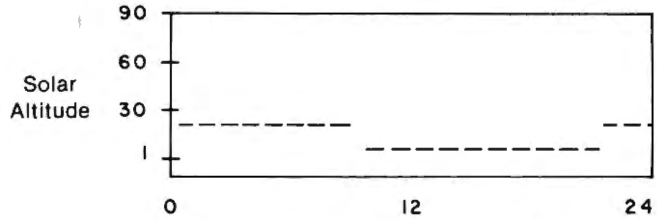
```
TAN(AZ) = SIN(LH)/(COS(LH)*SIN(LA) -
   TAN(DE)*COS(LA))
AZ = ATN(TA)*KC
```

TA is TAN(AZ) and AZ is the sun's azimuth. Remember to use conversion constant KC to keep degrees and radians straight, and to correct for quadrants disturbed by the ATN function.

An inventive programmer might use this alteration coupled with a reduced step value in line 210 to a tenth of an hour or less to track the sun's path for a solar energy device. The TRS-80 could conceivably be interfaced to any number of such devices.



Date 5/17/80 Latitude 43 Longitude 72
Time 2300 UT Solar Altitude 14.0668 EST 1800



Date 8/15/80 Latitude 81.45 Longitude 45
Completed
>Select Another Date (Y/N)?

Fig. 1.

Furthermore, many of the techniques employed in the program are useful in a variety of astrometric and navigational applications.

Graphic representation of solar altitude illustrates that the TRS-80 is not a toy. Rather, with a suitable program, it performs serious computing tasks.■

MULTI-USER OASIS HAS THE FEATURES PROS DEMAND. READ WHY.

(THEN COMPARE.)

NOW AVAILABLE FOR TRS 80 MOD II

Computer experts (the pros) usually have big computer experience. That's why when they shop system software for Z80 micros, they look for the big system features they're used to. And that's why they like Multi-User OASIS. You will too.

DATA INTEGRITY: FILE & AUTOMATIC RECORD LOCKING

The biggest challenge for any multi-user system is co-ordinating requests from several users to change the same record at the same time.

Without proper co-ordination, the confusion and problems of inaccurate or even destroyed data can be staggering.

Our File and Automatic Record Locking features solve these problems.

For example: normally all users can view a particular record at the same time. But, if that record is being updated by one user, automatic record locking will deny all other users access to the record until the up-date is completed. So records are always accurate, up-to-date and integrity is assured.

Pros demand file & automatic record locking. OASIS has it.

SYSTEM SECURITY: LOGON, PASSWORD & USER ACCOUNTING

Controlling who gets on your system and what they do once they're on it is the essence of system security.

Without this control, unauthorized users could access your programs and data and do what they like. A frightening prospect isn't it?

And multi-users can multiply the problem.

But with the Logon, Password and Privilege Level features of Multi-User OASIS, a system manager can specify for each user which programs and files may be accessed — and for what purpose.

Security is further enhanced by User Accounting — a feature that lets you keep a history of which user has been logged on, when and for how long.

Pros insist on these security features. OASIS has them.

EFFICIENCY: RE-ENTRANT BASIC

A multi-user system is often not even practical on computers limited to 64K memory.

OASIS Re-entrant BASIC makes it practical.

How?

Because all users use a single run-time BASIC module, to execute their compiled programs, less memory is needed. Even if you have more than 64K, your pay-off is cost saving and more efficient use of all the memory you have available — because it services more users.

Sound like a pro feature? It is. And OASIS has it.

AND LOTS MORE...

Multi-User OASIS supports as many as 16 terminals and can run in as little as 56K memory. Or, with bank switching, as much as 784K.

Multi-Tasking lets each user run more than one job at the same time.

And there's our BASIC — a compiler, interpreter and debugger all in one. An OASIS exclusive.

Still more: Editor; Hard & Floppy Disk Support; Keyed (ISAM), Direct & Sequential Files; Mail-Box; Scheduler; Spooler; all from OASIS.

Our documentation is recognized as some of the best, most extensive, in the industry. And, of course, there's plenty of application software.

Put it all together and it's easy to see why the real pros like OASIS. Join them. Send your order today.

OASIS

MAKES MICROS RUN LIKE MINIS

OASIS IS AVAILABLE FOR

SYSTEMS: Altos, Compucorp, Cromemco, Delta Products, Digital Group, Digital Microsystems; Dynabyte; Godbout; IBC; Index; Intersystems; North Star; Onyx; SD Systems; TRS 80 Mod II; Vector Graphic; Vorimex.

CONTROLLERS: Bell Controls, Cameo, Corvus, Konan; Micromation, Micropolis, Tarbell, Telefile, Thinkertoys/Cromemco.

Write for complete free Application Software Directory

PLEASE SEND ME:

Product	Price with Manual	Manual Only
OPERATING SYSTEM (Includes EXEC Language, File Management; User Accounting; Device Drivers, Print Spooler, General Text Editor etc.) SINGLE-USER / MULTI-USER	\$150 / 350	\$17.50 / 17.50
BASIC COMPILER/ INTERPRETER/DEBUGGER	100	15.00
RE-ENTRANT BASIC COMPILER/INTERPRETER/ DEBUGGER	150	15.00
DEVELOPMENT PACKAGE (Macro Assembler; Linkage Editor; Debugger)	150	25.00
TEXT EDITOR & SCRIPT PROCESSOR	150	15.00
DIAGNOSTIC & CONVERSION UTILITIES (Memory Test; Assembly Language Converters, File Recovery, Disk Test; File Copy from other OS, etc.)	100	15.00
COMMUNICATIONS PACKAGE (Terminal Emulator, File Send & Receive)	100	15.00
PACKAGE PRICE (All of Above) SINGLE-USER / MULTI-USER	500 / 850	60.00 / 60.00
FILE SORT	100	15.00
COBOL-ANSI '74	750	35.00

Order OASIS from:

Phase One Systems, Inc.
7700 Edgewater Drive, Suite 830
Oakland, CA 94621
Telephone (415) 562-8085
TWX 910-366-7139

NAME ______
STREET (NO BOX #) ______
CITY ______
STATE ______ ZIP ______
AMOUNT \$ ______
(Attach system description; add \$3 for shipping; California residents add sales tax)
☐ Check enclosed ☐ VISA
☐ UPS C.O.D. ☐ Mastercharge
Card Number ______
Expiration Date ______
Signature ______

A case for giving your computer a home of its own.

The Wandering 80

Don Hubert
613 Hartless Court
Hampton, VA 23669

Until the time I decided to buy my TRS-80, I never realized that the houses they build for humans don't include a computer room. Faced with this fact, I had to come up with a place to put the computer. After the kitchen table was vetoed, I got a lease on my wife's sewing room with the provisions that the computer goes when she needs to use the room. It seemed fair enough to me until I found myself moving the TRS-80 more than I was using it.

In any case, I hated to see all those cables and cords laying all over the table in an unprotessional manner, so I decided to make my '80 not only function as a computer but look like a computer as well.

I made the case out of scrap wood and plastic I had in my workshop, but all the materials are readily available from any lumber company or similar store. In addition, I designed the unit so maximum air flow would prevent overheating since the TRS-80 is not equipped with a cooling fan. For this reason, I used 1/2-inch square plastic grill used for fluorescent light covering for the top, rear removal panel, and bottom where wood was not used to connect the sides together.

Start construction by cutting out the side panels (A) from 1/2-inch plywood. (Make sure you have the good side of each facing out.) Next cut out the front panel (B), top panel (C), power panel (D), video monitor panel (E), back panel (F) and the rear door panel (G).

I assembled my unit by nailing (finishing nails) the top and power board first, followed by the front panel. This made the unit sturdy enough to move about without it falling apart. Next I cut out the computer panel from a piece of 1/4-inch wall panel board, putting the unfinished side facing up. (Give it a light sanding.) Note that this panel runs the complete width of the case so figure in the side panels before you cut.

At this point, it's a good idea to double check your work by putting the keyboard in the space provided. Also make sure

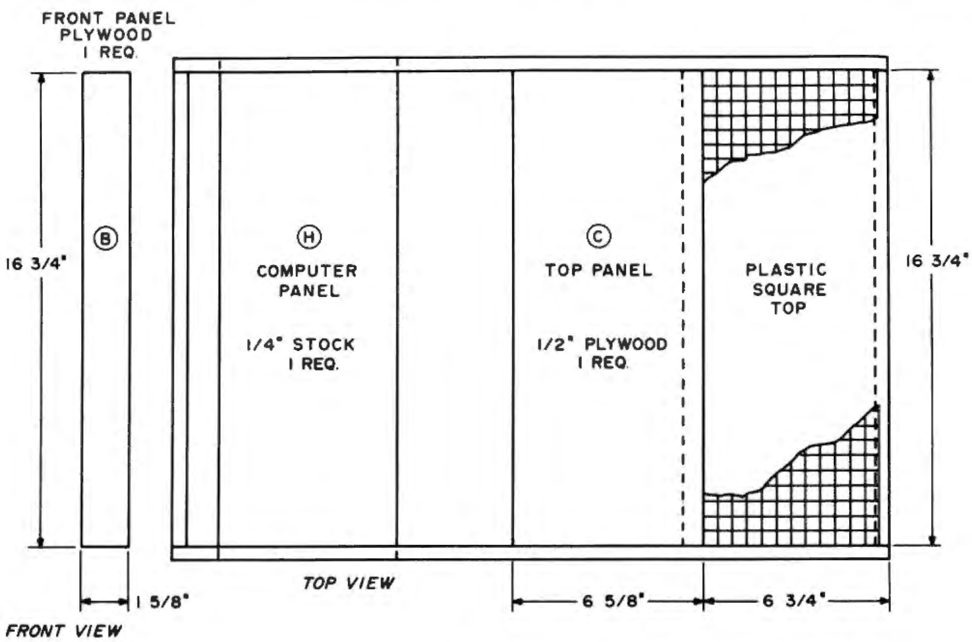


Fig. 1

Photo 1. The complete system, up and running: Note the hood to eliminate glare and feet on each corner to give a free flow of air for cooling.

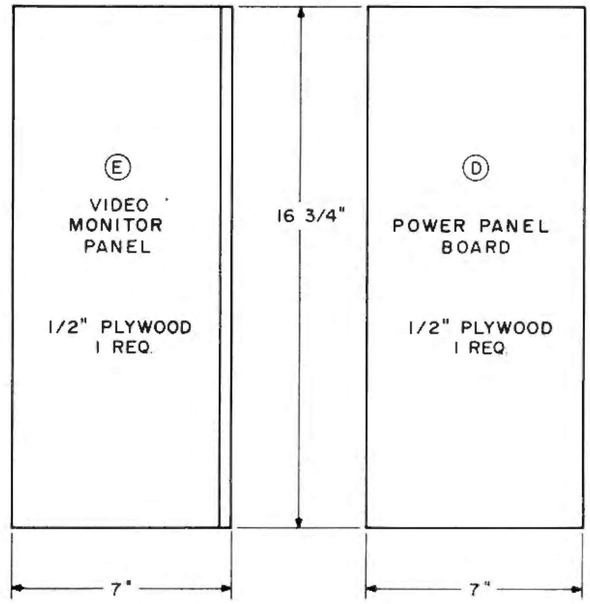


Fig. 2

the video monitor will slip in from the front and your side clearance is okay. If all works out well, proceed. If not, back to the saw.

Take the video monitor panel board (E) and the video monitor and place it in the case. You may need an extra set of hands to hold everything in place. Bring the video monitor forward so it is flush with the front slope of the case and not too tight at the top. Mark where the board is to be nailed. Cut a 1/2-inch strip the length of the video monitor board and glue it to keep the monitor from slipping off the board. Remove the monitor and secure the video monitor board with three nails on each side. Once again slip the video monitor in the case to be sure it fits okay. Replace the computer in the case and make sure you have enough room to reach the on/off and reset switch. Next mount the back panel (F) with two nails on each side.

Now we are ready to add the plastic cooling panels to the top rear of the case. Mark and cut the plastic square to fit between the top (C) and back panel (F). I used a variable speed sabre saw to cut the plastic. Place it on a firm surface and hold it steady when cutting as it will break quite easily. I used a glue gun to secure the plastic panel to the sides, top and back panel.

The removable rear door is fitted and the plastic glued on the rear door panel. The opening for the power cord can be cut after you complete the final power assembly. The door is held in place by glueing two small wooden stops on each side of the case to keep the door from falling into the case. I used two home-made latches that can be swiveled to hold the door in place.

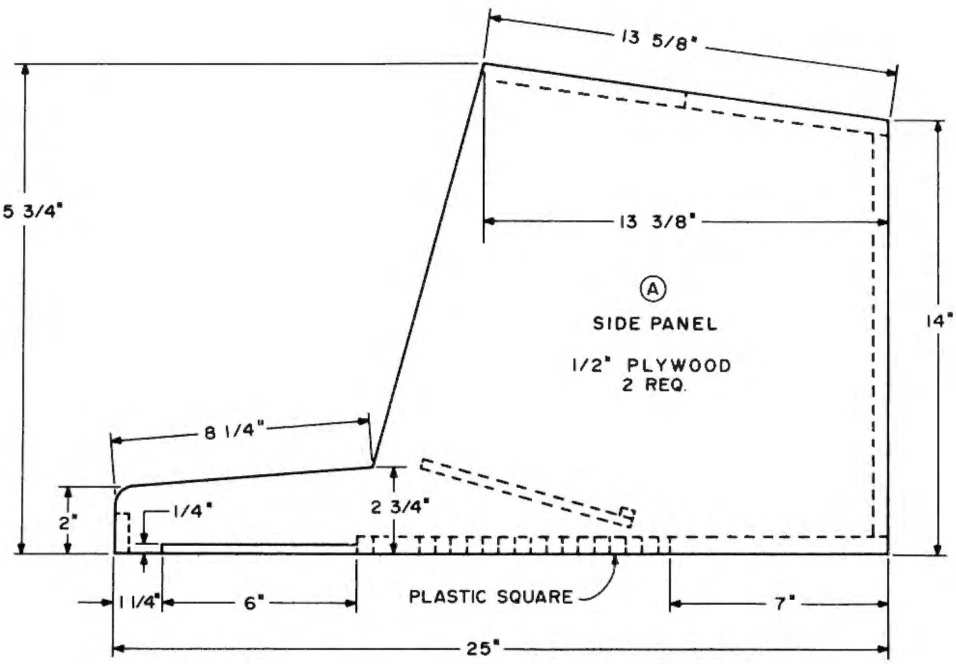
With the plastic I had left over, I cut a piece that would fit between the computer board (H) and the power panel (D) and glued it in place. This will prevent the cables from falling out when the unit is moved. If you use some other material be sure to drill holes for cooling.

