APRIL 1983
ISSUE NUMBER 56

PRICE U.S. $2.95
U.K. £1.25

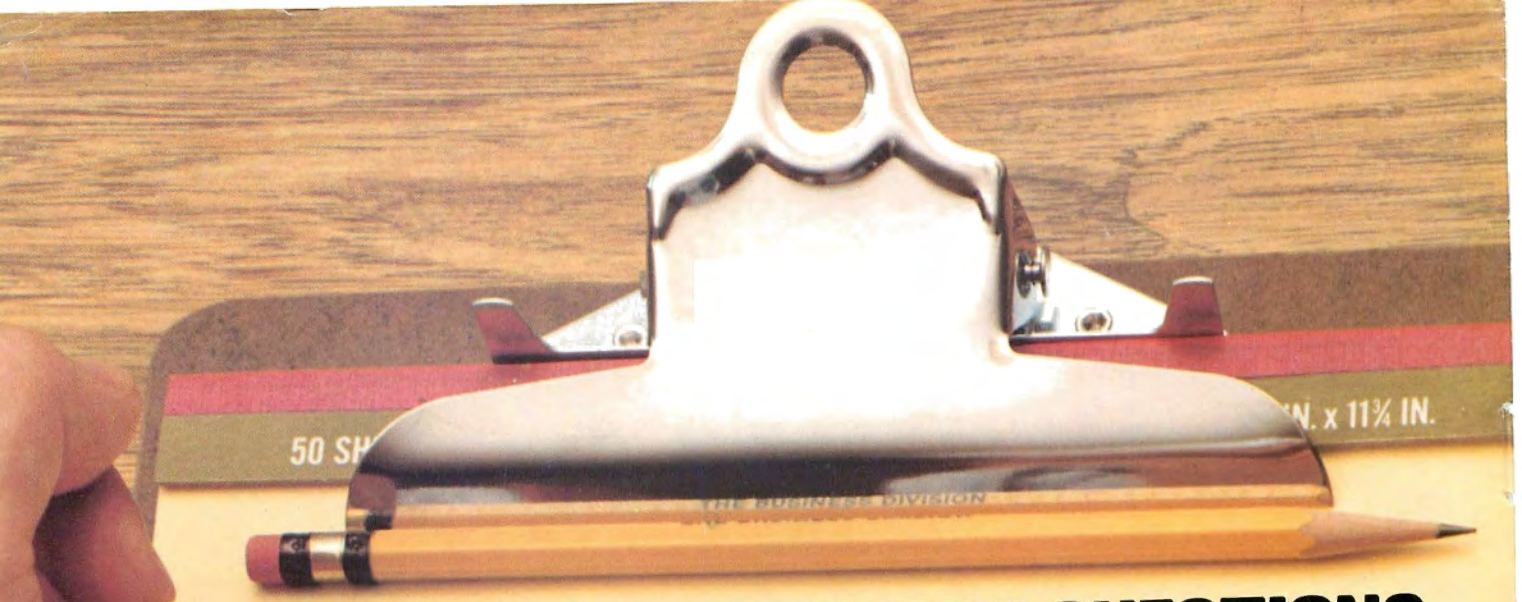
THE ORIGINAL MAGAZINE FOR TRS-80™* OWNERS

H&E COMPUTRONICS INC.

Cover Photo by Harry Peterson
H&E COMPUTRONICS INC.
50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

*TRS-80™ IS A TRADEMARK OF TANDY CORPORATION

U.S. POSTAGE
STANDARD
BULK RATE
Permit #58
New City, N.Y. 10956
FORWARDING & RETURN POSTAGE GUARANTEED



PLEASE ANSWER THESE 5 QUESTIONS

	YES	NO
1. Do you have information that must be kept organized and accessible?	☐	☐
2. Do you ever need to perform statistical analyses?	☐	☐
3. Would you like to have a tool that will allow you greater flexibility in managing your own or your company's money?	☐	☐
4. Could you use a mail list program that will be easy to use, maintain an unlimited number of names, allow you great sorting flexibility, and even interact with a word processor?	☐	☐
5. Do you own or have access to a TRS-80 microcomputer?	☐	☐

If you answered "yes" to any of these questions, we can be of assistance to you. Our Maxi Series of applications programs are designed to give you maximum versatility with a minimum of hassle. These programs were created for business use, but you'll find yourself using them for personal applications as well.

Thorough support — Of course, each program comes with in-depth, user-oriented documentation, and is menu-driven to make it easy to use. When necessary, the Maxi programs are compatible with each other, and, whenever pertinent, are interactive with the major word processing and spreadsheet programs published by other manufacturers. Also, we maintain a telephone support line to provide you with any assistance you might require.

Maxi Manager *by Dale Kubler*

Maxi Manager is a remarkable data base manager. Its fast machine language sort complements its large data storage capacity. The sophistication of its data entry, management, and printing capabilities makes Maxi Manager a versatile tool for many applications.

The program now includes Maxi Utility, which allows you to rescue files on diskettes that have been damaged by excessive wear or misuse and lets you expand, add, or delete fields from an existing data base. Let the unmatched capabilities of Maxi Manager handle your data management!

Model I & Model III. Minimum 1 disk drive required . 012-0196 $149.99

Maxi Cras *by Dale Kubler*

(Check Register Accounting System)

Maxi Cras is a system that will computerize check writing, recording and analysis for business and personal finance. The system features 223 income and expense accounts, each of which will handle an unlimited number of transactions. Extensive register and report printing capabilities make Maxi CRAS an indispensible tool for managing money effectively.

Model I & Model III. Minimum 2 drives required 012-0145 $99.95

Maxi Mail *by Dale Kubler*

Maxi Mail is a powerful mail list management system that:

1. Is easy to use.
2. Has virtually unlimited storage capacity.
3. Interfaces with the major word processors to generate form letters and other text.
4. Prints mailing labels up to four-across in any format desired.
5. Has virtually unlimited coding capabilities with thirteen fields of information for each record.

Maxi Mail is the most sophisticated user-oriented mail list program available.

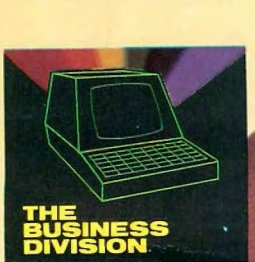
TRS-80 Model III only. 2 disk drives required 012-0148 $99.95

Maxi Stat *by David Walonick*

Maxi Stat is the most useful statistical analysis package on the market today. It was developed to allow maximum flexibility in designing customized analysis. Maxi Stat handles the three main components of statistical analysis:

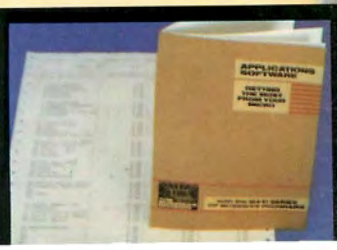
1. Complete menu-driven codebook creation and editing.
2. Menu-assisted data entry.
3. User-created control files to describe the statistical analyses to be performed and printed out on the variables of your choice.

Model I & Model III. Minimum 2 drives required 012-0153 $199.95



Send $1.00 for our 16 page booklet "Getting The Most From Your Micro" All 16 pages are packed with indepth explanations and printout samples from the Maxi Series of applications programs.

THE BUSINESS DIVISION
BOX 3435
LONGWOOD,FL 32750
(305) 830-8194



PUBLISHER
Howard Y. Gosman

BUSINESS MANAGER
Steven M. Kahan

EDITOR-IN-CHIEF
Hubert S. Howe, Jr.

BUSINESS EDITOR
Peter Shenkin

MANAGING EDITOR
Martin Leffler

CONTRIBUTING EDITORS
Leo M. Conrad
Richard Kaplan
Spencer Koenig
Joseph Rosenman
Gordon Speer
A. A. Wicks
Steven M. Zimmerman, Ph.D.

ADVERTISING DIRECTOR
Kevin Rushalko

SALES MANAGERS
Valerie Furci
Sheryl Prevot

DEALER SALES MANAGER
Janet Lasher

ART DIRECTOR
Edmund Khaleel

OFFICE MANAGER
Beatrice Kahn

SOFTWARE MANAGER
Darlene Bell

CUSTOMER SERVICE
Robert Williams

INVENTORY CONTROL
Michael Wiseltier

SHIPPING
Joan Gentry
Al Pizzo

PRODUCT DEVELOPMENT
Steven Kaplan
David Staub

PRODUCTION
Adele Damiano
Louis Wetstein

MARKETING MANAGER
Andrew Hofer

PROGRAMMING MANAGER
Roy Flynt
Nancy Rhodes

APRIL 1983 **ISSUE NUMBER 56**

CONTENTS

FEATURES

REGULAR DEPARTMENTS

All correspondence should be addressed to: The Editor, H & E Computronics, Inc., 50 North Pascack Road, Spring Valley, NY 10977. Unaccepted manuscripts will be returned if accompanied by sufficient first class postage. H & E Computronics will not be responsible for the return of unsolicited manuscripts, cassettes, floppy diskettes, program listings, etc. not submitted with a self-addressed, stamped envelope. Opinions expressed by the authors are not necessarily those of H & E Computronics, Inc.

For information about receiving copies of *COMPUTRONICS* in quantity contact:

U.S. and Canadian Distributor

H & E Computronics, Inc.
50 North Pascack Road
Spring Valley, New York 10977
Attention: Steven M. Kahan
Tel.: (914) 425-1535

International Distributor

Worldwide Media Service, Inc.
386 Park Avenue South
New York, New York 10016
Attenion: Sandra A. Joseph
Cable: WORLDMEDIA
Telex: 620430 (WUI)
Tel.: (212) 686-1520

BITS & PIECES

Howard Y. Gosman

This has been an exciting month, and this issue contains some great information about new computers and software, which you'll read about here and in "The Crystal Ball." We have news and predictions about no less than FOUR new TRS-80 computers (!), the new Model 16 operating system, and a VERY important tip for anyone considering the purchase of a Model 16 (see "The Crystal Ball"). This month, we're even excited about a new APPLE computer—the LISA has finally been unveiled!

MULTITASKING ON DESKTOP COMPUTERS

One of the most important features of the newest computers will be multitasking capabilities. This feature, previously available only for users of large minicomputers and mainframe machines, allows a computer to literally do two (or more) tasks at the same time (or at least it appears to). The microprocessor (Central Processing Unit, or CPU) of your computer can accomplish most tasks at fantastically high speeds, much faster than other parts of the system can keep up with it. Thus, in the ordinary operation of your microcomputer, the CPU spends most of its time waiting—it waits for the operator to input data, it waits for the printer to finish printing a line, it waits for the disk drive retrieve data from a file, etc. Since there's so much wasted "processor time" when running a computer, it is possible (with the right operating system) to have the processor two or more jobs at once. Whenever the processor has to wait for something while doing its primary task, it will then work on a secondary, or "background" task. For example, you use a word processor to type a long paper and edit it. When finished, you have the computer start printing the paper. While the printer is still printing, you can then start typing and editing a letter (a "foreground" task) while the computer continues to

continued on page 6

The H & E COMPUTRONICS MONTHLY NEWS MAGAZINE is published by H & E Computronics, Inc., 50 North Pascack Road, Spring Valley, New York 10977. *The H & E COMPUTRONICS MONTHLY NEWS MAGAZINE* is not sponsored, nor in any way officially sanctioned by Radio Shack, a division of Tandy Corporation.

The purpose of the *H & E COMPUTRONICS MONTHLY NEWS MAGAZINE* is to provide and exchange information related to the care, use, and application of the TRS-80™ computer systems. H & E COMPUTRONICS, Inc. does not take any financial responsibility for errors in published materials. Users are advised to check and edit vital programs carefully.

The H & E COMPUTRONICS MONTHLY NEWS MAGAZINE encourages comments, questions, and suggestions. H & E COMPUTRONICS will pay contributors for articles and programs published in the magazine.

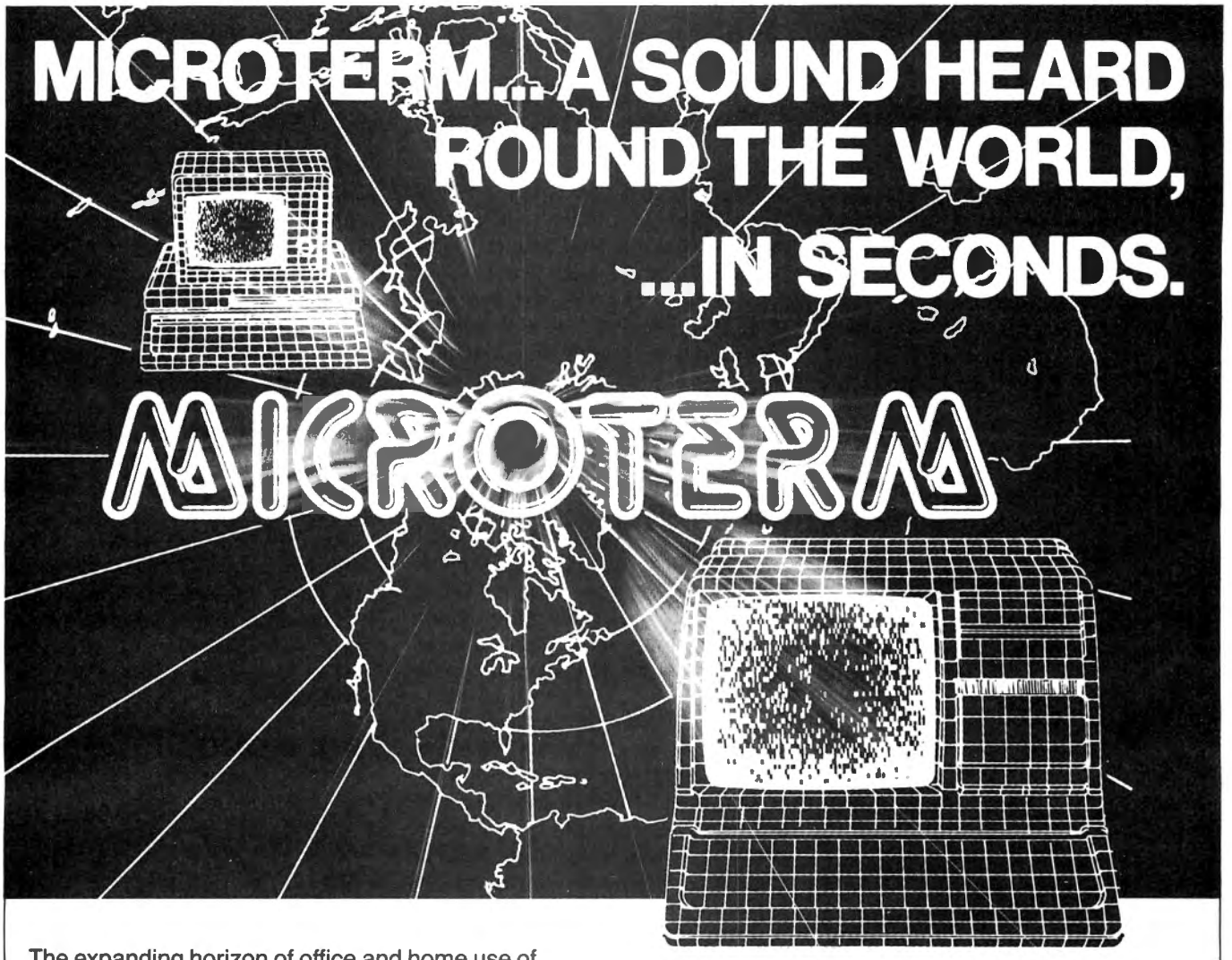
The H & E COMPUTRONICS MONTHLY NEWS MAGAZINE is typeset by Photonics, Ltd., 188 Highwood Ave., Tenafly, NJ 07670, and is printed by Kay Offset Printing Service, Inc., 154 Grand Street, New York, NY 10013.

SUBSCRIPTION RATES

$24 per year	SURFACE MAIL	U.S. Only
$36 per year	FIRST CLASS MAIL	U.S.
$36 per year	AIR MAIL	Canada and Mexico
$48 per year	AIR MAIL	Outside U.S., Canada and Mexico
$3 per copy	Single Copies	U.S., Canada and Mexico
$4 per copy	Single Copies	Outside U.S., Canada and Mexico

Foriegn subscriptions and sales should be remitted in U. S. funds drawn on a U.S. bank.

YOUR SUBSCRIPTION HAS EXPIRED IF . . . THE NUMBER ABOVE YOUR NAME AFTER THE DASH ON YOUR MAILING LABEL IS 56 (OR LESS). THE NUMBER FOLLOWING THE DASH TELLS YOU THE LAST ISSUE THAT YOU WILL RECEIVE. For example, if your subscription number is 16429-56, your subscription expires with this issue (issue #56).



The expanding horizon of office and home use of new communications and data services tying smart terminals into networks through telecommunications links makes the world need a high speed terminal program.

Busy computers shouldn't have to wait for data simply because an old fashioned link can't handle today and tomorrow's telecommunications. Slow terminal telecommunications can stand the smart operator's world on its ear.

Into today's fast paced world, Micro Systems Software presents MicroTerm, the first truly high speed terminal for this inquisitive world. Some computers can run at up to 4800 baud null-free (9600 in some direct-connect applications).

MicroTerm enables you to do more in less time, in both ASCII and the new "error-free" direct file mode. Its unique Macro-Key function allows you to have 10 user-defined keys that transmit up to 64 characters at a single stroke. You can even dial a phone number and transmit the buffer at a specified time completely unattended by the operator.

And while MicroTerm improves your computer's "ears" by outperforming any other telecommunications terminal program, its low price won't take a bite from your bit budget. It's only $79.95 retail.

You can't lose with MicroTerm's features, performance, price, documentation, or support.

It's the only terminal program enabling you to continue operations in the command mode while receiving additional data through the RS232 cable. And you can adjust video width, turn on the printer, open the buffer and do many other things and then return to the terminal mode without missing a thing.

Available for the TRS-80 Models I, II, III, 16, IBM PC, Zenith Z-100, and Apple II computers.

If these advantages are what you want in your world, communicate with your nearest MicroTerm dealer. For information contact: Micro Systems Software, Inc., 4301-18 Oak Circle, Boca Raton, Florida 33431,
Telephone Toll Free: 1-800-327-8724
In Florida (305) 983-3390

MICRO-SYSTEMS SOFTWARE, INC.
4301-18 Oak Circle, Boca Raton, Florida 33431, Telephone: (305)983-3390
Toll Free 1-800-327-8724

THE CRYSTAL BALL

News and Rumors of Interest to TRS-80 Owners

ANOTHER NEW PORTABLE COMPUTER

According to the Wall Street Journal (January 28, 1983), Tandy Corporation is coming out with a Japanese-made, five-pound computer with a full-size keyboard, a screen that can display about a third of a page of text and built-in programs that enable the machine to perform such tasks as text editing and record-keeping the moment it is turned on. In addition, a plug-in package, much like a video-game cartridge, is being prepared to make it possible for the computer to do spreadsheet financial analysis. The price of the new computer is expected to be about $700.

A few notes of clarification about portable computers: there are at least three different sizes of computers under the general classification of "portable" computers.

At the small end of the scale are the so-called "pocket" computers, or "handhelds" which look like big pocket calculators (too big for most pockets). These computers have a complete typewriter keyboard in miniature (which is too small to type on with both hands) and use a one-line LCD display to show text and numeric data.

At the large end of the scale are the OSBORNE-sized desktop portables, which weigh between twenty and thirty pounds and have disk drives and video screens built in. These machines are sometimes jokingly referred to as "sewing machine" computers because of their size, shape and weight. This type of computer generally does not have a built-in battery power supply. (Radio Shack will also soon release a portable version of the Model III computer, described in our last issue.)

In the middle are what are called "notebook" computers, which have full-size keyboards, but are about the size of a big three-ring notebook, and can be operated in your lap anywhere you go. These computers usually have an LCD display that will show about four lines of text at a time. Epson was the first company to heavily advertise a computer of this type, and several other companies (such as Hewlett-Packard and Teleram) have also created very good machines.

The new Radio Shack computer is also a "notebook" computer, but unlike previous lap-sized portable computers, this new portable has full desktop-computer power. The new unit is called the Micro Executive Work Station. Tandy expects to sell *350,000* of them the first year.

The new computer will run programs written on other Radio Shack computers. It can also serve as an appointment book and an automatic telephone dialer and directory. In addition, it will also be able to place calls to electronic information services (such as The Source) and fetch and store such data as the latest prices of stocks.

The new Tandy computer seems to be far superior to the new Epson and HewletMt-Packard computers (and its list price is lower too).

THE MODEL II HAS BEEN REPLACED

Tandy hasn't done a good job selling the TRS-80 Model 16, and they've replaced the Model II with the Model 12. The Model 12 actually sells for less money than the Model II, and has many more features (such as 80K of memory, which allowes the operating system to reside in the top 16K and leaves the other 64K for your programs and data). There's nothing *wrong* with the Model 16, but it's a bit expensive for the present market.

Although it *is* an excellent computer, don't buy a Model 16. Buy a Model 12 instead. Then, when you're ready, upgrade the Model 12 to a Model 16. The price comes out to be just about the same as if you purchased a Model 16—but the Model 12 has a more advanced keyboard (including eight function keys), and when upgraded to 512K, the Model 12 will still have *more* empty slots available for future upgrades than the Model 16 has. If you upgrade the Model 12 to a Model 16, you'll have a better computer than the regular Model 16. So forget the Model 16 and the Model II. The Model 12 will give you more "byte" for your money.

continued on page 6

THE BT ENTERPRISES COMPUTER CONNECTION

Specials for You!

SOFTWARE

ICPL

Interactive Control & Programming Language

The best DATA BASE MANAGER ever designed for the TRS-80 MOD I & III

"Unbelievable is the only word to describe it."

- Menu Facility
- Report Generation Facility
- Full Screen I/O Facility
- Fully Interactive from Basic Programs

ICL Interactive Control Language

*(113049) $59.00

LSO Library Support Option

*(113046) $79.00
*(113047) ICL & LSO $99.00

*product of XYZT Computer Dimensions, Inc.

DosAide

Ever need to go from your program to DOS without losing data? With DosAide you just hit both shift keys and your memory is saved to disk giving you full DOS functions (even go to BASIC) without losing your program data! For all Mod III DOS's.

(103015) $24.95

POWER DRAW

Now you can create those screen displays you have seen in other programs without the need for a video map and set/reset! Just draw it, save it, and merge it into your program! Mod I & III

(113005) $29.95

POWER DOT

If you have GRAFTRAX in your Epson and do not have POWERDOT then you have only half a printer. Now create printer graphics from your keyboard easily! Mod I & III

(113006) $39.95

HARDWARE

B.T. MOD III DISK UPGRADE KIT

Put Disk Drives into your TRS-80 MOD III. Easy to install with Photo Instruction Manual. Featuring the Holmes Disk Controller (assembled & tested) available with Hardware Real Time Clock and Battery built in. Software Clock routines included.

- 40/80 Single/Double Density Support
- Dual Sided and 8" Drive Support
- Drives are optional

(203103) without clock $269.50
(203104) with clock $329.50
(201583) 8" External Drive Cable $29.50

SPRINTER I & III

Shift your TRS-80 into high gear with the Holmes Sprinter clock speed up. Complete with 48K of NEC 150ns (fast!) RAM. All you need to break the speed limit.

(201576) Mod I $135.00
(203576) Mod III $135.00

3M DISKETTE PRODUCTS

Scotch Brand Diskette Products

Excellent and reliable.

5¼"SS/DD (500451) ..... $6.00/2 pk.
5¼"SS/DD (500452) $27.50/10
5¼"SS/DD (500455) $33.50/10
5¼"Head Cleaning Kit $24.00/kit

8"SS/SD (500461) $28.50/10
8"SS/DD (500462) $33.50/10
8"DS/DD (500465) $39.50/10
8"Head Cleaning Kit $26.00/kit

EPSON PRINTER CARTRIDGES

Epson Replacement Ribbons Cartridges

for MX 70/80 & FX 80

Black (500050) ..... $5.00 ea.
Red (500060) ..... $5.00 ea.
Blue (500070) ..... $5.00 ea.
Green (500080) ..... $5.00 ea.
Brown (500090) ..... $5.00 ea.

Epson Replacement Ribbon Carts for MX 100

Black (500550) ..... $11.95 ea.

B.T. Enterprise Dept. 2D
10B Carlough Road
Bohemia, N.Y. 11716
(516) 567-8155 (voice)
(516) 588-5836 (modem)
B.T. Enterprises is a division of Bi-Tech Enterprises Inc.

Dealer Inquiries Welcome
Prices Subject to Change
N.Y.S. Residents Add Tax

Orders Only
800 645 1165
N.Y. Call (516) 567-8155

American Express, Carte Blanche, Diners Club, MasterCard and VISA accepted.

THE CRYSTAL BALL
continued from page 4

GOODBYE TRSDOS — HELLO CP/M

Radio Shack is getting rid of TRSDOS and replacing it with CP/M. According to our sources, Radio Shack will soon be distributing its Model 12 and Model 16 with CP/M 3.0. As you know, the Model 12 comes with 80K of memory (not with the 64K found on the Model II). CP/M 3.0 will use that extra 16K to store the CP/M operating system, thus leaving the entire remaining 64K of memory for your programs and data. This will make it easy to use every available CP/M program on the market (such as WordStar and all those other great programs you have heard about but been unable to use because they only work under CP/M).

It seems that CP/M is dominating the market. Just about every computer is being distributed with CP/M and Radio Shack has decided to take the plunge.

CP/M 3.0 is the newest, most advanced version of CP/M. Older versions of CP/M contained the entire operating system within memory. CP/M 3.0 only stores the important parts of the operating system in memory. If other parts of the operating system are required, the computer accesses the disk for the additional instructions. If you don't know that that means . . . well, let's just say that CP/M 3.0 is far superior to and more efficient than the current versions of CP/M that can be purchased for the TRS-80 from companies such as Lifeboat Associates and Pickles & Trout.

CP/M 3.0 also adds some fantastic enhancements to the TRS-80. For example, more than one computer will be able to share the same hard disk drives.

AND YET ANOTHER COMPUTER FROM RADIO SHACK!

Radio Shack will soon take the Model III off the market. It will be replaced by two new computers. One is the portable Model III computer mentioned earlier. The other is a new computer—the TRS-80 Model IV. The new Model IV will be an upgraded Model III, with 80-track double-sided double density diskette drives, an 80-column screen and high resolution graphics, with an optional built-in 5 Megabyte hard disk drive.

BITS AND PIECES
continued from page 2

print out your long paper at the same time (the background task). Essentially, whenever the processor finds itself idle, it will use the time to work on background task(s). Since the processor may experience thousands of idle moments each second, all of the tasks being accomplished appear to actually be running at the same time.

A fully developed multitasking system could be the next big step forward in small business computers, and it could make some programs far more efficient. For instance, many people have purchased software to keep a schedule of appointments. This kind of package seems very useful at first, but may present problems to some users. An office which uses this kind of program may experience this type of scenario: the operator is using the computer to type a letter when the phone rings. The caller wants to make or cancel an appointment. The computer operator now has to save the text of the letter on diskette, remove the diskette from the machine, put a diskette with the appointments program in the computer, load and run the program, and then use the program to enter the new appointment in the schedule, or locate and change one already scheduled. Then, when finished, the updated schedule must be saved on disk, the operator swaps disks again, loads the word processor, loads the text file of the interrupted letter, and starts working on the letter again. Two seconds later, the phone rings again! As you can see, the operator will *never* finish the letter if the office is busy.

Enter the multitasking computer: the operator is typing the letter, when the phone rings. The caller

continued on page 8

Attention

BARGAIN HUNTERS

Receive Hundreds of Classified Ads Like These Every Month

HARD DISK DRIVE Diablo Mod 31 1.2 MByte std. density. Includes power supp. and cable, rack mount slides, amd manual. Excellent condition. $450. Ca... 1601 ...03

IMPACT PRINTER 165 CPS Serial and parallel interfaces-Eight Selectable character sizes-Single and double width characters-uses standard plain paper - same mechanism as the integral data ... 1 year old $589, syst... Sta... St., ... 250...

HEATHKIT H-11/DEC LSI-11 system, 32K Byte storage, reader 1 punch, video terminal, complete software. Cost $4500 assembled, $3500 kit. Like new. Sell for $2250. 305-962-6677. 2058 Griffin Rd., Ft. Lauderdale, FL 33312.

FOR SALE: Interdata (Perkin-Elmer) 7/16 Mini with 32KB core, front panel, 50A PWR supply. Includes HS tape reader, interfaces for LP, 2 (TTY), and RS-232 (Full duplex, programmable). Includes manuals and much SW (Basic, Fortran, OS, etc.,). $800 - After 6 P... (603)... 2035...

COMPUTER AUTOMATION ALPHA 16; 16 k-word core memory,RTC, PF-R. Modified Mod. ASR-33 TTY Manuals, utilities, assemblers and many option boards - 16 bit I/O Driver, 16 bit I/O, Asynch modem contr. 64 bit output, 10 bit A/D - D/A. Fairly complete documentation. Up and running in Fortran. Not much more than TTY at $1000. **Herb Sauer, 303-494-8724.**

FOR SALE: Heath H9 video terminal, excellent condition, $175 or best offer. You ship. **[214] 962-4484**

WANTED: DIGITAL Group 32K memory board without memory chips and Phi deck controller board (kit, assembled or not working). W... ...1510 NW 35th. T... 32...

PET COMPUTERS moving up to LSI-11. Pet business system priced to sell. PET 2001-16N Computer $800; 2040 Dual Floppy 340K (holds more data than 6 TRS-80 disks) $1,100. Digital cassettes (2) $60 each System complete with Text Editor, disk sort, database software, real estate software and more $2,100. Call **PAUL** (313)971-8447

COMPUTER SHOPPER, the new buy, sell, and trade publication, is ready to help you with the latest information on personal, small business and large-system computers, accessories and software.

Each ad-packed issue is full of bargains you are looking for. Included are ads from individuals throughout the United States who are selling their good, pre-owned equipment just so they can trade-up to new equipment coming on the market.

But, COMPUTER SHOPPER'S bargains won't be yours unless you subscribe. This useful, money-saving publication can become your way to communicate with other buyers, sellers, and traders all over the nation.

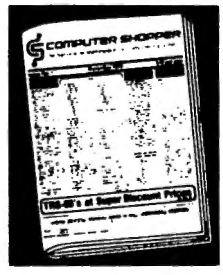
Whether you are a hobbyist or a part-time user, COMPUTER SHOPPER will put you in touch with the nationwide computer marketplace in time for you to take advantage of bargain opportunities.

Have something to sell? A COMPUTER SHOPPER subscriber probably wants to buy it.

Looking for a part, component or even a complete system? A COMPUTER SHOPPER subscriber probably wants to sell it.

COMPUTER SHOPPER is THE marketplace for anything in computers and is read by thousands of people who are ready to buy.

COMPUTER SHOPPER offers a unique format in which classified ads are categorized for fast location of specific items. Combining this with low individual ad rates — 12 cents a word — makes it the ideal place for buyers and sellers to communicate. And, its mix of individual, dealer, and manufacturer ads enable subscribers to find what they want at the best price possible.



COMPUTER SHOPPER will work for you in other ways, too. If you are just thinking about getting into computers, it can help you learn product availability and prices before you make a decision. And, through the timely ads, COMPUTER SHOPPER will keep you abreast of changes in the market which could create bargain opportunities for you.

BUT COMPUTER SHOPPER cannot work for you unless you subscribe.

Want to look us over first? We'll give you your first issue FREE and then bill you for the next 12. If you are not convinced COMPUTER SHOPPER suits your needs, just write "cancel" on the invoice and return it.

And, to let COMPUTER SHOPPER start working for you right now, with a paid subscription we'll also give you a FREE classified ad to sell your pre-owned equipment or to find equipment you want.

If you don't need to use the free classified ad now, use it anytime during your subscription.

Subscription: $10/year, 12 issues plus your first free one. Bank cards accepted. Money back guarantee.

The Nationwide Marketplace for Computer Equipment

COMPUTER SHOPPER

P O Box 23 ● Titusville, Florida 32780

Telephone 305-269-3211

MasterCharge or VISA orders only, call TOLL FREE 800-327-9920.

continued from page 6

wants to make an appointment. The computer operator, by touching a couple of keys, causes the letter to vanish, and the appointment schedule instantly appears. The operator quickly updates the schedule, says goodbye to the caller and hangs up. Then, just as quickly, the appointment schedule disappears, and the letter reappears, ready to be completed.

Advanced systems can allow users to do *many* tasks at the same time, interrupting one program to use another, and then returning to the first program — at the same point where you left it. The computer runs both programs at the same time. An exciting example of this type of computer is APPLE's new LISA computer, described below. After all this time, APPLE seems to have finally introduced a machine that even we die-hard TRS-80 lovers can get excited about!

In some cases, desktop systems will support more than one operator — the perfect example being Radio Shack's TRS-80 Model 16, which may be attached to two additional data terminals. Three operators can work at the same time on this system, reading data from the same disk files, sharing a printer, etc., all working with the same microprocessor (the 68000) in the Model 16. The biggest news about the Model 16 is that the Xenix operating system, which will allow multi-user operation, is now finally available (see below).

TRS-XENIX OPERATING SYSTEM

The TRS-80 Model 16 is finally ready to become one of the most powerful computers available for businesses, and all of Radio Shack's competitors are going to have trouble sleeping at night once people start installing these systems — no other company in the industry offers so much for so low a price, and if word gets around that the system is as versatile and easy to use as Radio Shack says, then every other company will be scrambling to develop multi-user systems. (SPECIAL NOTE: for a hot tip on why you should *not* buy a Model 16 now, see this month's "Crystal Ball" department, and read about the new TRS-80 Model 12!)

The big news is that the real Model 16 operating system is now finally available. The TRS-XENIX operating system is based on Western Electric's UNIX system — which many people feel will be the new standard operating system for the future, eventually displacing CP/M. TRS-XENIX, created by Microsoft, is not just a "UNIX-like" operating system, it actually *is* UNIX, written for the TRS-80 Model 16. The really exciting big feature of this system (and the Model 16) is it's multitasking abilities. The TRS-XENIX system will allow the connection of two of Radio Shack's DT-1 data terminals to the Model 16, so that *three* operators may simultaneously use the system for different tasks, all using the Model 16's microprocessor, and accessing the same files, all at the same time under the XENIX system. Originally, reports showed that a complete three-user system (consisting of a Model 16 with 256K of main memory, two DT-1 terminals and an 8.4 Megabyte Hard Disk drive) would cost about $16,000. Now, Radio Shack is advertising this system for only $10,391 (see the impressive full page ad on p. 21 of *The Wall Street Journal*, January 12). If you were to purchase a system with similar capabilities from IBM, it would cost more than $30,000.

Multi-user capability is one feature that all computers will eventually have to provide to stay competitive in selling small business computers. Here's Radio Shack's scenario: "Think of the possibilities. Your sales department could check inventory levels while someone in accounting updates your general ledger — at the same time someone else is working on payroll!" (And they all work using files on the same hard disk drive.) "All multi-user programs for accounting and inventory will be available in the coming weeks . . . plan on seeing progam for word processing, electronic filing, and . . . a new, second generation electronic spreadsheet from Microsoft . . . the ultimate tool for forecasting and modeling. It's so good, it was recently selected as *InfoWorld magazine's 1982 Software* Product of the Year."

A free brochure on the TRS-XENIX system and the Model 16 is available upon request from Radio Shack, Dept. 83-A-662, 1300 One

Tandy Center, Fort Worth, Texas 76102.

APPLE INTRODUCES THE LISA

Apple's most ambitious computer has finally been released, and the entire industry anxiously waits to see what the response of consumers will be. The Lisa, by far the most expensive Apple at just under $10,000, has features associated with even more expensive computers.

Using an approach similar to the Xerox Star System or other computers which use variations of the "Smalltalk" language, the Lisa uses very high-resolution graphics to create a picture of a desktop on screen, with symbols (or "icons") that represent file folders, disk drives, a wastebasket and other familiar office equipment. The operator uses a "mouse", which is a handheld cursor-movement device, to place a cursor on one of the symbols and select a particular operation. For example, you can move a cursor to indicate a file folder, and a cooresponding data file is then opened for your access.

Information from many different sources is mixed on screen all at the same time with "windows"—groups of simultaneous overlapping displays which can be opened and moved around on screen. An important feature of this kind of system is that numeric data, graphics and text can be passed back and forth between many different types of programs—for instance, data from a spreadsheet program can be used to generate graphs which can then be used by a word processing program. Many people predict that this type of system will be the new "wave of the future" in small computers, introducing a new era in which people can deal with computers in more human terms, instead of having to learn all kinds of special codes to put computers through their paces.

The Lisa includes a 12-inch high-resolution black and white monitor (720 by 364 pixels), one Megabyte (!) of main memory (that's 1000K of RAM!), two 1.7 Megabyte floppy diskette drives, a built-in 5 Megabyte hard disk drive, a typewriter-style keyboard with 10-key numeric keypad, and a computer mouse.

One reason for the high price tag is the inclusion of a lot of software with the Lisa package (everyone is learning something from Osborne, the first company to package standard software with every computer sold). With the Lisa comes LisaCalc (a VisiCalc-like spreadsheet), LisaWrite (word processor), LisaGraph (to graph data from LisaCalc), LisaDraw (a graphics package), LisaList (a database system), and LisaProject (a project management system). The Lisa will also run the BASIC, COBOL and PASCAL languages and will be able to run Microsoft's new XENIX multi-tasking operating system.

MAKING MONEY WITH YOUR COMPUTER

Anyone who's ever wanted to start their own part-time business will appreciate this encouraging and realistic look at how a personal computer can become the basis of a successful home money-making enterprise. The authors give 33 innovative computer ideas ranging from lost pet services to roommate referrals, from direct mail advertising to personal investment analysis, from a unit comparison shopping service to novelty publications, from a computerized babysitting listing to church record-keeping, from biorhythm charts to computer music. And that's just the beginning for an imaginative self-starter.

We have seen at least three or four of these "using a computer to make money books" over the past several years. This seems to be the best of them and it is worthwhile reading if that's what you are setting out to do.

There are plenty of helpful hints on advertising services—including tips on how to write response-pulling ad copy, how to handle newspaper advertising, and tips on developing and keeping customers.

Making Money With Your Microcomputer was written by Robert J. Traister and Rich Ingram and published by TAB BOOKS INC., Blue Ridge Summit PA 17214 (717) 794-2191. The price is $7.95. (It's not available through H & E Computronics, Inc. but it is available through local computer book stores or by calling the telephone number above)

continued on page 10

BITS AND PIECES
continued from page 9

ELECTRONIC ENCYCLOPEDIA

A computerized version of the *World Book Encyclopedia* is now available through the CompuServe Information Service. This project is intended to test the feasibility of this type of service, and enables computer users to search through a listing of more than 31,000 subject entries and retrive any part of the ten-million word text of the encyclopedia. Subscribers can select articles from a menu display that shows all articles whose titles begin with a key word you specify. A News Flashback feature covers current events and steers the reader to related encyclopedia articles, and a challenge feature tests the user's awareness of facts covered in various articles. The encyclopedia will be updated quarterly to insure that all information is current.

