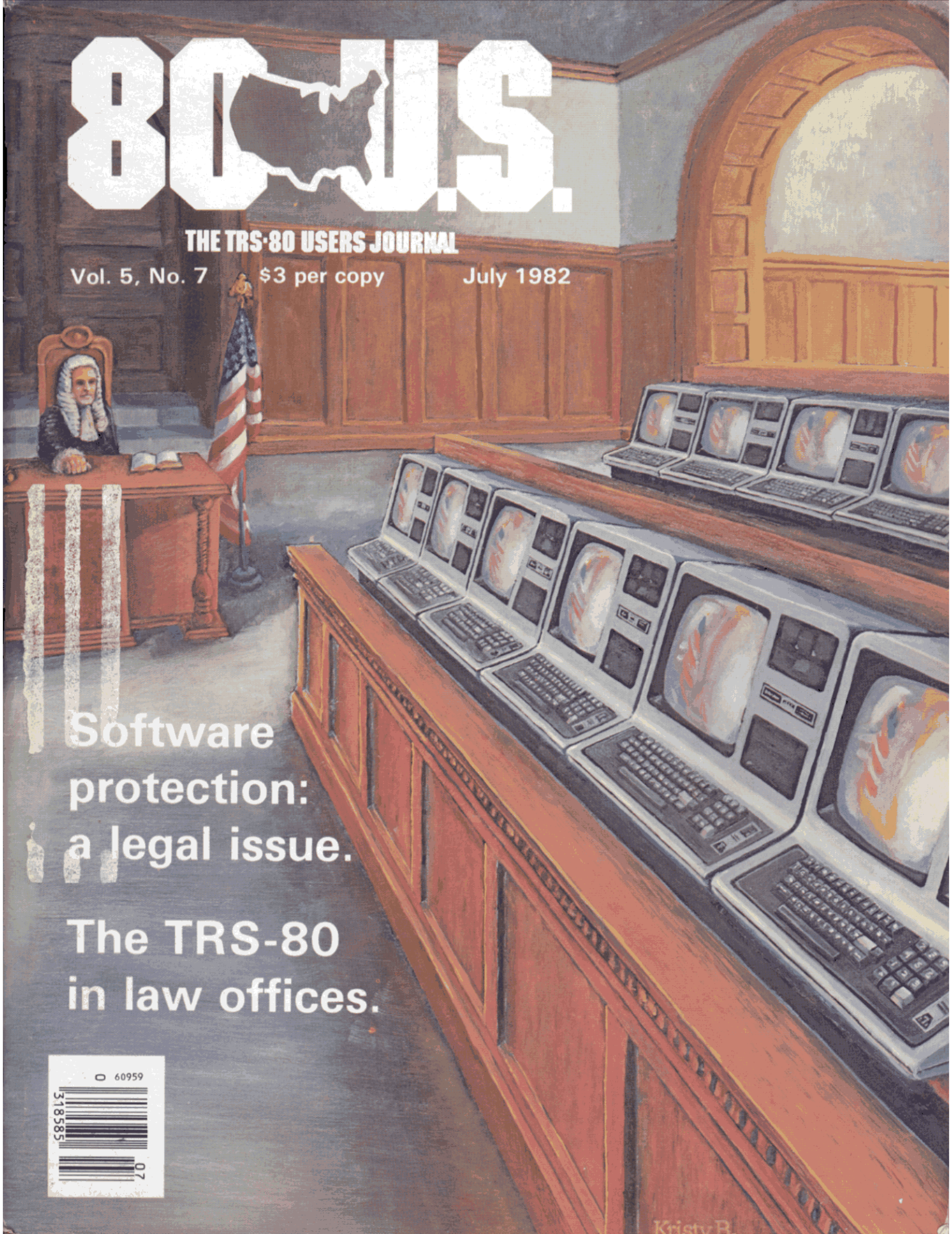
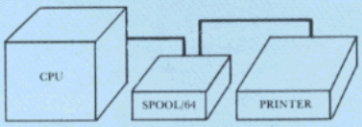

80 U.S.
THE TRS·80 USERS JOURNAL
Vol. 5, No. 7
$3 per copy
July 1982
Software protection: a legal issue.
The TRS-80 in law offices.
0 60959 31858507
Kristy B.