The entry hole for the cassette recorder cables can be put on either side of the case, as you desire. I put mine on the right side. Drill at least a one-inch hole to accommodate the plug of the recorder. Just be sure it is below the video monitor board.

Now you are ready to paint the case. One can of flat black spray paint should be enough to paint the case. Spray all parts of the case that are visible. Once the paint is dry I took silver duct tape and trimmed the case as shown in the photo to match the TRS-80 color scheme. You could use silver paint and accomplish the same thing.

To make the case look more professional and also improve its cooling, I added four feet to each corner. I used 1/2-inch diameter feet that have a nail in the center for attaching to the wood.

The next order of business is to make provisions for all those plugs that require power. I installed a single plug and receptacle box and found when I add-

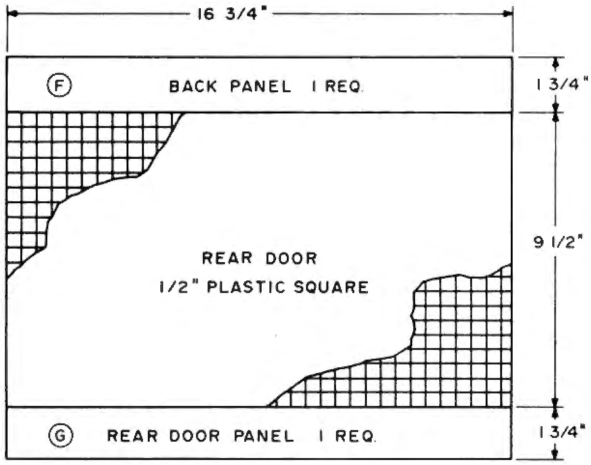


SIDE VIEW

Fig. 3a

ed an Acu-Data Tape Digitizer I was short a place to plug it in. I would highly recommend a four (or more) three-wire grounded power strip to allow for expansion. In my set-up, I have a three-wire grounded six-foot extension cord that I cut the receptacle from and used to connect to the receptacle located on the power panel. I plug the main power cable into a switched power strip so I have an on/off switch for the computer and accessories. You could add an on/off switch on the right or left rear side of the case if you desire. Just make sure it's rated large enough to handle the load. Whatever method you decide on, make sure you locate it so the video monitor doesn't obstruct its use.

That completes the basic case. However, I have my computer on a table under an overhead light. Because of the slope of the video monitor I got a glare on the screen that just had to go. To solve this problem I installed a moveable hood that can be moved in or out to remove the glare. I had some 1/4-inch scrap plastic that I cut six and 1/2 inches wide and long enough to fit over the top of the



REAR VIEW

Fig. 3b

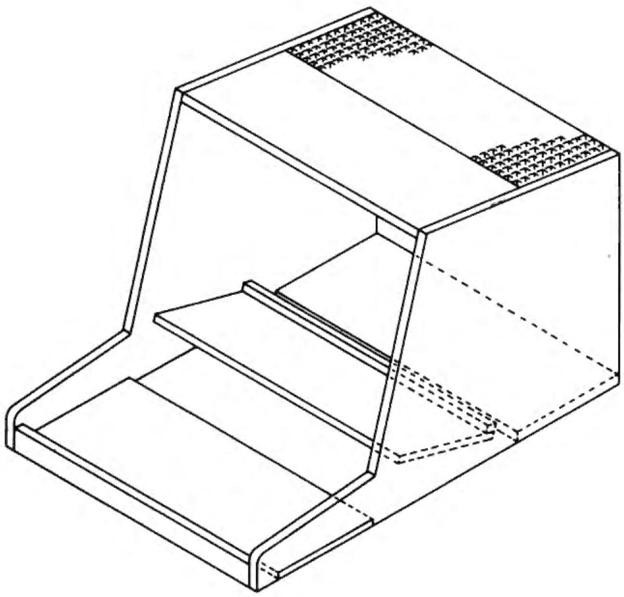
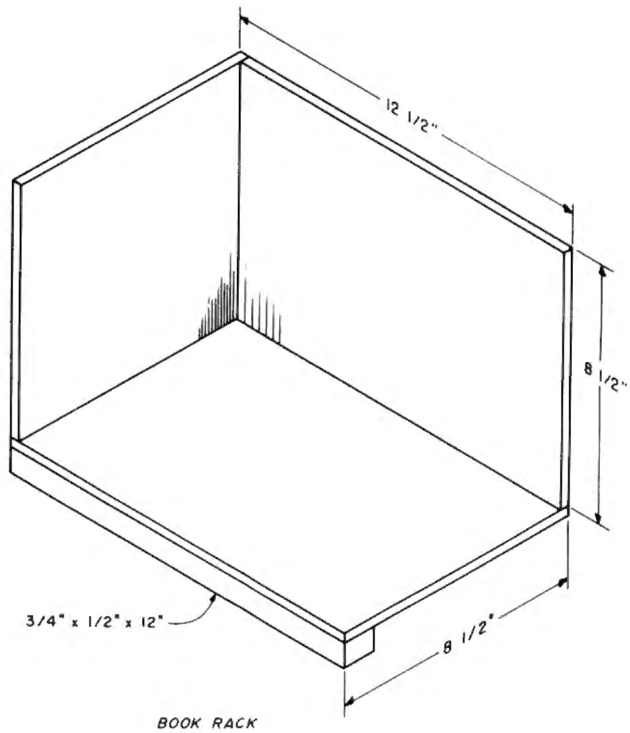


Fig. 4



BOOK RACK

Fig. 5

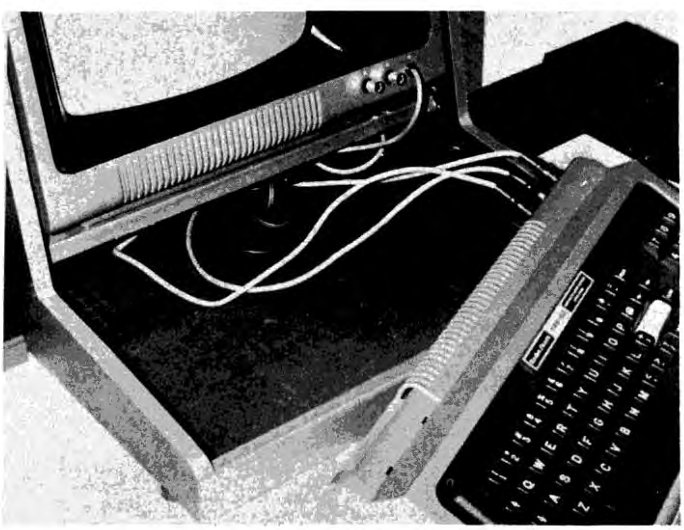


Photo 2. Make sure the video monitor and the keyboard fit properly. Note the plastic grill that supports the cables.

Photo 3. Make sure you can reach the on/off and reset switch.

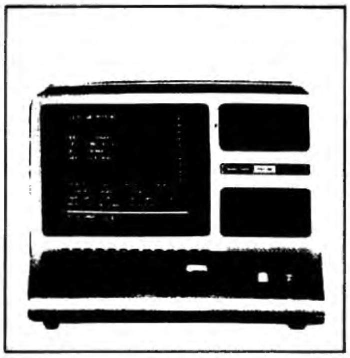
INTRODUCING:

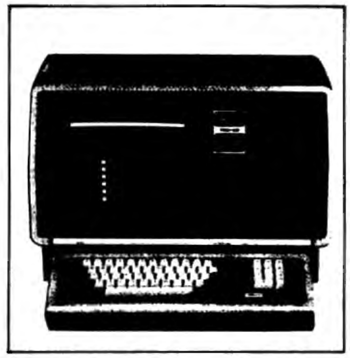
SPECTRONICS, INC.

(A Wholly Owned Subsidiary of Level IV Products)

– OFFERING –

The Full TRS-80* Line at Discount Prices, and 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 2-40 drives (E)	$1859
48K 2-80 drives (E)	$2119
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	Call
Anadex All	Call

DRIVES	COMPLETE	RAW
40 TRK MPI (E)	$309	$245
80 TRK MPI (E)	$429	$360
40/40 TRK MPI (E)	$445	$375
80/80 TRK MPI (E)	$579	$515

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.

SPECTRONICS, INC.
32461 SCHOOLCRAFT ROAD
LIVONIA, MICHIGAN 48150

MICHIGAN – (313) 525-6200
TOLL FREE – 800-521-3305

LEVEL IV PRODUCTS

14

*TRADEMARK OF TANDY CORP.

See List of Advertisers on page 322

case along with two side pieces. I used the glue gun to join them together. I then sprayed it black to match the case. To keep the hood from rubbing on the top of the case I installed two rails glued to the sides of the case. I used scrap metal rails but you could just put two pieces of wood to hold the hood above the top.

I also wanted to place all my reference books in order and yet be able to move them on a moment's notice. I made a three-sided bookcase that butts up against the left side of the case. I got a cheap metal book end that holds all the books upright and can be moved as a unit. I used plastic, again glueing the pieces together with the glue gun. You could make the bookcase out of 1/4-inch stock or similar material. Paint it black to match the case.

That completes the case, hood and bookcase. All that is needed now is to put the TRS-80 in the case and have fun. I can tell you straight out, it took longer to write this article on how to build the case than it did to build it. So don't be timid, jump right in and give your TRS-80 a professional computer set-up.■

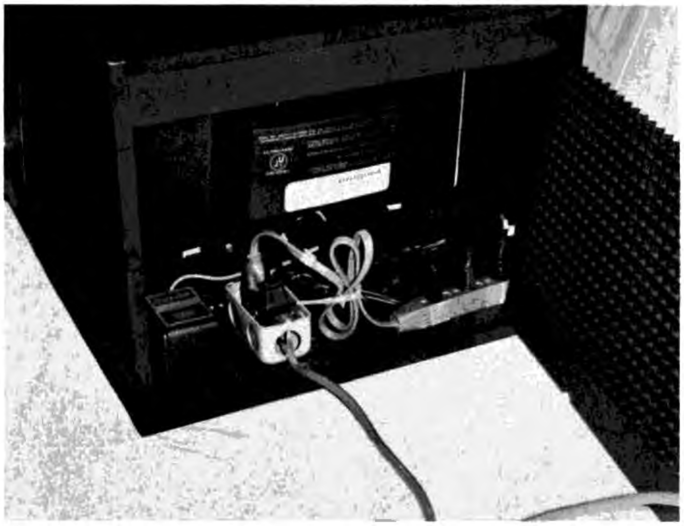


Photo 4. A rear view of the power hook-up and removable rear panel. Note the homemade latch to hold panel in place.

Photo 5. The unit is configured for easy moving with disconnecting components.

apple

Compare!

SOFTWARE for Apple II/II+

		SAVE
Language/Pascal System	$379	24%
DOS 3.3	$ 49	20%
Apple Writer	$ 59	21%
The Controller	$495	21%
Apple Pilot	$119	27%
DOS Tool Kit	$ 59	22%
Apple Fortran	$149	25%
Apple Plot	$ 49	30%
Tax Planner	$ 99	33%
Visicalc 3.2	$119	21%
Visicalc 3.3	$159	20%

Copy II Plus: Bit copy program. COPY II PLUS by Central Point Software. Will copy most protected programs for your backup in 45 seconds. $39.95

		SAVE
MICROSOFT: (all disk)		
Typing Tutor	$ 14	30%
Adventure	$ 23	24%
Fortran - 80	$149	25%
A.L.D.S.	$110	10%
BASIC Compiler	$299	25%
Olympic Decathlon	$ 19	24%
INSOFT:		
ALD System II	$110	10%
TransFORTH II	$110	10%
Accounting Software	$365	66%

A full professional quality integrated GL, A/R, A/P, Payroll package. Hot-line support available. Send for free sample printouts.

APPLE II PLUS:		SAVE
16K	$1049	22%
48K	$1099	29%
64K	$1269	27%
Apple ///		Call

All are 1981 models with Apple RAM. 64K unit is 48K unit with Microsoft 16K RAM board. 48K and 64K units include Applesoft and Integer BASIC's when used with DISK II.