AUTOMATED ELECTRONIC MAIL

A new communications utility is available that has some of the most advanced features ever seen in a microcomputer terminal program. The ACEMAIL system was designed to support the auto-dial/auto-answer features of the Hayes Stack Smartmodem, and is capable of receiving or making calls, and sending or receiving data completely unattended! With the ACEMAIL system, it is possible to program your computer to place a call at any time and date (redialing as many times as necessary to get on line), and once logged on, it can send or receive whatever files you have programmed it to transfer—all automatically.

A disk file can be created to store commands for a long list of outgoing calls, including the date, time, phone number, transmission speed, prompt delay and names of files to be transmitted. When the proper time comes, the program refers to the disk file, places the call, and makes the transmission.

The ACEMAIL system is presently available for TRS-80 Models I and III equipped with 300 or 1200 baud Smartmodems, and the Smartmodems themselves are available from the same source. The ACEMAIL 1200 Software costs $119; the ACEMAIL 300 Software is priced at $79. The Hayes Stack Smartmodem, 300 and 1200 baud models, are available for $239 and $619 respectively. The ACEMAIL system was created by ACE Computer Products of Florida, Inc., 1640 N.W. 3rd Street, Deerfield Beach, FL 33441; voice phone: (305) 427-1257; data phone: (305) 427-6300.

SMALL BUSINESS MANAGEMENT SYSTEM

Howe Software has released a SMALL BUSINESS MANAGEMENT SYSTEM designed to provide everything needed to manage a small business on a microcomputer. The system provides for order entry, prints invoices (on plain paper or preprinted forms) and mailing labels, allows recall of invoices from past orders, and keeps track of inventory, with a complete list of all products sold by product type and retail value. Bookkeeping features include a complete general ledger with records of income and expenses by categories.

A complete "installation package" customizes the program for the needs of each business. You define (or update at any time) product and price list (with a maximum of 1000 products) and build a chart of income and expense categories. Other custom items include sales tax rates, shipping and handling charges, automatic invoice numbering, and provision for price discounting.

The system is available for the TRS-80 Models I and III (priced at $350), and also for the TRS-80 Model II, IBM Personal Computer, Osborne-1, or any CP/M computer with 8-inch disk drives (all priced at $400). The system requires an 80 (or more) column printer with 10 character-per-inch print, and it will work on hard disk drives. For more information, contact Howe Software, 14 Lexington Road, New City, NY 10956; (914) 634-1821. (This program can also be ordered directly from H & E Computronics at 800-431-2818.) ■

PROGRAM PREVIEWS

A. A. Wicks

This Month: ARRANGER

Whenever I am looking for a program to review that may be of interest to our readers, I sometimes scan the back reaches of various computer publications. This is the part of the magazine where some of the typefaces are so small I feel that I am in training to read the Bible on the head of a pin. At these times I look for the small two-column ads, placed mostly by companies or individuals who are perhaps not in a position to place $6000-per-page four-color advertisements. I do this for two reasons — perhaps there is a gem to be found (among rocks); and, it would be nice if a worthy program were able to receive whatever benefits an exposure by my review might offer.

On one of these "back country" searches recently, I found "ARRANGER," a disk file indexing program offered by Triple-D Software of Layton, Utah — a small town just north of Salt Lake City. (Since seeing Triple-D's advertising in another magazine, I note that they are also advertising in this one.)

ARRANGER cannot unequivocally be classified as a "gem," but far from being a "rock" it is an exceptionally fine program for the purpose for which it was designed. I had some initial problems in using this program, but, as it turned out, these stemmed from my interpretation of unclear documentation rather than from any program fault or deficiency. Once these problems where out of the way, I had no further difficulties, and was impressed with the smoothness and impressive operation of the program. Incidentally, the cooperation of the people at Triple-D was outstanding in clarifying my problem, they were interested, concerned, and cooperative, and quickly supplied me with another disk (which turned out to be unnecessary).

INITIATION

ARRANGER is started by placing the ARRANGER disk in Drive 0 and resetting for a new "boot-up." No provision is made for a backup of the program disk itself, and the original disk has its own operating system. When I inquired regarding what DOS this might be, I was advised it was a Triple-D DOS, and ARRANGER is self-booting. I did not pursue this further, it works and these reviews are not particularly concerned with the inner workings of a program, except possibly as a passing interest. However, as the program would not back up, and would not provide a Directory from DOSPLUS, I assume the DOS is proprietory. This lack of a backup capability concerns me — somehow I believe a backed up disk is more comfortable to work with; but the program originators must be the ones to determine how useful this might be versus the moral issue of bootlegging possibilities. Later, there will be a description of the function "Backup" as it applies to ARRANGER.

INDEX PREPARATION

Prior to running and assembling a disk directory with ARRANGER, you must decide how you propose to format this directory, by naming your disks in accordance with some plan acceptable to ARRANGER. The name applies to the disk itself — not the programs thereon. The name will replace the current name. For example, the name may be "TRSDOS" before renaming. Afterwards, this could be a number, or any other name up to eight characters long. For instance, I have perhaps 200 disks, and prefer to number them physically (regardless of their data name), using press-on labels. But prior to ARRANGER, they carried disk data names as above — TRSDOS, NEWDOS, etc. With ARRANGER, it was an easy task to give the disks the same "electrical" name (which is actually a number in my case), as the physical one. Others may prefer to name their disks by classification, perhaps, such as GAMES1, UTILITY4, and so on. Once the method of indexing is determined, you may begin renaming your disks, using the Rename command. This is selected from a menu following

continued on page 12

PROGRAM PREVIEWS

continued from page 11

program initialization and date entry, by entering R. (The commands appear on the menu as <R>ename, etc., with the first letter providing the shortcut function entry. The Rename command is followed by a screen request to insert a disk and press <ENTER>. The old name is then displayed and a new one requested, which you now type in, and immediately the disk is rewritten with the new name. All this takes only a few seconds—less than the time to describe it. Proceeding in this way, a rapid renaming of all disks is accomplished easily. If your disks are physically distributed so that you can pick them up and insert them quickly, dozens of disks can be processed in 10 or 15 minutes.

In using Rename, if the name was previously on file (this is when you have ARRANGER established as your system index), you will be asked if you want the change stored in ARRANGER. If the reply is affirmative, the old name is deleted and the new name entered on the disk. If you should reply in the negative, the old name will remain in the file (but not on the disk), the ARRANGER index will still show the old name. This is useful if you have added a new disk name not wanted. Rename will correct the disk name and the Index.

Each of the menu commands may result in certain Error Messages, if the situation warrants. With Rename, one that can occur (as noted above), is "Disk Name Already Exists," which, if allowed will result in two identical names on file. Another error message will occur if an attempt is made to rename a write-protected disk, or an unrecognized DOS is encountered. The latter is not likely to occur, inasmuch as ARRANGER recognizes nine different DOS.

The preceding Renaming function is the "work" part of using ARRANGER. Once this is out of the way, the remaining menu functions are short and easy,and provide beneficial results.

OPERATION

The Add command adds the disk records to ARRANGER. That is, the disk name and all program file names on each disk are added with this command. Up to 45 names per disk are allowed, and in addition, pertinent information such as Free Grans, created and updated dates, number of files, whether a System or Data disk, density and DOS type, are all recorded for later information. All single density disks are recorded as "NEWDOS," although you may have no NEWDOS disks! System and Invisible files are not added. I like this, as otherwise you could have dozens of the same program name in the final index.

Add operates similarly to Rename. You are prompted to insert a disk and press <ENTER>. Once the directory is found and read in to memory, it is displayed and added to the ARRANGER file. This function too, has error messages. One of these, "Maximum Capacity Reached," occurs when 240 disks have been added (200 in single density), which is the maximum quantity. This is not restrictive to anyone wishing to extend beyond this number of disks, as long as this is planned in advance for your system of indexing. For example, your index numbering system could be such as 001-200 for the first series, and 201-400 for the

second. Another valuable flag is "Diskette (..name..) Already on File," which prevents duplication of this type.

If at any time, you may wish to update an ARRANGER index, that word, Update, will do this. This would be desirable, for instance, if new files have been added to a disk, or purged, etc. This function too, is interactive with a screen dialog.

We now come to some of the useful outputs of ARRANGER. Using View, all information about a disk is displayed, except the file names. Nine lines of information are provided, and to view more, the Up and Down arrows will scroll the display. The Print (alphabetical list), command allows you to view the file name information not provided with the preceding command. There are several interesting and useful options available here. One of these allows the output to be either viewed or printed. Also, a Sort, other than the alphabetical listing, may be requested. For example, if you only wish to see CMD-type files, this may be requested. When using a video output, you may use the arrow keys once more to scroll through the ARRANGER index. One nice feature is that the filenames are in rows on the screen, whereas the printer output is in three columns. If there are more than 250 filenames in memory, pressing Clear will continue the sort.

The Find command will search for any file name chosen. In fact, if you cannot recall the full name, it will search based upon the initial part of the file name, if you precede the portion of the name you enter with a question mark. This too, permits the locating of all file names that may start with a certain letter grouping. For example, you might have a number of biorhythm programs, all starting with "bio" that would be displayed when searched for with the command Find—?bio. Once located, the monitor will display the file name and the disk name (or number if "name" is a number), on which the program appears. As with the other commands, the screen may be scrolled if necessary.

The Locate command may sound as if it were another Find com-

continued on page 14

PROGRAM PREVIEWS
continued from page 13

mand, and in a sense it is. However, in this case Locate identifies disks in the total index system that have a specified amount of free space on them. When you specify <L>, the screen dialog requests the minimum amount of free space that it is desired to locate, in granules. Suppose you state "20" as your answer. A list of all disk names that qualify will be displayed, showing the number of granules each has on it. Of course, all disks with more than 20 available granules will be shown, but none will be shown that has less. This is an attractive feature, particularly when disk space is getting tight and you are scratching for a bit of space. It also permits full utilization of disk space.

The Scan command allows you to look through the directories of all disks on ARRANGER. All information relating to a disk will be displayed. Scrolling is controlled as with the other commands, by use of the arrow keys. In looking for some particular name while in the Scan mode, you have the option of starting at some selected name, rather than having to scan the entire file.

We now come to the Backup command. Earlier I said it was not possible to backup ARRANGER. True—when we discuss backup in the usual connotation. But in the ARRANGER sense, backup may be accomplished at any time ten or more disk records have been added to ARRANGER. Then, what occurs is that the ARRANGER and the disk index is backed up to another disk. You now have a backup from which we assume ARRANGER and the index records cannot be separated. I queried the authors about this rather unusual approach, and was told that they considered this "a passive means of protecting the program." I suppose it does accomplish that, and it does not handicap the owner of the program—but it would be a real annoyance to anyone receiving an unauthorized copy (together with someone else's index on it).

SPECIAL OPTIONS

There are some operating niceties with ARRANGER that are well worth mentioning. If you have used one of the menu selections, pressing any key will return the program to the previous selection. This markedly speeds up such functions as adding disks to the records, etc. If you press the Break key in response to a screen query, you are returned to the menu. Users having three or four disk drives may select these drives directly while in screen dialog. Also, there are several options available to use should difficulties occur because of an electrically damaged disk, for example. A lower-case driver is built in to ARRANGER, and Model I systems have keyboard debouncer software installed. No problems should occur with Model I units that have been modified by Radio Shack, and operation with the Radio Shack or other manufacturer's doublers installed, is trouble-free.

Earlier, mention was made of the Manual mode of ARRANGER. In the Manual mode, it is necessary that you type in the disk names during Add or Update. This could happen if you did not want to Rename all of your disks, but would really slow up information entry. There is a "toggle" command available that allows you to select between Manual and Automatic entry. The latter mode is as described for all of the functions previously listed.

INSTRUCTION SHEET

The instructions that accompany ARRANGER cannot be classified as a manual as they comprise only both sides of a sheet of light card stock 8 1/2 by 14 inches in size. However, this text is formatted to the equivalent of six pages of standard printed pages reduced to eight-point type size. This is undoubtedly small (about 1/8th inch), but nevertheless is in a clear, easy-to-read typeset font, with Bold section headings.

The composition of this instruction sheet is another matter. It is rife with spelling errors (and a few typographical ones). There is really no excuse for errors such as these, and I would urge the authors to obtain the services of a High School English teacher to look this material over before the next reprinting. The instructions themselves are confusing in places, and one important item that has been overlooked is an Introduction—something, in this

continued on page 17

EVERY FEW YEARS A UNIQUE PROGRAM ARRIVES

...SO PRACTICAL...SO USEFUL IT BECOMES HERALDED BY COMPUTER OWNERS INTERNATIONALLY AS A TRUE CLASSIC...AN INDUSTRY STANDARD...HIGHLY ACCLAIMED BY EVERY MAJOR TRS-80 PUBLICATION. YOU'VE SEEN IT WITH ONLY A HANDFUL OF TRS-80 PROGRAMS...NOW EXPERIENCE A LEGEND WITH THE NEW...

VERSION 3.0

SUPER UTILITY PLUS

MANIPULATE DISK SECTORS, MEMORY AND FILES • COPY FILES FROM AND TO NON-COMPATIBLE DOS'S • ZAP A DIRECTORY • ZAP A SECTOR • COPY A SECTOR AND COMPARE A SECTOR • COMPARE FILES • PURGE A DIRECTORY • DISPLAY DISK MAPS • EXAMINE FILE LOCATIONS • READ AN ALTER DATA ADDRESS MARKS • DOUBLE SIDED DRIVE SUPPORT LDOS, DOSPLUS, AND MULTIDOS • MORE • BOOTS ON MOD I OR MOD III 35, 40, OR 80 TRACKS • SINGLE OR DOUBLE DENSITY • TEST MEMORY • READS ALL CURRENT POPULAR DOS'S; TRDOS, LDOS,

1st PLACE 80-MICRO 1982 UTILITY PROGRAM OF THE YEAR

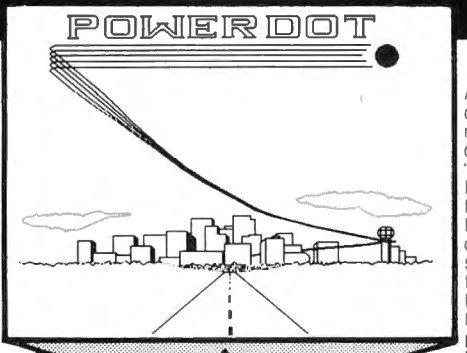
PROTECTED MEDIA
BACK UP COPY INCLUDED

NEWDOS, NEWDOS/80 2.0, DOSPLUS, MULTIDOS, DBLDOS • AUTO DENSITY RECOGNITION • AUTO DOUBLER RECOGNITION • AUTO DOS RECOGNITION • AUTO MODEL RECOGNITION MOD I OR MOD III • REPAIR DAMAGED DIRECTORIES • REPAIR BOOT SECTORS • REPAIR HIT & GAT TABLES • CREATE A SPECIAL OR CUSTOM FORMATTED DISK • REFORMAT WITHOUT ERASE • BUILT IN PRINTER ROUTINES • SOFTWARE BULK ERASE A DISKETTE • PATCHES TRSDOS AT OWNERS OPTION • EASILY CONFIGURED FOR ANY SYSTEM • MORE •

THE PROGRAM W/MANUAL 79.95 • THE TECH MANUAL 14.95 • THE BOOK INSIDE SU+ 19.95

POWERDOT

A GRAPHICS BREAKTHROUGH! Now the ULTIMATE in graphics design and editing. PowerDOT is the state-of-the-art in HI-RES Dot Mappers. Unmatched in quality and features. PowerDOT allows the user to create TRUE HI-RES printouts on the EPSON (Graftrax or Graftrax+ req.), or the C.Itoh PROWRITER from Leading Edge without ANY hardware modifications! The program has the ability to create drawings several times larger than the computer screen. You are only limited by disk storage. Your screen is merely a "picture window" to a much larger drawing area. You define the picture size. You select the print mode: hi-res, lo-res, or ultra-res. A very versatile utility and truly ahead of its time. Only **$49.95.**



POWERDRAW

A full screen graphics editor. PowerDRAW allows the user to draw directly to the screen utilizing cursor controlled movement. Graphics and text may be combined and saved directly to disk or tape, and recalled at a later time. The "SAVE" formats are: CONDENSED TOKENS, EDAS SOURCE, EDTASM SOURCE, BASIC DATA STATEMENTS, BASICS STRINGS, BASIC ARRAYS, and LOAD FILE FORMAT. The files may be run alone or merged into any type of program or application. Animation is also possible when screens are sequenced. Design custom forms, titles, pictures, and graphics for your applications. Received EXCELLENT reviews in INFOWORLD 6/82, 80-US 10/82, and 80-MICRO 10/82. Three printer drivers are included, and the program is JOYSTICK compatible. A GREAT value at only **$39.95.**

POWERMAIL

The most powerful, mailing system for the TRS80. Powermail is a highly sophisticated mass mailing system designed to run under all of the popular DOS's currently available for the Mod I or III. The program is written entirely in machine language for maximum operation speed, and occupies only 4K of the available RAM in your computer. There are no slow periods when Powermail is running. New features have been added to the program that others have always lacked. You now have the ability to keep track of mailings using the 24 flags that are incorporated into the Powermail program. The Powermail system will handle a file up to 8 megabytes, or 65,535 names, whichever is smaller. The program will run in as little as 32K and one disk drive, although 48K and 2 drives are desirable. The program will also sort the entire maximum file size and open up to 168 files simultaneously during the process. Only **$99.95**

PROGRAMS OF TOMORROW HERE TODAY

POWERDRIVER

New printer drivers for the new Radio Shack™ Superscripsit™. This new generation of custom printer drivers allows you to utilize all of SUPERSCRIPSIT'S features with your EPSON, PROWRITER and C.ITOH F-10 STARWRITER printers. You can now utilize all of the printers custom features such as compressed, expanded and proportional print, underlining, boldface, superscripting, and sub-scripting. All drivers can be called from within SUPERSCRIPSIT at "document open time." Order by printer designation as follows: Only **$29.95** each

POWERDRIVER E	EPSON MX70/80/100
POWERDRIVER P	C.ITOH PROWRITER
POWERDRIVER F	C.ITOH F-10

POWERSOFT
Products from Breeze/QSD, Inc.

Available from Selected Dealers Everywhere

11500 Stemmons Fwy., Suite 125
Dallas, Texas 75229
To order call toll free 1-800-527-7432
For product information (214) 484-2976

SORTING IT OUT

Arne Rohde

1.Introduction

In the series of articles on sorting and searching which have appeared in Computronics, we have seen a number of different algorithms which can be used to sort an array into a particular sequence. These algorithms have not been programmed as stand-alone programs, but as routines which could be used in other programs. Normally sorting programs are not written to show the sorting process as it occurs, but the TRS-80 with its memory mapped video display is suited to a real-time demonstration of sorting techniques.

The program presented here is based on the sorting algorithms presented in the series "Array of hope for BASIC programmers," and the technique was inspired by a sorting demonstration program contained on the MMSForth disk (MMSForth version 1.9). The sorting demonstration on the Forth disk contains several sorting routines, where the data being sorted are single byte values contained in the video display area.

2. The Sorting Program

The sorting demonstration program consists of two main sections. The first section contains the code to generate a random array of single-byte values which is to be sorted. The array is kept in memory, to allow the same set of data to be used for each of the sorting methods.

The values generated are in the range from 20H to 0BFH, in other words all the printable ASCII characters and the TRS-80 graphic characters. On Model I computers without a lower case modification, lower case alphabetical characters will not be shown, as they are converted to upper case when they are stored in the video RAM.

The menu allows the selection of each of the sorting routines, or the generation of a new random set of data. The main portion of the program consists of the seven different sorting routines. Each routine can sort up to 960 elements, leaving the bottom line on the screen free for displaying text during the sorting process.

When a sorting algorithm has been chosen, the number of elements to be sorted must be entered. For all except the insertion sort the screen is then filled with the randomly generated elements to be sorted, and the sorting process then begins.

The program can be used for purposes other than demonstrating various sorting algorithms. Since the sorting routines can be programmed in different programming languages, they can also be used to give an indication of the relative execution speed of various languages and compilers. The comparisons are perhaps of limited value, since all manipulations are performed on integer values.

The program has been run in the version shown under several compilers, and some of the algorithms have been programmed in other languages.

3. Execution Timings

All the timings have been performed on a Model I TRS-80 at its standard speed, running under Newdos/80 Version 2 with interrupts enabled. The timings may be improved slightly by disabling interrupts, and possibly also by running under the ROM BASIC without any DOS present.

On the Model III and on any other computers where wait states are introduced during screen scanning, these waits should be disabled. On the Model III, port 236, bit 5 is used to control the video wait function. If bit 5 is set to zero, then the video does not generate waits, if set to 1 then waits will be generated when accessing the video RAM area. Thus the instruction OUT 236,0 could be inserted in the program on the Model III to select no video wait, 64 character mode, and disables the alternate character set.

The timings have been performed using a stopwatch instead of the internal timer, and several of the times are averaged over more than one execution. Because of the slowness of interpreted BASIC, only a few times have been recorded for comparison purposes.

Each sort routine used the maximum 960 elements for the timing runs. The same array was used for the timings of the various algorithms used with each compiler.

The execution speed of some sorting algorithms are more dependent than others on the number of duplicates in the data to be sorted. With 960 characters to be sorted and only 160 different characters available there must be many duplicates. The timings should therefore not be considered as absolute values, but rather as guides to the relative speeds of the algorithms and compilers.

In the languages where the random generation function is not directly available, the random numbers have been chosen by filling the array with a portion of the ROM memory. This gives 256 different characters values in the array. This will have a slight influence on the sort timings for these languages. The languages where this technique have been used are Pascal-80, Fortran and assembler.

4. Interpreted BASIC

As could be expected, the interpreted BASIC version of the program is the slowest. The program has been written with spaces between words, and with a few comments in appropriate places, and this also affects the execution speed negatively.

The program was run with the BASIC supplied with Newdos/80 Version 2, and because of the lengthy execution times only a few of the sorting algorithms have been timed.

The times for interpreted BASIC were as follows:

Sorting method	Time in seconds for 960 elements
Insertion	7009
Bubble	10260
Shuttle	8755
Shell	438
Quicksort with insertion	391

continued on page 18

case, that would tell the reader what the program is for, what it does, and how it does it, succinctly, before the specific details are given. As it is, the instructions start out by telling in three short sentences how to start the program running, and then there is a discourse on how numbers (for disk names), should be entered with and without leading zeros. In general, the instructions continue in this disjointed manner, with no practical grouping of the information. On the scale of 1 to 10 as provided in these reviews, this documentation rates a 4.

As critical as I am of the documentation, the exact opposite must be said of the program operation. Every function operates cleanly, and with phenomenal speed. It is a pleasure to use a program in which machine language is directly at work, and this is obviously reflected in the alacrity of searches, sorts, and screen listings. As the speed of operation was amply demonstrated for me with a few hundred file names, I can easily accept the producer's claim that ARRANGER will sort 1500 file names in just 40 seconds, and find one program name from the same number of names in 30 seconds or less. Some error trapping would be desirable—I lost all that I had entered at one point with a "lockup" of the computer when I inadvertently addressed a drive that had no disk in it. ARRANGER does not include an "information line" as some disk index programs do; this may or may not be important to the user.

PROGRAM BACKGROUND

According to Triple-D, ARRANGER was three years in development. Rich Foy wrote the first version in BASIC for personal use. Then Dan Foy (at the time, 16 years old), assumed the task of writing the program in machine code as a challenge to his capabilities. He did this throughout his Junior and Senior years of High School before the program evolved satisfactorily in July 1982. Dick, Dave and Dan Foy are Triple-D (get it?).

The program is now entirely in machine code, and its source listing requires about 160K, and it requires 17 minutes to assemble it into a CMD file. The program makes only two calls to ROM—one for delay and the other for a keyboard driver. When a disk is updated, the entire file is re-written, and each time the program names are sorted, it is a total sort. Further development is taking place on the sort portion of the program, and tentative experiments have put it at about one-half again of its present rate of speed, which even now, as has been mentioned, is exceedingly fast.

SUMMARY EVALUATION

ARRANGER is a useful and attractively priced disk index program. The broad array of information gathering functions, and the alphabetized printout will make it a welcome addition to anyone who has large disk records to maintain, and wishes to handle these expeditiously and accurately, with an excellent printout of the results.

ARRANGER—Model I, Model III (specify), 32K, 1-drive min. Triple-D Software, P.O. Box 642, Layton, UT 84801—$29.95

A. A. Wicks
30646 Rigger Road
Agoura, CA 91301 ■

LETTERS TO THE EDITOR

Errors in TRSDOS Patches

I just received my February issue of *Computronics* and I am sorry to report that you made four errors in listing my patches (page 60), and I would appreciate it if you would inform your readers. I am also going to explain to them how to correct the errors if they were unfortunate enough to have entered them.

#4 should be:

PATCH *6 (ADD=5850, FIND=3A62,CHG=BF5F).

Users would have no problem with this one, since the patch would not be able to "find" the erroneous values.

#12 should be:

PATCH *5 (ADD=4F04, FIND=D0,CHG=C9).

Users who erroneously entered this patch can fix it by patching as follows:

PATCH *5 (ADD=4F04,FIND=D9, CHG=C9).

#17 should be:

PATCH *4 (ADD=4E28,FIND=20, CHG=18).

If users were successful in making this patch, two patches will be necessary to correct it. Since the address is wrong for the routine as published, it may not have worked when users tried to enter it. If they got a STRING NOT FOUND error, they merely will have to enter the correct patch. Otherwise:

PATCH *5 (ADD=4E28,FIND=18, CHG=20).

Then enter the correct patch #17.

#18 should be:

PATCH *0 (ADD=4EA9,FIND=CA, CHG=C3).

The correction routine is the same as that for #17. If the patch WAS entered and accepted, correct it by

PATCH *5 (ADD=4EA9,FIND=C3, CHG=CA)

and then enter the correct patch #18.

I know I don't need to point out that patches entered incorrectly can wreak havoc with a user's DOS. Please get these corrections to your readers in the next issue.

Jerome I. Weintraub
690 Mountain View Road
El Cajon, CA 92021 ■

continued from page 16

The bubble sort is by far the slowest, the time for 960 elements corresponds to almost 3 hours! The quicksort is more than 25 times faster with the chosen data, and this factor increases with increasing numbers of elements to be sorted. The insertion sort which is actually simpler to code than the more popular bubble sort is about 30 percent faster, but still about 18 times slower than quicksort.

The difference in time between the Shell sort and quicksort is not very great, the fastest execution of Shellsort being actually faster than the slowest execution of quicksort. The difference can be expected to be greater when more elements are to be sorted, but with the numbers chosen here there is very little difference in speed. Quicksort could possibly be speeded up by a better selection of the pivot element when each portion is being sorted.

Another problem with interpreted BASIC is that the execution speed of a GOTO or GOSUB is dependent on the location of the destination line relative to the source line, or its location relative to the start of the program. GOTO and GOSUB near the beginning of the program will be faster than near the end of the program. Quicksort is therefore at a disadvantage in this case, since it is positioned near the end of the program.

Variables in a BASIC program are allocated in memory in the sequence in which they are encountered when the program is executed. The variables which are used first will be referenced faster than variables which are allocated later, since the search for a variable name is a sequential search. Thus the execution speed of a sort algorithm will be faster if it is executed first, rather than being executed after all the others have been executed, since each of them allocate new variables.

BASIC program execution speed can be improved by removing all spaces and remark statements, but at the cost of making the programs almost illegible. The program size will, of course, also be smaller. Several compress programs are available to remove unnecessary spaces from BASIC programs.

5. Compressed-interpreted-BASIC

Some time ago I wrote a compress program which does more than remove uneccesary spaces and comment lines from BASIC programs. Lines are also combined into longer lines when they do not contain conditional statements. Thus the lines

```
51Ø AE = AM + 1
52Ø PLE = AE - 2
```

could be combined into a single line if line 520 is unreferenced, whereas the lines

```
129Ø IF JH>IH THEN 114Ø
13ØØ JL = JH - 1:AW = PEEK(JH)
```

cannot directly be combined even if line 1300 is unreferenced. However, they can be combined with the addition of an ELSE between the two lines.

The compress program also removes the effect of destination line location on GOTO and GOSUB statements, since these are recoded to give an indirect jump directly to the destination line. Thus all GOTO and GOSUB statements will require the same execution time, irrespective of their location and the location of the destination line.

Only two of the sorting algorithms were executed with the compressed program, but the times give an indication of the savings possible by simple means. The times were as follows:

Sorting method	Time in seconds for 960 elements
Shell	344
Quicksort with insertion	268

The difference between the Shell sort and the quicksort has become more pronounced, which seems to indicate that the quicksort in normal BASIC is being slowed down by the GOTO statements, since the program lines for this sort are located nearest the end of the program.

Compression and direct GOTO and GOSUB are two methods for speeding up execution of BASIC programs. Another method is compilation with one of the compilers available. The Microsoft compiler was one of the first available, but also one of the most expensive, so it does not exist in my program collection. ACCEL2 from Southern Software in England was one of the first compilers to appear at a reasonable price, and it has since been followed by ACCEL3.

6. ACCEL2 and ACCEL3

The ACCEL3 compiler is available in the USA from Allan Gelder in San Francisco for $99.95. It is a rewrite of ACCEL2 with a number of extra features. The compiler is resident in memory together with the program to be compiled, and the program is compiled directly to memory, replacing the original BASIC source.

The compilation process is faster in ACCEL3 than in ACCEL2, and took less than 5 seconds for the sort demonstration program. The compiled program was about the same size as the source, but to this must be added the portion of the compiler which must be resident at all times.

There are very few restrictions on programs to be compiled with ACCEL3 compared to interpreted BASIC, but not all statements are compiled. The uncompiled statements will still be executed, but they will be interpreted. Thus there is not much to be gained from compiling programs using only double precision real arithmetic, whereas programs using integer arithmetic can be speeded up considerably, as the sorting demonstration shows.

The times for ACCEL2 are slightly faster in several cases, but at the inconvenience of slower compilation, larger compiled programs, and more restrictions on the allowable BASIC program constructs. The program as shown was compiled with both ACCEL2 and ACCEL3 without any changes.

The sorting times were as follows

Sorting method	Time in seconds for 960 elements	
	ACCEL2	ACCEL3
Insertion	77.8	74.4
Selection	157.1	163.7
Bubble	196.8	231.1
Sifting	85.4	80.1
Shuttle	146.9	165.7
Shell	6.0	6.9
Quicksort with insertion	5.6	4.8

continued on page 20

continued from page 18

The improvements compared to interpred BASIC are very impressive, ranging from about 50 times to about 90 times faster. It is fascinating to watch a screen being filled with random characters which are then sorted within about 5 seconds.

The disadvantage of ACCEL3 compared to other compilers is that the disk BASIC interpreter must be present while the program is being executed, as must a portion of the compiler resident in high memory. The compiled program can be saved to a disk file, but it cannot be loaded and executed in the same way as assembly language programs. A compiler called ZBASIC is available with these features.

7. ZBASIC 2.2

ZBASIC is a BASIC compiler written by Andrew Gariepy and sold by Simutek in Tucson, AZ. The disk version of the program costs $89.95, and a tape version is available at a slightly lower price.

As mentioned above, ZBASIC has a few advantages compared to ACCEL3, but it certainly also has a number of disadvantages. Actually ZBASIC can be considered a new language, except for programs using only integer arithmetic. Most programs cannot be written and tested in interpreted BASIC and then compiled for increased speed, as is the case with ACCEL3. Instead, the programs must be written in the ZBASIC dialect, and compiled each time a correction has been made.

The process is simplified for smaller programs since the source program, the compiler, and the compiled program can be resident in memory at the same time. For larger programs the revised source program must be saved to disk, the compiled program saved to disk, control passed to the DOS, and the program loaded and executed from disk. This is a rather slow process, even if the compilation is relatively fast, being only a few seconds for the demonstration program.

The difference between BASIC and ZBASIC can best be illustrated with a few examples. In the ZBASIC dialect the commands AUTO and EDIT are used to generate a tone on the cassette port, CDBL and CINT will search for a particular byte in memory, DELETE gives a delay of a specified number of milliseconds, ERR and ERL move portions of memory, and FN provides the remainder after an integer division. There are other, smaller, differences. For example all string data in DATA statements must be enclosed in quotes, and FOR loops cannot specify a variable as STEP.

The demonstration program required a few minor changes to be compiled with ZBASIC. The Shell sort FOR loop starting at line 950 contains a variable as STEP, and this had to be changed for ZBASIC. The FOR statement is changed to an assignment (FI = I), and the NEXT statement in line 1010 is changed to an addition and a conditional GOTO.

One of the main advantages of ZBASIC is the execution speed for integer arithmetic, as can be seen from the following times for the sorting algorithms.

(see top of next column)

Sorting method	Time in seconds for 960 elements
Insertion	68.2
Selection	90.7
Bubble	127.8
Sifting	65.8
Shuttle	97.5
Shell	5.3
Quicksort with insertion	2.9

8. Forth

The MMSForth system from Miller Microcomputer Services of Natick, MASS contains a sorting demonstration which inspired the BASIC program reproduced here. The demonstration contains four sorting algorithms, but there are two variants of the quicksort. One of them contains embedded assembly language code for improving the execution speed.

The version of MMSForth used was 1.9 which has been superseded by version 2.0 some time ago, but since I have not upgraded to 2.0 I have not been able to compare the times for the later version.

One difference between the Forth version and the other versions of the program is that the random values are produced as they are moved to the screen. If the random values were generated first, the times for the sorting methods would be reduced by about 4 to 5 seconds each. No attempt has been made to modify the original demonstration program, either to move the data generation or to optimize the sorts.

The times for the four sorting methods were as follows:

Sorting method	Time in seconds for 960 elements
Insertion	379.7
Selection	262.8
Shell	29.5
Quicksort	15.1
Quicksort with assembler	10.4

There is an apparent anomaly between the relative times for insertion sorting and selection sorting, but this is apparently due to an inefficient coding of the insertion sort in the Forth demonstration, and perhaps a more efficient coding of the selection sort. I have not attempted to analyze the algorithms in detail, since there is some truth in the statement that Forth is a WOL (Write Only Language).

Compared to interpreted BASIC the execution times for Forth are impressive, but they cannot compete with the times for the two compiled versions of BASIC. Only the quicksort implementation with embedded assembler code is comparable in speed to the ACCEL3 and ZBASIC versions.

9. Pascal-80

One of my friends who owns a TRS-80 has the Ramware version of Pascal-80. A later, improved version is now sold by New Classics Software of Denville, NJ for $99. The Ramware version was used to code 3 of the sort algorithms.

The Pascal-80 implementation is a semi-compiler,

continued on page 22

BeaLin Corp. *OFFERS YOU...*

TRAX-SW

RETAIN YOUR *TRS-80 BLOCK GRAPHICS CAPABILITY WHILE ADDING ALL OF THE FEATURES OF **GRAFTRAX-PLUS SUCH AS UNDERLINE MODE, SUBSCRIPT/SUPERSCRIPT, LINE DRAWING GRAPHICS, ETC. PLUG IN BOARD SUPPLIED WITH OR WITHOUT **GRAFTRAX-PLUS. (Not available for serial MX-80 Printers.)

TRAX-SW ..... ($69.95)
TRAX-SW WITH GRAFTRAX-PLUS ..... ($129.95)
GRAFTRAX-PLUS ..... ($69.95)
GRAFTRAX 80 ..... ($65.00)

LNW MODEL I $1495.00

48K/4MHZ OPERATION

5-8 Double density controller supports Hi-Res B&W and Color TRS-80 Model I compatible; supplied with a NEC JB 1260 Hi-Res green phosphor monitor, DOSPLUS 3.4 and LNW Basic

LNW MODEL II $1995.00

125K & all Model I features, supports Model I systems and CPM.
Supplied with CPM and NEC JB 1260.

(Special prices for package purchases - call for quotation)

PRINTER SWITCHES

Manual Version-Mod I, III....$79.95 Mod II, XII, & XVI....$99.95

Manual/Software Selectable-Mod I, III...$119.95 Mod II, XII, & XVI..$129.95

Switch between two printers, use your computer to produce draft copies on one printer. Then make final copies on letter quality printer without switching cables.

(MODEL II & XVI requires NO cable modification.)

VERBATIM DATA LIFE DISKETTS (Box of 10).....$29.95

LNW EXPANSION SYSTEM.....$369.95

TEAC DISK DRIVES 40TK/SS/DD .. ($250.00)
New Thin Line drives available at an additional $10.00.
(one year warranty/price includes case and power supply)

HAYES SMARTMODEM $229.00
1200 BAUD VERSION $599.00

PAGE-IT $29.95

A disk utility that prints basic programs in easily readable format.
• COMPATIBLE WITH MOST OPERATING SYSTEMS • HIGHLIGHTS REM STATEMENTS • INDENTS CODE — LINE NUMBERS STAND OUT • UNPACKS BASIC CODE • DOUBLE SPACE OPTION • SUPPORTS SPECIAL OPTIONS OF 15 PRINTERS

(Call or write for information and prices.)

USE OUR DATA LINE TO BROWSE THROUGH THE MANY ITEMS WE HAVE AVAILABLE. YOU MAY PLACE ORDERS OR HAVE SPECIFIC QUESTIONS ANSWERED. DATA LINE FORMAT IS 300 BAUD, 7 BIT, NO PARITY. SATISFY ALL OF YOUR COMPUTING NEEDS, COMPLETE SYSTEMS, DISK DRIVES, PRINTERS, EPSON RIBBONS & REPACKS, PAPER PRODUCTS, PLUS MUCH MORE. CALL OR WRITE FOR FREE CATALOG.

Software shipping and handling — $3.50

PHONE VOICE (301) 490-2744 MODEM (301) 730-2229

*Trademark of Tandy Corp.
**Trademark of Epson

BeaLin Corp. 9335 Old Scaggsville Rd. Laurel, Md. 20707

VISA OR MASTERCARD ACCEPTED — DEALERS INQUIRIES WELCOME

continued from page 20

since the source code is compiled to an intermediate code, usually known as P-code. The P-code is then interpreted when the program is executed. In theory, and sometimes also in practice, the processor independent P-code can be moved from one processor to another, by merely coding the interpreter program for the new processor. This is not the intention in Pascal-80 which presumably uses P-code because it is simpler to compile a program to P-code than to machine code.

The disadvantage of compiling to P-code is that the code must be interpreted, and although it is much faster than interpreted BASIC, it is still considerably slower than machine language.

The times for the three sort algorithms coded in Pascal-80 were as follows:

Sorting method	Time in seconds for 960 elements
Insertion	988.9
Selection	799.1
Shell	72.0

The times for the sorts are about 7 times faster than interpreted BASIC, but 5 to 14 times slower than compiled BASIC. The advantages of Pascal-80 seem to be limited to the improved program structure which Pascal provides compared to BASIC. There seem to be no advantages in execution speed compared to compiled BASIC.

10. Fortran

Fortran (an acronym for Formula Translating System) is the oldest high-level computer language still in general use. Its ancestry dates back to 1954, and its age certainly shows, even though several updated versions have been produced since then. The latest version, known as Fortran-77, contains several extensions which improve the previous version, known as Fortran-IV.

Microsoft has written a Fortran compiler, sold by Radio Shack for the TRS-80 at $99.95. Three of the sorting algorithms have been coded for an old version of this compiler, and this coding process caused a few problems.