How many times must your TRS-80 stop just to talk to your printer?

Your computer talks faster than your printer can listen. Expensive time is lost forever. SPOOL/64, a 64k byte hardware print spooler, keeps your computer processing while your printer is printing. No more waiting. No more wasting.

SPOOL/64 buffers your print output and transmits the information to your printer at the printer's specific print rate. It buffers up to 13 minutes of print output (at 80 characters per second).

SPOOL/64 plugs directly into both your printer and



computer, accepts input from parallel centronics cables (serial RS-232 available soon), and its user programming feature allows you to download your own programs so SPOOL/64 can handle sophisticated print routines. And SPOOL/64, used with your software spooler, can give you the highest performance spooling available.

SPOOL/64 utilizes 64k of dynamic RAM, a Z-80 CPU, a 2716 ROM and a copy of the source program for the object code in the ROM is included.

SPOOL/64, priced at just $399, is the most cost/effective print spooler on the market. Evaluate performance. Evaluate price. Then let SPOOL/64 get your computer back talking to you. Call or write Apparat today for a free brochure. Dealer inquiries invited.

VISA MasterCard

Apparat, Inc.

4401 So. Tamarac Parkway, Denver, CO 80237 (303) 741-1778

"ON GOING SUPPORT FOR MICROCOMPUTERS"

#1

FORTRESS

Only the Fast Survive!

SSM SOFT SECTOR MARKETING, INCORPORATED

6250 Middlebelt • Garden City, MI 48135 • 800-521-6504/(313) 425-4020

Written by Larry Ashmun

Prices per Game: TRS-80 16K Level II Mod I/Mod III Cassette $15.95
TRS-80 32K Level II Mod I/Mod III Diskette $19.95

10% discount for 2 items, 15% for 3 or more. Please add $2.50 per order for postage & handling. Michigan residents add 4% sales tax. Outside USA (except Canada) please add $10.00 per order for postage & handling.

Sound effects are playable through the cassette AUX plug. High scores are automatically saved after each game on disk versions.

Call or write for our complete catalog.

VISA MasterCard AMERICAN EXPRESS

So Far, So Good...

Your Model III is a fine computer. So why settle for less than a fine disk storage system?

A fast, reliable disk storage system — no other device is so vital to the satisfactory performance of your computer. At Percom we build quality, high-performance disk drive systems. From gold-plated connector contacts to goldcad metal chassis structures. From proven design through 100% reliability testing. Percom disk systems are the standard by which others are judged — the industry's "gold standard," in a sense. And since Percom is the largest independent manufacturer of disk systems for microcomputers, you get Percom quality at very competitive prices. Add-on drives for TRS-80* computers start as low as $399. Complete first-drive systems for the Model III start at only $599. Put a quality Percom mini-disk storage system in your Model III. And save.

#1 For Your Model III

Percom TFD drives for your TRS-80* Model III computer are available in 40- and 80-track versions with single or dual heads. Single-head 40-track drives store 180 Kbytes; dual-head 40-track drives store 360 Kbytes. Eighty-track drives store slightly over twice these amounts. Of course these are formatted capacities. The Percom Model III controller handles up to four drives so it's possible to access almost three million bytes of on-line program and data files. You get Percom's OS-80/III Basic language DOS with each first-drive system, and your first drive may be either internal (add-in) or external (add-on). Percom TFD drives work with Model III TRSDOS and other Model III disk-operating systems. First-drive systems are pre-assembled. Installation is accomplished with simple tools.

#1 For Your Model I, Too

Percom TFD add-on drives for the TRS-80* Model I computer are available with the same features and the same quality control measures as TFD Model III drives. As for Model III drives, all Percom Model I drives are double-density rated. Install Percom's popular DOUBLER II in your Model I Expansion Interface and upgrade your Model I to provide the same storage capacity as the Model III.

Watch for Announcement of Percom's Low-Cost Hard Disk System for the Model III!

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

PERCOM

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

The Drive People
You get more out of Percom disk systems.
Expect it!

PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE.

*TRS-80 is a trademark of Tandy Radio Shack Corporation which has no relationship to Percom Data Company, Inc.
PERCOM, DOUBLER II and OS-80/III are trademarks of Percom Data Company, Inc.