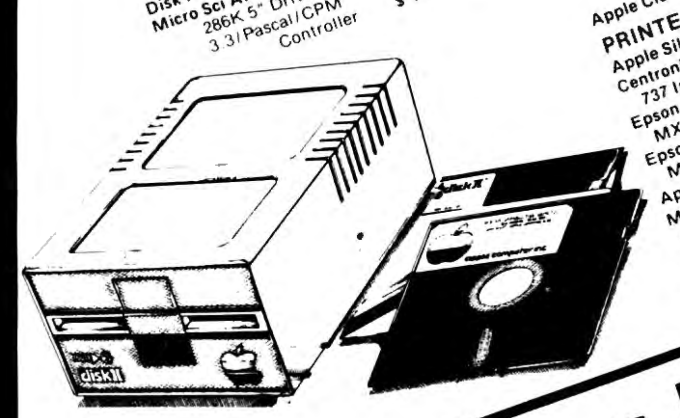
MONITORS:		SAVE
Sanyo 9" B&W	$ 169	30%
12" B&W	$ 249	21%
12" Green	$ 299	21%
13" Color	$ 449	20%
NEC 12" Green	$ 225	21%
RF Modulator, M&R	$ 25	38%
Serial/Parallel Interface, SSM AIO	$ 159	20%
Serial Interface Card, CCS	$ 129	35%
Parallel Interface Card, CCS	$ 99	37%
Apple IEEE Card	$ 339	25%
80 Column Video Cards:		
Apple Smarterm	$ 299	17%
Videx, Videoterm	$ 249	18%
Apple Clock/Calendar Card	$ 239	15%

HARDWARE for Apple II/II+

		SAVE
Disk II & 3.3 Controller	$ 499	23%
Disk II	$ 439	16%
Micro Sci A70 for Apple II 286K 5" Drive	$ 499	16%
3.3/Pascal/CPM Controller	$ 79	20%

PRINTERS:		
Apple Silentype w/Interface	$ 499	22%
Centronics 737	$ 759	26%
737 Interface & cable	$ 169	25%
Epson MX 80	$ 475	27%
MX 80 Interface & cable	$ 95	15%
Epson MX 100	Call	Call
MX100 Interface & cable	Call	Call
Apple Graphics Tablet	$695	13%
Microsoft:		
Z80 Softcard	$249	28%
16K RAM Card	$ 159	20%
Hayes Micromodem II™	$299	23%
10 Memorex 5" Diskettes	$ 25	45%
10 Maxell 5" Diskettes	$ 39	33%

NO SALES TAX



514

Computer Exchange

P O BOX 1380 Jacksonville, OR 97530

TOLL FREE (800)547-1289

TECHNICAL & OREGON: (503) 899-7297

Above prices for mail orders only. Our store showroom is 126 N.E. "F" St., Grants Pass, OR. Store prices, which include software service, differ from mail order prices. No mail order sales at store. CALL ORDER DESK.

ORDERING INFORMATION: Minimum order $100. Money Orders, Cashier Checks or Bank Wires Welcomed. Visa and MC orders add 3%. Personal checks accepted (allow up to 20 days to clear). Add 3% for shipping, handling and insurance. UPS ground is standard. Add 4% total for UPS Blue or 10% total for US Parcel Post. Include your telephone number. No COD's. Prices subject to change without notice. Order desk hours are 9 to 6 PDT, 10 to 3 Saturdays.

References: We have been an Apple dealer since 1978. Our bank reference is First Interstate Bank (503) 776-5620. We belong to the Chamber of Commerce (503) 772-6293.

apple computer Sales and Service

IMPORTANT NOTE:
We are an authorized dealer and repair center and will repair all Apple equipment regardless of where you purchased it, in or out of warranty. Normally our turnaround time on repairs is 24 hours. Call before sending equipment.

VISA master charge

Save.

See List of Advertisers on page 322

Loop around this brain teaser for awhile.

Digits for Fun

Ross A. Wirth
15906 E 96th St. N
Owasso, OK 74055

Learning should be fun, right? *Right!* I present to you a brain teaser—one that can be solved by a computer much easier than by human trial and error. The approach you take to solve this brain teaser, though, is what this article is all about.

Nine Digits of Fun

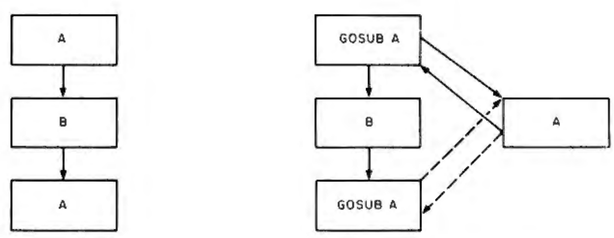
Construct a nine-digit number using the digits one through nine, using each digit only once. The nine-digit number you construct should be evenly divisible by nine. To make it a bit more difficult, the first eight digits must be a multiple of eight, the first seven digits a multiple of seven, and so on. For example:

123654
1 is a multiple of one (nothing new here)
12 is a multiple of two
123 is a multiple of three
1236 is a multiple of four
12365 is a multiple of five
123654 is a multiple of six

And so on up to nine.

Of the possible 362,880 combinations of nine digits, how many meet the above criteria? That's for you to find out.

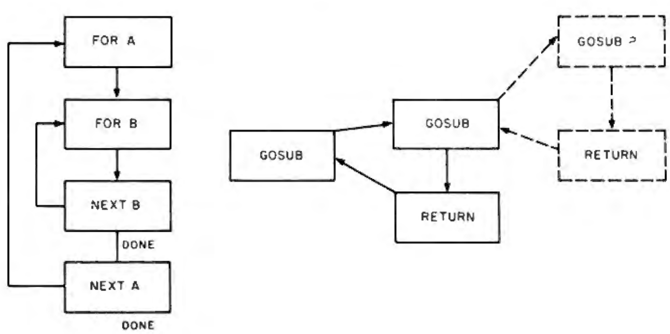
Now, put down the magazine and write a program to solve the nine-digit problem. If you are a genius you might try to solve it mentally. However, if you aren't, let your TRS-80 give it a try.



TOP DOWN FLOW

SUBROUTINE CALLS TO "A"

Figure 1



NESTED LOOPS

RECURSIVE CALLS TO ITSELF

Figure 2

```
1 REM SIMPLE EXAMPLE OF RECURSIVE CALLS TO A SUBROUTINE
5 CLS
10 INPUT"ENTER NUMBER OF LEVELS";LS
20 PRINT "START AT TOP"
30 GOSUB 100
40 PRINT "BACK AT TOP"
50 END
100 L=L+1
110 PRINT "DOWN TO LEVEL ";L
120 IF L<LS THEN GOSUB 100
130 L=L-1
140 PRINT "UP TO LEVEL ";L
150 RETURN
```

Program Listing 1

```
5 REM PROGRAMMED FUN #1 BY ROSS A. WIRTH
10 REM   TITLE -- 9 DIGITS OF FUN
15 REM   SAMPLE SOLUTION USING RECURSIVE CALLS
20 REM
30 CLS
35 DIM I(9),A(20,9)
40 L=0
45 I(0)=0
50 GOSUB 100
55 CLS
57 PRINT "THE 9 DIGIT ANSWERS ARE:"
58 PRINT
60 FOR J=1 TO AC
65   FOR L=1 TO 9
70     PRINT A(J,L);
75   NEXT L
80   PRINT
85 NEXT J
90 END
100 L=L+1
105 PRINT:PRINT"LEVEL ";L
110 I(L)=0
115 I(L)=I(L)+1
120   REM TEST TO SEE IF DIGIT IS ALREADY USED
130   C=0
135   FOR J=1 TO L:PRINT I(J);:NEXT J:PRINT
140   FOR J=0 TO (L-1)
150     IF I(L) = I(J) THEN C=C+1
160   NEXT J
170   IF C>0 THEN 330
175  REM TEST TO SEE IF PROPERLY DIVISIBLE
180   R=0
190   FOR J=1 TO L
200     R=(R*10) + I(J)
205     R=R - L
210     IF R > 0 THEN 205
215     IF R < 0 THEN R=R + L
220   NEXT J
230   IF R > .5 THEN 330
250   REM X IS NOW READY TO GO DOWN 1 MORE LEVEL
270   IF L=9 THEN 310
280   GOSUB 100
290   GOTO 330
300   REM NEW ANSWER FOUND
310   AC=AC+1
315   FOR II=1 TO 9
320     A(AC,II)=I(II)
325   NEXT II
330 IF I(L) < 9 THEN 115
340 L=L-1 :REM BACK OUT OF THE RECURSION
345 PRINT:PRINT "LEVEL ";L
350 RETURN
```

Program Listing 2

L	level of recursion (1 through 9)
I(9)	digits being tested
A(20,9)	answers
C	counter
AC	answer counter
J	loop counter (local)
R	remainder in divisability test

Table 1. Variables in Program Listing 2.

Lines	
5-45	initialization
50	first call
55-90	printing of solution
100	set level counter
105-115	initiate action for current level
120-170	test to see if the digit is already used (GOTO 330 if test fails)
175-230	test to see if "X" if a multiple of the level (GOTO 330 if test fails)
250	success at this level—let's go down one more
270	if already at level 9 an answer has been found
280	down one more level (GOSUB)
290	on the way back up loop around answer routine
300-325	new answer saved for later printing
330	if a possible digit still is untested at this level, let's go to 115 and try it out
340-350	all possible digits have been tried at this level. Let's go up a level and try some more at that level.

Note: Ideally lines 100, 330 and 340 would be part of a FOR-NEXT loop. The problem is that Level II does not allow a subscripted variable to be used as a loop counter in a FOR-NEXT loop. Thus, the longer format is used here.

Table 2. Logic description of Program Listing 2

Is There a Better Way?

The most common approach to solving the nine-digit problem is to write a series of nested For. . . Next loops. Each loop will check the usage of a digit more than once and the even divisibility of the number built to that point. Note: do both of these tests eight times (the first time, they are not needed).

Is there a better way? Sure, anytime you repeat a step, the common instructions can be linked into a subroutine. The execution continues along the same logic but it is easier to write. It's also easier to modify because you need to change the code only once, rather than as many times as the program references the same section of code. Fig. 1. illustrates the use of subroutines. You can do the same thing in our program; each set of tests can be referenced via a subroutine call.

There is also a way to simplify the common structure of the nested For. . . Next loops, recursion; we can have the subroutine call itself.

Fig. 2 shows what is meant by a subroutine calling itself. Enter Program Listing 1 into your TRS-80 and give it a try. As your program descends into itself, it prints the current level. All the program needs to know is how far down it is at the time and when it's supposed to stop and return to the top.

"The approach you take to solve this brain teaser, though, is what this article is all about."

How far down can you go? Originally, I thought that the limit would be some stack pointer internal to the Z-80 CPU. It appears, though, that the limit is the amount of available memory! With my Level II 16K, a line printer driver in upper memory and Program Listing 1, I can go down 3059 levels and back in a little over four minutes and down 3060 levels before I run out of memory. If you haven't entered and run Program Listing 1, you should do it now.

Now, back to the nine-digit problem. How can you use recursion to reduce the amount of program coding? Program Listing 2 shows the code I wrote to solve this brain teaser.

The run time can be significantly reduced by leaving out the prints that show the level and number being checked. For those interested in technical nature of this program, Table 1 shows the variable usage in Program Listing 2. Table 2 explains in detail the program logic.

Of the 362,880 possible combinations of nine digits, there is only one solution that meets the divisibility criteria. In a test run, some counters show that the duplicate digit usage test stops 2790 tries toward an answer. 1580 attempts to form a solution were stopped by failing the divisibility test. Thus, you can see that the tests drastically reduce the number of tries that actually get to the ninth level.

12-Digits of Fun

If you think you understand recursive programming, test your skills with the 12-digit problem described in Fig. 3. It seems a bit more complicated on the surface, but is really no more difficult than the nine-digit problem. The program I wrote to solve this brain teaser is Listing 3. But, try to solve it on your own before looking at the next page. ■

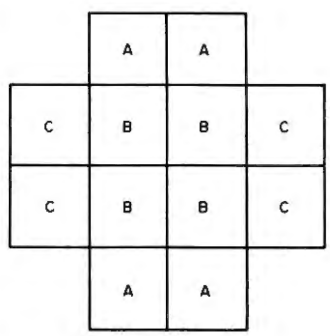
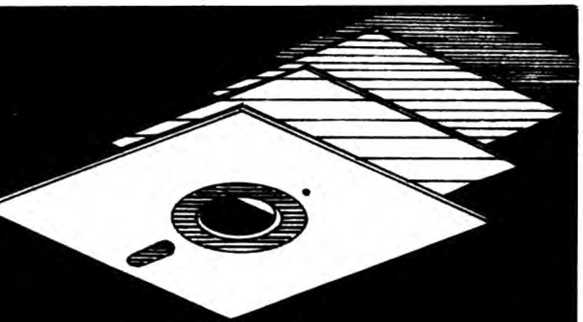


Fig. 3

Program Listing 3

```
5 REM PROGRAMMED FUN #1 BY ROSS A. WIRTH
10 REM   TITLE -- 12 DIGITS OF FUN
15 REM   SAMPLE SOLUTION USING RECURSIVE CALLS
20 REM
25 AN=26
30 L=0:F2$="##":F3$="###":C=0:AC=0:TC=0:J=0:X$="    ":X=
   0:Y=0
35 DIM I(12),A(220,12),SP(12)
40 CLS
45 I(0)=0
47 GOSUB 400:REM SETUP SCREEN
50 GOSUB 100
55 CLS
57 PRINT "THE 12 DIGIT ANSWERS ARE:"
58 PRINT
60 FOR J=1 TO TC
61   L=J
62   IF L>220 THEN L=L-220:GOTO 62
63   PRINT J;":  ";
65   FOR L=1 TO 12
70     PRINT A(J,L);
75   NEXT L
80   PRINT:PRINT
85 NEXT J
87 PRINT TC;" ANSWERS"
90 END
100 L=L+1
105 PRINT@7," ";:PRINT USING F3$;L;
110 I(L)=0
115 I(L)=I(L)+1
117 GOSUB 175:PRINT@SP(L)," ";:PRINT USING F2$;I(L);
118 IF I(L)>12 THEN 333
120   REM TEST TO SEE IF DIGIT IS ALREADY USED
130   C=0
140   FOR J=0 TO (L-1)
150     IF I(L) = I(J) THEN C=C+1
160   NEXT J
170   IF C>0 THEN 330
173 GOTO 270
175 REM TEST TO SEE HOW "26" IS FORMED
180 ON L GOTO 250,250,250,190,250,250,200,205,250,215,2
    50,220
190 X=I(1)+I(2)+I(3)+I(4)-AN:GOSUB 700:GOTO 250
200 X=I(2)+I(3)+I(6)+I(7)-AN:GOSUB 700:GOTO 250
205 X=I(5)+I(6)+I(7)+I(8)-AN:GOSUB 700
210 X=I(1)+I(4)+I(5)+I(8)-AN:GOSUB 700:GOTO 250
215 X=I(9)+I(2)+I(6)+I(10)-AN:GOSUB 700:GOTO 250
220 X=I(11)+I(3)+I(7)+I(12)-AN:GOSUB 700
230 X=I(9)+I(10)+I(11)+I(12)-AN:GOSUB 700
250 RETURN
270   IF L=12 THEN 310
280   GOSUB 100
290   GOTO 330
300   REM NEW ANSWER FOUND
310   AC=AC+1:TC=TC+1
312   IF AC>220 THEN AC=1
313 PRINT@71," ";:PRINT USING F3$;AC;
315   FOR II=1 TO 12
320     A(AC,II)=I(II)
325   NEXT II
330 IF I(L) < 12 THEN 115
333 I(L)=0
335 PRINT@SP(L)," ";:PRINT USING F2$;I(L);
340 L=L-1 :REM BACK OUT OF THE RECURSION
345 PRINT@7," ";:PRINT USING F3$;L;
350 RETURN
400 REM SETUP SCREEN
410 DATA 283,475,667,859,294,486
420 DATA 678,870,464,497,656,689
430 FOR I=1 TO 12:READ SP(I):NEXT I
440 DATA A1,B2,B3,A4,A5,B6,B7,A8,C9,C10,C11,C12
450 FOR I=1 TO 12
460   READ X$
470   PRINT@(SP(I)-67),X$;
475 NEXT I
480 FOR X=24 TO 113
490   FOR Y=7 TO 43 STEP 9
495     IF (X>45)AND(X<92) THEN 510
500       IF (Y=7)OR(Y=43) THEN 520
510     SET(X,Y)
520   NEXT Y
530 NEXT X
540 FOR Y=8 TO 42
550   FOR X=24 TO 112 STEP 22
555     IF (Y>15)AND(Y<35) THEN 570
560       IF (X<46)OR(X>91) THEN 580
570     SET(X,Y):SET(X+1,Y)
580   NEXT X
590 NEXT Y
600 PRINT@0,"LEVEL";
610 PRINT@64,"ANSWERS   0";
620 RETURN
700 IF X=0 THEN 750
710 IF X>0 THEN I(L)=13:GOTO 750
720 X=-X
730 IF I(L)>X THEN I(L)=13 ELSE I(L)=I(L)+X
750 RETURN
```