The Microsoft Fortran has been extended with PEEK and POKE functions which allow it to reference the screen area directly. However, if the byte being PEEKed has the most significant bit set it will be treated as a negative number. This causes all TRS-80 graphic characters to be treated as negative, and thus sorted before the printable ASCII characters.

It took a long time and several gray hairs to get the Shellsort to work. The compiler contained a bug in the comparison of two one-byte values, causing the sort routine to give unexpected results. Thus the statement

```
IF (PEEK(I) .LT. PEEK(I + 1)) GOTO 62Ø
```

will give incorrect results depending on the two one-byte values being PEEKed. The statement can be changed to the following

```
J = PEEK(I)
IF (J .LT. PEEK(I + 1)) GOTO 62Ø
```

which will function correctly. The two sets of statements should be equivalent, but did not give the same results. Since this was an old version of Fortran the bug may not be present in current versions.

The Microsoft Fortran is a true compiler, generating machine language in the compilation process. The generated code should therefore be fast and comparable in speed to the ACCEL3 and ZBASIC compilers. The times were as follows:

Sorting method	Time in seconds for 960 elements
Insertion	83.3
Bubble	175.5
Shell	5.7

The times are comparable with ACCEL and ZBASIC, and much faster than semi-compiled languages such as Pascal-80. The source program is much longer than the BASIC program, since Fortran only allows a single statement on each line. Also there are restrictions on the subscripts allowed for array references, on parameters for DO loops (the equivalent of Basoc FOR loops), and on IF statements which only allow a single statement and no ELSE clause.

For those who have been brought up on BASIC, it would probably be something of a disappointment to start programming in Fortran. Fortran also shares a disadvantage with most other compiled languages, that the correction process is long and tedious. First an editor is used to make the corrections, a compiler is then called to perform the compilation, a linker is used to link the program with the required modules from the Fortran relocatable library, and the load module can then finally be called to execute the program.

11. Assembly Language

Theoretically, assembly language should provide the fastest execution speed, but in practice optimizing compilers can often produce better code than programmers who only occasionally write assembly code. The Z80 has a complicated instruction set, and it can be difficult to remember all the instructions and the relative execution speed of each.

Another problem with assembly language is that to provide legible code, the code may have to be more comprehensive or slower than code which can be produced by a compiler. For example I would prefer not to write code which is self modifying, or which, as the interpreter for TRS-80 BASIC, contains jumps into the middle of instructions. For example, a compiler could concievably generate the following code:

```
        XOR     A       ;Routine entry 1
        DEFB    01H     ;Opcode for LD BC,data16
ENTR2   LD      A,01H   ;Routine entry 2
        DEFB    01H
ENTR3   LD      A,80H   ;Routine entry 3
        PUSH    AF
```

whereas most programmers would probably prefer the following version which uses one byte more for each entry.

```
        XOR     A       ;Routine entry 1
        JR      CONTIN  ;Continue
ENTR2   LD      A,01H   ;Routine entry 2
```

continued on page 24

New Release

Now supports Mailing Lists, Form Letters, "ZAP-PROCESSING", and 18 more printer drivers.

ZORLOF II

STILL ONLY $69.95

IF YOU STILL THINK YOU HAVE TO SPEND $200 FOR A GREAT WORD PROCESSING SYSTEM, THEN YOU NEED TO READ THIS AD!!

The Magnificent WORD PROCESSING SYSTEM

For the TRS-80 Model I and III

- Supports over 50 different popular printers including OKIDATA Microline 80, 82A, 83A, 84A, Qume, Centronics 737, 739, Radio Shack Line Printer IV, VI, Daisy Wheel II, EPSON MX-80, MX-100, Graftrax, Graftrax Plus, Gemini-10, Gemini-15, NEC PC-8023A-C, Spinwriter 5510, 5515, 5520, 5525, C. Itoh Prowriter 8510, Starwriter FP-1500, F-10, Tec 8500R, Smith-Corona TP-1, Brother HR-1, COMREX ComRiter CR-1, IDS Microprism 480, and Diablo 630.
- Supports proportional space right-margin justifying on Centronics 737, 739, Radio Shack Line Printer IV, Daisy Wheel II, Graftrax Plus, NEC PC-8023A-C, Spinwriter 5510, 5515, 5520, 5525, C. Itoh Prowriter 8510, Starwriter FP-1500, F-10, and Diablo 630.
- Powerful Mailing List and Mail-Merge capabilities for personalizing standard legal documents and Form Letters, handling infinite number of data records per run, infinite number of data fields per data record, and data fields as large as up to 1000 characters each.
- Brand new feature called "ZAP-PROCESSING", allows you to display and edit any type of data or program file in "ZAP" (byte-hexidecimal) format.
- Any character or symbol your printer can print, even dot graphics, can be used in mid-line printing with the Special Character feature.
- Written in fast Z80 machine language with type-ahead key-stroke buffering for speed typing.
- Single key-stroke control of all editing functions for ease of use.
- Continuous on-screen display of word count, line count, and free memory count.
- Superscripts, subscripts, underlined, bolded, expanded and condensed type styles - combine and intermix within a line.
- Automatically justifies and word-wraps on the screen as you type.
- Search, Replace, and Global Search and Replace.
- Odd and even page user-definable headers, footers, and page number lines, with automatic page numbering.
- User-definable linespacing, sheet size, top, bottom, left, and right margins.
- Move blocks of text and copy blocks of text from disk, to disk, and within the text.
- Examine disk directory on any disk and kill files while editing.
- Powerful full-screen editing features for EDTASM and BASIC files, including automatic renumbering of lines.
- Built in function to dump contents of screen to printer.
- Print-previewing formats text, inserts headers, automatically numbers pages, etc. on the screen without printing it on paper.
- Page by page pausing capability for sheet fed printers.
- Supports both parallel and serial printers.
- Printer control code access.
- Works with NEWDOS, NEWDOS80, TRSDOS, MULTIDOS, LDOS, and DOSPLUS - Single or Double Density.
- Compatible with most all available spelling checker programs.

GUARANTEE

Many word processing systems claim theirs are the best, but few would dare guarantee them. Not us! We are confident that **ZORLOF** is the most useful word processing system on the market for under $200. If you don't agree, return it within 30 days for a full refund.

Add $2.00 shipping & handling. Florida residents add 5% sales tax. Checks require 3 weeks to clear banks.

ANITEK

SEE YOUR LOCAL DEALER OR CALL **(305) 259-9397**

MasterCard VISA

ANITEK SOFTWARE PRODUCTS □ P.O. BOX 1136 □ MELBOURNE, FL. 32935 □ (305)259-9397

continued from page 22

```
          JR      CONTIN
ENTR3     LD      A,80H    ;Routine entry 3
CONTIN    PUSH    AF
```

The second version is much easier to read and understand, and should also be easier to modify if the need arises.

It is possible, without any attempt at optimization of code, to write assembly language programs which are considerably faster in execution than programs compiled with ZBASIC, as the following figures for four different sorting routines show.

Sorting method	Time in seconds for 960 elements
Insertion	11.3
Selection	15.7
Bubble	33.5
Sifting	11.3

The times are about 300 to 600 times faster than interpreted BASIC, and about 4 to 6 times faster than compiled ZBASIC code. The code could be improved, and made even faster. The bubble sort seems to be slower than necessary, and could probably be optimized to be almost as fast as the selection sort.

For execution speed alone it seems that assembly language should be the primary choice, but for speed of coding any of the high level compiled languages could be preferable. Perhaps the ZBASIC compiler is a suitable compromise since machine language code can be embedded directly in BASIC statements where speed is essential, but for most integer arithmetic applications the speed of the ZBASIC compiled code should be adequate.

PROGRAM LISTING

```
10 '
20 'SORT demonstration program
30 'Programmed by Arne Rohde, Pilevej 31, 7600 Struer
40 'Denmark, August 1981
50 DEFINT A-S,U-Z:DEFSTR T:B=15360:DIM S1(12),S2(12),R(959)
60 CLS:PRINT@512,"Generating random array": FOR I=0 TO 959:
R(I)=RND(160)+31: NEXT
70 CLS
80 PRINT "        S O R T I N G   D E M O N S T R A T I O N"
90 PRINT:PRINT "Choose sorting method to demonstrate."
100 PRINT "1 for insertion sort"
110 PRINT "2 for selection sort"
120 PRINT "3 for bubble sort"
130 PRINT "4 for sifting sort"
140 PRINT "5 for shuttle sort"
150 PRINT "6 for Shell sort"
160 PRINT "7 for quicksort"
170 PRINT"8 for new data array"
180 PRINT "9 to exit"
190 TK=INKEY$:IF TK<"1" OR TK>"9" THEN 190
200 IF TK="9" THEN 1400 ELSE IF TK="8" THEN 60
210 VT=VAL (TK)
220 LINE INPUT "Array size (1 - 960)? ";TK
230 AM=VAL(TK):IF AM<1 OR AM>960 THEN 70 ELSE AM=AM+B-1
240 CLS
250 ON VT GOTO 300, 400, 500, 600, 700, 900, 1100
260 GOTO 70
300 AE=B:PRINT @960,"Insertion sort";
310 AN=R(AE-B):POKE16383,AN:AN=PEEK(16383)
320 I = AE - 1
330 IF I>=B THEN IF AN < PEEK(I) THEN POKE I+1,PEEK(I): I = I
- 1: GOTO 330
340 POKE I+1,AN: AE=AE+1:IF AE<=AM THEN 310
350 GOTO 1500
400 FOR I=B TO AM:POKE I, R(I-B):NEXT:PRINT @960,"Selection
sort";
420 FOR J=B TO AM-1:NKV = PEEK(J)
430 FOR I=J+1 TO AM
440 IF PEEK(I) < NKV THEN NKV = PEEK(I): NIP = I
450 NEXT
460 IF NKV<PEEK(J) THEN AW=PEEK(J):POKE J,PEEK(NIP):POKE
NIP,AW
470 NEXT
480 GOTO 1500:REM Extraction completed
500 FOR I=B TO AM:POKE I, R(I-B): NEXT: PRINT @960, "Bubble
sort";
510 AE=AM+1
520 PLE = AE - 2
530 IF PLE < B THEN 580: REM sort complete
540 PE = PLE: PLE = -1: FOR I = B TO PE
550 IF PEEK(I) > PEEK(I + 1) THEN AW = PEEK(I): POKE I,PEEK(I
+ 1): POKE I + 1, AW: PLE = I - 1
560 NEXT I
570 GOTO 530: REM repeat until no exchange
580 GOTO 1500:REM Sort terminated
600 FOR I=B TO AM:POKE I, R(I-B): NEXT: PRINT @960, "Sifting
sort";
610 AE=AM+1
620 FI = B
630 FI = FI + 1: IF FI >= AE THEN 690: REM Forw pass end
640 IF PEEK(FI) >= PEEK(FI - 1) THEN 630
650 AW = PEEK(FI): POKE FI, PEEK(FI - 1): BI = FI
660 BI = BI - 1
670 IF BI > B THEN IF PEEK(BI - 1) > AW THEN POKE BI, PEEK(BI
- 1): GOTO 660
680 POKE BI, AW: GOTO 630
690 GOTO 1500: REM Sort end
700 FOR I=B TO AM:POKE I, R(I-B): NEXT: PRINT @960, "Shuttle
sort";
710 AE=AM+1
720 PLE = AE - 2: PFE = B
730 IF PLE < PFE THEN 820: REM sort complete
740 PE = PLE: PLE = -1: FOR I = PFE TO PE
750 IF PEEK(I) > PEEK(I + 1) THEN AW = PEEK(I): POKE I,PEEK(I
+ 1): POKE I + 1, AW: PLE = I - 1
760 NEXT I
770 IF PLE < PFE THEN 820
780 PE=PFE+1: PFE = PLE: FOR I=PLE + 1 TO PE STEP -1
790 IF PEEK(I) < PEEK(I-1) THEN AW=PEEK(I):POKE I,PEEK(I-1):
POKE I-1,AW: PFE = I-1
800 NEXT
810 GOTO 730: REM repeat until no exchange
820 GOTO 1500:REM Sort terminated
900 FOR I=B TO AM:POKE I, R(I-B): NEXT: PRINT @960,"Shell
sort";
910 AE=AM+1
920 DST = 1: IF AE - B< 2 THEN 1040 :REM Already in sequence
930 IF 4 * DST + 3 < AE - B THEN DST = DST * 2 + 1: GOTO 930
940 FOR I = B TO B + DST - 1
950 FOR FI = I TO AE - DST - 1 STEP DST
960 IF PEEK(FI) <= PEEK(FI + DST) THEN 1010: REM Sequence OK
970 AW = PEEK(FI + DST): POKE FI + DST, PEEK(FI)
980 BI = FI
990 IF BI - DST >= B THEN IF PEEK(BI - DST) > AW THEN POKE BI,
PEEK(BI - DST): BI = BI - DST: GOTO 990
1000 POKE BI, AW
1010 NEXT FI
1020 NEXT I
1030 DST = INT(DST / 2): IF DST > 0 THEN 940
```

```
104Ø GOTO 15ØØ:REM Sort completed
11ØØ FOR I=B TO AM:POKE I, R(I-B):NEXT: PRINT@96Ø, "Quicksort";
111Ø AE=AM+1
112Ø S1(Ø) = B: S2(Ø) = AE - 1: REM set array limit
113Ø SP = 1: REM One portion remaining
114Ø IF SP < 1 THEN 127Ø: REM No more portions - end
115Ø SP = SP - 1: IL = S1(SP): IH = S2(SP)
116Ø IF IH-IL < 11 THEN 128Ø
117Ø I = (IH + IL) / 2: REM Middle item index
118Ø AP = PEEK(I): REM Make pivot = A(middle)
119Ø JL = IL: JH = IH: REM indexes for search
12ØØ IF PEEK(JL) < AP THEN JL = JL + 1: GOTO 12ØØ
121Ø IF PEEK(JH) > AP THEN JH = JH - 1: GOTO 121Ø
122Ø IF JH<JL THEN 125Ø
123Ø AW = PEEK(JL): POKE JL, PEEK(JH): POKE JH, AW
124Ø JL = JL + 1: JH = JH - 1: IF JL<=JH THEN 12ØØ
125Ø IF (JL - 1 - IL) < (IH - JL) THEN S1(SP) = JL: S2(SP) =
IH: IH = JL-1 ELSE S1(SP) = IL: S2(SP) = JL - 1: IL = JL
126Ø SP = SP + 1: GOTO 116Ø: REM Use smallest portion
127Ø GOTO 15ØØ: REM Sort completed
128Ø JH=IL+1
129Ø IF JH>IH THEN 114Ø
13ØØ JL=JH-1:AW = PEEK(JH)
131Ø IF JL >= IL THEN IF AW < PEEK(JL) THEN POKE JL+1,
PEEK(JL): JL = JL - 1: GOTO 131Ø
132Ø POKE JL+1,AW: JH=JH+1: GOTO 129Ø
14ØØ END
15ØØ PRINT @1ØØØ,"Hit any key to continue";
151Ø TK=INKEY$:IF TK="" THEN 151Ø ELSE 7ØFile PG5604: Sorting it
out
```

12. Summary

The following table presents a summary of all the timings measured for the various sorting algorithms and the various languages.

	Insert	Select	Bubble	Sift	Shuttle	Shell	Quick
BASIC	7009		10260		8755	438	391
Compressed						344	268
ACCEL2	77.8	157.1	196.8	85.4	146.9	6.0	5.6
ACCEL3	74.4	163.7	231.1	80.1	165.7	6.9	4.8
ZBASIC	68.2	90.7	127.8	65.8	97.5	5.3	2.9
Forth	379.7	262.8				29.5	10.4
Pascal80	988.9	799.1				72.0	
Fortran	83.3		175.5			5.7	
Assembler	11.3	15.7	33.5	11.3			

13. Further Measurements

Several other languages and compilers are available for the TRS-80, but not all of them allow for direct reference to the screen area. I have the TRS-80 Cobol compiler, but it does not contain any instructions which allow for direct reference to memory. The sorting routines could be coded to sort an array and then write the result to the screen, but this would make the program completely different from the others.

Readers are welcome to try the BASIC program on any other BASIC compilers, or to recode the program in any other language which allows for direct memory references. It would be interesting to see the execution speeds for these programs.

Arne Rohde
Pilevej 31
7600 Struer, Denmark ■

PRACTICAL BUSINESS PROGRAMS

S. M. Zimmerman and L. M. Conrad

Month #4: CPA Program to Produce Balance Sheet and Profit and Loss Statement

This month continues the review of the second group of programs in our general ledger set. The first program in this group, TRANS, was reviewed last month. TRANS is used to input the initial values of a data set. This month's program is CPA, the program which produces the balance sheet and profit and loss statement. The last program in this group is TRIAL. The principal function of the TRIAL is to produce a trial balance to be used in the task of tracing down errors. We have found the trial balance program is seldom used, however when an error occurs it is needed.

With the addition of the CPA program you may now set up your chart of accounts, input your transactions, and produce a periodic set of accounting reports.

The publication schedule for the general ledger system is as follows:

No	Month	Program
1.	.....	GLMENU (Controls use of routines).
2.	.....	NAME & START (Initializes chart of accounts and firms name).
3.	.....	TRANSACTION (Inputs monthly transactions).
4.	.....	CPA (Produces Balance sheet & Income and Expense statements, Profit & Loss statement).
5.	.....	UPDATE & YEAR (Changes chart of accounts and performs year end closing tasks).
6.	.....	TRIAL (Produces trial balance).
7.	.....	BALANCE (Balances check book).
8.	.....	MILES & MOVE(Calculates gas mileage and moves files between disks).

The operational procedures of for CPA will be reviewed.

RUNNING THE PROGRAM

CPA is designed to work on a printer only. The intent is to produce an account report set for review by management. If you wish a set to work on the printer only we recommend you make two copies of CPA. Name one CPA and the other CRT. In the case of the CRT version change all the LPRINTs to PRINTs. You will also have to add a counter to stop the material on the screen so it may be read. We wrote such a version and found we never used it, so we dropped it from our accounting set.

The objective of CPA is to produce a profit and loss statement and a balance sheet. The title information will be printed out first:

THIS IS CPA PRINTER OUTPUT FOR P&L/BALANCE SHEET

The first question is necessary if you are using an operating system with its own line counter.

The next question is relative to the name of the current transaction file. This is the file you created when you ran the transaction program TRANS.

CURRENT TRANS?

The name we gave to this file in our example was MAY. The next question is relative to a line counter:

LINE COUNTER (Y/N)?

If your system does not have a built in line counter this question can be eliminated by leaving out line 40 when you copy the program.

OLD BAL?

The OLD BAL or old balance sheet information is the information you have to carry forward from the previous month. We named the old balance file APRIL2 last month and in the next question will name the new OLD BAL MAY2.

NEW OLD BAL?

The next question is relative to the disk used.

DISK?

You will next be asked for the date to be placed on the run.

DATE?

The profit and loss statement will be printed on your printer. When it is complete the balance sheet will be printed on the printer. You will be given the opportunity to line up your paper to print the balance sheet after the profit loss statement is complete.

SETUP FOR BALANCE SHEET AND ENTER?

When the program is complete you will automatically return to GLMENU.

EXAMINING THE PROGRAM

The program starts out with a CLEAR 290 to make room for the string variables to be used. If your chart of accounts is greater than 70, there is some possibility you will get an OUT OF STRING SPACE error. If you do, all that is necessary is to increase the 290 until the error disappears. If you increase 290 to far with only 32k in your system you could run out of memory. This is the reason we have kept this number low.

Line 20 defines the format to be used to print the information on your printer. If your business requires bigger numbers than the formats given, all you need to do is to increase the size and line up the headings. The size of the format used was selected so as to fit on an 80 column printer and perform the task for our business. To increase the format you add ## signs where wanted. In E$ the first set of pound signs is for the account number, the second set is for the debits and the third set is

for the credits.

Line 360 and 560 also contain formats which may have to be adjusted for businesses with bigger numbers. Line 360 is for the profit and loss statement, line 560 prints the totals.

As noted in the section on using the program, line 40 is included to allow the program to be used with operating systems such a DOSPLUS which have built in line counters.

Line 50 and 60 input the name of the files to be used and the disk upon which you want to write the files. Line 70 inputs the information from a file called NO. In this file is stored the number of current assets, fixed assets, etc.

Line 80 inputs the date and prints it on the printer. Line 90 inputs the values of the debit and credits associated with each of the accounts in the chart of accounts. Lines 100 and 110 input the values of the transactions. In line 100 we placed an upper limit of 999 transactions per period on our system by the TO 999 in the middle of the line. If you need to increase this number you may do so.

In line 120 we close the file we have just read in. At no time in our system are more than three files open at any one time. When you start your system you do not need a special file specification.

Lines 130-150 input and print the name to be placed on top of the accounting reports.

From this point the program jumps to line 270 for the purpose of closing the books. This task is complete by line 330 and the job of printing the profit and loss statement begins. We used the old term profit and loss in line 340 to title our statement. You need only change profit and loss to income and expense if you feel like it.

From this point on to line 630 the task of printing the profit and loss statement is performed. When ths job is complete the program is directed to go to line 170 to print the balance sheet. The task of printing the balance sheet is completed in line 260. At this point the program automatically returns to GLMENU.

PROGRAM LISTING

```
1Ø CLEAR 29Ø:REM    "CPA"
2Ø E$="### %              %  #,###,###.##
     ##,###,###.##"
3Ø CLS: PRINT "THIS IS CPA PRINTER ONLY OUTPUT FOR P&L/BALANCE
SHEET": INPUT "CURRENT TRANS";BZ$:NO$="NO":FF$="FILE"
4Ø INPUT "LINE COUNTER (Y/N)";LC$: IF LC$="Y" THEN
CMD"FORMS(T)"
5Ø INPUT "OLD BAL";NX$: INPUT "NEW OLD BAL ";NQ$: INPUT "DISK
";DZ$:FQ$=FF$+"N"
6Ø NP$=NX$+"P":NZ$=NQ$+"P"+":"+DZ$:NQ$=NQ$+":"+DZ$:SU=Ø
7Ø OPEN"I",1,NO$: FOR I=1 TO 6: INPUT #1,P:N(I)=P:SU=SU+P:
NEXT I: CLOSE 1:Z=SU: DIM X#(Z,2),S#(Z,2),QP(Z),QQ(Z)
8Ø INPUT "DATE";DA$: LPRINT DA$
9Ø OPEN"I",1,NX$: FOR I=1 TO SU: INPUT#1,X#(I,1),X#(I,2):
 NEXT I
1ØØ CLOSE 1: OPEN"I",1,BZ$: FOR I=1 TO 999: INPUT #1,P1%,P2%,
P3%,P4%,A1#,A2#:K=P3%:KK=P4%: IF P1%=Ø THEN 12Ø
11Ø X#(K,1)=X#(K,1)+A1#:X#(KK,2)=X#(KK,2)+A2#: NEXT I
12Ø CLOSE 1
13Ø OPEN"I",1,FQ$: FOR I=1 TO 5: INPUT #1,PP$: LPRINT PP$
14Ø NEXT I
15Ø CLOSE 1
```

continued on page 43

ASK RICHARD

Richard Kaplan

Last month you advised against using aluminum foil for mailing diskettes, stating that this can cause serious harm but achieves nothing positive. This seems to be quite a common practice. Are you sure that information was correct?

That point does deserve an explanation. Aluminum foil *will,* indeed, protect a diskette against magnetic fields.

Then why did you advise *against* using aluminum foil to mail diskettes?

The dangers of static electricity *are* magnified if the aluminum foil touches the surface of the diskette (inside the jacket). This is why I cautioned not to use aluminum foil. However, if you place the diskette in its paper jacket, the actual diskette surface will not be exposed. Thus, as long as a diskette is in its protective paper jacket, aluminum foil can, indeed, offer protection against damage.

You mentioned that aluminum foil provides protection against "magnetic fields." Aren't X-rays also a source of danger, as in the classic rubber stamp, "Magnetic Media — Do Not X-ray"?

There is little evidence to demonstrate that X-rays, such as those used in an airport or post office to inspect packages, cause measurable damage to a diskette. X-rays do not affect magnetic media, such as diskettes. Damage to floppy diskettes (erasure of information) *is* often caused by a source such as a magnet or a paper clip which has become magnetized.

Does the diskette with the disk operating system on it have to always remain in the drive, or is it used only when the computer is first turned on? — *Stan Smith; Clearwater, Florida*

If you load BASIC from a diskette ("Disk BASIC"), the operating system (*TRSDOS*) usually must remain in the first drive (drive 0, often called the "system drive"). This is necessary because not all of BASIC is loaded into the computer at one time. Some commands in a program require the computer to load instructions from the diskette in order to find out what the command means. Thus, as a general rule, a diskette containing the disk operating system must always remain in the system drive. But there are some exceptions to this rule: for example, during the operation of a utility such as BACKUP, you may temporarily have a non-system diskette in the system drive, as when copying a non-system diskette onto a blank diskette.

What did you mean when you said, "if" you load BASIC from disk?

Both the Model I and Model III already have a version of BASIC installed in the machine itself in the form of ROM (read-only memory, or "permanent" memory). This version of BASIC is commonly referred to as "Level II BASIC," or "Cassette BASIC," since it is the version which is used on a TRS-80 equipped only with a cassette recorder.

How is Cassette BASIC different from Disk BASIC?

Disk BASIC contains more features than Cassette BASIC does, such a built-in sorting (alphabetizing) command on the MODEL III.

How can I get to Cassette BASIC on my disk-based computer?

On the Model I or Model III (the Model II does not have Cassette BASIC) you can choose to use Cassette BASIC by pressing BREAK and simultaneously depressing the RESET key.

Why would I want to use Cassette BASIC on my diskette system?

Since Cassette BASIC is in "permanent" memory (ROM), it does not occupy any "temporary" memory (RAM). Therefore, more room is available for your programs and data.

I have heard computers referred to as "8-bit" or "16-bit." What does this mean, and is it significant in choosing a computer?

A bit is the smallest unit of information stored in a computer. (8 bits make up a byte, which is generally equal to one character in computer memory.) An 8-bit computer can only work with 8 bits of information in its primary "arithmetic center" at one time. When a computer can deal with more bits at one time, it is capable of more complex operations. A 16-bit computer, dealing with 16 bits at once, can perform more complex calculations with more information at one time, and 16-bit computers often seem to operate far faster than 8-bit computers. Another very important advantage of 16-bit computers is their ability to take advantage of far more memory space. An 8-bit computer can only directly "address" (use) a maximum of 64K of memory. A 16-bit computer is capable of addressing 512K of memory at once. A computer that has more memory space can perform much of its work on data that is present in memory, without having to repeatedly load and save data on a disk. For instance, if you have to search through a data file on a diskette for information, it will take much longer than if you are able to load the entire file into memory at once, and then just search through memory instead.

Are there computers which use more than 16-bits?

Yes. Generally, the number of bits a computer can deal with at once defines the "size category" into which a computer falls. *Micro*computers generally use 8 or 16 bits. *Mini*computers can handle 16 or sometimes 32 bits. *Mainframe* computers (which sometimes fill a whole room) generally use 32 bits and can often work with as much as 64 bits of information at one time when performing certain complex operations.

I know that a big selling point of the TRS-80 Model 16 is its 16-bit capability. Should I buy this computer because of this?

If you are thinking in terms of the future, *maybe.* In terms of the present, no.

Why should I not buy the Model 16 now because of its 16-bit capability?

Before any computer becomes useful, software must be available. Unfortunately, very little 16-bit software is currently available for the Model 16. Most software

operates by simulating a Model II on a Model 16. In particular, a 16-bit version of BASIC, which would take advantage of the 16-bit processor's greater capabilities, is not yet available for the Model 16. As a result, programs written in BASIC will have little noticable speed improvement (except possibly for faster diskette-access time.)

But if I am thinking about the future, wouldn't it be wise to purchase a Model 16 now, since it will eventually have 16-bit software available?

Probably not. Radio Shack recently introduced the *Model 12, which is apparently a replacement for the Model II. Some people feel that, when upgraded to Model 16 capability, the Model 12 may be a more advanced computer than the regular Model 16 is! In fact, rumors are circulating that the Model 16 may itself be discontinued. (See this month's "Crystal Ball" for more on the Model II/12/16 story.)*

You said earlier that 8-bit computers can use 64K of memory, and I know that the Model II has 64K. But if the Models I and III are also 8-bit computers, why can they only get a maximum of 48K?

The Model I and Model III both *do* have 64K of memory, but 16K is in the form of ROM which holds information that the computer needs to have access to, such as Cassette BASIC (discussed earlier) and instructions needed to generate characters and place them onto the video screen. Since the computer can only address 64K, and 16K is used for the ROM, only 48K of "addressing space" is left over for RAM.

Speaking of "memory for the video screen," I've often heard it said that my TRS-80 has a *memory mapped* video display. What does this mean?

A *memory mapped* video display is a feature of computers that have an area of memory set aside which reflects exactly what is appearing on the video screen at all times. In order to print a character on the screen, the character is simply placed somewhere in the video memory (the "map" of the screen), and that character instantly appears in a cooresponding position on the screen. Any character on the screen can be instantly changed simply by changing one character in the video memory. This feature makes it very easy to create complex and attractive screen displays, as when using features like BASIC's PRINT @ statement. Computers that have the ability to print the contents of the video display on a line printer don't actually read the screen itself—they just read the video map in memory.

What is the alternative to memory-mapping?

A computer without a memory-mapped video display would save some memory, but it would have to use some other method (generally through software) of addressing a specific point on the screen in order to create a "formatted" display. Computers without the ability to address a specific point on the screen (such as video displays for most computers that use "dumb" video terminals) can only add characters to the screen at the current position of the cursor. Video output is then merely a continuous stream of characters sent to

continued on page 38

BALL BEARINGS AND FOOTBALL

Gordon Speer

BALL BEARINGS

We have a local industry that manufactures bearing balls. They are made from steel wire by cutting into cylindrical shape, "heading" into spheres (the same type of machine is used to form heads on nails and bolts), grinding to exact shape slightly oversize, case hardening in a smoky furnace, and polishing to final size and finish in tumblers in a process which is well known to lapidarists. I assume they have lots of charts and tables similar to this one, but I suspect they still don't use the metric system much. This program produces a table on the video display which computes the volumes, masses, and number-per-kilogram of various sizes of bearing balls. The program contains quite a variety of formatting algorithms for you to study.

```
10Ø CLS :CLEAR 1ØØ
11Ø PRINT,"B A L L    B E A R I N G S"
12Ø PRINT"----INCHES----       CM          CC          GRAMS
    N/KG
13Ø DATA 1/8,5/32,3/16,7/32,1/4,9/32,5/16,3/8,7/16,1/2,5/8,3/4,7/8
14Ø FOR L=1 TO 13
15Ø READ S$                    'SIZE OF BALL BEARING
16Ø LET D=VAL(LEFT$(S$,1))/VAL(RIGHT$(S$,LEN(S$)-2))
'DIAMETER
17Ø LET CM=D*2.54              'DIAMETER IN CENTIMETERS
18Ø LET V=4/3*3.14*(CM/2)↑3    'VOLUME IN CC
19Ø LET M=V*7.85               'GRAMS EACH
2ØØ LET N=1ØØØ/M               'NUMBER PER KILOGRAM
21Ø PRINT S$;TAB(7)D;TAB(17-SGN(INT(CM)))CM;
22Ø PRINT TAB(27)V;TAB(39)M;
225 PRINT TAB(5Ø) USING"######";N
23Ø NEXT
24Ø GOTO 24Ø
```

FOOTBALL

Now that the on-again off-again football season is finally over here is a football game you can play on your Level-II computer and video display. If you are curious about the length of it (or any other program) you might like to know that the available memory (?MEM) after loading the program is 5716 bytes less than the available memory before loading it. Therefore your 16K computer will hold programs more than twice this size!

John Kemeny mentions Dartmouth Football in his 1972 book *Man and the Computer*, and I have an idea this version is descended from the Dartmouth-Yale game through many revisions. You should remember Dr. Kemeny, a Hungarian immigrant in 1940, who graduated first in his New York City High School class, graduated summa cum laude from Princeton, worked for Albert Einstein, and served as chairman of the Mathematics Department at Dartmouth before becoming President of Dartmouth College. If you see a car bearing New Hampshire license plates BASIC, that's him. He's co-inventor of the BASIC computer language.

The program plays a pretty good game of football, complete with field goal, safety, and two minute warning, but typical of the older basic programs does not use graphics, because of the limitations of the old teletype terminals originally used with BASIC.

```
1ØØ DIM S(2)
11Ø CLEAR 1ØØ
12Ø CLS
13Ø PRINT CHR$(23)
14Ø PRINT@462,"F O O T B A L L"
15Ø FOR N=1 TO 3ØØØ:NEXT
16Ø GOSUB 3Ø3Ø
17Ø PRINT
18Ø PRINT "THIS IS BIG 8 FOOTBALL
19Ø PRINT "I WILL QUARTERBACK OKLAHOMA
2ØØ PRINT "YOU WILL QUARTERBACK NEBRASKA
21Ø PRINT:PRINT "CALL PLAYS BY NUMBER:"
22Ø GOSUB 24Ø
23Ø GOTO 26Ø
24Ø PRINT"1=RUN","2=TRICKY RUN","3=SHORT PASS","4=LONG
PASS","5=PUNT","6=QUICK KICK","7=PLACE KICK (FIELD GOAL)"
25Ø RETURN
26Ø PRINT
27Ø Z=Ø
28Ø T=Ø
29Ø S(Ø)=Ø
3ØØ S(2)=Ø
31Ø PRINT "TOSS OF COIN:  H OR T?"
32Ø Z$=INKEY$
33Ø IF Z$="H" THEN 36Ø
34Ø IF Z$="T" THEN 36Ø
35Ø GOTO 32Ø
36Ø ON RND(2) GOTO 37Ø  ,42Ø
37Ø PRINT "OKLAHOMA WON THE TOSS"
38Ø GOSUB 3Ø1Ø
39Ø P=1
4ØØ X=4Ø
41Ø GOTO 46Ø
42Ø PRINT "NEBRASKA WON THE TOSS"
43Ø GOSUB 3Ø1Ø
44Ø P=-1
45Ø X=6Ø
46Ø Y=55+FIX((.5-RND(Ø))*1Ø)
47Ø GOSUB 3Ø1Ø
48Ø PRINT"KICKOFF IS GOOD FOR"Y"YARDS"
49Ø GOSUB 3Ø1Ø
5ØØ Y1=2Ø+FIX((RND(Ø)-.5)*1Ø)
51Ø IF RND(Ø)<.8 THEN 53Ø
52Ø Y1=Y1+FIX(RND(Ø)*8Ø)
53Ø Y=Y-Y1
54Ø PRINT "RETURN IS GOOD FOR"Y1"YARDS"
55Ø GOSUB 3Ø1Ø
56Ø GOTO 1ØØØ
57Ø P=1
58Ø X=2Ø
59Ø X1=2Ø
6ØØ D=1
61Ø PRINT "NEBRASKA'S BALL ON THEIR OWN 2Ø YARD LINE"
62Ø GOSUB 3Ø1Ø
63Ø GOTO 194Ø
64Ø GOSUB 24Ø
65Ø PRINT"NEXT PLAY-";
66Ø A$=INKEY$
67Ø Z=VAL(A$)
68Ø IF ABS(Z-4)<4 THEN 7ØØ
69Ø GOTO 66Ø
7ØØ R=RND(Ø)
71Ø R1=RND(Ø)
72Ø F=Ø
73Ø IF Z>4 THEN 83Ø
```

```
74Ø IF Z<3 THEN 1Ø2Ø
75Ø PRINT "IT'S A PASS-";
76Ø GOSUB 3Ø1Ø
77Ø IF Z=3 THEN 116Ø
78Ø Y=FIX(16Ø*(R1-.5)/3+3Ø)
79Ø IF R<.1 THEN 124Ø
8ØØ IF R<.25 THEN 131Ø
81Ø IF R<.7 THEN 127Ø
82Ø GOTO 121Ø
83Ø Y=FIX(1ØØ*(R-.5)/3+4Ø)
84Ø IF Z=7 THEN 277Ø
85Ø IF D=4 THEN 87Ø
86Ø Y=FIX(Y*1.3)
87Ø PRINT "PUNT IS GOOD FOR"Y"YARDS"
88Ø GOSUB 3Ø1Ø
89Ø IF X+P*Y<Ø THEN 235Ø
9ØØ IF X+P*Y>1ØØ THEN 165Ø
91Ø IF F=-1 THEN 93Ø
92Ø IF D<4 THEN 1ØØØ
93Ø Y1=5+FIX((.5-RND(Ø))*14)-F*FIX(Y*RND(Ø)/3)
94Ø IF RND(Ø)<.9 THEN 96Ø
95Ø Y1=Y1+FIX(RND(Ø)*8Ø)
96Ø Y=Y-Y1
97Ø F=-1
98Ø GOSUB 297Ø
99Ø PRINT "RUN BACK IS GOOD FOR"Y1"YARDS"
1ØØØ F=-1
1Ø1Ø GOTO 151Ø
1Ø2Ø PRINT "RUN - RUN - RUN
1Ø3Ø GOSUB 3Ø1Ø
1Ø4Ø IF Z=2 THEN 11ØØ
1Ø5Ø Y=FIX(32*(R-.5)/3+3)
1Ø6Ø IF R1<.Ø5 THEN 113Ø
1Ø7Ø IF R1<.95 THEN 133Ø
1Ø8Ø Y=FIX(RND(X)*6Ø)+Y
1Ø9Ø GOTO 112Ø
11ØØ Y=FIX(2Ø*R-5)
111Ø IF R1>.85 THEN 1Ø8Ø
112Ø IF RND(Ø)>.13 THEN 133Ø
113Ø F=-1
114Ø PRINT "FUMBLE  ***  FUMBLE"
115Ø GOTO 133Ø
116Ø Y=FIX(6Ø*(R1-.5)/3+8)
117Ø R=RND(Ø)
118Ø IF R<.Ø5 THEN 124Ø
119Ø IF R<.15 THEN 131Ø
12ØØ IF R<.55 THEN 127Ø
121Ø PRINT "COMPLETE"
122Ø IF RND(Ø)>.85 THEN 1Ø8Ø
123Ø GOTO 133Ø
124Ø PRINT "INTERCEPTED AFTER"Y"YARDS"
125Ø F=-1
126Ø GOTO 89Ø
127Ø PRINT "INCOMPLETE"
128Ø GOSUB 3Ø1Ø
129Ø Y=Ø
13ØØ GOTO 133Ø
131Ø PRINT "QUARTERBACK SACKED  ";
132Ø Y=-FIX(1Ø*R1)
133Ø GOSUB 297Ø
134Ø IF F=Ø THEN 136Ø
135Ø PRINT "AFTER ";
136Ø IF Y<Ø THEN 141Ø
137Ø IF Y=Ø THEN 147Ø
138Ø PRINT "GAIN OF"Y"YARDS"
139Ø FOR Q=1 TO 7ØØ:NEXT
```