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

☐ MODEL I ☐ MODEL III

Send to
PERCOM DATA COMPANY, INC. Dept. 8-U
11220 Pagemill Road, Dallas, TX. 75243

NAME

STREET

CITY STATE

ZIP PHONE NUMBER

After three years of selling my Model I and Model III programs, I've earned back my development costs.

So I can lower the price.

Now I'm offering my Model I and Model III programs for $75 each.

They've been checked out by thousands of TRS-80* users, most of whom get in touch with me, Irwin Taranto. Thousands of phone calls later, these systems are completely developed, checked out, glitch-free.

When people call, we've heard all the questions and we can answer them right off. I don't have to get on the phone and work through problems like I used to.

Since I'm getting off so easy, the least I can do is drop the price – 50% for General Ledger, 25% for the rest.

These are my Model I and Model III programs:

Accounts Payable It links to the General Ledger, calculates and prints checks and makes reports. It's an invoice-linked system.

Accounts Receivable It keeps track of billed and unbilled invoices, open and closed items and aging. It prints statements and links to the General Ledger.

General Ledger It keeps track of time by month, quarter, year and the previous three quarters. It even includes a Cash Journal.

Inventory Control It gives an immediate read-out on any item inquiry, including quantity and dollar total.

Invoicing It prints your detailed invoices and links to Accounts Receivable and the General Ledger.

Payroll It keeps the files, computes pay and deductions, prints forms and checks, figures taxes, overtime and piecework pay in any state tax routine, and prints the 941-A and W-2 forms.

They're all yours, for $75 each. You also need documentation when you run our systems. The Osborne books – one for Accounts Payable and Receivable, one for General Ledger, one for Payroll – cost $20 each. Our invoicing book costs $10.

Just send me the coupon, or call us toll free. We'll ship within 48 hours.

Please send me the following programs at $75 each.

		book ☑
Accounts Payable		
Accounts Receivable		
General Ledger		
Inventory Control		
Invoicing		
Payroll		
Add $4.50 per order for handling		
6% tax (California only)		
AMOUNT ENCLOSED		

If you need the books, add $20 each. The invoicing book is $10.

Mastercharge ☐ Visa ☐ No. _____ Expires _____

☐ Please send me information on other Taranto business programs, including TRS-80 Model II accounting systems.

Your name _____

Company name _____

Address _____

City/State/Zip _____

Taranto
& ASSOCIATES, INC.

The Total System Store,™ 121 Paul Drive, San Rafael, CA 94903. Mail inquiries to P.O. Box 6216, San Rafael, CA 94903. Outside California, toll free (800) 227-2868. In California, (415) 472-2670.

3

*A trademark of the Tandy Corporation

 Please address correspondence to: **80-U.S. Journal, 3838 South Warner Street, Tacoma, Washington 98409, telephone (206) 475-2219.**

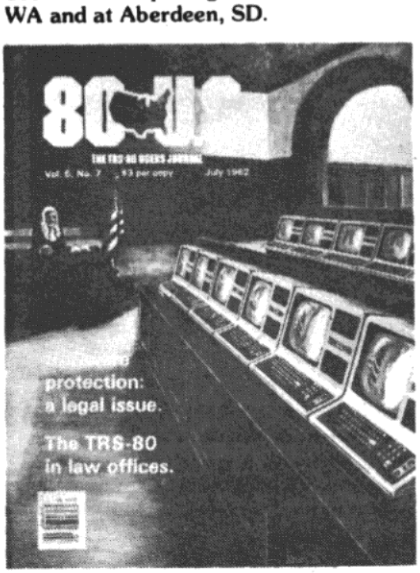
Advertisers: 80-U.S. Journal will accept relevant commercial advertising which pertains to, or is for use on, the Tandy Corporation microcomputers. Write for a current rate schedule.

Authors: We constantly seek material from contributors. Send your material (double spaced, upper/lower case, please) and allow approximately 4 to 6 weeks for review. Programs must be supplied in machine readable form on diskette or tape, clearly marked as to model and operating system. Text files may be on diskette. Media will be returned if return postage is provided. Cartoons and photographs are welcome. Generous compensation will be made for non-trivial works which are accepted for publication. 80-U.S. Journal pays upon acceptance rather than on publication.