UTILITY FORCE

$29.95

THE VIDEO MANAGER **FIRST IN A SERIES**

Our Video Manager will enhance your computer's performance with these features: vertical printing on screen; forms design direct from screen to printer; send entire screen or any part to your printer; save screen to disk as a file and more. Written so you can utilize it from DOS or BASIC. Available on disk for: TRS-80,* Mod I TRS DOS 2.3, Mod II TRS DOS 1.2 & 2.0, CP/M 2.0 or later, soon available for Mod III. For your convenience with Visa or M/C phone in your order. Please add $2 for handling and postage.

*™Tandy Corp.

SOFTWAREHOUSE INTERNATIONAL
5070 N. Sixth St. Suite 103B
Fresno, CA 93710 (209) 221-7877

TO RECEIVE A FREE CATALOG PLEASE ✓13

STOP...

. . . spending more time altering programs than you do enjoying them!

Get . . .

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 orgional 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 to:

BREEZE COMPUTING, INC., P.O. Box 1013, Berkley, MI 48072. (313) 288-9422. For $2.50 Breeze Computing, Inc. will send every owner, upon registration of Super Utility, one back up copy. *Foreign orders, please add $5.00 additional postage charge. Michigan residents add 4% sales tax. ✓76

THE PROGRAMMER'S GUILD MEANS ADVENTURE!!

SPIDER MOUNTAIN

SEARCH FOR GOLD AND GLORY IN THE TUNNELS OF SHELOB'S LAIR.

TAPE $14.95 **DISK $19.95**

LOST DUTCHMAN'S GOLD

YOU AND THE GHOST OF BACKPACK SAM AGAINST THE TREACHEROUS TERRAIN AND HOSTILE INDIANS OF THE SUPERSTITION MOUNTAINS.

TAPE $14.95 **DISK $19.95**

LOST SHIP

DISCOVER THE SECRET OF THE FLOATING PIRATE VESSEL IN THE MYSTERIOUS DEVIL'S TRIANGLE.

TAPE $14.95 **DISK $19.95**

DEATH DREADNAUGHT

CAN YOU ESCAPE THE CREATURE'S DEADLY PURSUIT AND MAKE IT TO THE SHUTTLE ALIVE? RATED **R** DUE TO VIOLENT DESCRIPTIONS.

TAPE $14.95 **DISK $19.95**

DRAGONQUEST

RESCUE THE PRINCESS BEFORE SUNDOWN FROM SMAEGOR THE TERRIBLE. MACHINE CODE, KEY—BOARD MACRO.

TAPE $15.95 **DISK $21.95**

THUNDER ROAD

HILARIOUS TAKEOFF ON THE "DUKES OF HAZARD" CAN YOU GET YOUR MOONSHINE TO KNAWBONE BEFORE SHERIFF BUBBA SHUTS YOU DOWN?

TAPE $14.95 **DISK $19.95**

DEADLY DUNGEON

DISCOVER THE SECRET OF THE DEADLY DUNGEON AND EMERGE VICTORIOUS AND WEALTHY.

TAPE $14.95 **DISK $19.95**

★ ★ ★ **NEW RELEASE** ★ ★ ★

TEMPLE OF THE SUN

EXPLORE AN ANCIENT AZTEC TEMPLE SEARCHING FOR THE SHAMAN'S SECRET.

TAPE $14.95 **DISK $19.95**

COMPUTER MAIL ORDER IS HERE!!!

THE Programmer's Guild ™ ✓364

P O Box 66 • Peterborough NH 03458 • 603-924-6065 Voice • 603-924-7920 Data

A place for everything.

All About Sorts—Part I

Len Gorney
Box 91 R.D. 5
Clarks Summit, PA 18411

Sorting is the process by which items are arranged into some kind of order depending upon a particular characteristic. The white pages of a telephone book, for example, are ordered alphabetically according to a person's last name; mailing labels are sorted according to their respective Zip Codes; student standings are arranged by grade point averages. In fact, most everything is sorted or arranged into some order.

My primary focus will be on some of the more commonplace and useful sorting techniques. The techniques I will examine fall into the general class of "internal" sorting routines, since the items to be sorted must reside in the main storage of the computer. External sorts are another subject, not covered here.

Choosing the Sort

The sorting process follows a fundamental sequence: Look at an item's key; determine the position of the item in a sorted list; and move the item to its proper position. A particular sorting method is chosen by a number of conditions involving time and space. Time is that limiting factor which tells us the speed of the sorting process. Space refers to the amount of memory required by the procedure.

The number and initial distribution of the items to be sorted is our first concern. The number of comparisons and exchanges made during the execution of the sort also bear on the choice. To determine the best sorting method we must consider the following:

- Number of items to be sorted.
- Initial distribution of the items.
- Storage requirements of the method.
- Number of comparisons.
- Number of exchanges.
- Total time spent in execution.

We will sort items which consist of a single "key" element by which the item is sorted (a single integer value). The set of items to be sorted is placed into an array (A) by simply reading integer values into the locations dimensioned for this array.

The main program common to all sorting techniques sets up the list of items to be sorted, calls the sorting subroutine and outputs the list of items after they have assumed their sorted order. The following variables and their descriptions will generally hold true for all the programmed routines.

Variable Name	Description
A	Array that holds the items to be sorted.
C	Number of comparisons.
E	Number of exchanges.
J, K, L	General pointers into the array A.
N	Number of items to be sorted.
P	Number of passes through the list.
T	Temporary location holding a particular item.

Program Listing 1

```
11000 DIM A(10), S(10)
1010 N = 10
1020 FOR X = 1 TO 4
1030   CLS:
       PRINT"ORIGINAL ITEMS ";
1040   FOR K = 1 TO N
1050     READ A(K):
         PRINT A(K);:
         S(K) = -999
1060   NEXT K
1070   GOSUB 1170
1080   C = 0:
       E = 0:
       GOSUB 1180
1090  CLS:
       PRINT"SORTED ITEMS  ";
1100   FOR K = 1 TO N
1110     PRINT S(K);
1120   NEXT K
1130   PRINT:
       PRINT"COMPARISONS =";C;"EXCHANGES =";E:
       GOSUB 1170
1140 NEXT X
1150 END
1160 DATA 0, 1, 2, 3, 4, 5, 6, 7, 8, 9:
     DATA 9, 8, 7, 6, 5, 4, 3, 2, 1, 0:
     DATA 8, 0, 9, 4, 3, 5, 7, 2, 6, 1:
     DATA -7, 3, -9, -7, 0, 6, 6, 4, 5, -1
1170 FOR T = 1 TO 1000:
     NEXT T:
     RETURN
1180 '
      LINEAR SELECTION

1190 FOR P = 1 TO N
1200   J = 0
1210   J = J + 1
1220 '
      CHECK FOR DUMMY VALUE OR END OF LIST

1230   IF A(J) = -999 THEN IF J = N THEN 1370
                                  ELSE 1210
1240 '
      PUT ASSUMED SMALLEST ITEM POINTED TO BY A(J) INTO
      TEMPORARY STORAGE (T).  K POINTS TO WHERE THIS ASSUMED
      SMALLEST ITEM RESIDES.

1250    T = A(J):
        K = J
1260 '
      SEARCH FOR A SMALLER ITEM THAN THE ONE AT A(J).

1270    FOR L = 1 TO N
1280 '
      SKIP DUMMY ITEM.

1290      IF A(L) = -999 THEN 1340
1300 '
      COMPARE A VALID ITEM AGAINST ASSUMED SMALLEST ITEM.
1310        C = C + 1:
```

Program continues

Linear Selection

Linear Selection treats the list of items to be sorted as a single succession of elements. A comparison is made by selecting the first item and then advancing from one

```
PASS # 1 ORIGINAL     SORTED
     8                  0
   -999               -999
     9                -999
     4                -999
     3                -999
     5                -999
     7                -999
     2                -999
     6                -999
     1                -999
PASS # 2 ORIGINAL     SORTED
     8                  0
   -999                 1
     9                -999
     4                -999
     3                -999
     5                -999
     7                -999
     2                -999
     6                -999
   -999               -999
         •
         •
         •
PASS # 10 ORIGINAL    SORTED
   -999                 0
   -999                 1
   -999                 2
   -999                 3
   -999                 4
   -999                 5
   -999                 6
   -999                 7
   -999                 8
   -999                 9
SORTED ITEMS   0 1 2 3 4 5 6 7 8 9
COMPARISONS = 55 EXCHANGES = 10
```

Fig. 1

Program Listing 1 continued

```
                 IF T <= A(L) THEN 1340
1320 '
        NEW ASSUMED SMALLEST ITEM IS AT A(L).
1330             T = A(L):
                 K = L
1340     NEXT L
1350 '
        DUMMY OUT A(K) AND PUT SMALLEST ITEM IN OUTPUT LIST (S).
1360     A(K) = -999:
         S(P) = T:
         E = E + 1
1370 '
        TAKE SNAPSHOTS.
1380     CLS:
         PRINT"PASS #";P,"ORIGINAL       SORTED"
1390     FOR KK = 1 TO N
1400       PRINT, A(KK), S(KK)
1410     NEXT KK
1420     GOSUB 1170
1430 NEXT P
1440 RETURN
1450 END
```


✓See List of Advertisers on page 322

position in the list to subsequent positions.

Linear selection assumes the first item in the list to be the smallest. This is then compared with the second through the Nth items until a smaller item is found or the end of the list is reached.

If a smaller item is found, the remaining list is compared to it. If not, then the first item is placed in the first position of the output (sorted) list. The position it occupied in the original list is converted to a dummy value.

For our purposes, the dummy value will be equal to −999. This dummy is necessary to avoid comparing the original value during subsequent passes of the sorting process.

The sorting procedure continues, choosing the next available item in the list as the next smallest item, and scanning the rest of the list (skipping the dummy items) until the next smallest is found. This item is placed in the output list and its position is filled with the dummy value.

Fig. 1 is an example of linear selection sorting. Program Listing 1 is the programmed routine that produces Fig. 1.

```
ORIGINAL ITEMS   8 0 9 4 3 5 7 2 6 1

PASS # 1 ITEMS   0 8 9 4 3 5 7 2 6 1

PASS # 2 ITEMS   0 1 9 4 3 5 7 2 6 8

PASS # 3 ITEMS   0 1 2 4 3 5 7 9 6 8

PASS # 4 ITEMS   0 1 2 3 4 5 7 9 6 8

PASS # 5 ITEMS   0 1 2 3 4 5 7 9 6 8

PASS # 6 ITEMS   0 1 2 3 4 5 7 9 6 8

PASS # 7 ITEMS   0 1 2 3 4 5 6 9 7 8

PASS # 8 ITEMS   0 1 2 3 4 5 6 7 9 8

PASS # 9 ITEMS   0 1 2 3 4 5 6 7 8 9

SORTED ITEMS     0 1 2 3 4 5 6 7 8 9

COMPARISONS = 45 EXCHANGES = 7
```

Fig. 2

```
ORIGINAL ITEMS   8 0 9 4 3 5 7 2 6 1

PASS # 2 ITEMS   0 8

PASS # 3 ITEMS   0 8 9

PASS # 4 ITEMS   0 4 8 9

PASS # 5 ITEMS   0 3 4 8 9

PASS # 6 ITEMS   0 3 4 5 8 9

PASS # 7 ITEMS   0 3 4 5 7 8 9

PASS # 8 ITEMS   0 2 3 4 5 7 8 9

PASS # 9 ITEMS   0 2 3 4 5 6 7 8 9

PASS # 10 ITEMS  0 1 2 3 4 5 6 7 8 9

SORTED ITEMS     0 1 2 3 4 5 6 7 8 9

COMPARISONS = 26 EXCHANGES = 28
```

Fig. 3

The number of comparisons made during any linear selection sort is equal to N^2: It does not matter if the items are already sorted in random order, or in reverse order. Since each item on the original list is transferred to an output list, the number of exchanges is equal to N. Extra storage requirements for the linear selection sort will also be equal to N; that is, the size of the original list must be duplicated for the output list.