continued on page 32

FOOTBALL

continued from page 31

```
140Ø GOTO 142Ø
141Ø PRINT "LOSS OF";-Y;"YARDS"
142Ø IF F>=Ø  THEN 151Ø
143Ø IF RND(Ø)>.5 THEN 151Ø
144Ø PRINT "FUMBLE RECOVERED"
145Ø F=Ø
146Ø GOTO 151Ø
147Ø PRINT "NO GAIN"
148Ø GOSUB 3Ø1Ø
149Ø IF F<Ø THEN 143Ø
15ØØ GOTO 177Ø
151Ø X=X+P*Y
152Ø IF X<=Ø THEN 224Ø
153Ø IF X>5Ø THEN 157Ø
154Ø PRINT "BALL ON NEBRASKA'S"X"YARD LINE"
155Ø GOSUB 3Ø1Ø
156Ø GOTO 177Ø
157Ø IF X>=1ØØ THEN 161Ø
158Ø PRINT "BALL ON OKLAHOMA'S"1ØØ-X"YARD LINE"
159Ø GOSUB 3Ø1Ø
16ØØ GOTO 177Ø
161Ø IF P<Ø THEN 167Ø
162Ø IF F<Ø THEN 165Ø
163Ø P=-1
164Ø GOTO 174Ø
165Ø PRINT "TOUCHBACK - OKLAHOMA"
166Ø GOTO 19ØØ
167Ø IF F<Ø THEN 174Ø
168Ø PRINT ">>>>>> SAFETY <<<<<<"
169Ø GOSUB 263Ø
17ØØ PRINT "OKLAHOMA MAKES A FREE KICK FROM THEIR 2Ø"
171Ø X=8Ø
172Ø P=-1
173Ø GOTO 46Ø
174Ø PRINT ">>>TOUCHDOWN<<< BIG RED!!!"
175Ø GOSUB 267Ø
176Ø GOTO 39Ø
177Ø D=D+1
178Ø IF F>=Ø THEN 186Ø
179Ø IF P>Ø THEN 183Ø
18ØØ PRINT "NEBRASKA'S BALL"
181Ø P=1
182Ø GOTO 193Ø
183Ø PRINT "OKLAHOMA'S BALL"
184Ø GOSUB 3Ø1Ø
185Ø GOTO 192Ø
186Ø IF P*(X-X1)>=1Ø THEN 193Ø
187Ø IF D<5 THEN 2Ø1Ø
188Ø IF P<Ø THEN 18ØØ
189Ø GOTO 183Ø
19ØØ X=8Ø
191Ø PRINT "SOONERS' BALL ON THEIR 2Ø"
192Ø P=-1
193Ø D=1
194Ø PRINT "FIRST DOWN"
195Ø GOSUB 3Ø1Ø
196Ø IF P*X<P*(5Ø+P*4Ø) THEN 199Ø
197Ø X1=5Ø+P*4Ø
198Ø GOTO 2Ø8Ø
199Ø X1=X
2ØØØ GOTO 2Ø8Ø
2Ø1Ø ON D GOTO 2Ø2Ø ,2Ø3Ø ,2Ø4Ø ,2Ø5Ø
2Ø2Ø PRINT"FIRST";:GOTO 2Ø6Ø
2Ø3Ø PRINT"SECOND";:GOTO 2Ø6Ø
2Ø4Ø PRINT"THIRD";:GOTO 2Ø6Ø
2Ø5Ø PRINT"FOURTH";
2Ø6Ø PRINT" DOWN AND"1Ø+P*(X1-X)"YARDS TO GO"
2Ø7Ø GOSUB 3Ø1Ø
2Ø8Ø PRINT
2Ø9Ø T=T+1
21ØØ IF T<>45 THEN 214Ø
211Ø PRINT
212Ø PRINT">>>>> TWO MINUTE WARNING <<<<<"
213Ø PRINT
214Ø IF T<5Ø THEN 222Ø
215Ø IF RND(Ø)>.2 THEN 222Ø
216Ø PRINT
217Ø PRINT "END OF THE GAME"
218Ø GOSUB 3Ø1Ø
219Ø GOSUB 3Ø3Ø
22ØØ RUN
221Ø RUN
222Ø IF P>0 THEN 64Ø
223Ø GOTO 24ØØ
224Ø IF F<Ø THEN 234Ø
225Ø IF P>0 THEN 228Ø
226Ø P=1
227Ø GOTO 237Ø
228Ø PRINT "SAFETY"
229Ø GOSUB 263Ø
23ØØ PRINT "NEBRASKA TAKES A FREE KICK FROM THEIR OWN 2Ø YARD
LINE"
231Ø X=2Ø
232Ø P=1
233Ø GOTO 46Ø
234Ø IF P>0 THEN 237Ø
235Ø PRINT "TOUCHBACK NEBRASKA"
236Ø GOTO 57Ø
237Ø PRINT "TOUCHDOWN OKLAHOMA"
238Ø GOSUB 267Ø
239Ø GOTO 44Ø
24ØØ P=-1
241Ø IF D>1 THEN 247Ø
242Ø IF RND(Ø)>.5 THEN 245Ø
243Ø Z=3
244Ø GOTO 7ØØ
245Ø Z=1
246Ø GOTO 7ØØ
247Ø IF D<4 THEN 254Ø
248Ø IF X<=3Ø THEN 251Ø
249Ø Z=5
25ØØ GOTO 7ØØ
251Ø IF 1Ø+X-X1<3 THEN 242Ø
252Ø Z=7
253Ø GOTO 7ØØ
254Ø IF 1Ø+X-X1<5 THEN 242Ø
255Ø IF X>X1 THEN 261Ø
256Ø IF RND(Ø)>.5 THEN 259Ø
257Ø Z=2
258Ø GOTO 7ØØ
259Ø Z=4
26ØØ GOTO 7ØØ
261Ø IF RND(Ø)>.25 THEN 259Ø
262Ø GOTO 257Ø
263Ø S(1-P)=S(1-P)+2
264Ø GOSUB 3Ø3Ø
265Ø FOR Q=1 TO 2ØØØ:NEXT:PRINT
266Ø RETURN
267Ø FOR Q=1 TO 2ØØØ:NEXT:IF RND(Ø)>.8 THEN 272Ø
268Ø PRINT "KICK IS GOOD!!"
269Ø S(1-P)=S(1-P)+7
27ØØ GOSUB 3Ø1Ø
```

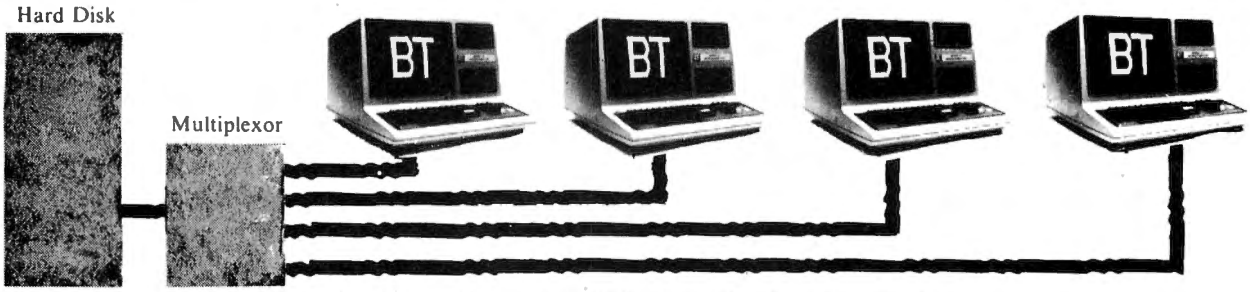
```
271Ø GOTO 264Ø
272Ø PRINT "KICK IS BLOCKED"
273Ø S(1-P)=S(1-P)+6
274Ø GOSUB 3Ø1Ø
275Ø GOTO 264Ø
276Ø PRINT
277Ø PRINT "PLACE KICK"
278Ø F=-1
279Ø IF R>.15 THEN 283Ø
28ØØ PRINT "KICK IS BLOCKED"
281Ø Y=-5
282Ø GOTO 151Ø
283Ø IF P*(X+P*Y)>=P*(P*6Ø+5Ø) THEN 29ØØ
284Ø IF P*(X+P*Y)<P*5Ø+5Ø THEN 288Ø
285Ø PRINT "KICK IS WIDE OF THE GOAL"
286Ø IF P=1 THEN 165Ø
287Ø GOTO 235Ø
288Ø PRINT "KICK IS SHORT"
289Ø GOTO 5ØØ
29ØØ PRINT "FIELD GOAL!!!!"
291Ø P=-P
292Ø S(1-P)=S(1-P)+3
293Ø GOSUB 3Ø1Ø
294Ø GOSUB 264Ø
295Ø IF P=1 THEN 44Ø
296Ø GOTO 39Ø
297Ø X2=X+P*Y
298Ø IF X2>=1ØØ THEN 161Ø
299Ø IF X2<=Ø THEN 224Ø
3ØØØ RETURN
3Ø1Ø FOR Q=1 TO 1ØØØ:NEXT
3Ø2Ø RETURN
3Ø3Ø 'SUBROUTINE FOR GRAPHICS
3Ø4Ø CLS:PRINT CHR$(23)
3Ø5Ø PRINT"            SCORE"
3Ø6Ø PRINT:PRINT"          NEBRASKA"S(2)
3Ø7Ø PRINT:PRINT"          OKLAHOMA"S(Ø)
3Ø8Ø PRINT@96Ø,"G 10 20 30 40 50 40 30 20 10 G";
3Ø9Ø PRINT@144,STRING$(14,14Ø);
31ØØ PRINT@272,STRING$(14,14Ø);
311Ø PRINT@4ØØ,STRING$(14,14Ø);
312Ø PRINT@448,STRING$(3Ø,14Ø);
313Ø PRINT@896,STRING$(3Ø,14Ø);
314Ø X=32:FOR Y=7 TO 19:SET(X,Y):NEXT
315Ø X=88:FOR Y=7 TO 19:SET(X,Y):NEXT
316Ø FOR X=Ø TO 12Ø STEP 12
317Ø FOR Y=22 TO 43
318Ø SET(X,Y)
319Ø NEXT
32ØØ NEXT
321Ø FOR N=1 TO 3ØØØ:NEXT
322Ø CLS:PRINT CHR$(23)
323Ø RETURN
324Ø '
325Ø 'ADAPTED  FOR LEVEL-II
326Ø 'BY MR GORDON E. SPEER
```

Gordon Speer
3304 Woodlawn Road
Sterling, IL 61081 ●

THE COMPUTER CONNECTION

Multiplexor for your Hard Disk

GET IT ALL TOGETHER



ADD COMPUTERS TO YOUR HARD DISK

Adding a hard disk to your computer is the first step in making your micro a powerful computer. Adding more computers to your hard disk makes it a real system. The Bi-Tech **MULTIPLEXOR** does just that. Joining a **MULTIPLEXOR** with your hard disk sub-system enables you to share a hard disk with many computers, and it does so without the need for additional sub-systems. The **MULTIPLEXOR** allows users to share programs and data present on the hard disk, thus affording a true data base to all of its users.

The Bi-Tech **MULTIPLEXOR:**

★ is capable of operating at a faster speed than the hard disk. There is no waiting time as with other systems.
★ offers the same data transfer rates as in a single computer to hard disk sub-system.
★ is totally transparent to the user, software, and hardware
★ is now available for most computers with a BT Hard Disk sub-system and also for many other hard disk systems.

MULTIPLEX VS MULTIUSER

Unlike multiuser systems, which share the microprocessor, the concept behind multiplexing is to share only the disk storage. This frees each computer to use as much processor time as required without affecting the other users. The Bi-Tech **MULTIPLEXOR** links multiple computers to a single hard disk sub-system.

MULTIPLEX VS NETWORK

The Bi-Tech **MULTIPLEXOR's** parallel transfer capability preserves the Hard Disk Drives fast transfer rate. No time is wasted on parallel/serial conversions as would be in a network scheme. Serial networks require one computer to be the host for the rest of the system. In a Multiplex system any computer can be run completely independent of the other computers. As such, each computer is not affected by a hardware or software failure on another computer. In addition, a network also requires expensive decoders at each station, but a Multiplex station needs only an inexpensive host adaptor to link into the Bi-Tech **MULTIPLEXOR.**

SPECIFICATIONS:

● Single Circuit Board ● No External Power required ● No Speed loss between CPU & Hard Disk ● 4 Users per Multiplexor may be daisy chained ● WD 1000/1001 Compatible ● Software independent ●

PRICES

Bi-Tech **MULTIPLEXOR** complete	$795.00
Bi-Tech Model I/III Host Adapter	$325.00
Bi-Tech Model II Host Adapter	$575.00
Bi-Tech IBM-PC Host Adapter	$595.00

Other computers available soon Call

B.T. Enterprise Dept. 2D
10B Carlough Road
Bohemia, N.Y. 11716
(516) 567-8155 (voice)
(516) 588-5836 (modem)
B.T. Enterprises is a division of Bi-Tech Enterprises Inc.

Dealer Inquiries Welcome
Prices Subject to Change
N.Y.S. Residents Add Tax

Orders Only
800 645 1165
N.Y. Call (516) 567-8155

American Express, Carte Blanche, Diners Club, MasterCard and VISA accepted.

THE COMPUTER CONNECTION

The Best of the "Muscle Micros"

DON'T JUDGE A COMPUTER BY IT'S COVER

...If you were to have a Microcomputer operating at 4 MHZ
...If you were to have a Hard Disk built in
...If you were to have a floppy disk drive built in
...If you were to have an Amber Phosphor Monitor built in
...If you had the finest Hard Disk Operating System available

THE BT HARD DISK MODEL III MICROCOMPUTER

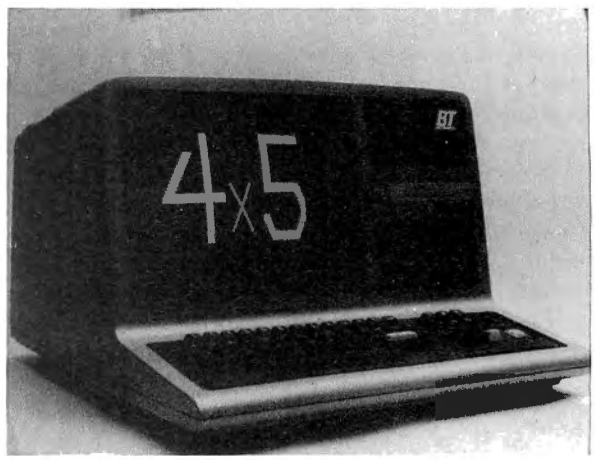
Starting with a basic TRS-80 Model III, add one Tandon 40 Track Dual Sided Disk Drive, 48K of High Grade NEC Memory, the famous Holmes Engineering Disk Controller System, with Real Time Clock, and the security of B.T. Enterprises engineering, assuring long life, and dependability!

The we really get started; include a 4 MHZ Module installed to make the Model III Microprocessor operate at twice the speed of a standard Radio Shack Model III. Also include the use of a Superdense Disk Drive for extra Floppy disk storage. Then have installed a Hard Disk Drive (Winchester Technology). When the system is configured, the Hard Disk Drive contains the operating system and becomes the master (primary) drive. The Hard Disk has 40 times the speed of data transfer as compared to a normal floppy disk as well as containing the storage capacity of more than 30 standard floppy disk drives.

Each unit is built and tested in our Long Island, New York assembly plant and delivered to your dealer complete with a Dos Plus 3.5 Hard Disk Operating System which we consider to have the finest features of any Operating System available.

AVAILABLE CONFIGURATION

4 x 5 5 Megabyte Formated Storage $3995.00*
4 x 10 10 Megabyte Formated Storage $4145.00*
4 x 15 15 Megabyte Formated Storage $4295.00*



Add $20.00 Shipping & Handling

B.T. HARD DISK SUB SYSTEMS

We want you to have all the power that your TRS-80 is capable. These Five, Ten and Fifteen Megabyte formated units are easy to use, just plug them into the expansion buss of your Model I or Model III. Comes complete with the fantastic DOS PLUS 3.5 Operating System!

SYSTEM PRICING

5 Megabyte Winchester Fixed Disk
201505 Model I Version $2399.95*
203505 Model III Version $2399.95*

10 Megabyte Winchester Fixed Disk
201510 Model I Version $2549.95*
203510 Model III Version $2549.95*

Add $10.00 shipping and handling

15 Megabyte Winchester Fixed Disk
201520 Model I Version $2699.95*
203520 Model III Version $2699.95*

*All B.T. Hard Disks Drives come complete with the following:
Postman Hard Disk Mailing System ($225 Value)
XYZT Library Suppot Option ($79.00 Value)
XYZT Interactive Control Language ($59.00 Value)
DosAide Mini-Dos Utility ($29.95 Value)

FOR YOUR DEMONSTRATION STOP IN AT THE FOLLOWING DEALERS

Bi-Tech Computer Stores
10 Carlough Rd.
Bohemia, N.Y. 11716
(516) 567-8155

Bi-Tech Computer Stores
360 Central Ave.
E. Rutherford, N.J. 07073
(201) 939-7341

Oak Tree Systems
20 Washington Ave.
Miller Place, N.Y. 11764
(516) 744-8270

80 Micro Computer Serv. Inc.
158 Remsen St.
Cohoes, N.Y. 12047
(518) 235-9007

Computer Serv. of Danbury
1 Franklin St.
Danbury, CT 06810
(203) 743-1299

CMD Performance
192 Broad St.
Windsor, CT 06095
(203) 688-3862

Micro-Images
146-03 25th Rd.
Flushing, N.Y. 11354
(212) 445-7124

Eberhard Engineering, P.C.
27 Pine Ridge Dr.
Smithtown, N.Y. 11787
(516) 543-7777

Micro Computer Center
28362 Margurite Pkwy Suite 2
Mission Viejo, CA 92629
(714) 495-3782

Services Computer Inc.
383 Chemin Pl. 6018
Drummondville, Quebec J2C5L9
(819) 472-4789

Informatique Lanaudiere Ltee
310 Des Loisirs
St-Ielix De Valois Quebec J0K 2M0
(514) 889-5133

Micro Informatique Laval Ltee
673 Clairette
Laual Quebec H7P 2Y3
(514) 622-6514

B.T. Enterprise Dept. 2D
10B Carlough Road
Bohemia, N.Y. 11716
(516) 567-8155 (voice)
(516) 588-5836 (modem)
B.T. Enterprises is a division of Bi-Tech Enterprises Inc.

Dealer Inquiries Welcome
Prices Subject to Change
N.Y.S. Residents Add Tax

Orders Only
800 645 1165
N.Y. Call (516) 567-8155

American Express, Carte Blanche, Diners Club, MasterCard and VISA accepted.

COLOR COMPUTER CORNER
55 COLOR COMPUTER PROGRAMS

One of the big problems facing all Color Computer users is where to find software and information about their computer. The Color Computer is a powerful and versitile machine, but support from Radio Shack and others has been so slow in coming that many owners feel a constant frustration when searching for software and books. Well, a few companies are finally starting to concentrate on this relatively undeveloped market, and one of them is ARCsoft Publishers, who now have several books devoted exclusively to the TRS-80 Color Computer. We're going to briefly cover one of those books this month.

55 COLOR COMPUTER PROGRAMS FOR THE HOME, SCHOOL, AND OFFICE

This book has a lot of good points to recommend it. The programs are generally quite useful, designed for small businesses, students and teachers, or anyone who wants to get more out of their Color Computer. Each program is clearly described, with a complete explanation of its operation, and the program listings themselves are presented in a large, uncommonly clear typeface (this is a very important point, as anyone who's ever typed in a program out of a book will understand). In addition to providing users with many practical programs, this book will also provide many insights for programming novices. The book is divided into seven sections by type of program. A highlighting of the most interesting programs follows.

Business programs for the Color Computer are a real rarity, so every program in this section is important. Here's some details of some of the best programs:

PROFIT ESTIMATOR helps answer a common question in the business office. Whether your business sells great quantities of wholesale merchandise, smaller quantities over the counter, or single items via mail order, you need to be able to get a fast estimate of your expected cash flow and potential profits. This(program provides an "instant advisor," allowing fast comparisons for quick decisions.

Given information about quantities of items, prices, quantity sold, discounts, etc., the program will calculate unit price, unit profit, gross profit, return percentages, sales needed to break even, and more (depending on which part of the program you're using).

INVOICING is a program that is designed to speed up the process of calculating all of the various totals needed to fill in an invoice. After entering information for each item sold, the program will display the total retail price of the items on the invoice, sales tax (if applicable), shipping charges and the total due. This data can then be quickly entered on the invoice, or, if you have a printer attached, you can easily modify the program to print out the results on paper.

WAGES is designed as a quick calculator of gross wages for all your employees. Given hours and pay rate for each employee, the program returns the gross pay for each, including overtime for employees who have worked over 40 hours.

INVENTORY REVIEW this time saving program is designed to produce a useful report of inventory levels and other helpful information. After entering product descriptions, names of suppliers, retail price and quantity presently on hand, the program will sort all of the information and then produce an inventory report, sorted alphabetically by manufacturer, then again sorted by product name. Whether you have results displayed on the video screen, or print them on your printer, the report will include total number of items in inventory, number of different items in various price ranges, and total retail value of the inventory.

Other programs in this section provide instant calculations of price markups, interest, unit price, and sales commissions, and will produce business graphs and charts.

Programs for students consist of a series of tests on state capitals, major U.S. cities, canadian provinces, the history of invention, moderate and advanced math "flash" quizzes, a hex chart, ASCII codes, and techniques for graphing.

Teachers will find a useful program for evaluating exam scores, an advanced version that alphabetizes scores by student's names, a universal testing program that accepts any type of questions and answers, and a class roll sorter.

Five music programs are provided: a simple piano program which plays notes when you press keys on the keyboard, a musical scale test, a keyboard music player that shows an animated keyboard while you play a tune, and two preprogrammed tunes: a minuet and a ragtime piece.

A wide variety of other programs are also included, from useful tools to games. Electric bill analysis accepts data for a whole year of bills and reports on your energy usage; two photography programs do calculations for flash exposures and closeups; a program that creates a sorted list of names and addresses; a program that synchronizes graphics on your computer with pretaped sounds from your cassette recorder; a savings account interest calculator; and a series of graphics and game programs: *DRAW STRAWS, UP IN LIGHTS, DICE, PASSWORD, CIVILIZATION, KALIEDOSCOPE, SKETCH PAD, BLINKING MAN, CHINESE WATER TORTURE, THE TUNNEL, PORTHOLE, WAVING WOMAN,* and *YELLOW TIME BOMB.*

The book ends up with a useful appendix that gives a series of charts of Color BASIC and Extended Color BASIC functions and statements, operators, video and graphics control codes, control keys, special characters and error messages.

This book will be a welcome addition to the library of any Color Computer owner, and by using these programs you'll become familiar with many programming techniques that you can put to use immediately in your own programs.

55 Color Computer Programs for the Home, School, and Office by Ron Clark, published by Arcsoft Publishers, Woodsboro MD 21798, 126 pages (paper), $9.95. Available from H & E Computronics, (800) 431-2818. ■

POCKET COMPUTER CORNER

Steven M. Zimmerman, Leo M. Conrad, and Stanley M. Zimmerman

This Month Inspection Experiment/Game

This pocket computer experiment/game is designed to aid in the understanding of the limitations of 100% inspection and in number recognition for a young person. Because the program is written on the pocket computer it may be played anytime by a child, or may be used in a professional meeting if enough pocket computers are available

The game consists of identifying how many three or less digit numbers of the numbers printed on the computer's display, are between a lower specification limit and an upper specification limit inclusive. For a childs, use it is suggest you make the game simpler by defining the lower limit as zero and asking the child to count the number of numbers less than 100, or the number of two digit numbers rather than three etc. The use of this program, like many other, is limited only by your imagination.

The computer is started with a seed number and then proceeds to produce a series of random numbers. As they are produced, the computer counts the number beyond the defined limits and prints the random number of the display with a PAUSE statement for a small amount of time. To play the game the player counts the number of numbers in the define range as the numbers are flashed on the display. When the game is over the players results may be compared to the computer's count.

PLAYING THE GAME

Make sure the computer is in the RUN or DEFinable modes before you try to run it. Typing R. ENTER starts the program in either mode. The first thing you will see is INSPECTION EXPERIMENT on the computer's display. The next thing you see is the question:

(N)EW (R)UN?

This menu allows you to start a new cycle with a new random number, or to run a new cycle using the random number currently in the computer. It is suggested you continue to use the current random number once you start the program. Use the (N)EW option only if you wish to control the results for some reason, or when the program is first placed in the computer.

Because of your computer's limited memory there are no built-in error traps to prevent you from making errors. You must answer with the single letter inside the parentheses as shown. If you make an error, the computer assumes you selected the (R)UN option and starts a new cycle.

If you selected the (N)EW option by typing N ENTER you will see:

SEED NUMBER?

A seed number is needed to start your random number generator. Selecting a small number will create a potential start up problem. The first few numbers generated will be zeros. To avoid this select a number seven digits long. Type 1111111 ENTER as your answer to the above question to duplicate our results.

The next question is:

NUMBER OF ITEMS?

There is no limit to how many items you wish flashed on the screen. It is suggested you start with a small number when used as a childs game so as not to lose the child's attention. When used as an inspection experiment you want to have at least 100 items flashed, so some fatigue factor enters into the inspection process. One concept the program can be used to demonstrate, is the boring nature of 100% inspection and the possible errors which occur. It is normal for individuals to count good items as bad, as well as bad items as good. For the sample run let's run 25 numbers. Type 25 ENTER to proceed to the next question:

LSL/USL?

LSL stands for lower specification limit and USL stands for upper specification limit. In the case of a child, if you wish them to count the number of two digits the lower specification limit would be 10, the smallest two digit number, and the upper specification limit would be 99, the largest two digit number. Let's use these limits for our experiment. Type 10 ENTER, and for the next question mark do as follows:

? 99 ENTER

Take care to notice the LSL is the lower limit of the items being counted by the computer, and USL is the upper limit of the items. This way of thinking may not be consistent with the manner in which you specificy the limits of quality of parts. It is convenient to use this approach in the program.

The computer now flashes 25 numbers on the display. Count the numbers between 10 and 99 inclusive with care. When the computer is finished you will see the instruction:

ENTER FOR COUNT?

After you have pressed the ENTER key you should see:

DEFECTIVES 3

If your computer happens to return to the first menu without stopping (this seems to happen to us often) you can still obtain the same results. Press the BREAK key and type R. 40 and you will see the results indicated above, or type Q ENTER and the value 3 will appear on the screen.

EXAMINING THE PROGRAM

This program is relatively short. Only 13 lines of code were needed to complete the routine. Following is a list of the variables have been used in this program:

Variable	Use
A$	To direct the flow from the first menu
A	As a counter in FOR/NEXT statement
B	The random seed number
C	The number of numbers to be flashed on display
D	The lower specification limit.
E	The upper specification limit.
F	The number flashed on the screen.
Q	A counter for number of defective items.
Z	A dummy to use to stop the program.

The letter A has two uses. Its first use is in the main menu. Once this use is complete the letter is reused for a counter in the FOR/NEXT loop producing the numbers on the display.

The program may be divided into three parts. Part I consists of lines 1, 2, 3, and 4. Line one prints the title of the program on the display and then uses an INPUT statement to route the program to either start (N)EW, or to make a (R)UN.

Line 2 inputs a new seed number if the player tells the computer to start new. The seed number is multiplied by 100,000 and then made into a postive number in line 2. This aids in getting the random number generator started just in case the player selected a small number.

Line 3 is used to select the number of items to be flashed on the display. There is no limit on this variable. Line 4 inputs the lower and upper specification limits. Careful selection of these limits will result in games or experiments for different purposes.

Part II of the program is contained in lines 14 through 20. This code prints the random number on the display and counts, using the variable Q, the number of numbers between the lower and upper specification limits. If you wish to change the size of the random number being produced, you need only change the value which is divided into B in line 17. In line 7 the value of B is divided by 100,000 and changed to an integer to yield a number between 0 and 999 inclusive.

In line 18 the number of numbers falling between the lower specification limit and the upper specification are counted. Notice the number between the limits are counted not the number outside the limits!

The next part of the program consists of lines 30 and 40. This code is used delay the program and to print the number of items inside the specification limits.

PROGRAM LISTING

```
1:PAUSE "INSPECTION EXPERIMENT":INPUT "(N)EW (R)UN ?" ;A$
2:IF A$=1"N" INPUT "SEED NUMBER?";B:B=10↑5*ABS (B)
3:INPUT "NUMBER OF ITEMS ";C
4:INPUT "LSL/USL ?";D,E
14:Q=0
15:FOR A=1 TO C
16:B=23B-INT (23B/(10↑8+1))*(10↑8+1)
17:F=INT (B/(10↑5))
18:IF F<=DIF F <= E LET Q=Q+1
19:PAUSE "  ";F
20:NEXT A
30:INPUT "ENTER FOR COUNT ?";Z
40:PRINT "DEFECTIVES ";Q:GOTO 1
```

SUMMARY

This program can be used both as an industrial experiment or as a game. It is useful for illustrating the problems associated with 100% inspection, or to aid a child in learning numbers.

Steven M. Zimmerman, Ph.D.
College of Business and Management Studies
University of South Alabama
Mobile, Alabama 36688

Leo M. Conrad
Imagineering Concepts
P.O. Box 9843
Mobile, Alabama 36691-0843

Stanley M. Zimmerman
Olensky Bros. Ccmputers
3763 Airport Blvd.
Mobile, Alabama 36608 ■

ASK RICHARD

continued from page 29

the screen (as with a teletype terminal that can only print a continuous stream of characters on paper).

I've been told that the CP/M operating system is supposed to run on just about any microcomputer, and that this is why programs written for CP/M will also run on any computer. But then someone else told me that there's a different version of CP/M for every machine. How can this be if its supposed to be the same system on all computers?

The CP/M operating system is supplied in a different version for every CP/M computer. It has to be different for every computer, because the hardware (the microprocessor and its supporting circuitry, and the machine's peripheral equipment) is different on each computer. When two computers use different processors, the "machine language" used by the two machines is different. The different versions of CP/M are written in the native machine language for each computer.

Put very simply, when you run the CP/M operating system on different computers, it makes them all act as if they were the same kind of computer—CP/M essentially acts as a "translator" between the machine itself and higher level languages such as MBASIC. A computer which is equipped with its own version of CP/M (and the MBASIC language) can then run *any* program written in MBASIC, no matter what computer the program was originally written for. A computer running CP/M "emulates", or mimics, all other computers that use CP/M.

Very often I've been told that I should purchase MBASIC to run a particular program. What is MBASIC, and why must I purchase it?

MBASIC is a version of the BASIC programming language written by Microsoft, Inc. Microsoft has become more or less an "industry standard," since they wrote the first (and most popular) commercial version of BASIC for microcomputers. MBASIC is Microsoft's BASIC which is designed to run on CP/M computers.

Because of the way in which MBASIC works, you need to have MBASIC present in your computer not only to write an MBASIC program, but also when you

run the program. Many commercial software programs are written in MBASIC because MBASIC allows easy corrections and updates, and also because many users of mass-marketed software like to be able to modify and customize programs for their specific needs. If a program is written in machine language instead of a language like MBASIC, modifications become extremely difficult, if not impossible.

What do "ASCII" and "EBCDIC" stand for?

ASCII (American Standard Code for Information Interchange) and EBCDIC (Extended Binary Coded Decimal Interchange Code) are two codes for representing data in a computer. EBCDIC is used mainly in large mainframe IBM computers. *All* present microcomputers use the ASCII code, in which every character is assigned a decimal number from 1 to 127. 95 of these characters are more or less standard among all computers (all 52 upper and lower case letters, numbers, and certain punctuation and special symbols). This standardization is what enables computers built by different manufacturers to "talk" with each other, and to use the same standard printers.

What is a "diskette emulator"?

A diskette emulator is a device (sometimes called a "SEMIDISK" or "RAMDISK"), usually available as a circuit board which is installed in a computer, which can hold a very large amount of information *in RAM memory*. When such a device is installed, it is usually designed so that is uses exactly the same commands to store and retrieve information as those used for floppy diskette drives. The result is tremendously decreased access time when using the emulator instead of a diskette drive.

Diskette emulators are available for many different computers. Their price is generally in the $500 to $2000 range.

I'm currently shopping around for a printer and, frankly, I'm confused by so many specifications. To begin with, should I buy a *dot matrix* or a *letter quality* printer?

Letter quality printers produce printed copy as good as (or better than) a typewriter's output. Unfortunately, they are expensive and relatively slow—from about 10 to 55 cps (characters per second).

Dot-matrix printers have letters formed by dots arranged in a grid, or matrix. Many people feel that output from this type of printer is unacceptable for business correspondence. However, dot-matrix printers are usually much faster than letter quality printers (often 120 cps or more), and they are also much more inexpensive.

What is a *bi-directional* printer?

A bi-directional printer prints from left to right on one line, then on the next line, prints backwards from right to left. Since time which would normally be spent returning the print head to the beginning of the line is saved, bi-directionality is a speed-increasing feature.

What is the difference between bi-directionality and *logic seeking*?

A bi-directional printer alternates printing lines forwards and backwards. A logic-seeking printer, on the other hand, calculates the direction to print which would require the shortest traveling distance for the print head. If a line printed from left to right only contained 5 characters, the printer would return the print head to the left and print from left to right instead of moving the print head over all the way to the right side of the paper first. Very often, a printer which is bi-directional will be logic-seeking as well.

What are *proportional spacing* and *justification*?

A proportionally-spaced character set is one in which different characters are different widths, depending upon their actual size. For example, a lower-case letter "i" takes up far less space than a capital "M". Proportionally-spaced printing resembles typesetting and appears more pleasing to the eye.

Justification (sometimes called "right-justification" refers to adding spaces in a printed line in order to line up both left and right margins exactly. For example, newspapers are always right-justified; the right margin always ends exactly at the same point due to small amounts of space added in between words, and sometimes between the letters in each word.

What width of paper should my printer be able to handle?

This is a very common question among new computer owners, and it illustrates a quite prevalent misconception. The width of the paper that your printer uses may have nothing to do with the number of characters that may be printed on each line, since different printers will print varying numbers of characters per line, and many dot-matrix printers allow you to select several different sizes of characters.

Then how many characters should my printer print on each line?

Some printers are available which only print 40 characters per line. I do not recommend purchasing these, since little software is available for such a printer.

An 80-character printer is the smallest line width for practical use. Much business software is designed for such a configuration, and a *compressed* printing mode (which will print 132 characters per line, in small type) is usually available for printing larger reports on dot-matrix printers.

For serious work, I recommend purchasing a 132-character printer. That is, if you plan to run your business with your computer or to generate large amounts of printed reports, 132 characters per line are usually required.

But didn't you just say that an 80-column printer with a compressed printing mode can print 132-columns?

Yes, but if you plan to use your printer for serious applications, it is usually worth the additional expense to purchase a full 132-character printer. In order to activate the compressed printing mode, a short command must be given to the computer, but sometimes this command cannot be given to the printer when using a commercially sold program.

Questions from readers on all aspects of personal computing are welcomed. Please enclose a self-addressed, stamped envelope with your request.

Richard Kaplan
H & E Computronics
50 North Pascack Road
Spring Valley, NY 10977 ■

BEGINNER'S CORNER

Spencer Koenig

Some not-so-BASIC Editor Functions

If you've stuck with me so far, you'll remember we've been going over the various edit functions for several of the most popular Disk Operating Systemd (DOSs). Last time I was just getting into MULTIDOS. This is a very interesting operating system in that it has some edit functions similar to those I've seen on mainframe editor packages (I've worked with WYLBUR on the New York City University IBM 360/370 system).

Just as an aside, I found working with WYLBUR a little mind boggling. It's such a huge program with so many options that it can be inhibiting to try to accomplish the simplest of commands. The manual, covering the edit functions as well as information on the operating system itself, must weigh 10 pounds plus. It's really immense.

MULTIDOS isn't as large or difficult to work with, but it does have an option similar to one of WYLBUR's (and yes, there is an ORVYL too). MULTIDOS has an option called GLOBAL EDITS. The global editing feature allows you to make changes throughout the whole program with one command. You can change variable names, integers, strings, and items contained in a data list. You can also compress strings, split lines or merge them, and change "reserved" words. See EXAMPLE 9.

EXAMPLE 9

MULTIDOS Global Edit Commands

-<ENTER> Enter Global Edit Mode

Once you're in Global Edit Mode, a message will appear which will ask you for the Target and line number. See EXAMPLE 10.

EXAMPLE 1Ø

Line	Target T =	REPLACED BY R =

In order to change any of the items mentioned above, you enter the name or label, which can be as many as 255 characters long. The item, however cannot be contained in quotes. If you look at EXAMPLE 11 you can see a list of the types of changes possible using MULTIDOS. It looks pretty incredible.

EXAMPLE 11

Changes available under Global Edit

Variables	1-255 characters ending with <ENTER>
Letters in quotes	precede the item with a " $ " sign
Changing the first letter of multi-lettered variables	enter the target character like this X' as X,X1,XRAY. It must be followed by a single quote
To change the second character leaving the first alone.	enter the target character enclosed in single quotes 'X'

(Notice: Lower case is not acceptable for variables or reserved words.)

To change reserved words or using arithmetic operators or any of the control codes	Changing "RUN" to -RUN you must use "<" and ">" to bracket the target item
Merging adjacent line numbers	Lines can only be merged to the lines after them i.e. line 1 and 2 become line 1. Answer the TARGET question with a "/" and the line # to be appended to.
Splitting lines	Lines can only be split into consecutive lines else run time errors will occur when references to lines in between the split lines occur. The in between lines will not be found. To use the split function answer the TARGET query with -###. -### is the word to be split. If the target is a reserved word then bracket it with "<" ">". If no target is specified then the line will split at the first colon encountered giving you line 1 and line 2.

Splitting lines can be hazardous. I suggest you look at EXAMPLE 12.

EXAMPLE 12

The splitting of lines:

```
1Ø PRINT"HELLO":PRINT"BYE"

T=-,LINE A= 1Ø, LINE B= 2Ø
```

The resulting lines look like this:

```
1Ø PRINT"HELLO"
2Ø PRINT"BYE"
```

Splitting lines at reserved words:

```
1Ø A$="HELLO":B$="BYE":PRINT A$:PRINT B$

     T=-<PRINT> B$, LINE A=1Ø. LINE B=2Ø
```

Leaving the results:

```
1Ø A$="HELLO":B$="BYE"
2Ø PRINT B$
```

Once you have entered in the TARGET, you will be asked to enter the range of lines within which to search. Afterwards you will enter in the REPLACEMENT. If by chance you made a mistake in the TARGET, simply hit the <ENTER> key in answer to the replacement question. This will return you to the TARGET query whereby you use the SHIFT+@ keys to delete the TARGET and then enter in a new one.

When you've passed this stage you will go on to the next (Thanks for that piece of info) where you will be asked (one more time) if you are sure you want these changes made (Y/N). This is also the last chance to correct any errors, so you'd better be sure. If you answer "Y," the fun begins, and you begin to see the changes take place under the title lines:

LINE TARGET REPLACE.

To break out of the Global Edit mode, hit <ENTER> after a successful change or <BREAK> at any of the prompts, and that about covers the subject as it relates to specific editor in some typical and well conceived operating systems (thank you Howard Cosell). Right? Wrong, keyboard breath (I like Hill Street Blues).

Now we get to the topic that I've been building up to for so long. Can you use an ordinary and/or not so ordinary word processor to do some of the same work more efficiently, less dangerously, and (sometimes) with less muss and fuss? Good question! The answer is a definite MAYBE, depending on circumstances and what you want to do.