SUBSCRIPTION PRICE: U.S.: $16 for one year, $31 for two years and $45 for three years. **Canada and Mexico:** $25 per year, no two or three year subscriptions are offered. **All other:** $30 per year via surface mail, $72 per year via airmail. Two and three year subscriptions are not offered.

ISSN Publication #0199-1035. The 80-U.S. Journal is published monthly by 80-Northwest Publishing, Inc., 3838 S. Warner St., Tacoma WA 98409-4698. Printed in the United States of America.

POSTMASTER: Please send change of address form 3579 to 80-U.S. Journal, 5615 West Cermak Road, Cicero, Illinois 60650. **Second Class postage PAID at Tacoma, WA and at Aberdeen, SD.**



OUR COVER is an oil painting by Kristy Boardway-Dickson, Gig Harbor, WA. She displays a visual conception of microcomputers and law.

80-U.S.
THE TRS·80 USERS JOURNAL

TRS-80 is a trademark of the Tandy Corp.

Vol. 5, No. 7 — July 1982

ARTICLES

80-U.S.

Publisher/Editor-in-Chief
I. Mike Schmidt

Editors
Cameron C. Brown
Donald Scarberry

Associate Editors
Terry R. Dettmann
Jim Klaproth

Contributing Editors
Spencer Hall
Robert W. Liddil

Advertising Sales
Margaret G. Farrell

Promotion
Robert P. Perez

Production
Catherine D. Doud
Sherry A. Miksell

Editorial Secretary
Eva R. Jones

Circulation
Kristine E. Schmidt

Accounting
Helene Dalton

RENEWING?

Check your label to be sure it's correct. By the way, your label shows the last issue in your current subscription in the top right corner. (Example: OCT 82 means October of 1982 is the last issue you'll get.) For uninterrupted service include your label with your order.

MOVING?

Please enclose your label or write your name and address as it appears on your label.

Name_____

Address_____

City/State_____

Write in new address:

Name_____

Address_____

City/State_____

80-U.S. Journal
Subscription Department
5615 West Cermak Road
Cicero, IL 60650

Editorial

Practically everyone involved with the production, manufacture, distribution and sale of computer software is acutely aware of the copyright infringement problem that permeates the market place. Our feature story in this issue, written by Richard Stern, addresses the status of the newly amended copyright laws as they apply to copyrighted software. I hope you find his interpretations of the law to be as enlightening as I have. While reading his material I began to wonder how much consideration has been given to computer crime in general.

Businesses, in an attempt to increase employee productivity, are adding substantial numbers of microcomputers to their computing repertoire. As a result, larger amounts of confidential company data become vulnerable to potential unauthorized use. Computer security must be a major consideration for any business that is considering adding or expanding computer facilities. Computer-related crime is increasing rapidly, and failure to recognize the importance of a controlled security system can be disastrous for many companies.

The first recorded computer-related crime occurred in 1964, and involved a programmer in Texas who stole his employer's computer programs, valued at $5 million. Fortunately, the programmer was caught and convicted, and he served a five-year prison sentence.

The first recorded federal computer-related crime occurred in 1966, when a programmer in a bank modified a computer program to disregard his overdrafts. Needless to say, he created a number of NSF checks until his act was discovered.

In 1968, it was discovered that an accountant had embezzled approximately $1 million over a six-year period. The number of recorded computer-related crimes has increased from eight in 1969, to scores in 1981. Notice, I said "recorded." How many criminal acts have been perpetrated by individuals who have never been caught? There may be no way to tell — until it's too late!

A practical security consideration is to reduce the number of potential perpetrators to as few people as possible. Computer systems are no more secure than their environment or procedures used during operation and maintenance. Adequate control of access, and sufficient procedures and discipline in maintaining them, must be implemented. Security precautions should be present in any computer facility where the threat of loss or damage is present.

It is my feeling that many security activities should be automated. Manufacturers of computer hardware should include security protective devices in their equipment, thereby providing "turnkey" protection. This would not be a total solution because the computer will still be vulnerable to maintenance people with sufficient skills, knowledge and access. Still, manufacturers can provide a large measure of protection which businesses can use as a springboard for developing adequate protection schemes.