Linear Selection with Exchange

Linear selection with exchange differs from the previous technique in that the final sorted items reside in the original list rather than in a separate output list. This saves a substantial amount of memory, namely, N locations that were required for the separate output. The linear selection with exchange sort is known as an in-place sorting process because the items are sorted in the original list.

The search for the smallest item in the list is the same as described above. At the end of each pass, however, the item, held in temporary storage (T), exchanges place with the first item in the original list. Subsequent passes begin with the next position, then the third, etc., since all previous positions contain previously sorted items. Fig. 2 represents the relevant locations given by the program in Program Listing 2.

The number of comparisons, regardless of distribution, is equal to $N^2/2$. The number of exchanges is equal to the number of passes throught the list, or N. There is no extra storage required. This alone makes the linear selection with exchange sort a particularly useful process. Also, the number of comparisons is halved, which increases its speed. One minor disadvantage of this sort is that the coding is a bit more complex.

Linear Insertion

An insertion sort orders items as they are selected. In other words, an insertion sort is quite similar to a manual sort: an item is selected and inserted into its proper position relative to those items already sorted. This position may be anywhere in the list of sorted items.

Linear insertion is quite useful when an already sorted list of items must be updated. For example, if a sorted mailing label list already exists and a new label arrives, this new label must be inserted into its proper position. When the new label's position is determined, a place is made for it by simply bumping forward the previously sorted items whose keys are greater than the new label's key.

Linear insertion operates in the following manner: The first item of the list is assumed to be the smallest and is placed at the beginning of the output list. The second item is then compared with the single item already in the sorted list. If the second item is smaller than the first, an exchange is made between the two. Otherwise, the second item is simply transferred to the second position of the output list. Subsequent items are inserted into their proper positions in the sorted list. When an item must be placed between two already sorted items, items that are larger than this new item are bumped forward on the list by one location. Program Listing 3 is the routine that outputs samples given in Fig. 3.

The efficiency of this sort may be increased by sorting items in-place; that is, the original list becomes the sorted list after the routine runs to completion. Pro-

Program Listing 2

```
1000 DIM A(10)
1010 N = 10
1020 FOR X = 1 TO 4
1030   CLS:
       PRINT"ORIGINAL ITEMS   ";
1040   FOR K = 1 TO N
1050     READ A(K):
         PRINT A(K);
1060   NEXT K
1070   GOSUB 1170
1080   C = 0:
       E = 0:
       GOSUB 1180
1090   PRINT:
       PRINT"SORTED ITEMS     ";
1100   FOR K = 1 TO N
1110     PRINT A(K);
1120   NEXT K
1130   PRINT:
       PRINT"COMPARISONS =";C;"EXCHANGES =";E:
       GOSUB 1170
1140 NEXT X
1150 END
1160 DATA 0, 1, 2, 3, 4, 5, 6, 7, 8, 9:
     DATA 9, 8, 7, 6, 5, 4, 3, 2, 1, 0:
     DATA 8, 0, 9, 4, 3, 5, 7, 2, 6, 1:
     DATA -7, 3, -9, -7, 0, 6, 6, 4, 5, -1
1170 FOR T = 1 TO 1000:
     NEXT T:
     RETURN
1180 '
        LINEAR SELECTION WITH EXCHANGE

1190 FOR P = 1 TO N-1
1200 '
        PUT ASSUMED SMALLEST ITEM POINTED TO BY A(P) INTO
        TEMPORARY STORAGE (T).  WHEN K = 0, ASSUMED SMALLEST
```

Program continues

Program continued

```
              ITEM IN A(P) IS, IN FACT, THE SMALLEST ITEM.
1210     T = A(P):
         K = 0
1220
          SEARCH FOR A SMALLER ITEM THAN THE ONE AT A(P).

1230     FOR L = P+1 TO N
1240       C = C + 1:
           IF T >= A(L) THEN T = A(L):
                             K = L
1250     NEXT L
1260 '
          SMALLEST ITEM IS AT A(K) IF K NOT EQUAL TO ZERO, ELSE
          SMALLEST ITEM IS AT A(P).

1270     IF K >< 0 THEN T = A(P):
                        A(P) = A(K):
                        A(K) = T:
                        E = E + 1
1280 '
          TAKE SNAPSSHOTS.

1290       PRINT:
           PRINT"PASS #";P;"ITEMS    ";
1300       FOR KK = 1 TO N
1310         PRINT A(KK);
1320       NEXT KK:
           GOSUB 1170
1330 NEXT P
1340 RETURN
1350 END
```

```
1000 DIM A(10), S(10)
1010 N = 10
1020 FOR X = 1 TO 4
1030   CLS:
       PRINT"ORIGINAL ITEMS    ";
1040   FOR K = 1 TO N
1050     READ A(K):
         PRINT A(K);:
         S(K) = -999
1060   NEXT K
1070   GOSUB 1170
1080   C = 0:
       E = 0:
       GOSUB 1180
1090   PRINT:
       PRINT"SORTED ITEMS      ";
1100   FOR K = 1 TO  N
1110     PRINT S(K);
1120   NEXT K
1130   PRINT:
       PRINT"COMPARISONS =";C;"EXCHANGES =";E:
       GOSUB 1170
1140 NEXT X
1150 END
1160 DATA 0, 1, 2, 3, 4, 5, 6, 7, 8, 9:
     DATA 9, 8, 7, 6, 5, 4, 3, 2, 1, 0:
     DATA 8, 0, 9, 4, 3, 5, 7, 2, 6, 1:
     DATA -7, 3, -9, -7, 0, 6, 6, 4, 5, -1
1170 FOR T = 1 TO 1000:
     NEXT T:
     RETURN
1180 '
        LINEAR INSERTION

1190 '
        INSERT FIRST ITEM INTO SORTED LIST.
        INITIALIZE END OF SORTED LIST POINTER (SP).

1200 S(1) = A(1):
     SP = 2
1210 FOR P = 2 TO N
1220   J = 1
1230 '
        FIND NEXT ITEM'S PROPER POSITION IN SORTED LIST.

1240   C = C + 1:
       IF A(P) < S(J) THEN 1270
1250     J = J + 1:
         IF J >< SP THEN 1240
                    ELSE S(SP) = A(P):
                         E = E + 1:
                         GOTO 1310
1260 '
        BUMP UP ITEMS TO MAKE ROOM FOR INSERTION OF NEW ITEM.

1270   FOR K = SP TO J+1 STEP -1
1280     S(K) = S(K-1):
         E = E + 1
1290   NEXT K
1300   S(K) = A(P)
1310   SP = SP + 1
1320 '
        TAKE SNAPSHOTS.

1330   PRINT:
       PRINT"PASS #";P;"ITEMS    ";
1340   FOR KK = 1 TO SP-1
1350     PRINT S(KK);
1360   NEXT KK:
       GOSUB 1170
1370 NEXT P
1380 RETURN
1390 END
```

Program Listing 3

gram Listing 4 shows the code for this revised routine. Fig. 4 is the sample output from the linear insertion with exchange sorting method.

Bubble Sort

The bubble sort can be compared to the activity that takes place within a glass of carbonated beverage; the lighter (smaller) bubbles (items) tend to rise to the top (beginning) of the glass (list), while the heavier (larger) bubbles (items) tend to remain at the bottom (end) of the glass (list).

The bubble sort selects an item and exchanges it with larger items on its way up the list. When it meets a smaller item, it stops "bubbling"; that is, the searching process for a new smaller item begins again with a different item. Another item is selected and the bubbling continues until all the items have bubbled to their proper positions in the list.

The bubble sort uses a flag (the variable F$) which allows the number of passes through the list to be variable rather than fixed. When the value of the flag is true (F$ = T) after a completed pass, this signals the computer that no exchanges were made during this pass; therefore, the items are in sorted order.

When the variable F$ is false, an exchange has been made and a subsequent pass is necessary.

The sample in Fig. 5 is taken from Program Listing 5. It should be readily apparent that the smaller items bubble up the list. Follow a particular item as it travels from its original position to its final postion in the sorted list.

Under average, that is, random condi-

```
ORIGINAL ITEMS    8 0 9 4 3 5 7 2 6 1
PASS # 1 ITEMS    8 0 9 4 3 5 7 2 6 1
PASS # 2 ITEMS    0 8 9 4 3 5 7 2 6 1
PASS # 3 ITEMS    0 8 9 4 3 5 7 2 6 1
PASS # 4 ITEMS    0 4 8 9 3 5 7 2 6 1
PASS # 5 ITEMS    0 3 4 8 9 5 7 2 6 1
PASS # 6 ITEMS    0 3 4 5 8 9 7 2 6 1
PASS # 7 ITEMS    0 3 4 5 7 8 9 2 6 1
PASS # 8 ITEMS    0 2 3 4 5 7 8 9 6 1
PASS # 9 ITEMS    0 2 3 4 5 6 7 8 9 1
PASS # 10 ITEMS   0 1 2 3 4 5 6 7 8 9
SORTED ITEMS      0 1 2 3 4 5 6 7 8 9
COMPARISONS = 36 EXCHANGES = 27
```

Fig. 4

```
ORIGINAL ITEMS    8 0 9 4 3 5 7 2 6 1
PASS # 1 ITEMS    0 8 1 9 4 3 5 7 2 6
PASS # 2 ITEMS    0 1 8 2 9 4 3 5 7 6
PASS # 3 ITEMS    0 1 2 8 3 9 4 5 6 7
PASS # 4 ITEMS    0 1 2 3 8 4 9 5 6 7
PASS # 5 ITEMS    0 1 2 3 4 8 5 9 6 7
PASS # 6 ITEMS    0 1 2 3 4 5 8 6 9 7
PASS # 7 ITEMS    0 1 2 3 4 5 6 8 7 9
PASS # 8 ITEMS    0 1 2 3 4 5 6 7 8 9
SORTED ITEMS      0 1 2 3 4 5 6 7 8 9
COMPARISONS = 45 EXCHANGES = 27
```

Fig. 5

Program Listing 4

```
1000 DIM A(10)
1010 N = 10
1020 FOR X = 1 TO 4
1030   CLS:
       PRINT"ORIGINAL ITEMS    ";
1040   FOR K = 1 TO N
1050     READ A(K):
         PRINT A(K);
1060   NEXT K
1070   GOSUB 1170
1080   C = 0:
       E = 0:
       GOSUB 1180
1090   PRINT:
       PRINT"SORTED ITEMS      ";
1100   FOR K = 1 TO N
1110     PRINT A(K);
1120   NEXT K
1130   PRINT:
       PRINT"COMPARISONS =";C;"EXCHANGES =";E:
       GOSUB 1170
1140 NEXT X
1150 END
1160 DATA 0, 1, 2, 3, 4, 5, 6, 7, 8, 9:
     DATA 9, 8, 7, 6, 5, 4, 3, 2, 1, 0:
     DATA 8, 0, 9, 4, 3, 5, 7, 2, 6, 1:
     DATA -7, 3, -9, -7, 0, 6, 6, 4, 5, -1
1170 FOR T = 1 TO 1000:
     NEXT T:
     RETURN
1180 '
       LINEAR INSERTION WITH EXCHANGE.

1190 FOR P = 1 TO N
1200 '
```

Program continues

Program continued

```
          TEMPORATY STORAGE (T) HOLDS ITEM TO INSERT.

1210   T = A(P)
1220 '
          FIND POSITION WHERE THIS ITEM ISS TO BE INSERTED.

1230   FOR K = P-1 TO 1 STEP -1
1240     C = C + 1
1250     IF T >= A(K) THEN A(K+1) = T:
                          GOTO 1290
                     ELSE A(K+1) = A(K):
                          E = E + 1
1260   NEXT K
1270 '
          ITEM IN TEMPORARY STORAGE IS THE SMALLEST ITEM IN LIST.

1280   A(1) = T
1290 '
          TAKE SNAPSHOTS.

1300   PRINT:
       PRINT"PASS #";P;"ITEMS    ";
1310   FOR KK = 1 TO N
1320     PRINT A(KK);
1330   NEXT KK:
       GOSUB 1170
1340 NEXT P
1350 RETURN
1360 END
```

```
1000 DIM A(10)
1010 N = 10
1020 FOR X = 1 TO 4
1030   CLS:
       PRINT"ORIGINAL ITEMS    ";
1040   FOR K = 1 TO N
1050     READ A(K):
         PRINT A(K);
1060   NEXT K
1070   GOSUB 1170
1080   C = 0:
       E = 0:
       GOSUB 1180
1090   PRINT:
       PRINT"SORTED ITEMS      ";
1100   FOR K = 1 TO N
1110     PRINT A(K);
1120   NEXT K
1130   PRINT:
       PRINT"COMPARISONS =";C;"EXCHANGES =";E:
       GOSUB 1170
1140 NEXT X
1150 END
1160 DATA 0, 1, 2, 3, 4, 5, 6, 7, 8, 9:
     DATA 9, 8, 7, 6, 5, 4, 3, 2, 1, 0:
     DATA 8, 0, 9, 4, 3, 5, 7, 2, 6, 1:
     DATA -7, 3, -9, -7, 0, 6, 6, 4, 5, -1
1170 FOR T = 1 TO 1000:
     NEXT T:
     RETURN
1180 '
       SHELLSORT.

1190 '
       DISTANCE PARAMETER (D) DETERMINATION.

1200 D = INT ( N / 2 ):
     PP = 0
1210 FOR P = 1 TO N-D
1220 '
       COMPARE THE J(TH) ITEM AGAINST THE ITEM D DISTANCE AWAY.

1230  J = P:
      EE = 0
1240  C = C + 1:
      IF A(J) <= A(J+D) THEN 1290
1250 '
       EXCHANGE ITEMS.

1260    T = A(J):
        A(J) = A(J+D):
        A(J+D) = T:
        J = J - D:
        E = E + 1:
                        EE = E
1270 '
       LOOP BACK TTO COOPARE AS LONG AS POINTER J IS VALID.

1280    IF J > 0 THEN 1240
1290 '
       TAKE SNAPSHOTS.

1300     PP = PP + 1:
          PRINT:
          PRINT"PASS #";PP;"ITEMS    ";
1310      FOR KK = 1 TO N
1320        PRINT A(KK);
1330      NEXT KK
1340     GOSUB 1170
1350 NEXT P
1360 D = INT ( D / 2 ):
     IF EE = 0 THEN 1380
1370 IF D > 0 THEN 1210
1380 RETURN
1390 END
```

Program Listing 5

tions of distribution, there will be $N^2/4$ comparisons and exchanges during the course of a bubble sort. A worst case condition (reverse initial order) will require $N^2/2$ comparisons and exchanges. The minimum number of exchanges for the best case (items already sorted) will be zero, while the minimum number of comparisons will be equal to N-1.