Most of the things that your specific operating system BASIC can do can be done using a word processor as long as there is good planning and careful consideration before, during and after program coding. First, let's look at the the can's and can'ts of using a word processor. See EXAMPLE 13.

EXAMPLE 13

COMMANDS	BASIC	WORD PROCESSOR
1.AUTO	YES	NO
2.RENUM ENTIRE PROGRAM	YES	NO
3.WORKS WITH NON-ASCII FILE	YES	NO
4.LIST SPECIFIC LINES (TOP OR LAST OR IN BETWEEN)	YES	YES (USING 'FIND STRING')
5.PAGE COMMAND	YES	SOME
6.USES LABELS WITH LINE NUMBERS	NO	YES (BUT MUST BE DELETED BEFORE RUNNING THE PROGRAM
7.MOVE BLOCK	SOME DO	YES (COPIED NOT RENUMBERED)
8.RELOCATE LINES OR BLOCKS	SOME DO	CAN BE COAXED SEE TEXT
9.MERGING LINES	SOME	YES
1Ø.MERGING PROGRAMS	YES	YES
11.GLOBAL VARIABLE EDITS	SOME	YES
12.CHANGING CHARACTER	YES	YES
13.CHANGE STRING TO NEW STRING	SOME	YES
14.INSERT CHARACTER	YES	YES
15.EXTEND LINE	YES	YES
16.SCROLL DISPLAY BY LINE	YES	YES
17.SCROLL DISPLAY BY PAGE	NO	SOME DO
18.FIND STRING	SOME	YES
19.MACRO (BLOCK) DELETE	YES	YES
2Ø.RANGE-OF-LINE DELETE	YES	SEE BLOCK DELETE
21.REDEFINE KEYS	NO	NO
22 BASIC LINES LONGER THAN 255 CHARACTERS	NO	YES (YOU SHOULD BE CAREFUL WITH THIS BECAUSE IT WILL RESULT IN ERRORS)

The next to last command is often found in those BASIC EDITORS, such as XBE/CMD from Computer Applications Unlimited or IRV from Creative Software. Sometimes these types of editors also let you add graphics to BASIC programs more easily.

OK. Now that we're past the comparison stage and you see what can't be done (easily), let's get down to work. The object of this adventure is to find a way to get around the limitations of the word processor and to make those commands that BASIC can do work with us.

To begin, the word processor I use is Electric Pencil. It produces an ASCII file without any additional appendages for any printers unless you want them. It isn't as fancy as some of the others around, such as Lazy Writer or Scripsit, but it is easy to use and manipulate. I mention this because I intend to give my famous short cuts and suggestions as if you too were using Pencil. If you are using those other guys, then you may have to make some adjustments to my instructions.

How, then, do we get around the limitations of commands 1-3? To get around the AUTO command, we have to create a file that contains nothing but line numbers. You do this when in BASIC by typing "AUTO" (big help huh) and putting a " ' " or REM marker on each line. You then save the file in ASCII format.

This brings us to command 3. Can we use a file other than an ASCII file? The answer is no, because word processors are essentially fancy string handlers. If a file isn't in STRING/ASCII format, then the word processor can't read it. The reason the E.P. can't read it is because it looks for an end-of-file marker that consists of two

zeros (00). Once it finds these it stops loading the file. If a file is saved in its normal "BINARY" format, it then consists of a bunch of 1's and 0's. The chances are good that there will be many many double 1's and especially double 0's.

Command 2 is no real problem. W.P's just can't renumber lines either. I suggest that if you need to renumber a program you do it in BASIC. It's a lot easier.

Commands 4 and 6, listing specific lines and/or using labels are, as I see it, related techniques. Here's why. When you're programming, it's a good idea to program in a modular fashion. By this, I mean that your program should be broken down into smaller sections that do some simple part of the overall task. Some texts refer to this as TOP DOWN programming. If you think this way when you're working, then you can give each section a specific and unique name or LABEL. Once you've assigned a LABEL to the section of code, you can find it again very easily using the W.P's FIND STRING command. See EXAMPLE 14.

EXAMPLE 14

```
I suggest you put the labels in front of the line number.

5' INITIALIZE
1Ø DEFINTA-Z
2Ø PRINT "HI THERE "

25'SECOND SECTION
```

After naming the section INITIALIZE, it then becomes very easy for Electric Pencil to find it again. This same technique can be used as well with specific lines, variables or variable strings.

One of the best techniques to learn is choosing your variable names carefully. If this is planned from the outset, then changing them globally becomes a minor problem. See EXAMPLE 15.

EXAMPLE 15

```
5'    INPUT SCORES
1Ø DIM TS(151)' test scores
2Ø AV=Ø       ' AVERAGE
3Ø FOR SC = 1 TO 15Ø ' 15Ø SCORES TO BE INPUT
4Ø INPUT " INPUT THE TEST SCORES PLEASE "; TS(SC)
5Ø NEXT
6Ø'
65'    OUTPUT AVERAGE
7Ø FOR SC=1 TO 15Ø
8Ø TS(Ø)=TS(Ø)+TS(SC)'THE NEXT SCORE IS ADDED TO THE PREVIOUS
9Ø AV = TS(Ø)/SC    ' AND THEN DIVIDED BY THE NUMBER OF SCORES
1ØØ PRINT AV
11Ø NEXT
```

Now, if I wanted to modify the name of AVerage to AS, the task would be greatly simplified, because AV is a unique variable that is easily read, understood, and found. I realize that this is a simple example, but I'm sure you can imagine how much easier life would be even as the programs become more complex.

What's required is to tell the W.P. to "FIND and REPLACE AV with AS" for as many times as it occurs in the program. The same kind of thing can be done for any string within the program as long as it's unique and the command is given correctly. By this I mean that you have to be sure you define the string you want changed, correctly. If you want to change AV and there is another AV within another string ("WAVE") and you specified it to Electric Pencil as FIND AV instead of FIND " AV ", you will have some problems. Remember, the space is a delimiter that must be specified.

The next thing that W.P's are good for is block deletes. You can specify a block to be deleted quite easily. The same holds true for block moves. What isn't done easily is renumbering the block of code. See EXAMPLE 16.

EXAMPLE 16

```
  HAVE                WANT
1Ø GOTO 1Ø          6Ø GOTO 6Ø
2Ø GOTO 2Ø          7Ø GOTO 7Ø
3Ø GOTO 3Ø          8Ø GOTO 8Ø
4Ø GOTO 4Ø          9Ø GOTO 9Ø
```

What we have is in column 1 labeled "HAVE" (clever huh?), and you can guess the rest. How do we go about this? Step 1 is to use the copy command in Electric Pencil to the new location. We then must place the cursor in front of each string and tell Electric Pencil to REPLACE 10 with 60 2 times and the same for 70 and 80 etc. I admit this must seem to be the second hardest way to accomplish this task, next to retyping the lines, but if you don't have a RENUM program it will make things go a little easier.

A suggestion by a new friend of mine can make this kind of problem easier. He suggested that if you number you modules with the first character alphabetic followed by the usual numeric sequence. See EXAMPLE 17.

EXAMPLE 17

```
A1ØØ 'THIS IS THE FIRST LINE OF THE FIRST MODULE
A11Ø 'THIS IS THE SECOND
A12Ø 'GUESS YOU CAN FIGURE OUT WHICH THIS IS

B1ØØ 'THIS IS THE FIRST LINE OF THE SECOND MODULE
B11Ø 'THIS IS THE SECOND OF THE SECOND
B12Ø 'THIS IS THE THIRD
```

When you want to relocate this section of code, all you have to do is find and replace "A", for example with "1" or "2" or whatever digit places the code correctly. Thanks, Jules Greenstein, where ever you are.

Merging lines is very easy. All you have to do is to delete the line numbers of the lines you'll be getting rid of, and then get rid of the carriage return of the line before it. See EXAMPLE 18

(see top of next column)

The problem of merging programs is our old friend RENUM. I suggest that you renumber one of the files you want merged prior to merging. Number it so that there are no conflicts between the two programs. Once

```
EXAMPLE 18
1Ø 'this is line 1Ø
2Ø 'this is line 2Ø

1- delete "2Ø"
2- replace the <CR> at the end of line 1Ø with a " : "

       This will leave you with:

1Ø' this is line 1Ø : this is line 2Ø
```

this is done, save them as ASCII files, and then simply load them into the W.P. buffers and save them together as one file.

Well, that about covers the subject. I admit it isn't perfect, and unless you want to spend some more "dough" on another utility, I suggest you consider this alternative.

Until next time, keep your 80 happy and well fed.

Spencer Koenig
153-27 73 Avenue
Flushing NY 11367 ●

PRACTICAL BUSINESS PROGRAMS
continued from page 27

```
16Ø GOTO 27Ø
17Ø IF LC$="Y" THEN CMD"FORMS(T)"
18Ø LPRINT "BALANCE SHEET":SS#=Ø:SX#=Ø:OPEN"I",1,FF$:W$="NO
         ACCOUNT      DEBIT              CREDIT": LPRINT W$
19Ø FOR I=1 TO SU: INPUT #1,A$: IF I<Z1 OR I>Z3 THEN LPRINT
USING E$;I,A$,X#(I,1),X#(I,2)
2ØØ SS#=SS#+X#(I,1):SX#=SX#+X#(I,2): NEXT I
21Ø LPRINT S$
22Ø LPRINT USING XG$;SS#,SX#: LPRINT XX$
23Ø CLOSE 1: OPEN"O",1,NQ$: FOR I=1 TO SU: FOR Q=1 TO 2:
RO#=INT(X#(I,Q)*1ØØ)/1ØØ:DI#=X#(I,Q)-RO#:X#(I,Q)=RO#: IF
DI#>.ØØ5 THEN X#(I,Q)=X#(I,Q)+.Ø1
24Ø PRINT #1,X#(I,Q);
25Ø NEXT Q
26Ø NEXT I: CLOSE 1:LOAD "GLMENU",R:REM INPUT TO DISK
27Ø Z1=N(1)+N(2):Z2=Z1+N(3):Z3=Z2+N(4):Z1=Z1+1
28Ø FOR I=1 TO SU: IF X#(I,1)>X#(I,2) THEN LET X#(I,1)=
X#(I,1)-X#(I,2):X#(I,2)=Ø
29Ø IF X#(I,1)<=X#(I,2) THEN LET X#(I,2)=X#(I,2)-X#(I,1):
X#(I,1)=Ø
3ØØ NEXT I
31Ø SA#=Ø:CC#=Ø
32Ø FOR I=Z1 TO Z2:SA#=SA#+X#(I,2)-X#(I,1): NEXT I
33Ø FOR I=Z2+1 TO Z3:CC#=CC#+X#(I,1)-X#(I,2): NEXT I
34Ø B$="        PROFIT & LOSS": C$="                PRESENT
                YEAR TO DATE"
35Ø D$="                          AMOUNT  PERCENT    AMOUNT
 PERCENT"
36Ø H$="### %             %  ###,###.##   #.##   ###,###.##
  #.##": LPRINT B$
37Ø LPRINT C$: LPRINT D$
38Ø SS#=Ø:SX#=Ø: OPEN"I",1,NP$
39Ø FOR I=Z1 TO Z3: INPUT #1,A#,B#: S#(I,1)=A#+X#(I,1):
S#(I,2)=B#+X#(I,2):SS#=SS#+S#(I,1 ):SX#=SX#+S#(I,2): NEXT
I: CLOSE 1
4ØØ OPEN"O",1,NZ$: FOR I=Z1 TO Z3: FOR Q=1 TO 2:
RO#=INT(S#(I,Q)*1ØØ)/1ØØ:DI#=S#(I,Q)-RO#:S#(I,Q)=RO#: IF
DI#>.ØØ5 THEN S#(I,Q)=S#(I,Q)+.Ø1
41Ø PRINT #1,S#(I,Q);
42Ø NEXT Q: NEXT I: CLOSE 1
43Ø FOR I=Z1 TO Z2: IF SA#>0 THEN QP(I)=X#(I,2)/SA#
44Ø IF SX#>0 THEN QQ(I)=S#(I,2)/SX#
45Ø NEXT I
46Ø FOR I=Z2+1 TO Z3: IF CC#>0 THEN QP(I)=X#(I,1)/CC#
47Ø IF SS#>0 THEN QQ(I)=S#(I,1)/SS#
48Ø NEXT I
49Ø OPEN"I",1,FF$: FOR I=1 TO Z1-1: INPUT #1,A$: NEXT I
5ØØ FF#=Ø:SD#=Ø: FOR I=Z1 TO Z2: FF#=FF#+S#(I,2):
SD#=SD#+X#(I,2)
51Ø INPUT #1,A$: IF X#(I,2)>0 THEN LPRINT USING H$;I,A$,
X#(I,2),QP(I),S#(I,2),QQ(I) ELSE S#(I,2)=S#(I,2)-X#(I,1):
LPRINT USING H$;I,A$,-X#(I,1),QP(I),S#(I,2),QQ(I)
52Ø NEXT I: LPRINT USING H$;Ø,"TOTAL SALES",SD#,1.,FF#,1.:
LPRINT" "
53Ø FOR I=Z2+1 TO Z3: INPUT #1,A$: IF X#(I,1)>0 THEN LPRINT
USING H$;I,A$,X#(I,1),QP(I),S#(I,1),QQ(I) ELSE S#(I,1)=
S#(I,1)-X#(I,2): LPRINT USING H$;I,A$,-X#(I,2),QP(I),
S#(I,1),QQ(I)
54Ø NEXT I
55Ø PR#=SA#-CC#:S$="                          ----------
      ----------": LPRINT S$
56Ø XG$="                  ##,###,###.##
    ##,###,###.##"
57Ø PS#=SX#-SS#: LPRINT USING XG$;PR#,PS#
58Ø XX$="                          ==========
      ==========": LPRINT XX$
59Ø CLOSE 1: FOR I=Z1 TO Z3: X#(I,1)=Ø: X#(I,2)=Ø: NEXT I:
IF PR#>0 THEN LET X#(SU,2)=X#(SU,2)+PR#
6ØØ IF PR#<Ø THEN X#(SU,1)=X#(SU,1)-PR#
61Ø IF X#(SU,1)>X#(SU,2) THEN LET X#(SU,1)=X#(SU,1)-X#(SU,2):
X#(SU,2)=Ø
62Ø IF X#(SU,2)>=X#(SU,1) THEN LET X#(SU,2)=X#(SU,2)-X#(SU,1):
X#(SU,1)=Ø
63Ø GOTO 17Ø
```

SUMMARY

The CPA program reviewed is the second program in group 2. This program produces a profit and loss statement and a balance sheet. With this new program added to the ones already available you can set up your chart of accounts, input your transactions, and produce a periodic set of accounting reports.

Steven M. Zimmerman, Ph.D.
College of Business and Management Studies
University of South Alabama
Mobile, Alabama 36688

Leo M. Conrad
Imagineering Concepts
P.O. Box 9843
Mobile, Alabama 36691-0843 ■

PRODUCT REVIEW: CASSETTE ENHANCEMENTS
LEMONS TECH SERVICES
KWIK SOFTWARE

Richard Kaplan

When the owner of an overworked cassette system considers purchasing a mass-storage device for his computer, the possibilities are enormous. Aside from the classic diskette drive, there are companies which promote stringy floppies, hard disk drives, tape drives (such as those used with mainframe computers), and even an 8-track cartridge drive to store programs. Unfortunately, each of these systems has one significant drawback: cost. Lemons Tech Services and Kwik Software each offer a unique alternatives; a "cassette loader" for $17.99 to increase cassette reliability and a "cassette operating system" for $26 which can decrease loading time by as much as a factor of 6.

Lemons Tech Services manufactures hardware enhancements for cassette owners; Kwik Software develops software. Although each company actively promotes the other's product and the software and hardware do complement each other, either product may be used independently. Thus, let's first examine the hardware aspect of this unique cassette enhancement.

THE LEMONS TECH LOADER

Lemons Tech offers a variety of loaders, varying in price from $17.99 to $29.99. Models are available which can run on either the Model I or Model III (although a switch may need to be aCpivated for Model III operation at high baud).

Essentially a cassette loader is a "black box" which eliminates the guesswork in setting the volume level when loading programs from cassette. The Lemons Tech loaders are designed to operate with the cassette volume level set at the highest setting. The loader's curcuitry monitors the volume level for you.

Setting It Up

Installing the loader is quite simple. The cassette cable plug which normally is inserted in the "ear" jack should be inserted into the loader; a wire attached to the loader itself plugs into the cassette recorder's ear jack. Once the volume level is adjusted to it highest setting, the computer operator need not know the loader is even set up. No batteries or external power is needed.

(NOTE: The loader's instructions make a point of the fact that the loader is designed for the Radio Shack CTR80 or CTR80A. Resistors are provided for modifying the CTF41 or CCR81, a modification which requires opening these recorders and soldering in new resistors. The loader may not function with other cassette recorders.)

What Does It Do?

The loader by itself does *not* increase program loading speed. Instead, it is designed as a stand-alone product which increases cassette reliability. Difficulties with volume settings and poor reproduction quality are greatly diminished. In addition, a jack for a speaker or earphone is provided to listen to program saves or loads without pulling plugs from the cassette recorder.

OPTIONAL FEATURES

A number of optional features are avilable for a quite nominal cost. For $19.99 ($2 above the standard MODEL LC), the MODEL LCM includes a built-in audio monitor for listening to program saves/loads without a speaker or earphone. Other models include a volume control and/or an audio monitor. Finaly, the "top-of-the-line" MODEL LLQ-2MC has an output jack for copying tapes from one recorder to another.

Another Hardware Addition

Another standalone product available as a cassette enhancement is the SOFTROL SOFT SWITCH for $18.99. This device has two primary functions: (1) An override switch enables freer tape positioning without pulling plugs from your cassette recorder; (2) A delayed motor-off feature automatically places gaps between your programs, allowing for easier program location and a lessened chance of data being damaged at the end of a program, when the motor is turned off.

An Analysis of Lemons Tech

Lemons Tech hardware appears to be very reliable, and it has a price which can't be beat. The instructions provided clearly indicate the manufacturer's competency in the area of cassette storage, noting subtle differences in the operation of the various Radio Shack cassette recorders. (Did you know the CTR80 and 80A allow you to rewind or fast-foward a tape while hooked up to a computer, while the CTR41 does not?) In short, Lemons Tech loaders are highly recommended to any cassette owner who has ever experienced loading difficulties.

THE KWIK SOFTWARE "OPERATING SYSTEM"

The heart of the KWIK Software offerings appears to be the cassette operating system. Actually, there are two different operating systems—*KOS3* for the Model III and *KWICOS* for the Model I. These operating systems can improve cassette loading time as much as 6X, and they add many disk-like commands to cassette BASIC.

Running The Program

KWICOS and KOS3 are machine-language programs which occupy 1700 bytes of memory. Program loading instructions are quite explicit, and a friendly but stern warning message warns the user the the program is

continued on page 61

BOOK REVIEW

Learning TRS-80 BASIC

for the Models I, II/16, and III by David A. Lien

Spencer Koenig

Learning TRS-80 BASIC is 544 pages long and is by far one of the most useful and well written books for the TRS-80 I've had the pleasure to read. I suggest that if your or your childs' school is in need of a well conceived and presented text for the Junior High to High school level (and beyond), then urge them to get this one. Now that you know my opinion on Mr. Lien's book, let me tell you a little about him and the book and why I feel this way.

If you have been involved with the TRS-80 for a little while, then the name David Lien should not be new to you. He wrote the "original Level I manual." The Level I manual set a standard for technical manuals. This standard said that they should be clear, be fun, and most of all be understandable.

This new book, *Learning TRS-80 BASIC* contains lots of fun cartoons that illustrate and point out myths and mistakes to watch out for that every beginner might make or misunderstand. It is by this low humor that it overcomes any computer phobia that the learner could have. This is especially important for the adults working with a computer for the first time. I say this about adults, because the fact is that I have not yet met a child or teenager who couldn't wait to get his hands on one of these amazing imagination machines. They run towards them without any reservations or fears.

Though it is a book on a technical subject, it is still a very entertaining and educational book, for the layman/beginner, on using the TRS-80 and learning BASIC. I am very happy to see that Mr. Lien hasn't abandoned the approach that, after all, made him famous. The fact that he hand holds the reader step by step from the very early stages of computing to the mid-intermediate stages, and does it so well, illustrates his talents as a teacher and writer.

The book is broken down into 5 sections (A through E). Section A is in 8 parts, followed by 4 secondary sections (B through E). In SECTION D we have 8 appendices and in SECTION E we have the INDEX that I wish could have been a part of the original documentation that came with the Model I, way back when. It's this kind of consideration that permeates this book and that makes it a cut above the rest.

Within the 8 major sections, the book is further broken down into 50 much smaller chapters. I like the fact that the chapters are so small because this allows the reader to progress at his or her own pace and not feel that there is too much information to "overcome" in order to get to the end of a chapter. Mental fatigue is a big problem, and this is generally the problem of many textbooks on the subject.

Many times assumptions are made about what the reader knows, and this leads to problems and frustrations. *Learning TRS-80 BASIC* gets around this by having all the information there in these small chapters, and it includes limited problems to solve as you go along, reviewing the material covered. Not only could you learn a great deal about BASIC from this text, but you could also learn quite a bit about teaching as well. This is important for parents, who need a text to help them help their younger children to learn computing. It can't be said about many texts I've come across.

Now for the *tour de force*. Perhaps the most important fact about this book is that it teaches about all of the Tandy machines available at the time of this writing (there are always rumors). The text also contains most of the documentation you'll need to do anything on these machines. The indices are what we've been waiting for from Tandy for a long, long time. Now we finally have them for each of the machines, and in one place.

In section A, part I, we have chapters 1-4. The smallest chapter is 6 pages and the largest is 10. In fact, the largest chapter I could find (at a glance) was about 13 pages in length. This is a good guide to follow for young readers and phobia prone adults. Keep them short and keep them happy ("them" refers to different "them"'s for those who are paranoid).

The subject is taken from the very beginning, starting with turning the computer ON. This section is called COMPUTER ETIQUETTE. It is here that you first come across one of the things I don't like about the book. In trying to cover so much material, Mr. Lien has come up with a novel approach, which I think is a good idea but difficult to do successfully because it is a compromise.

He explains Level II from the point of view of the Model III, which is fine, but in the right margins he inserts any differences to the Model I and the Model II/16, as well as additional information that may be relevant but better introduced as an aside. There are times when all this information becomes confusing and distracting. The text in these margins is also much smaller than the regular text by necessity and therefore harder to read, even though great care was taken in the layout to keep the pages as uncluttered as possible.

Although I recommend this book as a general text for Junior High School and up, I wish there were a larger format available for students (and for me too) that would be easier on the eyes and less constrained on the page.

Meanwhile, back in the chapter, we find that in a few short pages we've had a number of simple examples, 4 new commands, 1 statement, and some miscellaneous information.

Chapter 2, entitled "Expanded Program," does just that. A few more commands, statements and general information lead us to chapter 3, which covers the "EDITOR-First Semester." The pace for introduction of material is very natural and seems to come up quite conversationally. Buzzwords are not overlooked and are explained when encountered. I can't tell you how often I've found texts that explain a subject without explaining some of the language necessary to understand what the author is explaining.

PART II, Chapters 5-8, covers what is perhaps the root of computer phobia, "Math Operators" in 5, "Scientific Notation" in 6, "Order of Operations" in 7 and "Relational Operators" in 8. When I started "teaching computers" in my school, the first question kids asked me was "Do I have to know MATH," and the second was "Do I have to know how to type." All I could say was "NO," because I knew they would learn what they had to know when they had to know it, and it would make the whole experience more meaningful. They might even remember something about it later.

Mr. Lien treats the subject lightly and gives short examples and a problem to solve that is related to his examples. He doesn't really go into much depth (and shouldn't), but he does cover the subject clearly and quickly. This allows the beginner to see the potential and to get onto something more interesting.

Chapters 9-11 deal with input and output via the screen, calculator and immediate mode, and last but not least, cassette and disk commands and usage. There is no real depth to these, but enough information is given to allow the beginner to work with his equipment and at least get the job done.

From chapter 12 on, we first get into the meat of real programming through "For-Next Looping" with a brief introduction to incrementing. (He uses some of his earlier math examples again to show some applications of the new command—nice touch!) Chapter 13 expands on the For-Next with four new commands (List ###, Run ###, Delete ### and Cont).

Chapter 14 picks up on the Tab (sorry) function for pretty printing to the screen, and chapter 15 again expands on the For-Next. In 15, however, we come across the first introduction of material that we call "Programming technique." Here is the first mention of considerations for making programs more readable (line indentation in loops). I don't recall seeing this concept in Mr. Lien's earlier versions (I could be wrong), but I think it's extremely important that these kinds of considerations be demonstrated to the beginner early on.

Chapter 16 covers the integer function fully. Chapter 17 talks about "More Branching Statements" used in conjunction with the earlier INT function. Here we get to see some of the more powerful commands the BASIC language has to offer. I didn't think that Mr. Lien was careful enough at this point. Introduction of GOTO's these days is a touchy subject, and more could have been said and demonstrated about careful use of these commands without going into too much detail. The concept of subroutines and "CALLS" is also discussed here in a very cursory manner.

Chapter 17 could have been called "FUN with Random," but instead it's just plain old "Random Numbers." Demonstrations of the famous game coin flip fill out this chapter.

The final chapter in section 2 covers "READing DATA." Introduced are the ideas of; how to use the READ data statement; restoring data and reading string data. Some examples show the famous OD error and what to do about it.

Section 3 begins with the often entangled subject of "STRINGS." It's at this point that the learner starts to get more involved with the examples as they get larger and more complex. Chapter 20 is entitled "Intermediate BASIC" and re-covers most of the material already demonstrated with some new twists. Multi-statement lines, string variables, some short hand (? instead of Print) and the use of some special key to get around the keyboard are explained.

Chapter 21 covers "The Editor - Second Semester" and greatly expands on Chapter 3. At this point the user is beginning to feel like a real hacker, solving problems and getting the computer under control.

Chapters 22 and 23 start going into details of string handling, a subject that is more than just learning to use the TRS-80. ASCII is explained, and how the TRS-80 handles them is followed by "Strings in General."

Chapters 24, 25 and 26 start the real subject matter as we learn how to begin manipulating strings by taking them apart and putting them together from the beginning of the string, middle or end.

Well, by this time if you're not hooked on computing you never will be, and so it's time to start getting back into the dreaded subject of "MATH." Part 4, chapter 27 covers "what price precision?" The overhead of double precision is reviewed as is the many ways of doing similar numeric gymnastics. Introduced are the commands INT(n), FIX(n), ABS(n), LOG(n), and EXP(n). The chapter is mostly "try it and see." A little more depth is left to the last chapter of this section which include some of the TRIG functions.

Now, we finally come to the all important subject "Graphics and Display Formatting." We finally get to draw pictures using all those math routines we've had to endure. Chapter 30 discusses the video display. Chapter 31 covers line drawing using variables and introduces a little bit of animation. Chapter 33 explores the power of the Print@ function and 34 gets back to those wonderful TRIG functions I'm sure everybody skipped the first time.

Chapter 34 covers "Point," 35 covers getting a response into the computer via "INKEY$," and applications of these two commands are combined with "PRINT USING" for those of you with business computing in mind in chapters 36 and 37.

Surprisingly, in a whole section (6) of its own, we come to a topic that is usually covered very lightly and left to the reader to fend for himself. The subject is "ARRAYS" (chapter 38), how to manipulate them ("SEARCH and SORT" in chapter 39), and then how to use multi-dimensional arrays (chapter 40).

Chapters 41 and 42 get back to the fun stuff re-combining the older material for some advanced graphics techniques. Although this text isn't meant to really teach graphics, it certainly goes far to whet your appetite. There is a reference to an excellent book for graphics available at Radio Shack, cleverly entitled *TRS-80 GRAPHICS for the Model I and Model III.*

The remaining sections (7 and 8) of the book cover miscellaneous topics of using POKEs and PEEKs, memory organization, logical operators and USR functions. Section 8, for me, was a pleasant surprise, because it talked about "Program Control" but really discussed programmer control. In other words, it covered techniques usually reserved to other books, flow charting, debugging techniques, and figuring out what those error messages really meant.

Section B has answers to all those problems to be solved, followed by section C, which contains a number

PIRATE SOFTWARE, HBO DISHES, OFF-THE-AIR VTR, AND OTHER ALLEGED "CRIMES" OF THE COMPUTER AGE

Mike Shadick

To hear some say it, it would seem that the age of personal computery and video is only ushering in a veritable phlange of ways to either break, bypass, or slide by the electronically foreshortened arm of the law. Yet every one of the so-called crimes referred to in this editorial's title—not to mention many others, as well—are not in the least criminal from the standpoint of crimes *only in the minds of the profiteers.*

Who are the profiteers? Those who stand to gain greatly from the very computer/video revolution they seem so staunchly to oppose! Instead of meeting the new breed of consumer in the marketplace, the profiteers appear to prefer seeing him or her in court!

In effect, the profiteers are saying (if not shouting!), "Creative use of home computery and video is dipping into our profits, and we're gonna fight it with everything we've got!"

Personally, I find it very difficult to feel any compassion for them whatsoever.

Take the issue of "illicit" software duplication (pirating). There is one sure way in which profiteering software suppliers can not only get "one up" on the pirating game, but can even beat it! And what is that one way? By making frequent *improvements* in their software prorams, thereby assuring that whomever wants the best, the latest, the most up-to-date state-of-the-art programs available, will have no choice but to *buy* them.

An effective solution to the pirating problem, is it not? Yet it would seem that some software suppliers would rather bellyache about what they wish was, instead of creating their own ongoing reality in the form of frequent program improvements.

Well, so much for software suppliers. Heaven knows we need them! What we *don't* need is their "victim" laments. *And neither do they.*

Now let's move right along to home microwave antennas—the ubiquitous rooftop or balcony dishes. To hear the Home Box Office organization and others tell it, you'd think that the dishes were driving them all to the poorhouse! In the cable profiteers' attempts to keep their signals out of unpaying hands, they're trying all sorts of signal-scrambling techniques, only to be one-upped by home dish owners who are scrambling (or should I say, *un*scrambling), even faster.

Yet it is difficult to feel any compassion for the cable companies' "plight." After all, I've yet to find it inscribed in the Good Book (or any other), "Thou shalt send thy signal via microwave, and only microwave shalt thou use." Pardon my French, but *Hell,* there are any number of ways that HBO and the other such services could be getting their signals from here to there. The fact that they've chosen to use microwave relays is certainly no valid reason to point a blaming finger at home video buffs who also enjoy receiving the microwave media.

To accuse dish owners of signal pirating, then, is a lot like faulting Ma Bell's customers for putting modems on their phones. As long as nothing is screwed up, the phone lines are for the customer's use, in any way that he or she wishes.

The same is true of dishes, or it *should* be.

One final microjab, this one levelled at the foes of home video recording. You know—the TV networks, prerecorded cassette suppliers, and others who are trying (and in some cases apparently succeeding) to make it illegal to tape copyrighted material off the air, which is just about everything on the tube.

Their courtroom antics remind one of the teetotaling bar-smashers just before the enactment of Prohibition. Do the networks and others actually believe that they can legislate home video recording out of existence? For that's the only way in which their archaic wishes could ever come true!

Here, again, the answer seems obvious. The best way for the profiteers to "beat" the home video revolution is to plunge whole hog into the business of selling blank tapes! That's where a sizable chunk of their long-term profits lie. But it seems that they'd rather just bellyache.

So as I see it, my microfriends, the burden of proof as to what does or does not constitute home computer/video "crimes" rests not with the alleged perpetrators, but rather with the profiteers. If they're just too lazy (or intransigent) to go with the flow and thereby *benefit* from the new tools of a new age—well, then, that's just too bad for them.

At any rate, they have no right to take *their* problems out on *us.* American know-how and ingenuity belongs in the plants and on the drawing boards, not in the courts.

Mike Shadick
Cedar Square West, Apt. E-414
1515 South Fourth Street
Minneapolis, MN 55454 ■

(Editor's note: the views expressed in this article are not necessarily those of H & E Computronics, Inc.)

---

of program examples to study and take apart with your new found skills.

The remaining APPENDIX's A-G are real treasures, covering the differences between the three and a half machines from DISK BASIC to ASCII characters as well as most of the various special features that each machine contains.

All in all, this is an amazing accomplishment. To have so much information so well covered certainly puts most of the texts on the same subject to shame. The fact that all of the Tandy machines are covered so well makes it three times the bargain. You can be sure that, had it been a different author with lower standard, you'd have to buy three or four books at the same price as this one to get half the information. *Learning TRS-80 BASIC for the Models I, II/16, and III* is perhaps the computer book buy of decade.

Spencer Koenig
153-27 73 Avenue
Flushing, NY 11367 ■

Price: $19.95, Available from Compusoft Publishing, H & E Computronics (800-431-2818), and various computer and book stores.

SPACEX

Steve Brown

This unique game program is relatively short and with very simple modifications can run on any TRS-80. These modifications would be made on the calls to the built-in ROM routines (0060H and 01C9).

Once the game has started, erroneous data may not be entered because the input comes from the keyboard and only valid responses are checked.

The sound effects are fairly good and the graphics are excellent.

From the user's standpoint this program is very easy to use. After getting the program loaded the user presses the left arrow key to move left, the right arrow key to move right, or the space bar to fire. There are no complicated formulas to remember.

The game is tantalizing. The closer you get to winning the more difficult the game becomes. The 3 levels of difficulty and the shot counter at the end of the game allow everyone to play at a level they feel comfortable with.

PROGRAM ORGANIZATION

Basically the program consists of a control section, several major subroutines, and several minor subroutines (which may be thought of as simple functions—for example, INC40H increments the HL register pair 40H times).

USERS GUIDE

255 bombs will be spit out by the mother ship. They move horizontally until they are directly above the correct bomb "chute" then they move vertically into the bomb chute. When a bomb chute is full (contains 5 bombs) a bomb is dropped vertically. If the bomb hits your gun, the resulting explosion will destroy the planet. You may either shoot the bombs or move horizontally to avoid being hit by bombs.

UFOs will randomly appear at the far right or left side of the screen. They slowly drop vertically. If they reach the planet (the bottom of the screen) they explode and destroy the planet.

Your gun moves horizontally across the planet. To move left, press the left arrow key. The gun will continue to move left until you release the key. The right arrow key moves the gun to the right.

To fire your gun, press the space bar. The gun will continue to fire until you release the space bar. A count is kept of the number of shots fired and is displayed at the end of the game. Also the number of shots fired in the best game so far today is displayed at the end of the game.

To win the game you must shoot all 255 bombs or avoid being hit by them, and shoot all UFOs before they hit the ground.

To start the game:

1) Type "SYSTEM" and press ENTER
2) Type "SPACEX" and press ENTER
3) Load the tape
4) When the program has loaded, type "/18688" and press ENTER
5) Press "1" for an easy game, "2" for a medium game, or "3" for a hard game.

In playing the game, "U" represents your gun, "*" a bomb, "%" a UFO, and the mother ship and bomb chute are shown by graphic blocks.

VARIABLES

BEST: 4 bytes—Used by NEWWIN—Fewest number of shots taken to win a game (so far today).

BOMBFL: 5 Bytes—Used by FIGMOV and FIRE—Number of moves downward before bomb reaches the bottom of the screen. Referenced after a bomb chute is full (contains 5 bombs). There are 5 bomb chutes and each byte corresponds to a bomb chute.

CURBMB: 2 bytes—Used by FIGMOV—Current bomb address (screen location).

CURCOL: 1 byte—Used by COLADD and FIGMOV—Bomb chute that the current bomb is moving to.

CURLFT: 5 bytes—Used by FIGMOV—Number of downward moves before bottom of chute is reached. There are 5 chutes and each byte corresponds to a chute.

GUNADR: 2 bytes—Used by UMOVE, FIRE, and FIGMOV—Gun address (screen location).

GUNPOS: 1 byte—Used by UMOVE and FIRE—Relative gun position left to right, greater than 0 and less than 31.

LTOGO: 1 byte—Used by FIGMOV—Number of moves to the left before the bomb will be directly above the appropriate bomb chute. This variable is loaded into LTOGOL on the first move.

LTOGOL: 1 byte—Used by FIGMOV—Value is loaded from LTOGO. Then is decremented each time the bomb moves left until it is equal to zero (at this point the bomb is directly above the appropriate bomb chute).

SHOTS: 4 bytes—Used by FIRE and NEWWIN—Number of shots fired in current game. ASCII representation of a decimal number.

TMPCOL: 1 byte—Used by FIRE—Used to retain CURCOL until it is needed by the FIRE routine.

TMPLFT: 5 bytes—Used by FIGMOV—Value is loaded from CURLFT. Then is decremented each time the bomb moves downward until it is equal to zero (at this point the bomb is at the appropriate position in the chute).

UFOCTR: 1 byte—Used by RNDUFO—The RNDUFO routine is executed every eighth time it is called. This variable keeps count of how many times its been called since it was last executed.

UF1ADR: 2 bytes—Used by RNDUFO and FIRE—UFO #1 address (screen location) where UFO #1 is located.

UF1CTR: 1 byte—Used by RNDUFO—Number of downward moves before UFO #1 reaches the bottom of the screen.

UF2ADR: 2 bytes—Used by RNDUFO and FIRE—UFO #2 address (screen location) where UFO #2 is located.