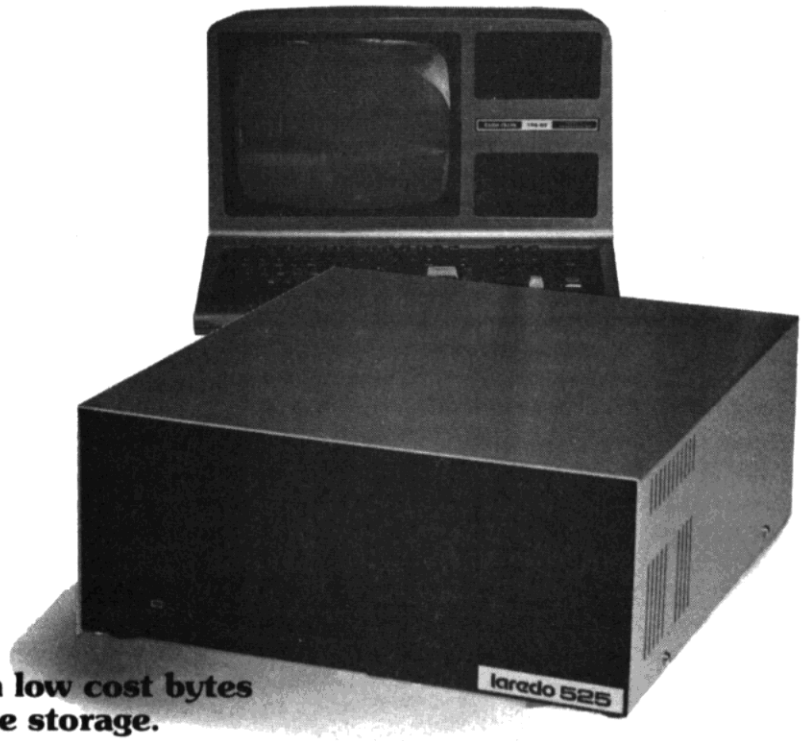
No comprehensive technological solution for adequately protecting computers is at hand (and none is expected for some years). In the meantime, the businessman must protect his system using currently available methods, and attempt to provide a reasonable degree of security. Limited accessibility would be a vast improvement over the present situation, where large numbers of computer users have the capability to subvert computers.

Have you considered the security of your computer system, software and data lately?

Don Scarberry

For TRS-80 Models I and III

Introducing the Lowest Cost Hard Disk System Ever . . Laredo's LS525.



$1995*

5 million low cost bytes of on-line storage.

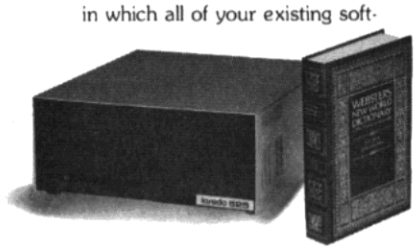
At $1995.00 affordable hard disk storage is now within your reach.

The LS525 hard disk system is easier to use than your printer . just plug it in switch it on and your disk storage problems are over forever.

The LS525 gives you unheard-of speed and reliability. For starters we built a "mainframe like" controller that can detect disk surface defects and relocate the data to alternate tracks. Talking about speed the LS525 can access files 10 times faster than your floppy.

This coupled with proven "winchester" technology is what we mean by reliability and speed.

We also provide a user-friendly TRSDOS compatable operating system in which all of your existing software can run on. It provides features that can make the LS525 look like one big 5M byte drive or four small 1.25M drives or whatever you like.



The LS525 is a compact (less than one half cubic foot) "color-keyed" package that will fit perfectly next to your TRS-80. So if your tired of seeing "Disk Space Full". . . . .

. . . there's no excuse.

Call toll free and order yours now.

(800) 538-5137

Inside California call (408) 980-1888.

* Model I or III host adapter $250 & LDOS $160 extra, specify when ordering

laredo systems inc.

. . .mass storage for personal computers.

Laredo Systems Inc. 2264 Calle De Luna, Santa Clara, California 95050

TRS-80 and TRSDOS are trademarks of Radio Shack a Tandy Co.