Though the bubble sort represents a substantial increase in sorting efficiency of those methods examined so far, the bubble sort is not the ultimate sorting technique. The methods that we have examined represent some of the most easily coded sort routines available. However, efficiency will increase and coding will become more involved as we venture further into more sophisticated sorting techniques.■

Next month: Shell sort, quick sort, vector sort, and merge sort routines.

✓See List of Advertisers on page 322

LOAD80
Armed and Ready

This month begins a column devoted to our exciting LOAD80 project. LOAD80, designed to save you time and effort by eliminating the need to type in the long programs featured in the magazine, has been enthusiastically received. This column will bring you news about LOAD80. We hope to provide you with an index to the programs which will appear on each month's tape. Also, there will be readers' comments, tape loading hints, suggestions and other valuable information about LOAD80.

LOAD80 tapes are recorded at a cassette baud rate of 500. Many of our readers who own Model IIIs have asked if there will be a 1500 baud version of LOAD80. At this time, there are no plans to offer this tape at the high baud rate. Mod. III owners should select the 'L' cassette baud rate before trying to load LOAD80. (No, we are not planning to offer a disk or a stringy floppy version either, sorry.)

Pgm #	Filename	Page #	Comments
1	FRKNSTN	92	None
2	TANK	104	None
3	SPCEMP	106	None
4	STRGURD	116	None
5	RACE	122	None
6	EPEXCR	126	None
7	EPSLCT	126	None
8	LASTTEST	136	None
9	MORSE	150	None
10	MESSAGE	150	None
11	DATGEN/SRC	168	Needs EDTASM
12	VIDSPC	180	None
13	RELSHP	192	None
14	VCESYN	208	None
15	SRTSOFSR	214	None
16	DISAS	240	None
17	PHODEX	248	None
18	SPRUSR/SRC	256	Needs EDTASM
19	DIRPROG	270	None
20	STRTRK	86	None
21	CMDFIL	278	None

The tape duplication of the LOAD80 cassettes is handled by Instant Software, with expert technicians working long hours to produce quality tapes. There always seem to be a few worms in every apple crate, however, and you may receive a LOAD80 tape that won't load. Here's what you should do:

1. Pull the Ear plug out of your cassette recorder and listen to the tape signal. If it wavers, or isn't there at all, you've got a bad tape.
2. If the signal sounds a little muted, not sharp and clear, you may need to have your tape recorder heads aligned.
3. If the tape is bad, return it for replacement to:

LOAD80
Attn: Laurie Eaton
80 Pine Street
Peterborough, NH 03458
(603) 924-6532

Due to the overwhelming response to our LOAD80 offering, delivery of the tapes does not begin until the middle of the month in which the magazine appears. So look for your August LOAD80 tape after the 15th of the month. Our Circulation Department is putting together a subscription plan for LOAD80. Look for details this fall in *80 Microcomputing*.

Sorry; the two source code listings of Gil Spencer's program in our July issue are not available to LOAD80 subscribers.

If you have any thoughts about LOAD80 you'd like to share, send them to this column at the above address. ■

80 microcomputing bookshelf

—BASIC & PASCAL—

new

● **PASCAL**—BK1188—by Paul M. Chirlian. Professor Chirlian's textbook combines a simple approach to the PASCAL language with comprehensive coverage on how a computer works, how to use a flowchart, working from a terminal as well as batch operation and debugging. Special attention is paid to idiosyncrasies of the language and syntax flowcharts abound for the convenience of the experienced programmer. Well indexed. $12.95*

● **INTRODUCTION TO TRS-80 LEVEL II BASIC AND COMPUTER PROGRAMMING**—BK1219—by Michael P. Zabinski. Written by an experienced educator, this is the book for those beginners who want to learn about computers without having to become an expert. It has practical programs, useful line-by-line comments, excellent flowcharts accompanied by line numbers and over 200 exercises which help the reader assess progress, reinforce comprehension, and provide valuable practical experience. $10.95.*

● **50 BASIC EXERCISES**—BK1192—by J. P. Lamoitier. This book is structured around the idea that the best way to learn a language is through actual practice. It contains 50 completely explained exercises: statement and analysis of the problem, flowcharts, programs and actual runs. Program subjects include mathematics, business, games, and operations research and are presented in varying levels of difficulty. This format enables anyone to learn BASIC rapidly, checking their progress at each step. $12.95*

● **THE BASIC HANDBOOK**—BK1174—by David Lien. This book is unique. It is a virtual ENCYCLOPEDIA of BASIC. While not favoring one computer over another, it explains over 250 BASIC words, how to use them and alternate strategies. If a computer does not possess the capabilities of a needed or specified word, there are often ways to accomplish the same function by using another word or combination of words. That's where the HANDBOOK comes in. It helps you get the most from your computer, be it a "bottom-of-the-line" micro or an oversized monster. $14.95.*

● **LEARNING LEVEL II**—BK1175—by David Lien. Written especially for the TRS-80, this book concentrates on Level II BASIC, exploring every important BASIC language capability. Updates are included for those who have studied the Level I User's Manual. Sections include: how to use the Editor, dual cassette operation, printers and peripheral devices, and the conversion of Level I programs to Level II. $15.95.*

● **BASIC BASIC (2ND EDITION)**—BK1026—by James S. Coan. This is a textbook which incorporates the learning of computer programming using the BASIC language with the teaching of mathematics. Over 100 sample programs illustrate the techniques of the BASIC language and every section is followed by practical problems. This second edition covers character string handling and the use of data files. $10.50.*

● **ADVANCED BASIC**—BK1000—Applications, including strings and files, coordinate geometry, area, sequences and series, simulation, graphing and games. $10.75*

● **SIXTY CHALLENGING PROBLEMS WITH BASIC SOLUTIONS (2nd Edition)**—BK1073—by Donald Spencer, provides the serious student of BASIC programming with interesting problems and solutions. No knowledge of math above algebra required. Includes a number of game programs, as well as programs for financial interest, conversions and numeric manipulations. $6.95.*

● **INTRODUCTION TO PASCAL**—BK1189—by Rodnay Zaks. A step-by-step introduction for anyone wanting to learn the language quickly and completely. Each concept is explained simply and in a logical order. All features of the language are presented in a clear, easy-to-understand format with exercises to test the reader at the end of each chapter. It describes both standard PASCAL and UCSD PASCAL, the most widely used dialect for small computers. No computer or programming experience is necessary. $14.95.*

● **PROGRAMMING IN PASCAL**—BK1140—by Peter Grogono. The computer programming language PASCAL was the first language to embody in a coherent way the concepts of structured programming, which has been defined by Edsger Dijkstra and C.A.R. Hoare. As such, it is a landmark in the development of programming languages. PASCAL was developed by Niklaus Wirth in Zurich; it is derived from the language ALGOL 60 but is more powerful and easier to use. PASCAL is now widely accepted as a useful language that can be efficiently implemented, and as an excellent teaching tool. It does not assume knowledge of any other programming language; it is therefore suitable for an introductory course. $12.95.*

—GAMES—

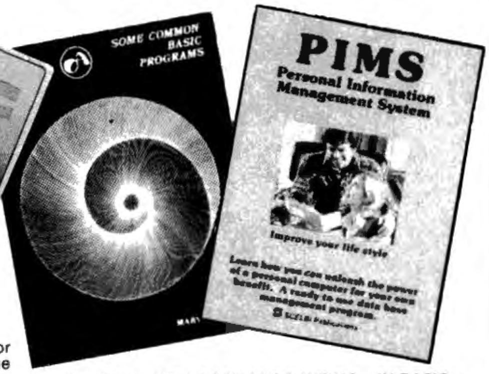
● **40 COMPUTER GAMES**—BK7381—Forty games in all in nine different categories. Games for large and small systems, and even a section on calculator games. Many versions of BASIC used and a wide variety of systems represented. A must for the serious computer gamesman. $7.95*

● **BASIC COMPUTER GAMES**—BK1074—Okay, so once you get your computer and are running in BASIC, then what? Then you need some programs in BASIC, that's what. This book has 101 games for you from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. $7.50.*

● **MORE BASIC COMPUTER GAMES**—BK1182—edited by David H. Ahl. More fun in BASIC! 84 new games from the people who brought you *BASIC Computer Games*. Includes such favorites as Minotaur (battle the mythical beast) and Eliza (unload your troubles on the doctor at bargain rates). Complete with game description, listing and sample run. $7.50.*

● **WHAT TO DO AFTER YOU HIT RETURN**—BK1071—PCC's first book of computer games . . . 48 different computer games you can play in BASIC . . . programs, descriptions, many illustrations. Lunar Landing, Hammurabi, King, Civel 2, Qubic 5, Taxman, Star Trek, Crash, Market, etc. $10.95.*

—BUSINESS—



● **PAYROLL WITH COST ACCOUNTING—IN BASIC**—BK1001—by L. Poole & M. Borchers, includes program listings with remarks, descriptions, discussions of the principle behind each program, file layouts, and a complete user's manual with step-by-step instructions, flowcharts, and simple reports and CRT displays. Payroll and cost accounting features include separate payrolls for up to 10 companies, time-tested interactive data entry, easy correction of data entry errors, job costing (labor of distribution), check printing with full deduction and pay detail, and 16 different printed reports, including W-2 and 941 (in **CBASIC**). $20.00.*

● **SOME COMMON BASIC PROGRAMS**—BK1053—published by Adam Osborne & Associates, Inc. Perfect for non-technical computerists requiring ready-to-use programs. Business programs, plus miscellaneous programs. Invaluable for the user who is not an experienced programmer. All will operate in the stand-alone mode. $14.99 paperback.

● **PIMS: PERSONAL INFORMATION MANAGEMENT SYSTEM**—BK1009—Learn how to unleash the power of a personal computer for your own benefit in this ready-to-use data-base management program. $11.95.*

—MONEYMAKING—

● **HOW TO MAKE MONEY WITH COMPUTERS**—BK1003—In 10 information-packed chapters, Jerry Felsen describes more than 30 computer-related, money-making, high profit, low capital investment opportunities. $15.00.*

● **HOW TO SELL ANYTHING TO ANYBODY**—BK7306—According to *The Guinness Book of World Records*, the author, Joe Girard, is "the world's greatest salesman." This book reveals how he made a fortune—and how you can, too. $2.25.*

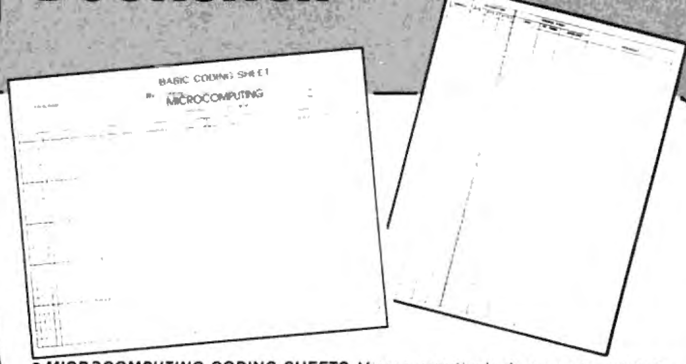
● **THE INCREDIBLE SECRET MONEY MACHINE**—BK1178—by Don Lancaster. A different kind of "cookbook" from Don Lancaster. Want to slash taxes? Get free vacations? Win at investments? Make money from something that you *like* to do? You'll find this book essential to give you the key insider details of what is really involved in starting up your own money machine. $5.95.*

*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 $1.00 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



—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.*

● **VOL. I COMPONENT TESTERS**—LB7359— . . . how to build transistor testers (8), diode testers (3), IC testers (3), voltmeters and VTVMs (9), ohmmeters (8 different kinds), inductance (3), capacity (9), Q measurement, crystal checking (6), temperature (2), aural meters for the blind (3) and all sorts of miscellaneous data on meters . . . using them, making them more versatile, making standards . . . invaluable book. $4.95.*

● **VOL. II AUDIO FREQUENCY TESTERS**—LB7360— . . . jam packed with all kinds of audio frequency test equipment. If you're into SSB, RTTY, SSTV, etc., this book is a must for you . . . a good book for hi-fi addicts and experimenters, too! $4.95.*

● **VOL. III RADIO FREQUENCY TESTERS**—LB7361—Radio frequency waves, the common denominator of Amateur Radio. Such items as SWR, antenna impedance, line impedance, rf output and field strength; detailed instructions on testing these items includes sections on signal generators, crystal calibrators, grid dip oscillators, noise generators, dummy loads and much more. $4.95.*

● **VOL. IV IC TEST EQUIPMENT**—LB7362—Become a troubleshooting wizard! In this fourth volume of the 73 TEST EQUIPMENT LIBRARY are 42 home construction projects for building test equipment to work with your ham station and in servicing digital equipment. Plus a cumulative index for all four volumes for the 73 TEST EQUIPMENT LIBRARY. $4.95.*

● ***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

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.