UF2CTR: 1 byte—Used by RNDUFO—Number of downward moves before UFO #2 reaches the bottom of the screen.

```
                ORG     4900H
DELAY           EQU     0060H
CLEAR           EQU     01C9H
BEGIN           CALL    INIT
MAIN            CALL    FIGMOV
                CALL    FIGMOV
                CALL    FIGMOV
                CALL    FIGMOV
                CALL    UMOVE
                CALL    FIRE
                CALL    RNDUFO
                JP      MAIN
L1              CALL    CLEAR
                CALL    NEWWIN
                LD      HL,MSG3
                CALL    WORL
                LD      HL,WINMSG
                LD      DE,3D10H
                LD      BC,0018H
                LDIR
                LD      HL,CHAMP
                LD      DE,3D90H
                LD      BC,0018H
                LDIR
                LD      HL,RESTRT
                LD      DE,3E90H
                LD      BC,000BH
                LDIR
L2              LD      A,(3840H)
                BIT     0,A
                JR      Z,L2
                JP      BEGIN
L3              CALL    CLEAR
                LD      HL,MSG2
                CALL    WORL
                JP      BEGIN
INIT            LD      HL,3C28H
                LD      (CURBMB),HL
                LD      HL,LTOGO
                LD      (HL),16
                INC     HL
                LD      (HL),16
                INC     HL
                LD      (HL),0
                INC     HL
                LD      B,10
L4              LD      (HL),5
                INC     HL
                DJNZ    L4
                CALL    CLEAR
                LD      HL,GUNPOS
                LD      (HL),16
                INC     HL
                LD      (HL),0
                INC     HL
                LD      (HL),0
                INC     HL
                LD      (HL),4
                LD      HL,BOMBFL
                LD      B,5
L5              LD      (HL),8
                INC     HL
                DJNZ    L5
                LD      HL,SHOTS
                LD      B,4
L6              LD      (HL),30H
                INC     HL
                DJNZ    L6
                CALL    CLEAR
L7              LD      HL,EASY
                LD      DE,3D10H
                LD      BC,0008H
                LDIR
                LD      HL,MED
                LD      DE,3D50H
                LD      BC,000AH
                LDIR
                LD      HL,HARD
                LD      DE,3D90H
                LD      BC,0008H
                LDIR
L8              LD      A,(3810H)
                OR      A
                JR      Z,L8
                BIT     1,A
                JR      Z,L9
                LD      HL,FIGMOV+1
                LD      (HL),0BH
                LD      HL,L31+1
                LD      (HL),7AH
                LD      HL,L33+1
                LD      (HL),6
                JP      L11
L9              BIT     2,A
                JR      Z,L10
                LD      HL,FIGMOV+1
                LD      (HL),9
                LD      HL,L31+1
                LD      (HL),77H
                LD      HL,L33+1
                LD      (HL),8
                JP      L11
L10             BIT     3,A
                JP      Z,L8
                LD      HL,FIGMOV+1
                LD      (HL),7
                LD      HL,L31+1
                LD      (HL),74H
                LD      HL,L33+1
                LD      (HL),0AH
L11             CALL    CLEAR
                LD      A,8
                OUT     (0FFH),A
                LD      HL,3C4CH
                LD      B,4
                LD      A,0BFH
L12             PUSH    HL
                POP     IX
                LD      (IX+0),A
                LD      (IX+8),A
                LD      (IX+16),A
                LD      (IX+24),A
                CALL    INC40H
                DJNZ    L12
                LD      HL,3C2AH
                LD      (HL),0BBH
                INC     HL
                INC     HL
                LD      (HL),0BFH
                INC     HL
                INC     HL
                LD      (HL),0B7H
                LD      HL,3C3AH
                LD      (HL),32H
                INC     HL
                INC     HL
                LD      (HL),35H
                INC     HL
                INC     HL
                LD      (HL),35H
                LD      HL,3FE0H
                LD      (GUNADR),HL
                LD      (HL),55H
                RET
FIGMOV          LD      B,1
                CALL    DELAY
                LD      A,20H
                LD      (3D88H),A
                LD      (3D90H),A
                LD      (3D98H),A
                LD      (3DA0H),A
                LD      (3DA8H),A
                LD      HL,CURLFT
                CALL    COLADD
                OR      A
                JP      Z,L19
                LD      HL,LTOGOL
                LD      A,(HL)
                OR      A
                JR      Z,L13
                LD      HL,(CURBMB)
                LD      A,28H
                CP      L
                CALL    Z,CNTER
                LD      HL,(CURBMB)
                LD      (HL),20H
                DEC     HL
                DEC     HL
                LD      (HL),2AH
                LD      (CURBMB),HL
                LD      HL,LTOGOL
                DEC     (HL)
                RET
L13             LD      HL,(CURBMB)
                LD      A,28H
                CP      L
                JR      NZ,L14
                LD      A,3CH
                CP      H
                JR      NZ,L14
                CALL    CNTER
L14             LD      HL,TMPLFT
                CALL    COLADD
                OR      A
                JR      Z,L15
                DEC     (HL)
                LD      HL,(CURBMB)
                LD      (HL),20H
                CALL    INC40H
                LD      (HL),2AH
                LD      (CURBMB),HL
                RET
L15             LD      HL,3C28H
                LD      (CURBMB),HL
                LD      HL,CURLFT
                LD      DE,CURCOL
                LD      A,(DE)
                ADD     A,L
                LD      L,A
                DEC     (HL)
                LD      A,(HL)
```

```
Ø215Ø         CALL   INCØ5H
Ø216Ø         LD     (HL),A
Ø217Ø L16     LD     HL,CURCOL
Ø218Ø         INC    (HL)
Ø219Ø         LD     A,(HL)
Ø221Ø         CP     5
Ø222Ø         JR     NZ,L17
Ø223Ø         LD     (HL),Ø
Ø224Ø L17     LD     HL,LTOGO
Ø225Ø         LD     A,(HL)
Ø226Ø         OR     A
Ø227Ø         JR     NZ,L18
Ø228Ø         LD     (HL),14H
Ø229Ø L18     DEC    (HL)
Ø23ØØ         DEC    (HL)
Ø231Ø         DEC    (HL)
Ø232Ø         DEC    (HL)
Ø233Ø         LD     A,(HL)
Ø234Ø         INC    HL
Ø235Ø         LD     (HL),A
Ø236Ø         RET
Ø237Ø L19     LD     HL,BOMBFL
Ø238Ø         CALL   COLADD
Ø239Ø         LD     B,A
Ø24ØØ         LD     D,H
Ø241Ø         LD     E,L
Ø242Ø         LD     HL,3D88H
Ø243Ø         LD     A,(CURCOL)
Ø244Ø         SLA    A
Ø245Ø         SLA    A
Ø246Ø         SLA    A
Ø247Ø         ADD    A,L
Ø248Ø         LD     L,A
Ø249Ø         LD     A,8
Ø25ØØ         SUB    B
Ø251Ø         LD     B,A
Ø252Ø L2Ø     CALL   INC4ØH
Ø253Ø         DJNZ   L2Ø
Ø254Ø         LD     (HL),2ØH
Ø255Ø         CALL   INC4ØH
Ø256Ø         LD     (HL),2AH
Ø257Ø         EX     DE,HL
Ø258Ø         DEC    (HL)
Ø259Ø         EX     DE,HL
Ø26ØØ         RET    NZ
Ø261Ø         LD     (HL),2ØH
Ø262Ø         CALL   INC4ØH
Ø263Ø         LD     (HL),2ØH
Ø264Ø         PUSH   HL
Ø265Ø         LD     HL,Ø28ØH
Ø266Ø         CALL   SOUND
Ø267Ø         POP    HL
Ø268Ø         DEC    HL
Ø269Ø         DEC    HL
Ø27ØØ         LD     DE,(GUNADR)
Ø271Ø         LD     A,E
Ø272Ø         CP     L
Ø273Ø         JP     Z,L45
Ø274Ø         INC    L
Ø275Ø         INC    L
Ø276Ø         CP     L
Ø277Ø         JP     Z,L45
Ø278Ø         INC    L
Ø279Ø         INC    L
Ø28ØØ         CP     L
Ø281Ø         JP     Z,L45
Ø282Ø         LD     HL,BOMBFL
Ø283Ø         CALL   COLADD
Ø284Ø         LD     (HL),8
Ø285Ø         JP     L16
Ø286Ø UMOVE   LD     A,(384ØH)
Ø287Ø         BIT    5,A
Ø288Ø         JR     Z,L21
Ø289Ø         LD     A,(GUNPOS)
Ø29ØØ         OR     A
Ø291Ø         RET    Z
Ø292Ø         LD     HL,GUNPOS
Ø293Ø         DEC    (HL)
Ø294Ø         LD     HL,(GUNADR)
Ø295Ø         LD     (HL),2ØH
Ø296Ø         DEC    HL
Ø297Ø         DEC    HL
Ø298Ø         LD     (HL),55H
Ø299Ø         LD     (GUNADR),HL
Ø3ØØØ         RET
Ø3Ø1Ø L21     BIT    6,A
Ø3Ø2Ø         RET    Z
Ø3Ø3Ø         LD     HL,GUNPOS
Ø3Ø4Ø         LD     A,(HL)
Ø3Ø6Ø         CP     31
Ø3Ø7Ø         RET    Z
Ø3Ø8Ø         INC    (HL)
Ø3Ø9Ø         LD     HL,(GUNADR)
Ø31ØØ         LD     (HL),2ØH
Ø311Ø         INC    HL
Ø312Ø         INC    HL
Ø313Ø         LD     (HL),55H
Ø314Ø         LD     (GUNADR),HL
Ø315Ø         RET
Ø316Ø FIRE    LD     A,(384ØH)
Ø317Ø         BIT    7,A
Ø318Ø         RET    Z
Ø319Ø         CALL   SHTCNT
Ø32ØØ         LD     HL,(GUNADR)
Ø321Ø         CALL   DEC4ØH
Ø322Ø         LD     B,Ø9H
Ø323Ø L22     PUSH   BC
Ø324Ø         LD     (HL),22H
Ø325Ø         LD     B,2
Ø326Ø         CALL   DELAY
Ø327Ø         LD     (HL),2ØH
Ø328Ø         CALL   DEC4ØH
Ø329Ø         POP    BC
Ø33ØØ         DJNZ   L22
Ø331Ø         LD     A,(GUNPOS)
Ø332Ø         LD     HL,TMPCOL
Ø333Ø         LD     (HL),1
Ø334Ø         LD     HL,3D4ØH
Ø335Ø         LD     DE,ØØØ8H
Ø336Ø         LD     B,5
Ø337Ø         LD     C,Ø
Ø338Ø L23     INC    C
Ø339Ø         INC    C
Ø34ØØ         INC    C
Ø341Ø         INC    C
Ø342Ø         ADD    HL,DE
Ø343Ø         CP     C
Ø344Ø         JP     Z,L25
Ø345Ø         PUSH   HL
Ø346Ø         LD     HL,TMPCOL
Ø347Ø         INC    (HL)
Ø348Ø         POP    HL
Ø349Ø         DJNZ   L23
Ø351Ø         CP     2
Ø352Ø         JP     Z,L24
Ø354Ø         CP     26
Ø355Ø         RET    NZ
Ø356Ø         LD     A,(UF1CTR)
Ø357Ø         OR     A
Ø358Ø         RET    Z
Ø359Ø         LD     HL,UF1CTR
Ø36ØØ         LD     (HL),Ø
Ø361Ø         LD     HL,(UF1ADR)
Ø362Ø         LD     (HL),2ØH
Ø363Ø         LD     HL,Ø88ØH
Ø364Ø         CALL   SOUND
Ø365Ø         RET
Ø366Ø L24     LD     A,(UF2CTR)
Ø367Ø         OR     A
Ø368Ø         RET    Z
Ø369Ø         LD     HL,UF2CTR
Ø37ØØ         LD     (HL),Ø
Ø371Ø         LD     HL,(UF2ADR)
Ø372Ø         LD     (HL),2ØH
Ø373Ø         LD     HL,Ø88ØH
Ø374Ø         CALL   SOUND
Ø375Ø         RET
Ø376Ø L25     LD     A,(HL)
Ø378Ø         CP     2ØH
Ø379Ø         RET    Z
Ø38ØØ         PUSH HL
Ø381Ø         LD     HL,Ø1Ø4H
Ø382Ø         CALL   SOUND
Ø383Ø         POP    HL
Ø384Ø         LD     B,4
Ø385Ø L26     PUSH   BC
Ø386Ø         LD     D,H
Ø387Ø         LD     E,L
Ø388Ø         CALL   DEC4ØH
Ø389Ø         LD     A,(HL)
Ø39ØØ         LD     (DE),A
Ø391Ø         POP    BC
Ø392Ø         DJNZ   L26
Ø393Ø         LD     (HL),2ØH
Ø394Ø         LD     A,(TMPCOL)
Ø395Ø         LD     B,A
Ø396Ø         LD     HL,CURLFT-1
Ø397Ø         LD     DE,BOMBFL-1
Ø398Ø L27     INC    HL
Ø399Ø         INC    DE
Ø4ØØØ         DJNZ   L27
Ø4Ø1Ø         INC    (HL)
Ø4Ø2Ø         CALL   INCØ5H
Ø4Ø3Ø         INC    (HL)
Ø4Ø4Ø         LD     A,(DE)
Ø4Ø5Ø         AND    8
Ø4Ø6Ø         RET    NZ
Ø4Ø7Ø         LD     A,8
Ø4Ø8Ø         LD     (DE),A
Ø4Ø9Ø         JP     L16
Ø41ØØ         RET
Ø411Ø CNTER   LD     HL,3C3EH
Ø412Ø         LD     A,(HL)
Ø414Ø         CP     48
Ø415Ø         JR     Z,L28
Ø416Ø         DEC    (HL)
Ø417Ø         RET
Ø418Ø L28     LD     (HL),39H
Ø419Ø         DEC    HL
Ø42ØØ         DEC    HL
Ø421Ø         LD     A,(HL)
Ø422Ø         CP     48
Ø423Ø         JR     Z,L29
Ø424Ø         DEC    (HL)
```

```
0425Ø          RET
0426Ø L29      LD      (HL),39H
0427Ø          DEC     HL
0428Ø          DEC     HL
0429Ø          LD      A,(HL)
0430Ø          CP      48
0431Ø          JP      Z,L1
0432Ø          DEC     (HL)
0433Ø          LD      HL,FIGMOV+1
0434Ø          DEC     (HL)
0435Ø          DEC     (HL)
0436Ø          LD      HL,L31+1
0437Ø          DEC     (HL)
0438Ø          DEC     (HL)
0439Ø          LD      HL,L33+1
0440Ø          DEC     (HL)
0441Ø          DEC     (HL)
0442Ø          RET
0443Ø RNDUFO   LD      HL,UFOCNT
0444Ø          LD      A,(HL)
0445Ø          OR      A
0446Ø          JR      Z,L3Ø
0447Ø          DEC     (HL)
0448Ø          RET
0449Ø L3Ø      LD      (HL),8
0450Ø          LD      HL,UF2CTR
0451Ø          LD      A,(HL)
0452Ø          OR      A
0453Ø          JP      NZ,L34
0454Ø          LD      A,R
0455Ø L31      CP      74H
0457Ø          JP      M,L32
0458Ø          LD      (HL),14
0459Ø          LD      HL,3C04H
0460Ø          LD      (UF2ADR),HL
0461Ø L32      LD      HL,UF1CTR
0462Ø          LD      A,(HL)
0463Ø          OR      A
0464Ø          JP      NZ,L35
0465Ø          LD      A,R
0466Ø L33      CP      1Ø
0468Ø          RET     P
0469Ø          LD      (HL),14
0470Ø          LD      HL,3C34H
0471Ø          LD      (UF1ADR),HL
0472Ø          RET
0473Ø L34      LD      HL,(UF2ADR)
0474Ø          LD      (HL),2ØH
0475Ø          CALL    INC4ØH
0476Ø          LD      (HL),25H
0477Ø          LD      (UF2ADR),HL
0478Ø          LD      HL,UF2CTR
0479Ø          DEC     (HL)
0480Ø          LD      A,(HL)
0481Ø          LD      DE,(UF2ADR)
0482Ø          OR      A
0483Ø          JP      Z,L45
0484Ø          JR      L32
0485Ø L35      LD      HL,(UF1ADR)
0486Ø          LD      (HL),2ØH
0487Ø          CALL    INC4ØH
0488Ø          LD      (HL),25H
0489Ø          LD      (UF1ADR),HL
0490Ø          LD      HL,UF1CTR
0491Ø          DEC     (HL)
0492Ø          LD      A,(HL)
0493Ø          LD      DE,(UF1ADR)
0494Ø          OR      A
0495Ø          JP      Z,L45
0496Ø          RET
0497Ø SOUND    LD      E,Ø
0498Ø          LD      D,H
0499Ø          SET     7,D
0500Ø          LD      A,9
0501Ø L36      LD      B,L
0502Ø L37      OUT     (ØFFH),A
0503Ø          DEC     DE
0504Ø          BIT     7,D
0505Ø          RET     Z
0506Ø          DJNZ    L37
0507Ø          CPL
0508Ø          OR      8
0509Ø          AND     11
0510Ø          JR      L36
0513Ø INC4ØH   PUSH    DE
0514Ø          LD      DE,ØØ4ØH
0515Ø          SCF
0516Ø          CCF
0517Ø          ADC     HL,DE
0518Ø          POP     DE
0519Ø          RET
0520Ø DEC4ØH   LD      B,4ØH
0521Ø L38      DEC     HL
0522Ø          DJNZ    L38
0523Ø          RET
0524Ø COLADD   LD      DE,CURCOL
0525Ø          LD      A,(DE)
0526Ø          ADD     A,L
0527Ø          LD      L,A
0528Ø          LD      A,(HL)
0529Ø          RET
0530Ø INCØ5H   INC     HL
0531Ø          INC     HL
0532Ø          INC     HL
0533Ø          INC     HL
0534Ø          INC     HL
0535Ø          RET
0545Ø WORL     LD      BC,ØØØCH
0546Ø          LD      DE,3D1ØH
0547Ø          LDIR
0548Ø          LD      B,2ØH
0549Ø L39      PUSH    BC
0550Ø          CALL    DELAY
0551Ø          POP     BC
0552Ø          DJNZ    L39
0553Ø          RET
0554Ø SHTCNT   LD      HL,SHOTS+3
0555Ø L4Ø      LD      A,(HL)
0557Ø          CP      39H
0558Ø          JR      Z,L41
0559Ø          INC     (HL)
0560Ø          RET
0561Ø L41      LD      (HL),3ØH
0562Ø          DEC     HL
0563Ø          JR      L4Ø
0564Ø NEWWIN   LD      B,4·
0565Ø          LD      HL,SHOTS
0566Ø          LD      DE,BEST
0567Ø L42      LD      A,(HL)
0568Ø          LD      C,A
0569Ø          LD      A,(DE)
0570Ø          CP      C
0571Ø          JR      Z,L43
0572Ø          JP      P,L44
0573Ø          RET     M
0574Ø L43      INC     HL
0575Ø          INC     DE
0576Ø          DJNZ    L42
0577Ø L44      LD      HL,SHOTS
0578Ø          LD      DE,BEST
0579Ø          LD      BC,ØØØ4H
0580Ø          LDIR
0581Ø          RET
0582Ø L45      LD      B,64
0583Ø          LD      A,128
0584Ø          PUSH    DE
0585Ø          POP     IX
0586Ø          DEC     IX
0587Ø          DEC     IX
0588Ø L46      LD      (IX+Ø),A
0589Ø          LD      (IX+2),A
0590Ø          LD      (IX+4),A
0591Ø          PUSH    BC
0592Ø          PUSH    AF
0593Ø          LD      HL,Ø38ØH
0594Ø          CALL    SOUND
0595Ø          POP     AF
0596Ø          POP     BC
0597Ø          INC     A
0598Ø          DJNZ    L46
0604Ø          JP      L3
0606Ø CURBMB   DEFS    2
0607Ø LTOGO    DEFS    1
0608Ø LTOGOL   DEFS    1
0609Ø CURCOL   DEFS    1
0610Ø CURLFT   DEFS    5
0611Ø TMPLFT   DEFS    5
0612Ø GUNADR   DEFS    2
0613Ø GUNPOS   DEFS    1
0614Ø UF2CTR   DEFS    1
0615Ø UF1CTR   DEFS    1
0616Ø UFOCNT   DEFS    1
0617Ø UF2ADR   DEFS    2
0618Ø UF1ADR   DEFS    2
0619Ø BOMBFL   DEFS    5
0620Ø TMPCOL   DEFS    1
0621Ø WINMSG   DEFM    '# OF SHOTS U TOOK : '
0622Ø SHOTS    DEFS    4
0623Ø CHAMP    DEFM    'BEST SO FAR TODAY : '
0624Ø BEST     DEFM    '9999'
0625Ø RESTRT   DEFM    'PRESS ENTER'
0626Ø EASY     DEFM    '1 - EASY'
Ø6262 MED      DEFM    '2 - MEDIUM'
Ø6264 HARD     DEFM    '3 - HARD'
0627Ø MSG2     DEFM    'U LOSE HUMAN'
0628Ø MSG3     DEFM    'U WIN HUMAN'
0629Ø          END     BEGIN
```

SELF-MODIFYING CODE

FIGMOV+1: Speed variable—This byte controls the speed of the game.

L31+1: Controls when UFO #1 will appear on the screen.

L33+1: Controls when UFO #2 will appear on the screen.

Steve Brown
3740 Park South Ct., Apt. #7
Topeka, Kansas 66609 ■

GOOD PROGRAMS DO IT ON A DISK

Ken Hipple

So you've finally done it. You went out and got yourself a disk drive or two. Just think, no more forced coffee breaks while waiting for Galaxy Invasion or an Adventure to load. Let's see, all you need to do is use the TAPE command to load the program from tape onto a disk and you're ready to go. For the last time you wait for the program to load from tape. There, it's done. Now all you need to do is type the name, press ENTER, and in no time you are running around trying to shoot a bunch of robots. What do you mean you're not shooting any robots? What do you mean the system reboots every time you try to discover the secret of the Fun House? You mean you didn't know that most tape-based machine language programs won't run from a disk? Well, most won't, but there is an easy way to correct this problem.

THE PROBLEM

The problem is that most of these programs load into memory at a point where they write over part of the disk operating system. When this occurs something is going to happen and it usually isn't going to be what you want. You can relocate the program so that it loads above the DOS but it probably still won't run since most machine language programs will only work if they are loaded into the right locations. What you could do after relocating the program is to change all jumps, calls, and other such commands to reflect the program's new position in memory but, trust me, you don't want to try this. So what is left to do? I'll give you a hint. Once the program you want to run from disk is loaded into memory you don't need the DOS anymore. Now that you know what to do to get Pyramid 2000 onto disk, let me talk about how to get cat hair out of your keyboard. What? You still don't understand about the disk? Well, maybe I'll talk about cat hair next time.

THE SOLUTION

The process for getting a tape-based program to run from disk is easy (it may not sound like it but it is). As a guess, I would say that 99 to 100 percent of all machine language programs that you can get loaded onto disk can be made to run from disk using the following method. The first thing to do is to find the loading address and entry point. Next, the program is relocated to a position above the DOS. Now a short section of code is added to the program and the new program is dumped back to disk with the new entry point being the section of code you added. That's all there is to it. The secret to the whole affair is that short section of added code. What this code does is move the whole program from its new position back to its old position and then transfer control to the old entry point. That gives you an idea of what you need to do and why, so let's get down to the details. While I'll be talking about TRSDOS in particular, the method itself should work with any DOS. The items that might be different would be the names of the commands, utilities used, and the way that the loading address and entry point of a file are stored.

STEP-BY-STEP

The very first thing to do is to get the program from tape onto disk. To do this use the TAPE command. Get the program tape ready to play and enter the following command (<ENTER> means press the white ENTER key): TAPE (S=T,D=D) <ENTER>. You will be prompted for the cassette speed. Press "H" for a high speed cassette or "L" for a low speed cassette. Now press PLAY on your tape deck and then any key on the keyboard. TRSDOS will now load the programs from tape and then write it to disk. Note that some programs will not load onto disk this way (Space Warp and Cosmic Patrol are the only two I know of that will not load).

Once the program is on disk, you are ready to find the loading address and entry point. If you have a utility that will do this for you then use it, otherwise you can use DEBUG to do it. To use DEBUG, first make a note of the name the program was saved under and then get into DEBUG by typing the following command:

DEBUG <ENTER>.

Once in DEBUG, press the "F" key and you will then be asked for a filespec. Type in the name of the program and press <ENTER>. The first record of the file will be displayed on the screen. Look at the two lines at the top of the screen. They should be in one of two formats. In the first format (example 1), *(Editor's note: all the examples are printed at the end of the article, pages 54 and 55.)* the loading address will be the second two bytes (ignore the information to the left of the colon). These are the two bytes underlined in example 1. As with all addresses, the loading address is in the least significant byte (LSB), most significant byte (MSB) order. In other words, if the second two bytes are 0043 then the actual address is 4300H. In the second format, the loading address is in the third pair of bytes on the second line. These are the two bytes underlined in example 2. Now that you have the loading address, write it down somewhere with the note that it is the original loading address. Next you need to find the entry point. To do this, hold down the semicolon key (;). DEBUG will start putting records from the file onto the screen. When it stops, you have reached the last record. This record should contain the entry point. To find it, start at the top of the screen and begin looking for the byte sequence 0202. The two bytes immediately following the two 02 bytes are the entry point. These are the two bytes underlined in example 3. Again, they are in LSB-MSB order. Write this address down with the note that this is the original entry point. I have run across one case where the 0202 bytes sequence was in the next to last record. If you don't find it in the last record, press the hyphen key and page backward one record at a time until you do find the two 02 bytes. After you have found the entry point look at the bytes just before the 0202 bytes. If they are 00, then count how many zero bytes (00 is one zero byte) there are between the 0202 bytes and the first non-zero byte preceding the 0202 bytes. Make a note of this number as the number of zero bytes. Now exit DEBUG by pressing the BREAK key

twice.

You are now going to relocate the program. The only decision to make here is where to relocate it to. You do not want to get in the way of any of the DOS, so any address above 6000 hex should work as long as there is enough memory left to load the program into. I have a 48K machine and usually use A000 hex. To relocate the program type in the following command:

RELO PROGRAM/CMD (ADD=0A000) <ENTER>.

Substitute the address you want to use for the 0A000 address. If the address you use starts with a letter, be sure to precede it with a zero, otherwise the zero is not needed. In place of PROGRAM/CMD, use the name of your program. Write down the address the program was relocated to with the note that this is the new loading address. Now, to make it easier to find the end of the program in a later step, enter the command:

CLEAR <ENTER>.

When this is done load the program using the LOAD command:

(LOAD PROGRAM/CMD <ENTER>).

When the program is loaded go back into DEBUG. A note on DEBUG—when displaying memory the two bytes on the far left side of the screen are the starting address for the line of memory that follows them. Press "D" and you will be asked for the address to be displayed. Enter the new loading address that you made a note of above. Now press "S" to get a full page display. Use the semicolon key to page through memory until you find the end of the program. When you start seeing nothing but zeros, you have found the end. Now use the hyphen (-) key to get back to the first non-zero bytes of memory. This is the end of the program, unless you earlier counted some zero bytes. If you did, then start at the last non-zero byte showing and count off the number of zero bytes you have marked down and this is the end of the program. The first byte after the end of the program is where you are going to add our section of code. Write down this address as the new entry point. To add the new code, press "M". You will be asked for the address to be modified. Enter the address of the new entry point and then press the spacebar (anything other than a space after the address terminates the modify mode). A cursor will appear on the byte you specified. Type in the following with the appropriate addresses entered:

21nnnn11oooo01ssssEDB0C3eeee <ENTER>.

The new loading address is nnnn. The original loading address is oooo. The size of the program is ssss. Remember to enter all of these values in LSB-MSB order. The size of the program can be found by subtracting the first memory location of the program from the last one and adding 1. If hexadecimal subtraction isn't for you, then use 4000H for a 16K program or 8000H for a 32K program. Make a note of the last memory location used by the code you added and call it the end location. With this done, exit DEBUG by pressing "Q". You are now ready to dump the program back to disk. Enter the following command:

DUMP PROGRAM/CMD

(START=nnnn,END=eeee,TRA=tttt) <ENTER>.

For the program name you might want to use a new name so that if you made a mistake you won't have to read the original program from tape again. The new loading address is nnnn. The end location is eeee (noted above). The new entry point, here called the transfer address, is tttt. Here you enter the addresses in their correct order. If an address starts with a letter be sure to put a 0 before it. That was it. You are now ready to run your program from disk.

AN EXAMPLE

To show you an actual example of what is done, I'll set up Adventure 8 to run from disk. After the TAPE command is used to load the program onto disk, I get into DEBUG, press "F", and answer the filespec question with ADVENT/CMD. DEBUG will load the first record of the file and display it on the screen. Example 1 is a screen dump of this. Since the file name and date are not on the first line, that means that I have the format where the second two bytes contain the loading address. These bytes are 0043 and are underlined in the example.

I write this down as:

original loading address = 4300H.

Next, I hold down the semicolon key until DEBUG stops putting new records onto the screen. This is the last record of the file. Starting at the top and looking for the 0202 byte sequence, I find it at the location marked 3EF4. The entry point byte sequence follows it and is 5043. I note this as:

original entry point = 4350H.

Example 3 is a screen dump of this, with the entry point underlined. This is one of those programs that have zero bytes on the end of it so I count these and find that there are 35 of them. I write this down as: number of zero bytes = 35. Pressing BREAK twice, I exit DEBUG back to TRSDOS READY. I relocate the program to A000H with the command RELO ADVENT/CMD (ADD=0A000). I note A000 as the new loading address. When this is done I enter the command CLEAR followed by LOAD ADVENT/CMD. With this finished I get back into DEBUG. First I press "S" and then "D" and I answer the prompt with A000. Paging through memory I find that the last non-zero byte is at memory location DCDC. I count down the number of zero bytes I noted earlier and find that the program ends at memory location DCFFH.

Example 4 is a screen dump of this page of memory. To find the size of the program, I subtract the new loading address (A000H) from the ending address (DCFFH) and add 1. This gives me 3D00H and I write this down as the length of the file. The first byte after the end of the program is DD00H (which is on the next page of memory). I write down this address as the new entry point. This is where I want to add my code, so I press "M" and answer the prompt with DD00 followed by a space. The cursor appears and I type in the following line:

2100A011004301003DEDB0C35043 <ENTER>.

Example 5 is a screen dump of how the screen looks after this is done. When entering the above line, remember that the addresses and the file length are in LSB-MSB order. Looking to find the new ending address I see that it is DD0DH, and I write this down. To find this address I looked at the start of the line and saw that DEBUG had it labeled DD00, which is the location of the first byte of memory on that line. Next I counted over to the last non-zero byte which is byte 13 or in hex, byte D. This is by calling the first byte number 0. If

hexadecimal is not for you, then just use the value for the start of the next line, in this case DD10H. Pressing "Q", I again exit DEBUG and then issue the following command:

DUMP ADVENT/CMD
(START=0A000, END=0DD0D,TRA=0DD00).

For the starting address I used the new loading address. The end address is the new ending location and the transfer address is the new entry point. When the dump is done, Adventure 8 will work from disk.

SUMMARY

This method for getting a tape-based program to run from disk is quick and easy to do. Not counting the time spent loading the program from tape, it takes me about 5 minutes. Its limitations are slight. The main one is that the program must be capable of being loaded onto disk. Since most programs will load onto disk even if they won't run from it, this is not a serious problem. Of course, the program won't have any of the additional capabilities that the regular disk version may have. Other than these two items, this method works very well. I have seen programs that will do most of the work involved for you, but considering the small amount of time and effort needed to do it by hand, I don't feel that they are worth buying. I hope you find this method as easy to use and useful as I have. Have fun doing it on a disk.

Ken Hipple
127 Plum Tree Road
Brandon, MS 39042 ■

```
000100: 0102 0043 C371 4FC3 0E51 C396 4EC3 904B ...C.qO..Q..N..K
000110: C353 4CC3 7E4C C3F8 4CC3 614E C3EA 4EC3 .SL..L..L.aN..N.
000120: 244F C329 57C3 0E57 C3EE 55C3 2A54 C3E0 $O.)W..W..U.*T..
000130: 53C3 6943 C327 47C3 2547 C316 47C3 9447 S.iC.'G.%G..G..G
000140: C3F3 47C3 FE46 C302 47C3 544A C35E 4ADD ..G..F..G.TJ..J.
000150: 6601 19DD F33E 4306 01FE 7028 0531 D759 f....>C...p(.1.Y
000160: 0600 78CD F347 32D2 57B7 C44B 43CD 0C48 ..x..G2.W..KC..H
000170: 225B 43CD BC4A CD8F 43CD F347 CD12 4318 ".C..J..C..G..C.
000180: 09CD E448 CD00 43CD 7C48 9732 5C59 CD00 ...H..C..H.2.Y..
000190: 4318 EECD F347 21CD 43CD 1647 2AF7 58CD C....G!.C..G*.X.
0001A0: 5E4A 21FD 43CD 1647 2AF3 58CD 5E4A DD2A .J!.C..G*.X..J.*
0001B0: 5359 01FD FFDD 09DD 7E01 DD77 02DD 7E00 SY.........w....
0001C0: DD77 01DD 3600 2E21 0844 CD16 47CD 9447 .w..6..!.D..G..G
0001D0: C92A 2041 4456 454E 5455 5245 202A 2028 .* ADVENTURE * (
0001E0: 5665 7273 696F 6E3A 2038 2E32 2920 4164 Version: 8.2) Ad
0001F0: 7665 6E74 7572 6520 6E75 6D62 6572 3A20 venture number:
```

Example 1

```
000100: 0510 4144 564E 3820 2020 3038 2F30 342F ..ADVN8   08/04/
000110: 3832 0100 00A0 C371 4FC3 0E51 C396 4EC3 82.....qO..Q..N.
000120: 904B C353 4CC3 7E4C C3F8 4CC3 614E C3EA .K.SL..L..L.aN..
000130: 4EC3 244F C329 57C3 0E57 C3EE 55C3 2A54 N.$O.)W..W..U.*T
000140: C3E0 53C3 6943 C327 47C3 2547 C316 47C3 ..S.iC.'G.%G..G.
000150: 9447 C3F3 47C3 FE46 C302 47C3 544A C35E .G..G..F..G.TJ..
000160: 4ADD 6601 19DD F33E 4306 01FE 7028 0531 J.f....>C...p(.1
000170: D759 0600 78CD F347 32D2 57B7 C44B 43CD .Y..x..G2.W..KC.
000180: 0C48 225B 43CD BC4A CD8F 43CD F347 CD12 .H".C..J..C..G..
000190: 4318 09CD E448 CD00 43CD 7C48 9732 5C59 C....H..C..H.2.Y
0001A0: CD00 4318 EECD F347 21CD 43CD 1447 2AF7 ..C....G!.C..G*.
0001B0: 58CD 5E4A 21FD 43CD 1647 2AF3 58CD 5E4A X..J!.C..G*.X..J
0001C0: DD2A 5359 01FD FFDD 09DD 7E01 DD77 02DD .*SY.........w..
0001D0: 7E00 DD77 01DD 3600 2E21 0844 CD16 47CD ...w..6..!.D..G.
0001E0: 9447 C92A 2041 4456 454E 5455 5245 202A .G.* ADVENTURE *
0001F0: 2028 5665 7273 696F 6E3A 2038 2E32 2920  (Version: 8.2)
```

Example 2

```
003E00: 0000 0000 0000 0000 0000 000B 0000 0000 ................
003E10: 0014 0015 0000 0000 0009 0000 0000 00FA ................
003E20: 5DFB 5D0C 5E19 5E1B 5E24 5E32 5E43 5E6D .........$.2.C.m
003E30: 5EA4 5EE1 5EF9 5E10 5F32 5F54 5F6D 5F82 .........2.T.m..
003E40: 5F98 5FA0 5FA9 5FC4 5FE7 5F07 602D 6061 ............-a
003E50: 608B 609E 60A9 60B1 60BD 60C7 60D0 60E7 .....`.`.`.`.`.`.`
003E60: 6003 6109 6116 6132 6146 61A0 61B5 61B7 .a.a.a2aFa.a.a.
003E70: 61B9 61D1 61E2 61FB 610C 6211 6230 623C a.a.a.a.a.b.b0b<
003E80: 626B 6288 629A 62AF 62D2 62F2 6206 6312 bkb.b.b.b.b.b.c.
003E90: 6338 6359 6364 6370 637C 637E 6384 63A2 c8cYcdcpc.c.c.c.
003EA0: 63AB 63B5 63C5 63D8 63E4 630B 6427 644F c.c.c.c.c.c.d'dO
003EB0: 6466 6495 64BB 64CA 64E8 64FF 6418 6537 dfd.d.d.d.d.d.e7
003EC0: 6552 6555 656C 6582 65B9 65F6 6509 6634 eReUele.e.e.e.f4
003ED0: 6600 0000 0000 0000 0000 0000 0000 0000 f...............
003EE0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
003EF0: 0000 0000 0202 5043 2054 616E 6479 2020 ......PC Tandy
```

Example 3

```
DC00  0008 0000 0000 0000  0000 0000 0000 0000 ................
DC10  0000 0000 0000 000B  0000 0000 0014 0015 ................
DC20  0000 0000 0009 0000  0000 00FA 5DFB 5D0C ................
DC30  5E19 5E1B 5E24 5E32  5E43 5E6D 5EA4 5EE1 .....$.2.C.m....
DC40  5EF9 5E10 5F32 5F54  5F6D 5F82 5F98 5FA0 .....2.T.m......
DC50  5FA9 5FC4 5FE7 5F07  602D 6061 608B 609E .........-a`.`.
DC60  60A9 60B1 60BD 60C7  60D0 60E7 6003 6109 `.`.`.`.`.`.a.a.
DC70  6116 6132 6146 61A0  61B5 61B7 61B9 61D1 a.a2aFa.a.a.a.a.
DC80  61E2 61FB 610C 6211  6230 623C 626B 6288 a.a.a.b.b0b<bkb.
DC90  629A 62AF 62D2 62F2  6206 6312 6338 6359 b.b.b.b.b.c.c8cY
DCA0  6364 6370 637C 637E  6384 63A2 63AB 63B5 cdcpc.c.c.c.c.c.
DCB0  63C5 63D8 63E4 630B  6427 644F 6466 6495 c.c.c.c.d'dOdfd
DCC0  64BB 64CA 64E8 64FF  6418 6537 6552 6555 d.d.d.d.d.e7eReU
DCD0  656C 6582 65B9 65F6  6509 6634 6600 0000 ele.e.e.e.f4f...
DCE0  0000 0000 0000 0000  0000 0000 0000 0000 ................
DCF0  0000 0000 0000 0000  0000 0000 0000 0000 ................
```

Example 4

```
DD00  2100 A011 0043 0100  3DED B0C3 5043 0000 !....C..=...PC...
DD10  0000 0000 0000 0000  0000 0000 0000 0000 ................
DD20  0000 0000 0000 0000  0000 0000 0000 0000 ................
DD30  0000 0000 0000 0000  0000 0000 0000 0000 ................
DD40  0000 0000 0000 0000  0000 0000 0000 0000 ................
DD50  0000 0000 0000 0000  0000 0000 0000 0000 ................
DD60  0000 0000 0000 0000  0000 0000 0000 0000 ................
DD70  0000 0000 0000 0000  0000 0000 0000 0000 ................
DD80  0000 0000 0000 0000  0000 0000 0000 0000 ................
DD90  0000 0000 0000 0000  0000 0000 0000 0000 ................
DDA0  0000 0000 0000 0000  0000 0000 0000 0000 ................
DDB0  0000 0000 0000 0000  0000 0000 0000 0000 ................
DDC0  0000 0000 0000 0000  0000 0000 0000 0000 ................
DDD0  0000 0000 0000 0000  0000 0000 0000 0000 ................
DDE0  0000 0000 0000 0000  0000 0000 0000 0000 ................
DDF0  0000 0000 0000 0000  0000 0000 0000 0000 ................
```

Example 5

RUBIK'S CUBE and INTERVAL

William H. Patrick

RUBIK'S CUBE

Want to get your cube back into the proper form? This program will show and tell you how. You may start with a standard cube or enter the colors from your own. You can try to work the cube or back up steps, having the reverse moves printed on the screen to get you back to where you started. Or you can have the computer tell you, in simple English, how to work it. No, it won't beat the kid down the block, but it does work. So take your cube, go into your TRS-80 room, lock the door, and END THE FRUSTRATION. Who's to know that you haven't finally figured out the blasted thing?