4

I read with great interest your review of spelling checkers in the February 1982 issue. As our Hexspell Spelling Checker was included in the review, I would like to make a few comments.

Having purchased and tested the other two products mentioned, I think the author generally did a good job of explaining their features. There is one area where I think something should be added however: that is the relative speed of the various programs.

Hexspell displays the text at reading speed as it is checked. Ensuring that the document is read through carefully before it is sent out does as much to increase accuracy as the spelling check itself. Hexspell allows a simultaneous check by human and computer, each looking for the sort of error it detects best.

People either love or hate this feature. Some people want the computer "to do it all," and Hexspell is not for them. But for people who welcome the chance to read through the document again, Hexspell is plenty fast enough. In fact, our latest version goes too fast for many people and has an adjustable delay feature to slow it down. In a system where the computer does it all, any time the program takes is wasted. In a system where you "read along" with the computer, you are both doing something useful at the same time, so the program needs only to be "fast enough."

When your review appeared the original Hexspell had been on sale for over a year. In fact, it was replaced at the end of 1981 by a much enhanced version called Hexspell 2. This has many additional features, such as a programmable character set, a word list extendable to 50,000 words and its own lower case driver. It uses a newer version of the Microsoft compiler and so includes the error-checking omitted from the original version. I hope you get a chance to review this new version in some future issue.

Bernard J. Hughes
Hexagon Systems
Vancouver, BC, Canada

I have just finished adapting the short-leader program *(80-U.S. Journal, April 1982)* to one of my data handling programs where I have a need for cassette backups to my Stringy Floppy data files. Congratulations to Kenneth Smith for an excellent idea.

Since I have the new ROM in my TRS-80, the program will not work as written. The following changes to listing 1 will adapt the program to the new ROM:

Delete lines 280—420
Delete the label at line 430
Change the jump address at line 440 to 207CH vice 209BH
Change the jump address at line 480 to 2079H vice 2096H

This reduces the program to 19 bytes and changes the program lines in listing 2 as follows:

```
270 FOR I=A TO A+18:READX:POKE I,X:NEXT I
280 DATA 217,225,217,254,35,194,124,32,205,254,1
290 DATA 6,40,205,137,2,195,121,32
(Delete line 300)
320 POKE A+12,X
```

I would also like to suggest that the program ORG be located at 4040H. This area of the communication region is used for DOS and would be available since this program will interfere with DOS. In using an ORG in this region, the program is not affected by memory size, thus ORG location or MEM SIZE does not have to be adjusted.

James A. Sladek
Norfolk, VA

I have tried to use listing 2 on page 56 of the April 1982 issue and all I get is SN ERROR on every line that has a print statement. I have checked my listing from 190 to 340 three times and no mistakes. Can you help? I have a late Model I Level II.

Gene K. Lynch
Boulder, MT

(Gene is referring to the article called "Short leaders for data tapes", by Kenneth R. Smith, whose reply follows):

Dear Mr. Lynch

The staff of 80-U.S. Journal *has forwarded me a copy of the note you sent them concerning problems using the short-leader patch I wrote for the Model I. I have rechecked the code and tried it again on a Model I. It works fine on the Model I tried, which, I am told, has one of the oldest sets of ROM available. Since your machine is newer, I suspect that the problem is that you have newer ROM chips which are incompatible with the patch.*

Here is a short program you can run on your Model I to ascertain whether or not the patch will work:

```
10 FOR I=8306 TO 8336
20 X=PEEK(I)
30 PRINT I,X
40 NEXT I
50 END
```

After you enter and run the program, the output should be:

8306	254	8317	30	8327	230
8307	64	8318	229	8328	63
8308	32	8319	33	8329	50
8309	25	8320	0	8330	166
8310	205	8321	60	8331	64
8311	1	8322	25	8332	225
8312	43	8323	34	8333	207
8313	254	8324	32	8334	44
8314	4	8325	64	8335	254
8315	210	8326	123	8336	35
8316	74				

If your output varies from this, the patch will not operate. You will, as you already know, receive the syntax error message.