● **OPERATING SYSTEMS: CONCEPTS AND PRINCIPLES**—BK1193—Presents an overview of the basic operating system types, their components and capabilities. $7.95.*

● **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.*

—6502—

*● **THE APPLE II USER'S GUIDE**—BK1220—by Lon Poole, Martin McNiff, and Steven Cook. This guide is the key to unlocking the full power of your Apple II or Apple II Plus. Topics include: "Applesoft and Integer BASIC Programming"—especially how to make the best use of Apple's sound, color and graphics capabilities, "Machine Level Programming," "Hardware Features"—which covers the disk drive and printer, and "Advanced Programming"—describing high resolution graphics techniques and other advanced applications. Well organized and easy to use. $15.00.*

● **PROGRAMMING THE 6502 (Third Edition)**—BK1005—Rodnay Zaks has designed a self-contained text to learn programming, using the 6502. It can be used by a person who has never programmed before, and should be of value to anyone using the 6502. The many exercises will allow you to test yourself and practice the concepts presented. $12.95.*

● **6502 APPLICATIONS BOOK**—BK1006—Rodnay Zaks presents practical-application techniques for the 6502 microprocessor, assuming an elementary knowledge of microprocessor programming. You will build and design your own domestic-use systems and peripherals. Self-test exercises included. $12.95.*

● **6502 ASSEMBLY LANGUAGE PROGRAMMING**—BK1176—by Lance A. Leventhal. This book provides comprehensive coverage of the 6502 microprocessor assembly language. Leventhal covers over 80 programming examples from simple memory load loops to complete design projects. Features include 6502 assembler conventions, input/output devices and interfacing methods, and programming the 6502 interrupt system. $16.99.*

● **6502 SOFTWARE GOURMET GUIDE AND COOKBOOK**—BK1055—by Robert Findley. This book introduces the BASIC language programmer into the realm of machine-language programming. The description of the 6502 structure and instruction set, various routines, subroutines and programs are the ingredients in this cookbook. "Recipes" are included to help you put together exactly the programs to suit your taste. $12.95.*

—8080/8080A—6800—

* ● **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.*

* ● **68000 MICROPROCESSOR HANDBOOK**—BK1216—by Gerry Kane. Whether you're currently using the 68000, planning to use it, or simply curious about one of the newest and most powerful microprocessors, this handbook has all the answers. A clear presentation of signal conversions, timing diagram conventions, functional logic, three different instruction set tables, exception processing, and family support devices provides more information about the 68000 than the manufacturer's data sheets. A stand alone reference book which can also be used as a supplement to *An Introduction to Microcomputers Vol. 2—Some Real Microprocessors*. $6.99.*

● **8080A/8085 ASSEMBLY LANGUAGE PROGRAMMING**—by Lance Leventhal—BK1004—Assembly language programming for the 8080A/8085 is explained with a description of the functions of assemblers and assembly instructions, and a discussion of basic software development concepts. Many fully debugged, practical programs are included as is a special section on structured programming. $15.99.*

● **8080 SOFTWARE GOURMET GUIDE AND COOKBOOK**—BK1102—If you have been spending too much time developing simple routines for your 8080, try this new book by Scelbi Computing and Robert Findley. Describes sorting, searching, and many other routines for the 8080 user. $12.95.*

*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 $1.00 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

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.*

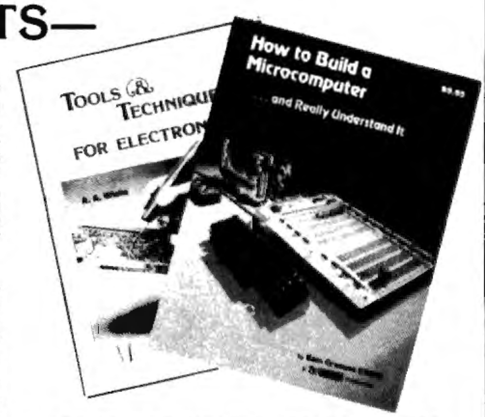
• **AN INTRODUCTION TO MICROCOMPUTERS, VOL. 0**—BK1130—The Beginner's Book—Written for readers who know nothing about computers—for those who have an interest in how to use computers—and for everyone else who must live with computers and should know a little about them. The first in a series of 4 volumes, 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. $22.50.* *Available end of August.*

• **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.*

• **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.

• **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.*

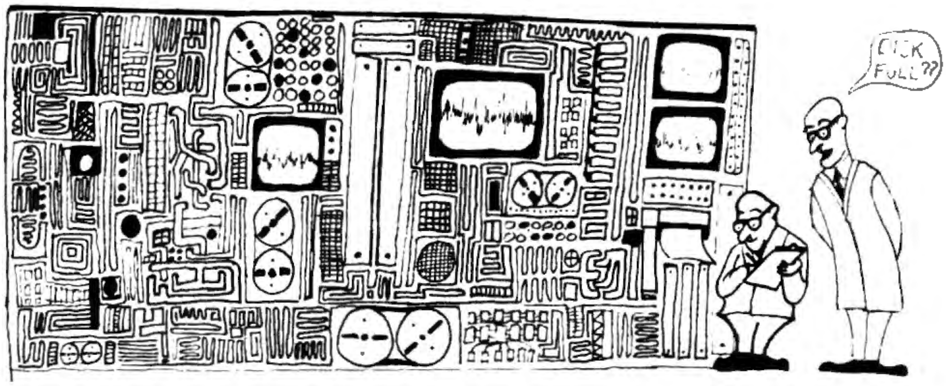
• **HOW TO BUILD A MICROCOMPUTER—AND REALLY UNDERSTAND IT**—BK7325—by Sam Creason. The electronics hobbyist who wants to build his own microcomputer system now has a practical "How-To" guidebook. This book is a combination technical manual and programming guide that takes the hobbyist step-by-step through 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 $1.00 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)

MPI Disk Drives

Model B/51 250K, 1 Side, 40 Tracks. **$321.00**
B/52 500K, 2 Sides, 40 Tracks per side. $439.00
B/91 500K, 1 side, 80 Tracks. $439.00
B/92 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)

Here's just a few of the features of MPI Disk Drives

- 40/80 Tracks
- 5 ms track-to-track
- Auto-eject
- Hi-Temp stability
- Speed constant ‹1½%
- Single/double density
- Optical sensors—no switches
- 102K per disk

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

Verbatim	10 for **$31.00**
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	**$139.00**
TRSDOS Manual	**$5.95**
Super Utility	**$49.00**
Make 80	**$14.00**

TRS-80

16K Model III	**$899.00**
48K Model III w/2 Disks	**$1900.00**
Expansion Interface OK	**$274.00**

Printers

Centronics 737-1	**$815.00**
IDS 460	**$1219.00**
Epson MX80	**$550.00**
Okidata M80	**$499.00**
Microline 82	**$650.00**
Microline 83	**$950.00**

Okidata prices include tractors

NEC Spinwriter 5530 (freight collect)	**$2579.00**

16K Memory Kits

Prime NEC 200ns dynamic RAM. Comes with complete instructions	**$39.00**
MPI Service Manual	**$3.00**
MPI Engineering Manual	**$30.00**

MIDWEST COMPUTER PERIPHERALS® ✓8

1467 S. MICHIGAN
CHICAGO, ILLINOIS 60605

TRS 80 TANDY CORP

FREE! UPS GROUND SHIPPING ON ALL ORDERS OVER $100.00

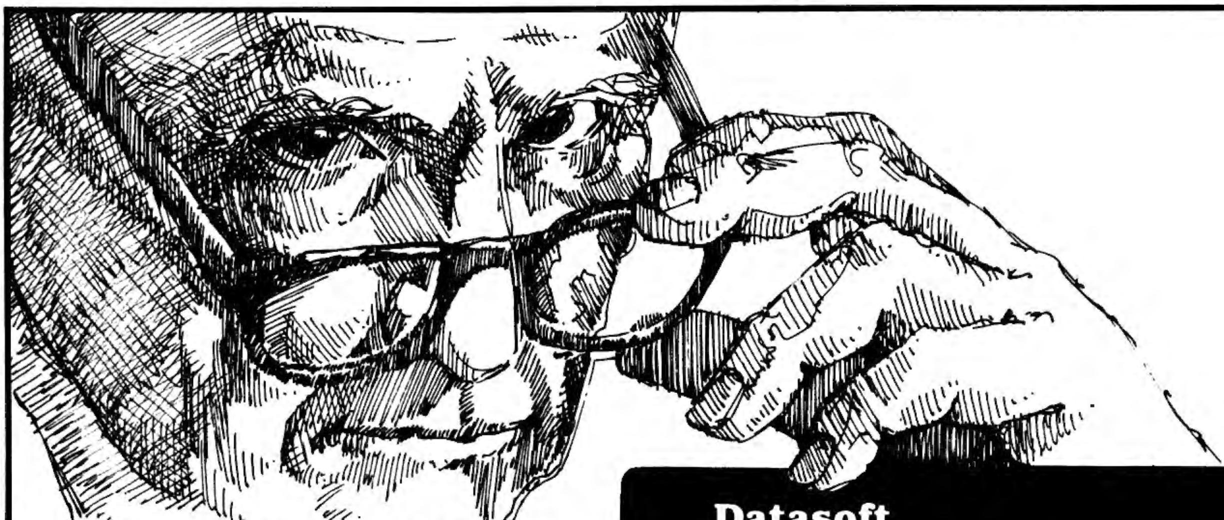
Quantity	Description	$ each	Total
		6% IL Tax	
		TOTAL	

(Minimum order $50.00)

☐ Check enclosed
Bill my ☐ Visa ☐ Am Ex
☐ Master Charge
Acct. No. ____________ Exp. ______

Name ____________
Address ____________
City ____________ State ______ Zip ______

Not Responsible for Typographical Errors. Prices Subject to Change w/o notice



Buy By Mail and Save!

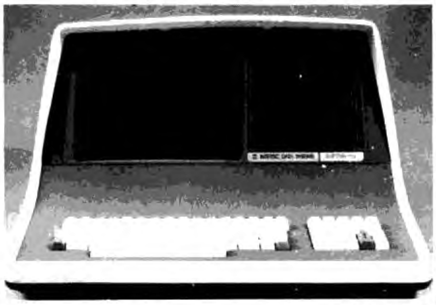
COMPUTERS

ZENITH | data systems

List $2895

OUR PRICE $2295

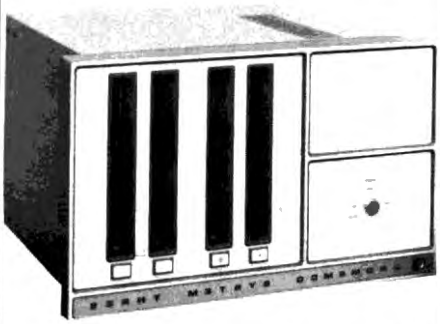
SUPERBRAIN
By INTERTEC



w/64K Dbl Density, List $3495 $2595

w/64K Quad Density, List $3995 $2995

CROMEMCO
SYSTEM THREE



64K List $7995 OUR PRICE $6395

NORTH STAR
HORIZON TWO

Horizon 2 32DD List $3095 $2289

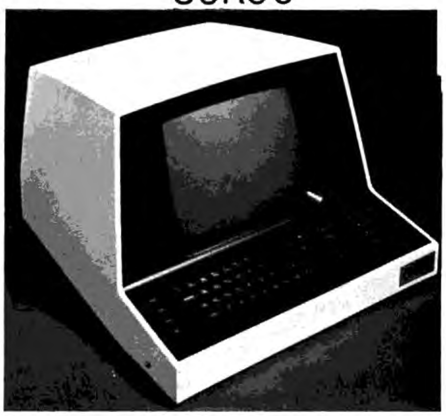
Call for latest NorthStar and Intersystem Prices

TERMINALS

TeleVideo

Televideo 912C $679
920C $799
950 .... $999

SOROC



Soroc 120 List $995 .... $689
IQ135 .... $719
140 .... $994

ZENITH Z19

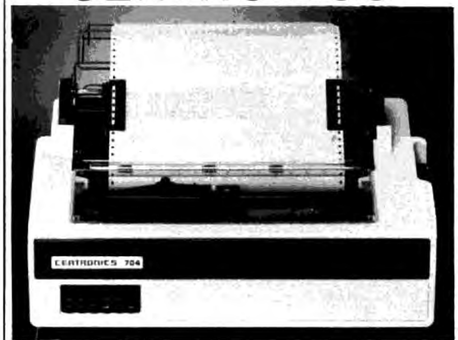
$719

INTERTUBE III

Intertube III List $895 $729

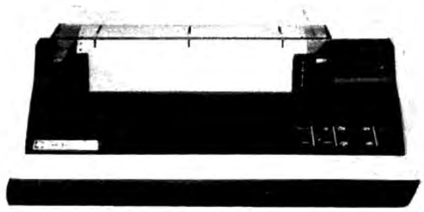
PRINTERS

CENTRONICS



739-1, List $995 .... $769
704-9 RS232 .... $1495
704-11 Parallel $1569

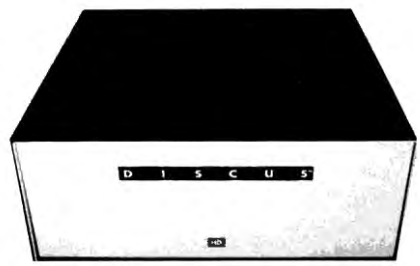
TI 810



TI 810, List $1895 $1489

DISK SYSTEMS

MORROW



Thinker Toys Discus 2D $939
Dual Discus 2D $1559
Discus 2 + 2, List $1549 $1259
M26 Hard Disk, List $4995 $3949

COMPUTERS WHOLESALE

VISA P.O. Box 144 Camillus, NY 13031 master charge

(315) 472-2582

Most items in stock for immediate delivery Factory sealed cartons w/full factory warranty NYS residents add appropriate sales tax Prices do not include shipping VISA and Master Charge add 3% COD orders require 25% deposit Prices subject to change without notice