This program is based on *The Simple Solution to Rubik's Cube by James G. Nourse,* Stanford University, and the program draws upon the article "Rubik's Cube" by Gordon Speer in the October 1981 issue of *Computronics.*

```
Ø CLS: PRINT CHR$(23): PRINT@ 2Ø6, "RUBIK'S CUBE";
1 PRINT@ 328,"WILLIAM H. PATRICK";
2 PRINT@ 39Ø, "RT7 PARADISE CAMP RD.";
3 PRINT@ 454, "HARRODSBURG, KENTUCKY"
4 PRINT@ 516,"        4Ø33Ø";
5 PRINT@ 644,"BASED ON SOLUTION  BY-"
6 PRINT@ 716, "JAMES G. NOURSE ";: PRINT @ 776,"STANFORD
UNIVERSITY";
7 REM     JANUARY 1982
8 REM    "RUBIK'S CUBE" AN ARTICLE BY GORDON SPEER
                  (33Ø4 WOODLAWN DR., STERLING, IL 61Ø81)
                  PUBLISHED IN OCTOBER 1981 *COMPUTRONICS*
                   WAS USED IN DEVELOPING THIS PROGRAM.
9 FOR II=1 TO 4ØØØ:NEXT
1Ø CLS' RUBIK'S CUBE
2Ø CLEAR 2ØØØ
3Ø DEFSTR A,C,T
31 DIM C(3,3,3)
32 MC=1
33 DIM MV$(2ØØ) 'MAX MOVES
4Ø GOSUB 1ØØØ ' INTRODUCTION
5Ø GOSUB 2ØØØ ' SET UP CUBE
54 CLS
55 GOSUB 1ØØØØ
6Ø GOSUB 3ØØØ 'INPUT MOVES
65 GOTO 55
8Ø END
1ØØØ REM ********** INSTRUCTIONS **********
1Ø1Ø PRINT "THE FOLLOWING KEYS ARE TO BE USED -"
1Ø15 PRINT: PRINT TAB(1),"KEY","FUNCTION (NOTICE NUMERIC PAD)"
1Ø2Ø PRINT TAB(1),"8","TOP OF CUBE"
1Ø3Ø PRINT TAB(1),"4","LEFT OF CUBE"
1Ø4Ø PRINT TAB(1),"5","FRONT OF CUBE"
1Ø5Ø PRINT TAB(1),"6","RIGHT OF CUBE"
1Ø6Ø PRINT TAB(1),"2","BOTTOM OF CUBE"
1Ø65 PRINT TAB(1),"T","TRANSVERSE ROTATION": PRINT
TAB(1),"L","LONGITUDINAL ROTATION": PRINT TAB(1),"H",
"HORIZONTAL ROTATION";
1Ø7Ø PRINT
1Ø8Ø PRINT TAB(1),"Ø","CLOCKWISE"
1Ø9Ø PRINT TAB(1),".", "COUNTER- CLOCKWISE"
11ØØ PRINT "ENTER A FACE FOLLOWED BY A DIRECTION."
111Ø PRINT:INPUT "<ENTER> TO CONTINUE";PG$
112Ø CLS: PRINT "TWO ADDITIONAL FEATURES-"
113Ø PRINT: PRINT TAB(1), "M#","HAVE YOUR LAST # MOVES
REVERSED."
114Ø PRINT TAB(1),"??","HAVE THE CUBE WORKED FOR YOU."
1142 PRINT
1145 PRINT TAB(1),"##","HAVE CUBE RANDOMLY MOVED"
115Ø PRINT@96Ø,;:INPUT"<ENTER> TO BEGIN";PG$: RETURN
116Ø REM            *** END OF INSTRUCTIONS ***
2ØØØ REM ********** SET UP CUBE **********
2Ø1Ø INPUT" SET UP STANDARD CUBE OR INPUT YOURS (S/Y)";D$
2Ø2Ø IF D$="S" THEN 25ØØ
2Ø3Ø IF D$="Y" THEN 21ØØ
2Ø4Ø PRINT"ENTER  'S' FOR STANDARD CUBE, 'Y' TO ENTER YOURS."
2Ø5Ø GOTO 2Ø1Ø
21ØØ CLS:REM SET UP YOUR CUBE
211Ø PRINT "THINK OF YOUR CUBE AS BEING MADE UP OF 27 SMALL
CUBES": PRINT : PRINT "THE IDENTIFICATION OF EACH CUBE IS 3
CHARACTERS-"
212Ø PRINT TAB(1),"FIRST","TOP OR BOTTOM COLOR": PRINT
TAB(1), "SECOND","LEFT OR RIGHT COLOR": PRINT TAB(1),
"THIRD","FRONT OR BACK COLOR"
2125 INPUT "<ENTER> TO CONTINUE";PG$:CLS: PRINT "ENTER CUBES
-": PRINT: PRINT TAB(1),"TOP LAYER FIRST": PRINT TAB(1),"FRONT
ROW FIRST": PRINT TAB(1),"LEFT  CUBE FIRST"
213Ø PRINT@ 512+128,"IF A COLOR IS NOT VISIBLE, SIMPLY PRESS
<ENTER>";
214Ø FOR V=1 TO 3:FOR D=1 TO 3:FOR H=1 TO 3
2143 PRINT@ 32Ø,"CUBE";(V-1)*9+(D-1)*3+H;
215Ø PRINT@ 384,;:INPUT"TOP OR BOTTOM COLOR";A$:IF A$="" THEN
A$="-"
216Ø PRINT@ 448,;:INPUT "LEFT OR RIGHT COLOR";B$:IF B$=""
THEN B$="-"
217Ø PRINT@ 512, ;:INPUT "FRONT OR BACK COLOR";C$:IF C$=""
THEN C$="-"
218Ø C(V,H,D)=A$+B$+C$
2181 A$="":B$="":C$=""
2185 PRINT@ 384,STRING$(192,32);
219Ø IF LEN(C(V,H,D))<>3 THEN PRINT"ENTER THAT CUBE AGAIN.";:
GOTO 215Ø
22ØØ NEXT H,D,V
221Ø GOTO 2999
25ØØ REM SET UP STANDARD CUBE
251Ø FOR V=1 TO 3:FOR H=1 TO 3:FOR D=1 TO 3: READ C(V,H,D):
        NEXT D,H,V
252Ø DATA RGW,RG-,RGY,R-W,R--,R-Y,ROW,RO-,ROY,-GW,-G-,-GY,
--W,---,--Y,- OW,-O-,-OY,BGW,BG-,BGY,B-W,B--,B-Y,BOW,BO-,BOY
2999 RETURN
3ØØØ REM  **** INPUT MOVES FROM KEYBOARD ****
3Ø1Ø N$=INKEY$:IF N$="" THEN 3Ø1Ø ELSE PRINT@ 822,N$;"  ";
3Ø2Ø M$=INKEY$:IF M$ ="" THEN 3Ø2Ø ELSE PRINT@ 826,M$;
3Ø21 IF( N$="8" OR N$="4" OR N$="5" OR N$="6" OR N$="2" OR
N$="T" OR N$="L" OR N$="H") AND (M$= "Ø" OR  M$=".") THEN
MV$(MC)=N$+M$:MC=MC+1
3Ø25 PRINT@ 822,"     ";
3Ø3Ø GOSUB 3Ø49
3Ø47 RETURN
3Ø48 REM        **** END OF INPUT ****
3Ø49 REM *****   DETERMINE MOVE TO MAKE   *****
3Ø5Ø PRINT@ 822,N$;M$;: PRINT@ 96Ø,STRING$(3Ø,32);: PRINT @
8Ø4, "MOVE";MC-1;:IF N$="8" AND M$="Ø" THEN GOSUB 14Ø1Ø:
GOTO 3999
3Ø51 IF N$="2" AND M$="Ø" THEN GOSUB 14Ø2Ø: GOTO 3999
3Ø52 IF N$="4" AND M$="." THEN GOSUB 14Ø3Ø: GOTO 3999
3Ø53 IF N$="4" AND M$="Ø" THEN GOSUB 14Ø4Ø: GOTO 3999
3Ø54 IF N$="6" AND M$="Ø" THEN GOSUB 14Ø5Ø: GOTO 3999
```

```
3Ø55 IF N$="6" AND M$="." THEN GOSUB 14Ø6Ø: GOTO 3999
3Ø56 IF N$="5" AND M$="Ø" THEN GOSUB 14Ø7Ø: GOTO 3999
3Ø57 IF N$="5" AND M$="." THEN GOSUB 14Ø8Ø: GOTO 3999
3Ø58 IF N$="8" AND M$="." THEN GOSUB 14Ø9Ø: GOTO 3999
3Ø59 IF N$="2" AND M$="." THEN GOSUB 141ØØ: GOTO 3999
3Ø6Ø IF N$="L" AND M$="." THEN GOSUB 1411Ø: GOTO 3999
3Ø61 IF N$="L" AND M$="Ø" THEN GOSUB 1412Ø: GOTO 3999
3Ø62 IF N$="H" AND M$="Ø" THEN GOSUB 1413Ø: GOTO 3999
3Ø63 IF N$="H" AND M$="." THEN GOSUB 1414Ø: GOTO 3999
3Ø64 IF N$="T" AND M$="Ø" THEN GOSUB 1415Ø: GOTO 3999
3Ø65 IF N$="T" AND M$="." THEN GOSUB 1416Ø: GOTO 3999
3Ø66 IF N$="#" THEN GOSUB 45ØØ: GOTO 3999 ELSE IF N$="?"
THEN GOSUB 5ØØØ
3Ø67 IF N$="M" THEN GOSUB 4ØØØ: GOTO 3999
3Ø68 PRINT@ 96Ø,"INPUT ERROR  ";N$;M$;:FOR ZX=1 TO 5ØØ:
NEXT: GOTO 3Ø1Ø
35ØØ REM     REPLAY LAST MOVES
3999 RETURN '      **** RETURN FROM DETERMINING MOVE ****
4ØØØ REM   ***** REMAKE MOVES FROM MEMORY *****
4Ø1Ø Q=VAL(M$):XX=MC-1
4Ø15 PRINT@ 96Ø,"REVERSING MOVES "XX "TO" XX-Q;
4Ø17 FOR ZZ=1 TO 5ØØ:NEXT
4Ø2Ø FOR II= XX TO XX-Q+1 STEP -1
4Ø21 MC=MC-1
4Ø3Ø N$=LEFT$(MV$(II),1)
4Ø4Ø P$= RIGHT$(MV$(II),1)
4Ø5Ø IF P$="Ø" THEN M$="." ELSE M$="Ø"
4Ø6Ø GOSUB 3Ø49
4Ø61 GOSUB 1ØØØØ
4Ø63 IF INKEY$="" THEN 4Ø63
4Ø65 NEXT II
4Ø66 IF MC<Ø THEN MC=Ø
4Ø7Ø RETURN
45ØØ REM **** MIX UP CUBE *****
45Ø5 RANDOM
45Ø6 FOR XX=1 TO 5Ø
451Ø W=RND(5)
452Ø ON W GOTO 453Ø,454Ø,455Ø,456Ø,457Ø
453Ø N$="8": GOTO 458Ø
454Ø N$="4": GOTO 458Ø
455Ø N$="5": GOTO 458Ø
456Ø N$="6": GOTO 458Ø
457Ø N$="2"
458Ø W=RND(2)
459Ø IF W=1 THEN M$="Ø" ELSE M$="."
46ØØ GOSUB 3Ø49
461Ø NEXT XX
4615 PRINT@ 96Ø,STRING$(3Ø,32);
462Ø RETURN '  *** END OF RANDOM MOVES ***
5ØØØ REM  ********** WORK CUBE ***************
5ØØ1 PRINT@ 96Ø,"SINGLE STEP  Y/N?";:Q$=INKEY$:IF Q$=""
THEN 5ØØ1 ELSE IF Q$="Y" THEN ZFLAG=1
5Ø05 PRINT@ 96Ø,"         TOP EDGES        ";:REM ****** TOP
FRONT EDGE ********
5Ø1Ø P1$=LEFT$(C(1,2,2),1):P2$=RIGHT$(C(2,2,1),1):P3$="-"
5Ø15 GOSUB 99ØØ '  FIND CUBE
5Ø2Ø IF X=1 AND Y=2 AND Z=1 THEN GOTO 51ØØ 'CUBE IS CORRECTLY
POSITIONED
5Ø25 IF X=3 AND Z<>1 THEN N$="2":M$="Ø": GOSUB 3Ø49: GOSUB
1ØØØØ: GOTO 5Ø15
5Ø3Ø IF Z=1  THEN N$="5":M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ:
GOTO 5Ø15
5Ø35 IF X=2 AND Y=1 THEN N$="4":M$=".": GOSUB 3Ø49: GOSUB
1ØØØØ: N$="2":M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ:N$="4":M$="Ø":
GOSUB 3Ø49: GOSUB 1ØØØØ: GOTO 5Ø15
5Ø4Ø IF X=2 AND Y=3 THEN N$="6":M$="Ø": GOSUB 3Ø49:
GOSUB 1ØØØØ: N$="2":M$=".": GOSUB 3Ø49: GOSUB 1ØØØØ: N$="6":
M$=".": GOSUB 3Ø49: GOSUB 1ØØØØ: GOTO 5Ø15
5Ø45 IF X=1 AND Y=1 THEN N$="4":M$ "Ø": GOSUB 3Ø49: GOSUB
1ØØØØ: N$="5":M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ:N$="4":M$=".":
GOSUB 3Ø49: GOSUB 1ØØØØ: GOTO 5Ø15
5Ø5Ø IF X=1 AND Y=3 THEN N$="6":M$=".": GOSUB 3Ø49: GOSUB
1ØØØØ: N$="5":M$=".": GOSUB 3Ø49: GOSUB 1ØØØØ:N$="6":M$="Ø":
GOSUB 3Ø49: GOSUB 1ØØØØ: GOTO 5Ø15
5Ø55 IF X=1 AND Z=3  THEN N$="H":M$="Ø": GOSUB 3Ø49: GOSUB
1ØØØØ:N$="6":M$=".": GOSUB 3Ø49: GOSUB 1ØØØØ:N$="H":M$=".":
GOSUB 3Ø49: GOSUB 1ØØØØ: GOTO 5Ø15
51ØØ A$=LEFT$(C(1,2,2),1)+"-"+ RIGHT$(C(2,2,1),1): IF
C(1,2,1)=A$ THEN 515Ø '  CUBE IS CORRECTLY ORIENTED
511Ø N$="5":M$=".": GOSUB 3Ø49: GOSUB 1ØØØØ
512Ø N$="8":M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ
513Ø N$="4":M$=".": GOSUB 3Ø49: GOSUB 1ØØØØ
514Ø N$="8":M$=".": GOSUB 3Ø49: GOSUB 1ØØØØ: GOTO 51ØØ
515Ø A$=LEFT$(C(1,2,2),1)+MID$(C(2,3,2),2,1)+"-" :
B$=LEFT$(C(1,2,2),1)+"-"+ RIGHT$(C(2,2,3),1) :
C$=LEFT$(C(1,2,2),1)+MID$(C(2,1,2),2,1)+"-" : IF C(1,3,2)=A$
AND C(1,2,3)=B$ AND C(1,1,2)=C$ THEN GOTO 52ØØ ' CORRECT
EDGES
516Ø PRINT@ 96Ø,"FINISHED ONE";: N$="H":M$="Ø": GOSUB 3Ø49:
GOSUB 1ØØØØ: GOTO 5Ø05
52ØØ PRINT @ 96Ø,"          TOP CORNERS   ";: REM ***** TOP
CORNERS *****
52Ø5 P1$=LEFT$(C(1,2,2),1): P2$=MID$(C(2,1,2),2,1):
P3$=RIGHT$(C(2,2 ,1),1)
521Ø GOSUB 99ØØ
5215 IF X=1 AND Y=1 AND Z=1 THEN 53ØØ'REM CORRECT POSITION
522Ø IF X=1 AND Y=1 AND Z=3 THEN N$="4":M$=".": GOSUB 3Ø49:
GOSUB 1ØØØØ:N$="2":M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ: N$="4":
M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ: GOTO 52Ø5
5225 IF X=1 AND Y=3 AND Z=3 THEN N$="6":M$="Ø": GOSUB 3Ø49:
GOSUB 1ØØØØ:N$="2":M$=".": GOSUB 3Ø49: GOSUB 1ØØØØ: N$="6":
M$=".": GOSUB 3Ø49: GOSUB 1ØØØØ: GOTO 52Ø5
523Ø IF X=1 AND Y=3 AND Z=1 THEN N$="6":M$=".": GOSUB 3Ø49:
GOSUB 1ØØØØ:N$="2":M$=".": GOSUB 3Ø49: GOSUB 1ØØØØ: N$="6":
:M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ: GOTO 52Ø5
524Ø IF (X=3  AND Y<>1) OR (X=3 AND Z<>1) THEN N$="2":
M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ: GOTO 52Ø5
5245 N$="4":M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ:N$="2":M$="Ø":
GOSUB 3Ø49: GOSUB 1ØØØØ:N$="4":M$=".": GOSUB 3Ø49: GOSUB
1ØØØØ: GOTO 52Ø5
53ØØ  IF LEFT$(C(1,1,1),1)=LEFT$(C(1,2,2),1) AND
MID$(C(1,1,1),2,1)=MID$(C(2,1,2),2,1) AND
RIGHT$(C(1,1,1),1)=RIGHT$(C(2,2,1),1) THEN 535Ø ' CORRECT
ORIENTATION
53Ø5 Z$="H.6.2Ø2Ø6Ø5Ø2Ø2Ø5.HØ": FOR I=1 TO 19 STEP 2:
N$=LEFT$(MID$(Z$,I,2),1): M$=RIGHT$(MID$(Z$,I,2),1): GOSUB
3Ø49: GOSUB 1ØØØØ:NEXT: GOTO 52Ø5
535Ø IF LEFT$(C(1,2,2),1)=LEFT$(C(1,3,1),1) AND
LEFT$(C(1,2,2),1)=LEFT$(C(1,3,3),1) AND
LEFT$(C(1,2,2),1)=LEFT$(C(1,1,3),1) AND
MID$(C(2,3,2),2,1)=MID$(C(1,3,1),2,1) AND
MID$(C(2,3,2),2,1)=MID$(C(1,3,3),2,1) AND
MID$(C(2,1,2),2,1) = MID$(C(1,1,3),2,1) THEN 536Ø
5355 N$="H":M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ: GOTO 52Ø5
536Ø IF RIGHT$(C(2,2,1),1)=RIGHT$(C(1,3,1),1) AND
RIGHT$(C(1,3,3),1)=RIGHT$(C(2,2,3),1) AND
RIGHT$(C(1,1,3),1)=RIGHT$(C(2,2,3),1) THEN 54ØØ ELSE 5355
54ØØ PRINT @ 96Ø,"       SECOND LEVEL   ";: REM *****
SECOND LEVEL *****
54Ø5 N$="L":M$=".": GOSUB 3Ø49: GOSUB 1ØØØØ: N$="L":
M$=".": GOSUB 3Ø49: GOSUB 1ØØØØ
541Ø P1$=RIGHT$(C(2,2,1),1): P2$=MID$(C(2,3,2),2,1):P3$="-"
5415 GOSUB 99ØØ
542Ø IF RIGHT$(C(2,3,1),1)=RIGHT$(C(2,2,1),1) AND
MID$(C(2,3,1),2,1)=MID$(C(2,3,2),2,1) THEN 55ØØ ' CUBE IS
CORRECT
```

```
5425 IF X=2 AND Y=3 AND Z=1 THEN GOSUB 549Ø: N$="8": M$=".":
GOSUB 3Ø49: GOSUB 1ØØØØ: GOSUB 549Ø: GOTO 541Ø ' ORIENT A
CORRECTLY POSITIONED CUBE
544Ø IF X=1 AND Y=1 AND Z=2 THEN GOSUB 549Ø: GOTO 541Ø
'POSITION FROM TLE
545Ø  IF X=1 AND Y<>1 THEN N$="8":M$="Ø": GOSUB 3Ø49: GOSUB
1ØØØØ: GOTO 541Ø ' WAS IN TOP BUT NOT TLE
5452 IF X=2 AND Y=3 THEN N$="H":M$="Ø": GOSUB 3Ø49: GOSUB
1ØØØØ: GOSUB 549Ø: N$="H":M$=".": GOSUB 3Ø49: GOSUB 1ØØØØ:
GOTO 541Ø
5455 IF X=2 THEN N$="H":M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ:
GOTO 541Ø
5489 REM *** SUB TO MOVE TLE TO FRE
549Ø Z$="6Ø8Ø6.8.5.8.5Ø":FOR I=1 TO 13 STEP 2:
N$=LEFT$(MID$(Z$,I,2),1): M$=RIGHT$(MID$(Z$,I,2),1):
GOSUB 3Ø49: GOSUB 1ØØØØ: NEXT: RETURN
55ØØ IF MID$(C(2,3,3),2,1)=MID$(C(2,3,2),2,1) AND
RIGHT$(C(2,3,3),1)=RIGHT$(C(2,2,3),1) AND
MID$(C(2,1,3),2,1)=MID$(C(2,1,2),2,1) AND
RIGHT$(C(2,1,3),1)=RIGHT$(C(2,2,3),1) AND
MID$(C(2,1,1),2,1)=MID$(C(2,1,2),2,1) THEN 55Ø1 ELSE 55Ø5
55Ø1 IF RIGHT$(C(2,1,1),1)=RIGHT$(C(2,2,1),1) THEN 56ØØ
' 2ND LEVEL IS FINISHED
55Ø5 N$="H":M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ: GOTO 541Ø
56ØØ REM  ********** THIRD LEVEL **********
561Ø PRINT@ 96Ø,"BOTTOM CORNERS          ";: FOR ZZ=1
TO 5ØØ: NEXT
56Ø5 N$="L":M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ: N$="L": M$="Ø":
GOSUB 3Ø49: GOSUB 1ØØØØ ' TURN CUBE OVER
561Ø REM POSITION CORNERS
5615 REM POSITION  3,1,3
562Ø P1$=MID$(C(2,1,2),2,1): P2$=RIGHT$(C(2,2,3),1):
P3$=LEFT$(C(3,2 ,2),1)
5625 GOSUB 99ØØ
563Ø IF X=3 AND Y=1 AND Z=3 THEN 564Ø
5635 N$="2":M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ: GOTO 562Ø
564Ø  REM POSITION 3,1,1
5645 P1$=LEFT$(C(3,2,2),1): P2$=MID$(C(2,1,2),2,1):
P3$=RIGHT$(C(2,2,1),1)
565Ø GOSUB 99ØØ
5655 IF X=3 AND Y=1 AND Z=1 THEN 5675' REM DFL IS CORRECT
566Ø IF X=3 AND Y=3 AND Z=1 THEN GOSUB 6ØØ1: GOTO 564Ø
5665 IF X=3 AND Y=3 AND Z=3 THEN GOSUB 6ØØ3: GOTO 564Ø
567Ø PRINT"SOMETHING'S HAYWIRE":END
5675 REM POSITION 3,3,1
568Ø P1$=LEFT$(C(3,2,2),1): P2$=MID$(C(2,3,2),2,1):
P3$=RIGHT$(C(2,2,1),1)
5685 GOSUB 99ØØ
569Ø IF X=3 AND Y=3 AND Z=1 THEN 61ØØ
5695 N$="H":M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ: GOSUB 6ØØ1
57ØØ GOTO 61ØØ ' CORNERS ARE POSITIONED
6ØØØ REM SWAP DFL WITH DFR
6ØØ1 Z$="6.2.6Ø5Ø2Ø5.6.2Ø6Ø2Ø2Ø":FOR II=1 TO 21 STEP 2:
N$=LEFT$(MID$(Z$,II,2),1): M$=RIGHT$(MID$(Z$,II,2),1):
GOSUB 3Ø49: GOSUB 1ØØØØ: NEXT: RETURN
6ØØ3 REM SWAP DFL WITH DBR
6ØØ4 Z$="6.2.6Ø5Ø2Ø2Ø5.6.2Ø6Ø2Ø": FOR II =1 TO 21 STEP 2:
N$=LEFT$(MID$(Z$,II,2),1): M$=RIGHT$(MID$(Z$,II,2),1):
GOSUB 3Ø49: GOSUB 1ØØØØ: NEXT: RETURN
61ØØ REM ORIENT BOTTOM CORNERS
61Ø5  REM COUNT NUMBER OF BOTTOM CORNERS THAT MATCH
611Ø BC=Ø
6115 B$=LEFT$(C(3,2,2),1) ' REM COLOR TO MATCH
6125 IF LEFT$(C(3,1,1),1)=B$ THEN BC=BC+1
6135 IF LEFT$(C(3,3,1),1)=B$ THEN BC=BC+1
614Ø IF LEFT$(C(3,3,3),1)=B$ THEN BC=BC+1
6145 IF LEFT$(C(3,1,3),1)=B$ THEN BC=BC+1
615Ø IF BC=4 THEN 7ØØØ 'REM BOTTOM CORNERS ARE CORRECT
6155 IF BC=2 THEN 616Ø ELSE 63ØØ
616Ø IF LEFT$(C(3,1,3),1)=B$ AND LEFT$(C(3,3,1),1)=B$ AND
RIGHT$(C(3,1,1),1)=B$ AND MID$(C(3,3,3),2,1)=B$ THEN
GOSUB 69ØØ: GOTO 61Ø5' PATTERN 6
6165 IF LEFT$(C(3,1,3),1)=B$ AND LEFT$(C(3,3,3),1)=B$ AND
RIGHT$(C(3,1,1),1)=B$ AND RIGHT$(C(3,3,1),1)=B$ THEN GOSUB
69ØØ: GOTO 61Ø5 ' PATTERN 5
617Ø IF LEFT$(C(3,3,1),1)=B$ AND LEFT$(C(3,3,3),1)=B$ AND
RIGHT$(C(3,1,1),1)=B$ AND RIGHT$(C(3,1,3),1)=B$ THEN GOSUB
69ØØ: GOTO 61Ø5' PATTERN 4
6175 N$="H":M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ:
6299 GOTO 61Ø5
63ØØ IF BC=1 THEN 63Ø5 ELSE 66ØØ
63Ø5 IF LEFT$(C(3,1,1),1)=B$ AND RIGHT$(C(3,3,1),1)=B$ AND
RIGHT$(C(3,1,3),1)=B$ AND MID$(C(3,3,3),2,1)=B$ THEN
GOSUB 69ØØ: GOTO 61Ø5 'PATTERN 1
631Ø IF LEFT$(C(3,1,1),1)=B$ AND RIGHT$(C(3,3,3),1)=B$ AND
MID$(C(3,1,3),2,1)=B$ AND MID$(C(3,3,1),2,1)=B$ THEN GOSUB
69ØØ: GOTO 61Ø5 ' PATTERN 2
6315 N$="H":M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ
6599 GOTO 61Ø5
66ØØ REM BC=Ø
66Ø5 IF RIGHT$(C(3,1,3),1)=B$ AND RIGHT$(C(3,3,3),1)=B$ AND
MID$(C(3,1,1),2,1)=B$ AND MID$(C(3,3,1),2,1)=B$ THEN GOSUB
69ØØ: GOTO 61Ø5 ' REM PATTERN 3
661Ø IF MID$(C(3,1,1),2,1)=B$ AND MID$(C(3,3,1),2,1)=B$ AND
MID$(C(3,3,3),2,1)=B$ AND MID$(C(3,1,3),2,1)=B$ THEN GOSUB
69ØØ: GOTO 61Ø5 ' PATTERN 7
67ØØ N$="H":M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ
6899 GOTO 61Ø5
69ØØ Z$="6.2.6Ø2.6.2Ø2Ø6Ø2Ø2Ø": FOR ZZ=1 TO 19 STEP 2:
N$=LEFT$(MID$(Z$,ZZ,2),1): M$=RIGHT$(MID$(Z$,ZZ,2),1):
GOSUB 3Ø49: GOSUB 1ØØØØ:NEXT: RETURN
7ØØØ PRINT @ 96Ø,"BOTTOM EDGES           ";: FOR ZZ=1 TO
5ØØ: NEXT
7ØØ2 B$=LEFT$(C(3,2,2),1) 'BOTTOM COLOR
7ØØ5 ZT=Ø
7Ø1Ø IF RIGHT$(C(3,2,1),1)=RIGHT$(C(2,2,1),1) OR
RIGHT$(C(2,2,1),1)=LEFT$(C(3,2,1),1) THEN 71ØØ' FRONT EDGE
CORRECTLY POSITIONED
7Ø15 N$="H":M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ:ZT=ZT+1
7Ø2Ø IF ZT<5 THEN 7Ø1Ø ELSE GOSUB 79ØØ: GOTO 7ØØ5
71ØØ REM CHECK ALL POSITIONS
711Ø IF(MID$(C(2,1,2),2,1)=MID$(C(3,1,2),2,1) OR
MID$(C(2,1,2),2,1)=LEFT$(C(3,1,2),1)) AND
(MID$(C(3,3,2),2,1)=MID$(C(2,3,2),2,1) OR
MID$(C(2,3,2),2,1)=LEFT$(C(3,3,2),1)) AND
(RIGHT$(C(2,2,3),1)=RIGHT$(C(3,2,3),1) OR
LEFT$(C(3,2,3),1)= RIGHT$(C(2,2,3),1)) THEN 8ØØØ
7111 GOSUB 79ØØ: GOTO 71ØØ
79ØØ Z$="4.6Ø5Ø4Ø6.2Ø2Ø4.6Ø5Ø4Ø6.": FOR II=1 TO 23 STEP 2:
N$=LEFT$(MID$(Z$,II,2),1): M$=RIGHT$(MID$(Z$,II,2),1): GOSUB
3Ø49: GOSUB 1ØØØØ: NEXT II: RETURN
8ØØØ PRINT@ 96Ø,"      FINAL STEP       ";: FOR II=1 TO
5ØØ: NEXT
8ØØ5 REM MATCH BOTTOM PATTERNS
8Ø1Ø IF RIGHT$(C(3,2,1),1)=B$ AND RIGHT$(C(3,2,3),1)=B$ AND
MID$(C(3,3,2),2,1)=B$ AND MID$(C(3,1,2),2,1)=B$ THEN 81ØØ
' REM PATTERN 1
8Ø15 IF LEFT$(C(3,1,2),1)=B$ AND LEFT$(C(3,3,2),1)=B$ AND
RIGHT$(C(3,2,1),1)=B$ AND RIGHT$(C(3,2,3),1)=B$ THEN 82ØØ
'PATTERN 2
8Ø2Ø IF LEFT$(C(3,2,1),1)=B$ AND LEFT$(C(3,3,2),1)=B$ AND
MID$(C(3,1,2),2,1)=B$ AND RIGHT$(C(3,2,3),1)=B$ THEN 83ØØ
'PATTERN 3
8Ø3Ø IF LEFT$(C(3,2,1),1)=B$ AND LEFT$(C(3,3,2),1)=B$ AND
LEFT$(C(3,2,3),1)=B$ AND LEFT$(C(3,1,2),1)=B$ THEN  84ØØ
' FINISHED!!
8Ø5Ø N$="H":M$="Ø": GOSUB 3Ø49: GOSUB 1ØØØØ: GOTO 8Ø05
81ØØ Z$="4.6Ø5Ø5Ø4Ø6.2Ø2Ø4.6Ø5Ø4Ø6.2Ø2Ø4.6Ø5Ø5Ø4Ø6.2.": FOR
```

```
II=1 TO 43 STEP 2: N$=LEFT$(MID$(Z$,II,2),1):
M$=RIGHT$(MID$(Z$,II,2),1): GOSUB 3049: GOSUB 10000:
NEXT II: GOTO 8005
8200 Z$="4.6050406.2404.6050406.2404.6050504006.2404.6050406.
2404.6050406.2020": FOR II=1 TO 63 STEP 2:
N$=LEFT$(MID$(Z$,II,2),1): M$=RIGHT$(MID$(Z$,II,2),1):
GOSUB 3049: GOSUB 10000:NEXT: GOTO 8005
8300 Z$="4.6050406.2.4.605.406.2.4.6050504006.4.6050406.
2020404.6050400 06.":FOR II=1 TO 59 STEP 2:
N$=LEFT$(MID$(Z$,II,2),1): M$=RIGHT$(MID$(Z$,II,2),1):
GOSUB 3049: GOSUB 10000: NEXT: GOTO 7000
8400 PRINT @ 960,STRING$(26,32);:Q$=" FINISHED
!!!!!!!!!!!!!!!!": FOR II=1 TO LEN(Q$): PRINT @
958+II*2,MID$(Q$,II,1);: FOR XX=1 TO 100: NEXT XX: NEXT II:
GOTO 8400
9900 REM ******** FIND A CUBE, RETURN LOCATION AS X,Y,Z
*******
9905 F1$=P1$+P2$+P3$: F2$=P1$+P3$+P2$: F3$=P2$+P1$+P3$:
F4$=P2$+P3$+P1$: F5$=P3$+P1$+P2$: F6$=P3$+P2$+P1$
9910 FOR X=1 TO 3:FOR Y=1 TO 3: FOR Z=1 TO 3
9915 C$=C(X,Y,Z)
9920 IF C$=F1$ OR C$=F2$ OR C$=F3$ OR C$=F4$ OR C$=F5$ OR
C$=F6$ THEN 9940
9925 NEXT Z,Y,X
9930 PRINT @960,X;Y;Z;:IF INKEY$="" THEN 9930
9940 RETURN ' REM ***** CUBE IS AT X,Y,Z
10000 REM ********* PRINT CUBE **********
10010 PRINT CHR$(23);
10020 PRINT@ 64,;
10030 V=1:W=1
10040 FOR D=3 TO 1 STEP-1
10050 PRINT@272-64*D,;
10060 FOR H=1 TO 3
10070 GOSUB 10500
10080 NEXT H,D
10090 FOR V=1 TO 3
10100 PRINT@258+64*V,;
10110 H=1:W=2
10120 FOR D=3 TO 1 STEP-1
10130 GOSUB 10500
10140 NEXT D
10150 PRINT@272+64*V,;
10160 D=1:W=3
10170 FOR H=1 TO 3
10180 GOSUB 10500
10190 NEXT H
10200 PRINT@288+64*V,;
10210 H=3:W=2
10220 FOR D=1 TO 3
10230 GOSUB 10500
10240 NEXT D
10250 PRINT@304+64*V,;
10260 D=3:W=3
10270 FOR H=3 TO 1 STEP-1
10280 GOSUB 10500
10290 NEXT H
10300 NEXT V
10310 V=3:W=1
10320 FOR D=1 TO 3
10330 PRINT@528+64*D,;
10340 FOR H=1 TO 3
10350 GOSUB 10500
10360 NEXT H,D
10370 REM FRAME
10380 IF POINT(1,12)=-1 THEN 10495
```

continued on page 60

RUBIK'S CUBE

continued from page 59

```
1Ø39Ø PRINT@164,"RUBIK'S CUBE";: PRINT@16,STRING$(8,131);:
PRINT@258,STRING$(3Ø,131);: PRINT@514,STRING$(3Ø,131);:
PRINT@784,STRING$(8,131);
1Ø4ØØ FOR Y=12 TO 24:SET(1,Y):SET(93,Y):SET(124,Y):NEXT
1Ø41Ø FOR Y=Ø TO 36:SET(29,Y):SET(61,Y):NEXT
1Ø495 IF ZFLAG=1 THEN IF INKEY$="" THEN 1Ø495
1Ø499 RETURN
1Ø5ØØ PRINT" ";MID$(C(V,H,D),W,1);: RETURN
1Ø999 REM          *** END OF PRINT ***
14ØØØ REM ************ MOVES ****************
14Ø1Ø V=1: PRINT@ 96Ø,"TOP CLOCKWISE";: GOSUB 17ØØØ: RETURN
'REM 8Ø   OK
14Ø2Ø V=3: PRINT@ 96Ø,"BOTTOM CLOCKWISE";: GOSUB 17Ø5Ø:
RETURN 'REM 2Ø   ØK
14Ø3Ø H=1: PRINT@96Ø,"LEFT COUNTER CLOCKWISE";: GOSUB 15ØØØ:
RETURN 'REM 4.    OK
14Ø4Ø H=1: PRINT@ 96Ø,"LEFT CLOCKWISE";: GOSUB 15Ø5Ø: RETURN
'REM 4Ø    OK
14Ø5Ø H=3: PRINT@ 96Ø,"RIGHT CLOCKWISE";: GOSUB 15ØØØ: RETURN
'REM 6Ø   OK
14Ø6Ø H=3: PRINT@96Ø,"RIGHT COUNTER CLOCKWISE";: GOSUB 15Ø5Ø:
RETURN 'REM 6.   OK
14Ø7Ø D=1: PRINT@ 96Ø,"FRONT CLOCKWISE";: GOSUB 16ØØØ: RETURN
'REM 5Ø   OK
14Ø8Ø D=1: PRINT@ 96Ø,"FRONT COUNTER CLOCKWISE";: GOSUB
16Ø5Ø: RETURN 'REM 5.   OK
14Ø9Ø V=1: PRINT@ 96Ø,"TOP COUNTER CLOCKWISE";: GOSUB 17Ø5Ø:
RETURN 'REM 8.   OK
141ØØ V=3: PRINT@96Ø,"BOTTOM COUNTER CLOCKWISE";: GOSUB
17ØØØ: RETURN 'REM 2.
1411Ø FOR H=1 TO 3: PRINT@96Ø,"LONGITUDINALLY COUNTER ";:
GOSUB 15Ø5Ø:NEXT: RETURN 'REM L.
1412Ø FOR H=1 TO 3: PRINT@96Ø,"LONGITUDINALLY CLOCKWISE";:
GOSUB 15ØØØ:NEXT: RETURN 'REM LØ
1413Ø FOR V=1 TO 3: PRINT@96Ø,"HORIZONALLY CLOCKWISE";: GOSUB
17ØØØ: NEXT: RETURN 'REM HØ
1414Ø FOR V=1 TO 3: PRINT@96Ø, "HORIZONTALLY COUNTER
CLOCKWISE";: GOSUB 17Ø5Ø: NEXT: RETURN 'REM H.
1415Ø FORD=1 TO 3: PRINT@ 96Ø,"TRANSVERSELY CLOCKWISE";:
GOSUB 16ØØØ: NEXT: RETURN 'REM TØ TØ
1416Ø FOR D=1 TO 3: PRINT@ 96Ø,"TRANSVERSELY COUNTER
CLOCKWISE";: GOSUB 16Ø5Ø: NEXT: RETURN 'REM T.
14999 REM          *** END OF MOVES ***
15ØØØ REM ******SUBROUTINES USED BY MOVES******
15Ø1Ø REM #### ROTATE LONGITUDINALLY FRONT TO BACK #####
15Ø2Ø T=C(1,H,3): TT=C(1,H,2): C(1,H,3)=C(1,H,1):
C(1,H,2)=C(2,H,1)
15Ø3Ø C(1,H,1)=C(3,H,1): C(2,H,1)=C(3,H,2): C(3,H,1)=
C(3,H,3): C(3,H,2)=C(2,H,3): C(3,H,3)=T: C(2,H,3)=TT
15Ø4Ø GOSUB 151ØØ
15Ø45 RETURN ' REM         END OF RL F TO B
15Ø5Ø REM #### ROTATE LONGITUDINALLY BACK TO FRONT ####
15Ø6Ø T=C(1, H,1): TT=C(1,H,2): C(1,H,1)=C(1,H,3):
C(1,H,2)=C(2,H,3): C(1,H,3) =C(3,H,3): C(2,H,3)=C(3,H,2):
C(3,H,3)=C(3,H,1): C(3,H,2)=C(2,H,1): C(3,H,1)=T: C(2,H,1)=TT
15Ø7Ø GOSUB 151ØØ
15Ø8Ø RETURN'  REM         END OF RL B TO F
151ØØ REM  %%%%  EXCHANGE LONGITUDINALLY %%%%%
1511Ø FOR V=1 TO 3: FOR D=1 TO 3: C(V,H,D)=MID$(C(V,H,D),3,1)
+ MID$(C(V,H,D),2,1) + MID$(C(V,H,D),1,1): NEXT D,V
1512Ø RETURN ' REM         END OF EL
16ØØØ REM
16Ø1Ø REM #### ROTATE TRANSVERSELY CLOCKWISE ####
16Ø2Ø T=C(1,3,D): TT=C(1,2,D): C(1,3,D)=C(1,1,D):
C(1,2,D)=C(2,1,D): C(1,1,D)=C(3,1,D): C(2,1,D)=C(3,2,D):
C(3,1,D)=C(3,3,D): C(3,2,D)= C(2,3,D):C(3,3,D)=T:C(2,3,D)=TT
16Ø3Ø GOSUB 161ØØ
16Ø4Ø RETURN ' REM          END OF T CW
16Ø5Ø REM #### ROTATE TRANSVERSELY COUNTERCLOCKWISE  ####
16Ø6Ø T=C(1,1,D): TT=C(1,2,D): C(1,1,D)=C(1,3,D):
C(1,2,D)=C(2,3,D): C(1,3,D)=C(3,3,D): C(2,3,D)=C(3,2,D):
C(3,3,D)=C(3,1,D): C(3,2,D)= C(2,1,D): C(3,1,D)=T:
C(2,1,D)=TT
16Ø7Ø GOSUB 161ØØ
16Ø8Ø RETURN ' REM          END OF T CCW
161ØØ REM %%%% EXCHANGE TRANSVERSELY %%%%
1611Ø FOR V=1 TO 3: FOR H=1 TO 3: C(V,H,D)=MID$(C(V,H,D),2,1)
+ MID$(C(V,H,D),1,1) + MID$(C(V,H,D),3,1): NEXT H,V
1612Ø RETURN ' REM          END OF ET
17ØØØ REM #### ROTATE HORIZONTALLY LEFT ####
17Ø1Ø T=C(V,1,1):  TT=C(V,2,1):  C(V,1,1)=C(V,3,1):
 C(V,2,1)=C(V,3,2):  C(V,3,1)=C(V,3,3):  C(V,3,2)=C(V,2,3):
 C(V,3,3)=C(V,1,3):  C(V,2,3)= C(V,1,2): C(V,1,3)=T:
C(V,1,2)=TT
17Ø2Ø GOSUB 171ØØ
17Ø3Ø RETURN ' REM          END OF RHL
17Ø5Ø REM #### ROTATE HORIZONTALLY RIGHT ####
17Ø6Ø T=C(V,1,1): TT=C(V,1,2): C(V,1,1)=C(V,1,3):
C(V,1,2)=C(V,2,3): C(V,1,3)=C(V,3,3): C(V,2,3)=C(V,3,2):
C(V,3,3)=C(V,3,1): C(V,3,2)= C(V,2,1): C(V,3,1)=T:
C(V,2,1)=TT
17Ø7Ø GOSUB 171ØØ
17Ø8Ø RETURN ' REM          END OF RHR
171ØØ REM %%%% EXCHANGE HORIZONTALLY %%%%
1711Ø FOR D=1 TO 3: FOR H=1 TO 3: C(V,H,D)=MID$(C(V,H,D),1,1)
+ MID$(C(V,H,D),3,1) + MID$(C(V,H,D),2,1): NEXT H,D
1712Ø RETURN ' REM           END OF EH
17999 REM    *** END OF SUBROUTINES USED BY MOVES ***
```