GIMME FIVE!
BIG FIVE, THAT IS

DEFENSE COMMAND

Several months ago the Kromorfkrom Empire invaded our planet and stole some of our newly developed and highly efficient "Krotnium" Star Cruiser fuel cells. Your mission is to infiltrate the Kromorfkrom Empire and pass yourself off as the commanding officer of one of their fuel vessels. Eventually you will be discovered and then it's battle time! Sound and joystick control make this another winner.

Cat. No. 3853	Mod. I & III, 16K, cass	**$15.95**
Cat. No. 3854	Mod I & III, 32K, disk	**$19.95**

STELLAR ESCORT

Five billion light years from Kromorfkrom, Federation forces have been at war with the hostile Cretonian Empire. Unfortunately, the Cretonians attacked by surprise and now your forces are almost out of supplies. It's up to you, the Escort fighter pilot, to save the Federation. Sound and joystick option are included.

Cat. No. 3855	Mod I & III, 16K, cass	**$15.95**
Cat. No. 3856	Mod I & III, 32K, disk	**$19.95**

GALAXY INVASION

Cruel and crafty invaders have been spotted in battle formation warping towards Earth. You must quickly eliminate the aliens as they swoop down upon you. Includes sound and joystick action.

Cat. No. 3847	Mod I & III, 16K, cass	**$15.95**
Cat. No. 3848	Mod I & III, 32K, disk	**$19.95**

ATTACK FORCE

In this fast-paced machine language game, eight alien Ramships are warping toward your ship. You must dodge them and fire missiles to destroy them before they get you! Sound and Joystick options are included.

Cat. No. 3849	Mod I & III, 16K, cass	**$15.95**
Cat. No. 3850	Mod I & III, 32K, disk	**$19.95**

ROBOT ATTACK

Evil robots from the planet Jidya have overtaken one of Earth's valuable Space Stations. Space Central is counting on YOU to invade the station and conquer the robots. You must act quickly and boldly in order to carry out your mission. ROBOT ATTACK features sound effects and either keyboard or joystick control.

Cat. No. 3851	Mod I & III, 16K, Cass.	**$15.95**
Cat. No. 3852	Mod I & III, 32K, disk	**$19.95**

COSMIC FIGHTER

Draft those pesky aliens! Your mission is to clear the skies of the invading aliens. As soon as space is clear, along comes another set. Keep shooting but watch out, your fuel is getting low. Good luck. Includes sound and joystick capability.

Cat. No. 3213	Mod I & III, 16K, cass	**$15.95**
Cat. No. 3213	Mod I & III, 32K, disk	**$19.95**

METEOR MISSION 2

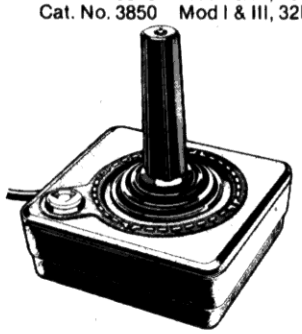
Emergency! Your astronauts are in trouble. You must maneuver through the asteroids and meteors in order to save your men and get them to the space station. Complete with sound and joystick option.

Cat. No. 3214	Mod I & III, 16K, cass	**$15.95**
Cat. No. 3215	Mod I & III, 32K, disk	**$19.95**

SUPER NOVA

SUPER NOVA is a fast paced real-time game for one or two players. The object is to destroy as many asteroids and aliens as possible without getting destroyed. Hitting a large asteroid causes it to break into smaller asteroids. Aliens and their flagship will appear on the screen and try to shoot you out of the sky.

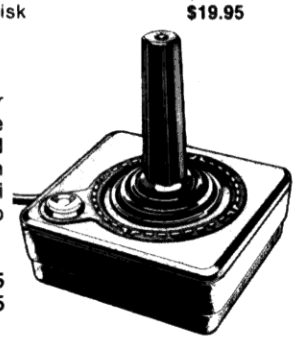
Cat. No. 3845	Mod I & III, 16K, cass	**$15.95**
Cat. No. 3846	Mod I & III, 32K, disk	**$19.95**



TRISSTICK

Model I and Model III owners may now use a joystick for better response in those fast-action programs. The TRISSTICK is compatible with all BIG FIVE software and comes complete with instructions for easy implementation into your own programs. It features the famous Atari joystick and plugs right in to your machine. There's no modification or wiring needed!