✓24

✓See List of Advertisers on page 322

LIST OF ADVERTISERS

RS	Number	Page
452	A M Electronics	272, 273
129	Aardvark Tech Services	287
56	ABM Products	166
229	Access Unlimited	124, 125
34	Acorn Software Products	59
513	Addison Wesley Publishing Company	264
222	Adventure International	46
224	Adventure International	46
97	Adventure International	47
387	Aerocomp, Inc.	284, 285
69	Alpha Byte Stores	21, 33
210	Alpha Products Company	23
550	Alpha Products Company	25
551	Alpha Products Company	27
124	Alphanetics	41
138	Alternate Source, The	90, 158, 241, 296
396	American Business Computers	220
520	American Business Computers	220
521	American Business Computers	220
522	American Business Computers	220
523	American Business Computers	220
524	American Business Computers	221
525	American Business Computers	221
526	American Business Computers	221
527	American Business Computers	221
528	American Business Computers	221
227	American Computer Development	120
529	American Small Business Computers	219
461	Ancie Labs	261
264	Apparat, Inc.	139, 255
*	Archbold Electronics	198
445	AT-80	227
146	Audio Video Systems	160
48	Automated Simulations	129
164	Avalon Hill Game Co.	80
122	B.T. Enterprises	141, 258
*	BAP$ Software	226
201	Barstrann Corp.	67
49	Basics & Beyond, Inc.	293
186	Baysian Investment Services	246
237	BCCompco	283
351	Berg Works	210
357	Big Five Software Co.	93
76	Breeze Computing Inc.	307
448	Business Problem Solvers	160
340	COM Business Computer Systems, Inc.	80
145	C&S Electronics Mart	184
483	Cantech Data	244
62	Cecdat, Inc.	265
348	Charles Mann & Assoc.	82
32	Cload Magazine	147
502	Cload Magazine	146
459	CMD Micro	276
263	Communications Electronic	320
496	Complete Computer Services	306
100	CompuCover	113
424	Compuserve	100, 101
471	Computer Answer, The	232
107	Computer Applications Unlimited	171
107	Computer Applications Unlimited	276
199	Computer Case Company	202
443	Computer Crossroads	261
372	Computer Discount of America	64
514	Computer Exchange	303
22	Computer Information Exchange	138
130	Computer Plus	151
36	Computer Sales and Service	160
212	Computer Shopper	184
111	Computer Store, The	257
24	Computer Wholesale	321
198	Computer Wit Inc.	246
392	Computex	251
9	Computronics, Inc.	172-179
204	ComSoft	197
10	Contract Services Assoc.	201
52	Coolidge & Assoc.	216
378	Cornucopia Software	183
426	Corvus Systems	127
233	Cottage Software	140
298	CPU Shop, The	247
512	Creative Computing	232
484	Culpeper Computer Designs	105
121	Custom Electronics	226
*	Cybernetics, Inc.	218
330	Dan G. Haney & Assoc., Inc.	82
464	Dan G. Haney & Assoc., Inc.	288
540	Data Resources Corp.	17
274	Data Trans.	275
226	Digibyte Systems Corp.	211
368	Digital Systems Eng.	202, 293
440	Discount Software Group	67
30	Displayed Video	280
*	Dosware Inc.	302
425	Dow Jones Information Services	31
242	Dynamic Software	314
5	E-Z Software	281
477	EDU-Ware	306
117	EBG & Assoc.	281
174	Educational Software Library	82
510	Eigen Systems	312
	80 Microcomputing	
	Books	315, 316, 317
	Boxes	271
	Dealers	128
	Encyclopedia	114, 115
	Load80	34
	Moving	77
	Subscriptions	50, 98
	University MicroFilm	98
33	80 US Journal	157
58	Electronic Specialists	53
456	Eltech Associates	235
173	ESI Lynx	79
278	ESI Lynx	159
404	Epson America Inc.	89
166	Epson America Inc.	79
3	Exatron	CIV
500	Exatron	75
225	Fantastic Software	94
141	FEC Ltd.	144
	Fine Arts Festival	121
462	Fort Worth Computers and Video	105
39	G & L Software Enterprises	279
*	Galactic Software Ltd.	53
79	Allen Gelder Software	118, 296
75	Godbout Electronics	158
218	Good-Lyddon Data Systems	289
181	Gosub International Hardware Division	82
284	Gosub Software	269
73	Graphic Source, The	227
325	Wayne Green Books	78
223	H & H Trading Company	41
*	HW Electronics	319
125	Hexagon Systems	309
401	Holmes Engineering	227
103	Howe Software	243
37	IJG Computer Services	72, 73
467	IMB/Illustrated Memory Banks	250
176	Imprint Software	78
305	Insiders Software Consultants Inc.	225
	Instant Software	
533	Climate Computer	283
530	Game Programs	291
532	Master Directory	212
2	Programmer's Kit	98, 227, 252, 312, 313, 314
246	Interface, Inc.	90
287	Interlude	97
180	Jenson Assoc.	78
193	Joe Computer	289, 296
85	Johnson Associates	279
190	JPC Products	161
171	J & S Software	79
432	Keyline Computer Products	110
490	King Electronics	244
149	Kogyosha Company, Ltd./NIK	287
26	Laredo Systems, Inc.	133
183	Lifeboat Associates	80
391	Lindbergh Systems	204
53	LNW Research	253
15	Lobo Drives, International	CIII
35	Logical Systems Inc./LSI	69
11	Lt. Data	245
162	Macrotronics, Inc.	80
268	Maine Software	153
509	Management Services	312
87	Management Systems Software	53
167	Marck	80
128	Med. Systems Software	162, 163
421	Medfield Computer Software	88
104	Mercer Systems Inc.	140
20	Meta Technologies Corp.	11, 13, 15
542	MetaTronics Corp.	9
54	Micro Architect	281
205	Micro Blajak Systems Inc.	94
89	Micro Learningware	94
16	Micro Mainframe	269
72	Micro Management Systems Inc.	109
423	Micro Management Systems Inc.	223
68	Micro Matrix	182
29	Micro Mega	187
310	Micro Mint	227
310	Micro Mint	302
495	Micro Pro Systems	306
493	Micro Systems Software	232
384	Micro Systems Software Inc.	6, 266, 267
109	MicroWorks, The	242
476	Micro-80, Inc.	88
379	Micro-Design	302
335	Micro-Design	82
470	Micro-Grip Ltd.	235
160	Micro-Labs Inc.	82
28	Microcomputer Technology Inc./MTI	134, 135
306	Microperipheral Corp.	277
8	Midwest Computer Peripherals	318
112	Miller Microcomputer Services	95
98	Minis 'n Micros Inc.	113
221	MISOSYS	200
451	MTS Enterprises	311
144	Mumford Micro Systems	167
430	National Computer Shows	233
135	NDM Designs	41
480	Nelson Software Systems, Inc.	242
194	NEBS/New England Business Service, Inc.	108
389	Omega Sales Co.	205
367	Omikron	111
350	Omni Resources	80
105	Omnitek	276
178	On-Line Computer Telephone Directory	79
427	ONACKI	229
460	Options-80	252
296	Orange Micro Inc.	119
376	Orion Instruments	204
345	Ozark Mountain Software	78
127	PC Newsletter	271
66	PSM Inc.	241
370	Pacific Exchanges	216
370	Pacific Exchanges	144
336	Pairr Products	79
64	Pan American Electronics	83
344	Pascal/Z Users Group	80
207	Pensadyne Computer Services	210
1	Percom Data Company, Inc.	CII
429	Percom Data Company, Inc.	3
258	Percom Data Company, Inc.	3
508	Percom Data Company, Inc.	3
51	Perry Oil and Gas	207
*	Phase I Systems	297
131	Plus Computer Technology, Inc.	65
435	Precision Prototypes	145
507	Pro 80 Systems	215
408	Process Control Tech.	250
17	Program Store/Realsoft	148, 149
364	Programmer's Guild, The	307
110	Programs Unlimited	196
277	Progressive Electronics	212
441	Prosoft	186
27	Quality Software Distributors	217
269	Quant Systems	118
271	Quant Systems	91
170	RSI	78
41	Racet Computes	185
133	Radio Ranch, Inc.	259
4	Radio Shack	37
179	Radio Shack	78
446	Rak Electronics	313
492	Ram Intelligence	244
236	Rand's Inc.	155
*	Realty Software Company	67
67	Recob Service Inc.	252
463	Registry Service, The	224
70	Remsoft Inc.	143
499	Remtron	105
276	Richcraft Engineering Ltd.	67
479	RMI Software Inc.	246
143	Sales Data, Inc.	276
342	Allen Saville	82
291	Scientific Engineering Labs	224
331	Scorpion Systems	79
373	SID/Southern Innovation Design	71
19	Simutek Computer Products	199
332	Singular Systems	79
91	68 Micro Journal	226
*	Snappware	190, 191
434	Soft Sector Marketing Inc.	239
515	Soft Sector Marketing Inc.	236, 237, 238
449	Soft Tools	288
177	Softswap	80
356	Software Affair	99
299	Software Concepts	203
13	Software House International	307
329	Software House International	82
380	Software Innovations	213
454	Sound Center/Radio Shack	105
132	Spectral Associates	264
14	Spectronics, Inc.	301
275	Speedway Electronics	259
438	Stocking Source, The	83, 99
428	Street Electronics Corp.	213
82	Sturdivant & Dunn Inc.	289
150	Sublogic	206
403	Superior Software, Inc.	144
231	Synergistic Solar Inc.	121
139	Syntex Electronic Innovations	87
358	Syracuse R & D Center	243
175	System Research Services	79
148	Tab Sales Company	153
447	Tallgrass Technologies Corporation	105
161	Tandy/Radio Shack	82
45	Taranto & Associates	81
147	Task Computer Applications	130
505	Task Computer Applications	130
6	Total Access	63
162	Trinity Computer Group, The	78
444	Trinity Computer Group, The	314
18	Twenty First Century Software	193
494	Used Computer Exchange	289
*	V R Data Corporation	228
137	Van Horn Office Supplies	198
343	Votrax	82
363	Walonick Associates	246
433	Warlock Warehouse	91
450	White River Products	218

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 October 31, 1981

Please help us to bring you a better magazine—by answering these questions.

I. What is your age?
- ☐ A. under 18
- ☐ B. 18-22
- ☐ C. 23-40
- ☐ D. 41-60
- ☐ E. over 60

II. What is your occupation?
- ☐ 1. Professional
- ☐ 2. Engineer
- ☐ 3. Data processing
- ☐ 4. Business
- ☐ 5. Education
- ☐ 6. Technician
- ☐ 7. Student
- ☐ 8. Other

III. What are your primary applications of your TRS-80 (check only two)?
- ☐ A. Business
- ☐ B. Games
- ☐ C. Home
- ☐ D. Education
- ☐ E. Scientific
- ☐ F. Control
- ☐ G. Music

IV. Your TRS-80, is it a
- ☐ 1. Level I
- ☐ 2. Level II
- ☐ 3. Model II
- ☐ 4. Don't own one yet

V. What peripherals do you have (check all that apply)?
- ☐ A. Expansion interface
- ☐ B. Disk
- ☐ C. Printer

VI. How much have you spent on hardware?
- ☐ 1. less than $500
- ☐ 2. $500-1,000
- ☐ 3. $1,000-2,000
- ☐ 4. $2,000-4,000
- ☐ 5. $4,000-6,000
- ☐ 6. more than $6,000

VII. How much have you spent on software?
- ☐ A. less than $100
- ☐ B. $100-250
- ☐ C. $250-500
- ☐ D. $500-1,000
- ☐ E. more than $1,000

VIII. What is your level of education?
- ☐ 1. Post-graduate
- ☐ 2. College
- ☐ 3. High school

IX. How many people read your copy of 80 Microcomputing?
- ☐ A. 1
- ☐ B. 2
- ☐ C. 3
- ☐ D. 4 or more

X. If you are not a subscriber please circle number 500.

80 Microcomputing • POB 2743 • Clinton IA 52735

80 Microcomputing ● August 1981

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 ______

BOOKS

80 Microcomputing
Peterborough NH 03458

Please send me the following 80 Microcomputing products:

Qty.	Catalog #	Title	Unit Price	Total

Add $1 shipping/handling ______

Total ______

Enclosed $ ______ ☐ Check ☐ M.O.
Bill: ☐ AE ☐ MC ☐ VISA

Card # ______________________

Exp date ______ Interbank # ______

Signature ______________________

Name ______________________

Address ______________________

City ______________ State ______ Zip ______

Please allow 4–6 weeks for delivery

No C.O.D. orders accepted.

8-81

SUBSCRIPTION

80 Microcomputing POB 981
Farmingdale NY 11737

BEAT THE PRICE INCREASE

Subscribe by October 1, 1981

Please allow 4–6 weeks for delivery

☐ New Subscription ☐ Renewal
☐ 1 year—$18
☐ Enclosed $ ______ ☐ Check ☐ M.O.
Bill: ☐ MC ☐ VISA ☐ AE ☐ Me

Card # ______________ Exp. date ______

Signature ______________ Interbank # ______

Name ______________________

Address ______________________

City ______________ State ______ Zip ______

Canada: $20 1 year only, US funds
Other foreign: $28 one year only, US funds

318B9

All previous subscription offers are void on October 1, 1981

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 CARD

FIRST CLASS PERMIT NO. 217 CLINTON IA 52735

POSTAGE WILL BE PAID BY ADDRESSEE

80 microcomputing T.M.

POB 2743
Clinton IA 52735

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

Expand your TRS-80*

With The LOBO LX-80 Expansion Interface

Now you can realize all the power and potential of your TRS-80*, Model 1. If it's add-on memory you need, your LX-80 can accommodate up to four 5¼-inch, single- or double-density 35, 40 or 80 track mini-floppies, four 8-inch floppies (single or double sided), and up to eight Winchester fixed disk-drives (5¼". 8", 14").

LOBO's powerful new LDOS™ operating system, provided with your LX-80, allows for the use of any eight drives, in any combination, single or double density.

And there's more ... lots more. There are two parallel ports (standard), two serial ports (optional), a keyboard ROM override switch, and a 32K memory expansion (optional). Send for a free LX-80/TRS-80 cost performance comparison chart.

For the full story on how the LX-80 can expand your TRS-80, see your nearest LOBO dealer, or write or call:

*TRS-80 is a registered trademark of Radio Shack, a Tandy Company.

LOBO drives INTERNATIONAL

LOBO DRIVES INT'L
354 South Fairview Ave.
Goleta, CA 93117
(805) 683-1576

✓15

See us at the N.C.C., booth #4211