INTERVAL

This program will tell you your dates or ownership for weekly interval ownership of condominiums or other "time-shared" facilities.

```
Ø CLS: PRINT"INTERVAL OWNERSHIP: ":PRINT TAB(1),"WEEKS BEGIN
WITH FIRST SATURDAY OF THE YEAR"
1 U$="%   % ###    % % ###  ":U1$="% % ##"
2 CR=Ø
5 PRINT@ 7*64,"BY WILLIAM H. PATRICK":PRINT "RT7 PARADISE
CAMP ROAD": PRINT "HARRODSBURG, KENTUCKY 4Ø33Ø"
1Ø PRINT: INPUT "DESIRED YEAR (1981 TO 2Ø99)";DY
2Ø IF DY<1981 OR DY>2Ø99 THEN 1Ø
3Ø AD=DY-198Ø
4Ø IF DY/4=INT(DY/4) THEN 6Ø
5Ø AD=AD+INT(AD/4)
6Ø AD=AD+4
7Ø R=AD-INT(AD/7)*7 'R IS NUMBER OF DAY OF WEEK YEAR BEGINS
ON
8Ø W=1
9Ø IF R=Ø THEN ST=1 ELSE ST=8-R
1ØØ READ M$,M
1Ø5 IF DY/4=INT(DY/4) AND M$="FEBRUARY" AND CR=Ø THEN M=M+1
:CR=1
11Ø PRINT USING U$;"WEEK",W,M$,ST;
12Ø ED=ST+7
13Ø IF ED<=M THEN 2ØØ
14Ø ED=ED-M
15Ø READ M$,M
2ØØ PRINT USING U1$;M$,ED
21Ø ST=ED
22Ø W=W+1
23Ø IF W>52 THEN 1ØØØ
24Ø GOTO 1Ø5
```

```
500 DATA JANUARY,31,FEBRUARY,28,MARCH,31,APRIL,30,MAY,31,
JUNE,30,JULY,31,AUGUST,31,SEPTEMBER,30,OCTOBER,31,NOVEMBER,30 ,
DECEMBER,31,JANUARY,31
1000 INPUT "ANOTHER YEAR (Y,N)";A$
1001 IF A$="N" THEN END
1002 IF A$="Y" THEN RUN
1003 GOTO 1000
```

William H. Patrick
Rte 7, Paradise Camp Rd.
Harrodsburg, KY 40330 ■

CASSETTE ENHANCEMENTS

continued from page 44

solely for his personal use. The inclusion of the user's name and address when first running the program is undoubtedly an effective deterrent to piracy.

The "KWIK" Storage Format

Certainly the most significant feature of these cassette operating systems is the KWIK storage format. Programs stored in the manner are saved at 2200 baud on the Model III (almost a 50% improvement over 1500 baud) and from 2 times to 6 times faster on the Model I. Loading time saved becomes more dramatic as program length increases, as 20 seconds of "gear shifting" is required to achieve the maximum 3000 baud. Typically, a 4-minute Model I program would load in 56 seconds under the KWIK format.

The KWIK storage procedure includes an enhanced verification process. Even if no speed were gained, KOS3 would be worth $26 for this feature alone.

"Disk Features"

A number of features are available in both KWICOS and KOS3 which simulate a disk operating system. Programs saved on cassette may be given passwords and even comments, and a "directory", including program lengths, may be obtained. In addition, graphics are used to indicate what process is occurring at any given time. Even error messages are improved, such as "RECORDER NOT READY."

A keyboard debounce feature and a variable-speed LIST command are available. In addition, KOS3 (Model III) adds a simple command to deactivate the BREAK key, simplified I/O routing (easily route your printer output to RS-232, for example), a command to display the real-time clock, and provisions for maintaining the current date.

CONCLUSION

It is possible to continue reciting commands, features, and praise of both the Lemons Tech loaders and the KWIK Software cassette operating system. However, I feel that the information presented here speaks for itself. For $43.95 complete ($26 for KOS3 or KWICOS + $17.99 for a loader), this combination of hardware and software is *the* mass storage answer for the computerist on a budget. If you are still using a cassette system and do not plan to expand in the near future, there is *no excuse* not to purchase these products.

continued on page 62

COMPUTRONICS CLASSIFIEDS

CLASSIFIED ADVERTISING

Introductory Rates (per insertion)	1×	3×	6×	12×
Special discount Price (to 25 words)	$20	$15	$12	$10
Special Discount Add'l charge/word	$.45	$.40	$.35	$.30

To figure cost of ad, consider words like "a," "the," "etc." as one word each. Telephone number with area code counts as two words. Please type or print your ad and send along with payment in full (check, money order, Visa, Master Card, or American Express) to *H & E Computronics, Inc., Classified Advertising Department, 50 North Pascack Road, Spring Valley, NY 10977.* Your ad will begin in the next available issue.

PROGRAMMER'S AIDE: easy way to keep track of variables. 5 books per package, $4.95. Cypress Computer, 5244 Oxford, Cypress CA 90630. Cash, check, or money order.

STOCK ANALYSIS PROGRAMS FOR TRS-80. For more information, write to:
R. DeCrick
7655 Whispering Brook C21
Portage, MI 49081

YOUR AD CAN APPEAR HERE for as little as $10 per month.

DIABLO HyType I PRINTER, word processing quality line printer with tractor feed attachment and full set of typewheels and ribbons, $1600. Interfaces to expansion port of TRS-80 Model I. Complete software drivers for 10 or 12 character per inch printing, with or without automatic page eject. Original cost over $2300. Selling because we will no longer be using Model I. (914) 634-1821.

PROGRAMS PUBLISHED IN Computronics on diskettes. Do you want to avoid typing in all those programs published each month in *Computronics?* For $12.00, you can have a diskette containing **all** the programs published in any single issue *with corrections.* Specify the issue number you want and whether you want Model I or Model III media (not available for other models). Order from Box A, H & E Computronics, 50 North Pascack Road, Spring Valley, NY 10977, or call toll-free order number (800) 431-2818 outside New York state; inside New York, call (914) 425-1535. Add $3.00 shipping and handling charge.

COMPUTRONICS Issue #37 (August 1981) program diskette contains the following programs: Calendar, Labels, Pizza, Diving, Payroll, Citysort, Graph Print and Text Print, and Alien Invasion Force.

COMPUTRONICS Issue #41 (January 1982) program diskette contains the following programs: Amortization of Loans, Matrix, Gauss Elimination, Linear Equations, Spellout, Currency Conversions, Density, and 110 Baud ASCII Radio Teletype Receive Program.

COMPUTRONICS Issue #42 (February 1982) program diskette contains the following programs: The Graphic Pie, Mortgage Comparisons, Statement of Income, Chainer, Windcrab, Horsepower, and Permute.

COMPUTRONICS Issue #43 (March 1982) program diskette contains the following programs: Interest Formulas, Depreciation under the 1981 Economic Recovery Tax Act, Electric, Sieve, and Horizon, and CLINTEST, a series of programs to help diabetics monitor their sugar levels, including Menu, Clintest, Foodlog, Calcount, Master and Print.

COMPUTRONICS Issue #44 (April 1982) program diskette contains the following programs: Check writer, Castle Adventure, Graphic Combinations, the Ladder, National Debt, Box, Roots, and April Fool.

COMPUTRONICS Issue #46 (June 1982) program diskette contains the following programs: Break Even Analysis, Coin Inventory program, Chase, Reaction Time, USA, Selling Price, Loan Processor, No Star Baseball, and Metric Conversion Calculator.

COMPUTRONICS Issue #48 (August 1982) program diskette contains the following programs: Information Retrieval System, including KW, Sort, Merge, and List; the Transportation method of linear programming; the Beale Treasure, including Filer, Write1, Write2, Write3, Process/Let, Process/L1, and Process/L2; and the Versatile Peeker.

COMPUTRONICS Issue #49 (September 1982) program diskette contains the following programs: the Simplex method of linear programming; Records and Files, including Schedule, Schedule/Sub, Schedule/Seq, Schedule/Ran; Monthly Expenditure Information package, including Update, Mstex, Monthre, and Report; and Scramble.

COMPUTRONICS Issue #50 (October 1982) program diskette contains the following programs: Cash flow analysis for real estate and other investments; Epson MX-80 Graftrax Initialization program; Kaleidoscope, Gascost, Permile, and Balloon by Gordon Speer.

COMPUTRONICS Issue #51 (November 1982) program diskette contains the following programs: the Cardwriter, the Graphical Method of Linear Programming, MACS-MAP, and Phone bill sorting program.

COMPUTRONICS Issue #52 (December 1982) program diskette contains the following programs: Regression Analysis with Confidence and Prediction Limits, Change baud rates on system tapes, PERT—Program Evaluation and Review Technique, Grid, and Monogram.

COMPUTRONICS Issue #53 (January 1983) program diskette contains the following programs: General Ledger Menu, Alpha Program, Roulette, Serious EDTASM, Surname Conversion, Jack and Jill, and Disk Checkbook Maintenance System.

COMPUTRONICS Issue #54 (February 1983) program diskette contains the following programs: Startup routines for General Ledger package, CLOCK/BAS, Findit, Boxer.

COMPUTRONICS Issue #55 (March 1983) program diskette contains the following programs: TRANS program to input transactions, Michigan and Graphics for Epson, Type, Alphabet Puzzle, Fraction Calculator, Memory Display, two programs for bit-image graphics, and Bowling Statistics Ledger.

CASSETTE ENHANCEMENTS
continued from page 61

Richard Kaplan
H & E Computronics
50 N. Pascack Road
Spring Valley, NY 10977

Loaders available from:

Lemons Tech Services
P.O. Box 429
Buffalo, MO 65622
(417) 345-7643

KWIK software available from:

KWIK Software
Box 328
Bolivar, MO 65613
(417) 326-7154 ■

The Original Magazine for Owners of the TRS-80™ MicroComputer

MODEL I • MODEL II/16 • MODEL III • POCKET COMPUTER • COLOR COMPUTER

Software for TRS-80 Owners

H & E COMPUTRONICS INC.

Monthly Newsmagazine for TRS-80 Owners

MONTHLY NEWS MAGAZINE

- PRACTICAL APPLICATIONS
- NEW EXPANDED BUSINESS SECTIONS
- GAMBLING
- GAMES
- EDUCATION
- PERSONAL FINANCE
- BEGINNER'S CORNER
- NEW PRODUCTS
- SOFTWARE EXCHANGE
- MARKET PLACE
- QUESTIONS & ANSWERS
- PROGRAM PRINT OUTS

...and MORE!

YOUR CHOICE FREE

with your Subscription or Renewal

A. ★★★★ $1000 ★★★★

More than $1000 worth of valuable coupons for free merchandise, discount software, hardware at a fraction over cost and computer supplies.

OR

B. **A Word Processor, Data Management System and Cleanup**

(A maze game) All on cassette. (Add $3 for diskette, add $5 for modified MOD-II/16 diskette version—N/A on color computer or pocket computer.)

H & E COMPUTRONICS INC.

50 North Pascack Road
Spring Valley, New York 10977

CALL TOLL FREE 800-431-2818

(Outside of New York State)

24 HOUR ORDER LINE
914-425-1535

MasterCharge VISA American Express Card

☐ One Year Magazine Subscription $24 ☐ New ☐ Renewal
☐ Two Year Magazine Subscription $48.00 ☐ New ☐ Renewal
☐ Sample Issue $4. ☐ Mod II/16 Newsletter Subscription $18

Your Choice: $1000 offer (see box A) ☐ or Word Processor/Data Management ☐
Model I ☐ Model II/16 ☐ Model III ☐ Color Computer ☐ Pocket Computer ☐

NEW! NEW! **MOD-II/16 NEWSLETTER** $18/Year (or 12 issues)

Name _____ Address _____ City _____

State _____ Zip _____ Signature _____

Credit Card Number _____ Expiration Date _____

Add $12/Year (Canada, Mexico)—Add $24/ Year Air Mail outside of U.S.A., Canada, and Mexico
All Prices and Specifications Subject to Change

* TRS-80 is a trademark of the Radio Shack Division of Tandy Corp.

Form **1040A** Department of the Treasury—Internal Revenue Service **U.S. Individual Income Tax Return** **1982** (99) OMB No. 1545-0085

Use IRS label. Otherwise, please print or type.	Your first name and initial (if joint return, also give spouse's name and initial)	Last name	Your social security number
	Present home address (Number and street, including apartment number, or rural route)		Spouse's social security no.
	City, town or post office, State and ZIP code	Your occupation ▶ / Spouse's occupation ▶	

Presidential Election Campaign ▶ Do you want $1 to go to this fund? . . . . . Yes ☐ No ☐ Note: *Checking "Yes" will not increase your tax or reduce your refund.*
If joint return, does your spouse want $1 to go to this fund? . . . Yes ☐ No ☐

M&E COMPUTRONICS INC.

• • EVERYTHING FOR YOUR TRS-80™ • ATARI™ • APPLE™ • PET™ • CP/M™ • XEROX™ • IBM™ • OSBORNE™ • • • • KAYPRO • •

* TRS-80 is a trademark of the Radio Shack Division of Tandy Corp. * ATARI is a trademark of Atari Inc. *APPLE is a trademark of Apple Corp. * PET is a trademark of Commodore * CP/M is a trademark of Digital Research *XEROX is a trademark of Xerox Corp. * IBM is a trademark of IBM Corp. * OSBORNE is a trademark of Osborne Corp.

BUSINESS PAC 100

100 Ready-To-Run Business Programs

★ *All orders processed within 24-Hours*
★ *30-Day money back guarantee*

(ON CASSETTE OR DISKETTE).....Includes 128 Page Users Manual.....
Inventory Control.....Payroll.....Bookkeeping System.....Stock Calculations.....
Checkbook Maintenance.....Accounts Receivable.....Accounts Payable.....

BUSINESS 100 PROGRAM LIST

	NAME	DESCRIPTION
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	REQWITH	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
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

☐ **TRS-80 Cassette Version** $99.95
☐ **TRS-80 (Mod-I or III), Pet, Apple or Atari Versions** $99.95
☐ **TRS-80 Mod-II, IBM, Osborne and CP/M Versions** $149.95

ADD $3.00 FOR SHIPPING IN UPS AREAS
ADD $4.00 FOR C.O.D. OR NON-UPS AREAS
ADD $5.00 TO CANADA AND MEXICO
ADD PROPER POSTAGE OUTSIDE OF U.S., CANADA AND MEXICO

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

AMERICAN EXPRESS · VISA · master charge

24 HOUR ORDER LINE
(914) 425-1535

M&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

ASK FOR OUR 64-PAGE CATALOG
DEALER INQUIRIES INVITED

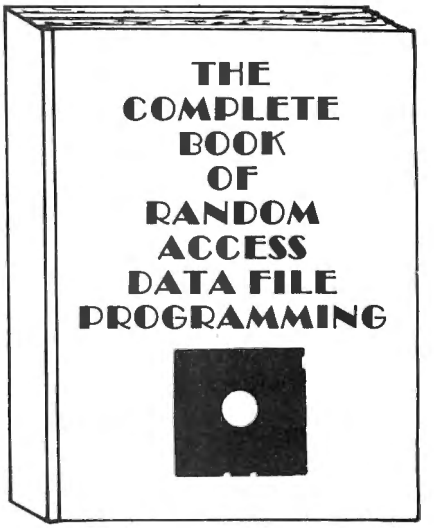
ALL PRICES & SPECIFICATIONS SUBJECT TO CHANGE
DELIVERY SUBJECT TO AVAILABILITY

The Complete Book Of Random Access Data File Programming

For TRS-80*, IBM Personal Computer*, Osborne*, and all Microsoft BASIC* computers

The last word on disk random access and file handling techniques, this series is intended for everyone — beginning programmers, businessmen and professionals will learn how to create custom programs to handle inventories, mailing lists, work scheduling, record keeping, and many other tasks, while more experienced programmers will learn advanced, professional programming techniques for faster, more efficient data storage and retrieval.

Although random access file handling is a matter of some complexity, the subject has been treated in a simple and down-to-earth fashion, so that anyone with some small familiarity with programming in Microsoft BASIC will be able to cope with the material. Each stage of learning uses a sample program as a starting point. The programs grow in capability and complexity as the books progress into all of the various aspects of file handling and record manipulation. An extensive effort has been made to keep the material coherent and every program line is explained in detail.



Volume I: Basic File Handling Techniques

- The writing of a Menu to Summarize program functions
- The writing of a screen format to accept data for records
- The creation of the basic record
- The FIELD and LSET routines for buffer preparation
- The writing of the record to disk in a random access mode
- The ability to change or edit a record
- The LPRINT capability from disk using three different formats
- Deleting a record from a random file
- Sorting the random file
- Searching the random file by name or key field
- The ability to search in a "NEXT or PRIOR" fashion
- The ability to purge deleted records from a disk file
- The ability to calculate with data from a disk file
- The provision for future expansion of the data fields
- The use of flags to prevent program crashes
- Date setting, printer on-line and many other routines to make a program run like a commercially-written program

Volume II: Advanced File Handling Techniques

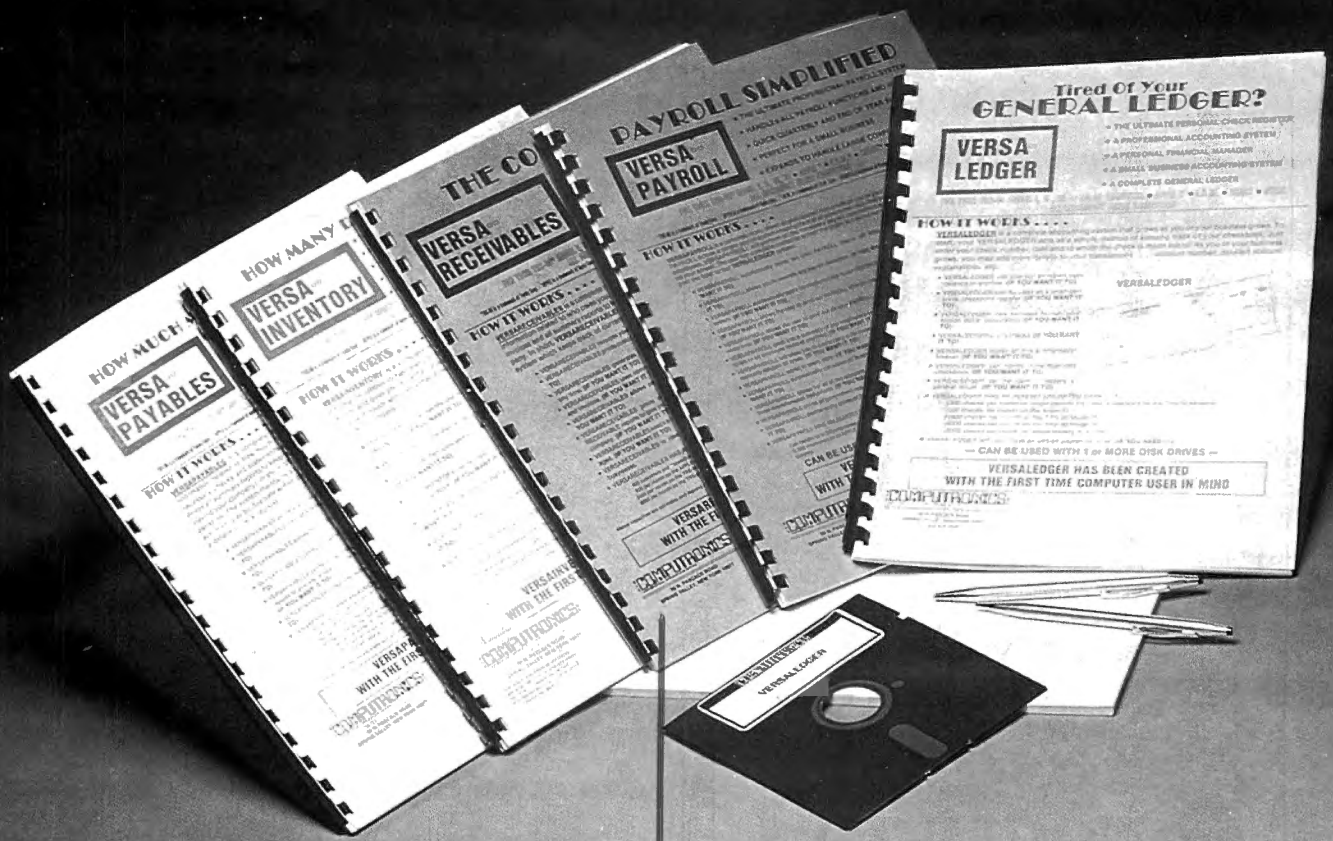
- Blocking & de-blocking, Shell-Metzner sort, In-place screen editing, recovery of deleted record space
- Alpha-index record retrieval, fast machine/BASIC sort
- Linked list record structure and sort-merge, deleted record removal and file reorganization
- Multi-key file reorganization and record searching
- Relational database programming--comprehensive self-balancing accounting system with printouts
- Hashcoded data file manipulation--(probably the fastest method of data retrieval). Hashing the input key and recovery method explained
- Span-blocking techniques (allows; creation of records longer than 256 bytes without wasted space

The Complete Book Of Random Access Data File Programming

Volume I: Basic File Handling Techniques	**$29.95**
optional Vol. I Program Disk for Model I/III	**$28.50**
optional Vol. I Program Disk for Model II	**$32.50**
Volume II: Advanced File Handling Techniques	**$29.95**
optional Vol. II Program Disks for Models I, II or III	**$49.95**

Introducing the Most Powerful Business Software Ever!

TRS-80™ (Model I, II, III, or 16) • APPLE™ • IBM™ • OSBORNE™ • CP/M™ • XEROX™



The VERSABUSINESS™ Series

Each VERSABUSINESS module can be purchased and used independently, or can be linked in any combination to form a complete, coordinated business system.

VERSARECEIVABLES™ $99.95

VERSARECEIVABLES™ is a complete menu-driven accounts receivable, invoicing, and monthly statement-generating system. It keeps track of all information related to who owes you or your company money, and can provide automatic billing for past due accounts. VERSARECEIVABLES™ prints all necessary statements, invoices, and summary reports and can be linked with VERSALEDGER II™ and VERSAINVENTORY™.

VERSAPAYABLES™ $99.95

VERSAPAYABLES™ is designed to keep track of current and aged payables, keeping you in touch with all information regarding how much money your company owes, and to whom. VERSAPAYABLES™ maintains a complete record on each vendor, prints checks, check registers, vouchers, transaction reports, aged payables reports, vendor reports, and more. With VERSAPAYABLES™, you can even let your computer automatically select which vouchers are to be paid.

VERSAPAYROLL™ $99.95

VERSAPAYROLL™ is a powerful and sophisticated, but easy to use payroll system that keeps track of all government-required payroll information. Complete employee records are maintained, and all necessary payroll calculations are performed automatically, with totals displayed on screen for operator approval. A payroll can be run totally, automatically, or the operator can intervene to prevent a check from being printed, or to alter information on it. If desired, totals may be posted to the VERSALEDGER II™ system.

VERSAINVENTORY™ $99.95

VERSAINVENTORY™ is a complete inventory control system that gives you instant access to data on any item. VERSAINVENTORY™ keeps track of all information related to what items are in stock, out of stock, on backorder, etc., stores sales and pricing data, alerts you when an item falls below a preset reorder point, and allows you to enter and print invoices directly or to link with the VERSARECEIVABLES™ system. VERSAINVENTORY™ prints all needed inventory listings, reports of items below reorder point, inventory value reports, period and year-to-date sales reports, price lists, inventory checklists, etc.

VERSALEDGER II™ $149.95

VERSALEDGER II™ is a complete accounting system that grows as your business grows. VERSALEDGER II™ can be used as a simple personal checkbook register, expanded to a small business bookkeeping system or developed into a large corporate general ledger system **without any additional software.**

- VERSALEDGER II™ gives you almost unlimited storage capacity (300 to 10,000 entries per month, depending on the system),
- stores all check and general ledger information forever,
- prints tractor-feed checks,
- handles multiple checkbooks and general ledgers,
- prints 17 customized accounting reports including check registers, balance sheets, income statements, transaction reports, account listings, etc.

VERSALEDGER II™ comes with a professionally-written 160 page manual designed for first-time users. The VERSALEDGER II™ manual will help you become quickly familiar with VERSALEDGER II™, using complete sample data files supplied on diskette and more than 50 pages of sample printouts.

SATISFACTION GUARANTEED!

Every **VERSABUSINESS™** module is guaranteed to outperform all other competitive systems, and at a fraction of their cost. If you are not satisfied with any **VERSABUSINESS™** module, you may return it within 30 days for a refund. Manuals for any **VERSABUSINESS™** module may be purchased for $25 each, credited toward a later purchase of that module.

To Order:

Write or call Toll-free (800) 431-2818
(N.Y.S. residents call 914-425-1535)

* add $3 for shipping in UPS areas
* add $4 for C.O.D. or non-UPS areas
* add $5 to CANADA or MEXICO
* add proper postage elsewhere

DEALER INQUIRIES WELCOME

All prices and specifications subject to change / Delivery subject to availability.

COMPUTRONICS INC.

50 N. PASCACK ROAD, SPRING VALLEY, N.Y. 10977

* TRS-80 is a trademark of the Radio Shack Division of Tandy Corp. - *APPLE is a trademark of Apple Corp. - *IBM is a trademark of IBM Corp. - *OSBORNE is a trademark of Osborne Corp. *CP/M is a trademark of Digital Research - *XEROX is a trademark of Xerox Corp.

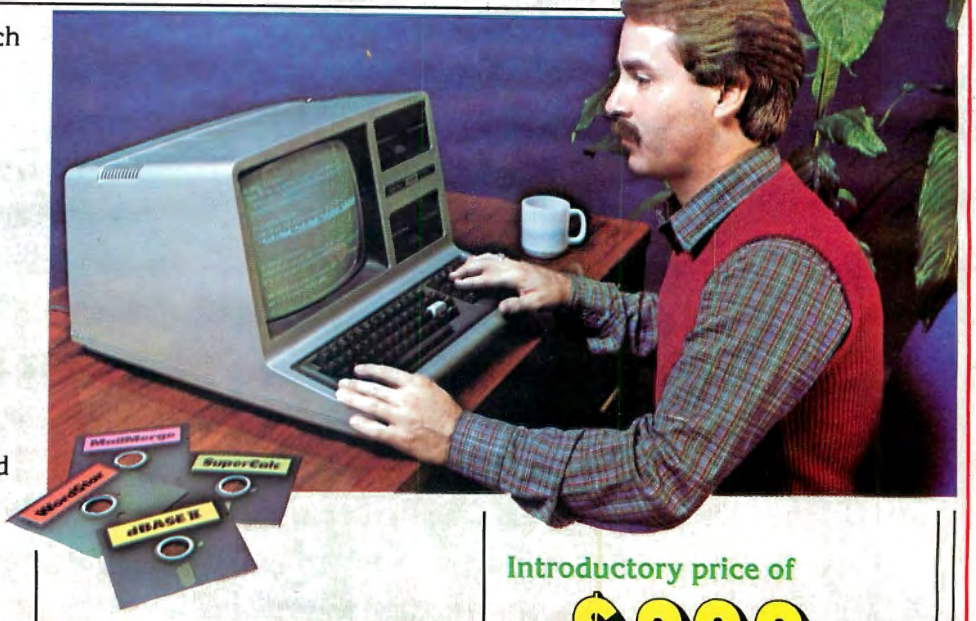
ADVERTISING DIRECTORY

Put 64K CP/M® 2.2 in your TRS-80 Model III and tap into 2,000 business programs.

Now you can run programs such as WordStar, dBASE II, SuperCalc, MailMerge and virtually thousands of other CP/M-based programs on your TRS-80 Model III.

CP/M 2.2 is the industry standard operating system that gives you access right now to over 2,000 off-the-shelf business programs.

Our plug-in Shuffleboard III comes with 16K of RAM, giving your Model III the power of full 64K CP/M 2.2 without interference of the ROM or video memory. In fact, the Shuffleboard will appear transparent in the TRS-80 mode and will not interfere with any DOS operation.



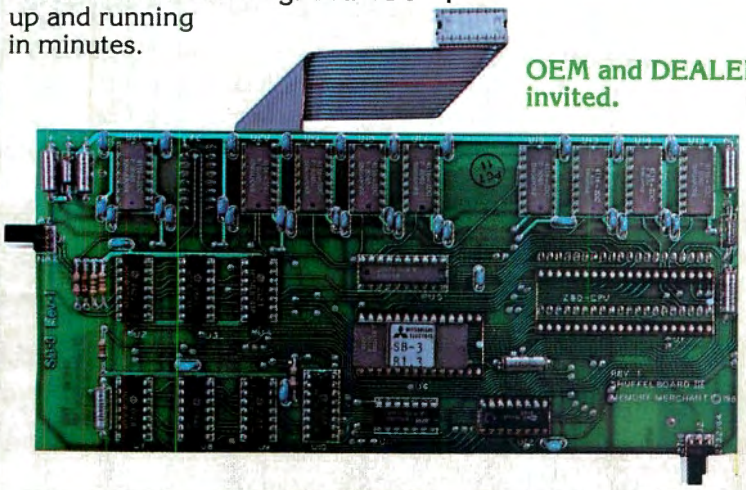
READ and WRITE Osborne, Xerox and IBM personal computer software plus many more popular formats.

Unfortunately, there is no standardized CP/M format for 5¼" diskettes. But we have developed a way to READ/WRITE and RUN standard programs under the following single-sided formats: Osborne 1 S/D, Xerox 820 S/D, IBM PC* D/D for CP/M 86 only, Superbrain D/D, Kapro II D/D, HP 125 D/D and TeleVideo D/D.

*Will Read and Write Only.

Easy plug-in installation.

It's so simple. The Shuffleboard III plugs into two existing sockets inside your Model III. There are no permanent modifications, no cut traces and no soldering. You'll be up and running in minutes.



New Products.

80 x 24 VIDEO BOARD: Features dual intensity screen, programmable cursor control for block, underline & blink rate, on-board bell with audible keyclick, battery-operated real time calendar/clock, full ASCII character set plus 256 special character graphics, dual RS-232 outputs and composite video output.

FLOPPY DISK CONTROLLER: Now you can access 5¼" and 8" floppy disk drives in any combination up to 4 drives of S/D density, S/D sided. Tap into a wealth of CP/M software which comes on 8" IBM 3740 format or Pickles & Trout CP/M for the Model II.

SOFTWARE: Additional CP/M software programs are available. Call or write for details.

OEM and DEALER inquiries invited.

Introductory price of

$299.

The Shuffleboard III comes fully burned-in and tested complete with 64K CP/M 2.2 and MBASIC 80 interpreter, plus software manuals and a first class user's manual — with a 1-year limited warranty and 15-day no-risk free trial — for only $299.

See the Shuffleboard III at your dealer's now.

Once you see what the Shuffleboard can do for your Model III you'll want one at once. If your dealer does not yet stock the Shuffleboard have him give us a call. Or send check, money order, VISA or MASTERCARD number (sorry, no COD's) plus $5 shipping per board ($17 outside the USA & Canada)* directly to the address below. Cal. residents please add sales tax. Credit card purchases can be phoned in directly and we'll ship from stock.

(415) 483-1008

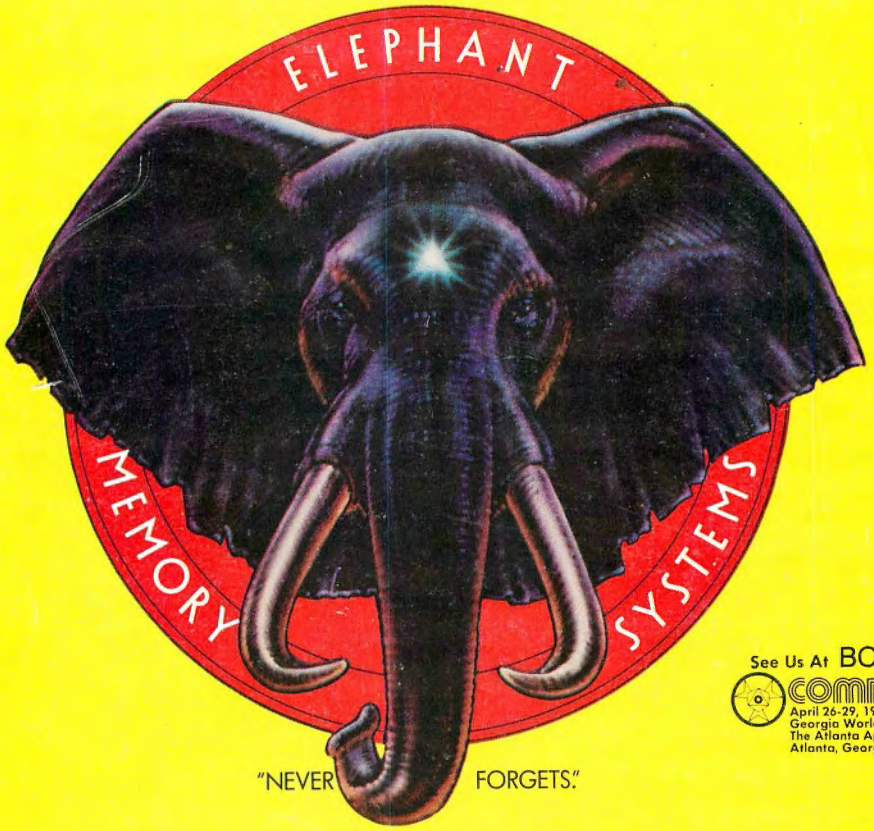
*Air mail shipments to Canada & all other countries.

Memory Merchant™

14666 Doolittle Drive San Leandro, CA 94577
(415) 483-1008

WordStar & MailMerge are trademarks of MicroPro.
SuperCalc is a trademark of SORCIM.
dBASE II is a trademark of Ashton-Tate.
CP/M is a trademark of Digital Research.
TRS-80 is a trademark of Tandy Corporation.
IBM is a trademark of IBM Corporation.

REMEMBER:



See Us At BOOTH #1146
COMDEX/SPRING '83
April 26-29, 1983
Georgia World Congress Center and
The Atlanta Apparel Mart
Atlanta, Georgia

MORE THAN JUST ANOTHER PRETTY FACE.

Says who? Says ANSI.

Specifically, subcommittee X3B8 of the American National Standards Institute (ANSI) says so. The fact is all Elephant™ floppies meet or exceed the specs required to meet or exceed all their standards.

But just who is "subcommittee X3B8" to issue such pronouncements?

They're a group of people representing a large, well-balanced cross section of disciplines—from academia, government agencies, and the computer industry. People from places like IBM, Hewlett-Packard, 3M, Lawrence Livermore Labs, The U.S. Department of Defense, Honeywell and The Association of Computer Programmers and Analysts. In short, it's a bunch of high-caliber nitpickers whose mission, it seems, in order to make better disks for consumers, is also to make life miserable for everyone in the disk-making business.

How? By gathering together periodically (often, one suspects, under the full moon) to concoct more and more rules to increase the quality of flexible disks. Their most recent rule book runs over 20 single-spaced pages—listing, and insisting upon—hundreds upon hundreds of standards a disk must meet in order to be blessed by ANSI. (And thereby be taken seriously by people who take disks seriously.)

In fact, if you'd like a copy of this formidable document, for free, just let us know and we'll send you one. Because once you know what it takes to make an Elephant for ANSI . . .

We think you'll want us to make some Elephants for you.

ELEPHANT.™ HEAVY DUTY DISKS.

For a free poster-size portrait of our powerful pachyderm, please write us.
Distributed Exclusively by Leading Edge Products, Inc., 225 Turnpike Street, Canton, Massachusetts 02021
Call: toll-free 1-800-343-6833; or in Massachusetts call collect (617) 828-8150. Telex 951-624.