Cat. No. 3939	Model I	**$39.95**
Cat. No. 3929	Model III	**$39.95**



"EXTRA SPECIAL" SPECIAL

Purchase any three or more BIG FIVE programs and take 10% off. Combined with FREE SHIPPING offer, it's just too good to be true.

HOW TO ORDER

Write or phone. Pay by check, M/C, VISA, or COD (add $1.50 for COD). (800) 423-5387 (213) 886-9200. Offer expires July 31, 1982. Mention this ad and we pay shipping (UPS ground only).
HW Electronics 19511 Business Center Dr., Dept. U7, Northridge, CA 91324

WHEN IN SOUTHERN CALIFORNIA, VISIT OUR RETAIL STORES

HW ELECTRONICS

19511 Business Center Dr.
Northridge, CA 91324

2301 Artesia Blvd.
Redondo Beach, CA 90277

444 S. Indian
Palm Springs, CA 92262

I purchased my Model I in September 1978 and sold it in August 1981 when I purchased a 2-disk Model III. The Model III ROM is similar in many ways to the Model I ROM, but the PRINT routines differ mainly in sequence. I suspect that Radio Shack has made some changes in the Model I ROM also.

Thank you for bringing this incompatability problem to my attention. I apologize for an inconvenience my article may have caused you.

Kenneth R. Smith

I recently acquired a war games tape called "Jagdstaffel." I thought it would run on my Model III, 48K TRS-80, but it apparently is an old tape and was only intended to run on older Radio Shack computers.

I would like to write the authors, David A. Wesley and Ross W. Maker, but they neglected to write their address on the material accompanying the cassette. Have you ever heard of either of these authors or this program?

Roy A. Hoffman, Jr
54½ South Main Street
Janesville, WI 53545

We haven't, but if anyone else has, could you please contact Mr. Hoffman at the address above? —Ed.

A real weird thing happened to me which may be of interest to your readers and maybe you can explain it further.

When I decided to have my Color Computer upgraded from 4K to 32K with extended BASIC, I thought I would type and CSAVE some programs which included extended BASIC commands to save me time later. All went well until I tried to run them. I could load them into my upgraded computer and list them and everything looked good at this point, but they bombed out whenever the computer came to a line that contained an extended BASIC command. (*To get the programs to run...*) I would load the program and try to run it. If, for example, I got the error on line 50, I would then type EDIT 50 and all I had to do was hit the enter key, which would cause the computer to reprogram my line as I had written it. I think the reason is that even though the upgraded computer sees the line as I had typed it, it was unable to understand it. After editing the whole program it was a simple matter to save it again, only this time the computer would understand it...

James A. Nyman
Millinocket, ME

Your 4K BASIC probably didn't know how to make "tokens" out of extended BASIC commands, and the act of editing the lines did it for you. —Ed.

"They shouldn't have touched my Dosplus"

"I only left my keyboard for a few minutes ...when I returned, I found Stamitz from accounting and Miss Sashshay from the secretarial pool fondling my DOSPLUS 3.4. Now if I've told them once, I've told them a hundred times...use my coffee cup. Borrow my key to the employee lounge. Bend my paper clips but, leave my DOSPLUS alone!! Did they listen? Nooooo! Well, I guess I lost my head. Both Stamitz and Sashshay are doing fine. They should be released from the hospital any day now. For me, it's an entirely different story."

Signed,
0076697

Why DOSPLUS creates such fanatically touchy users is not so hard to understand. DOSPLUS 3.4 turns the TRS-80 into something altogether better.

DOSPLUS 3.4 lets your TRS-80 work a lot faster — 5-12 times faster with more accuracy, efficiency and dependability. DOSPLUS 3.4 also has a lot of features that users such as 0076697 find positively endearing. For instance, the ability to read 40 track disks in 80 track drives, and an easy to read operating guide that makes using DOSPLUS as easy as...well, bending a paper clip.

So to increase productivity, to increase the speed and accuracy of your TRS-80. DOSPLUS 3.4 Only $149.95! Still the best DOS for the cost!

P.S. "Get your own DOSPLUS"
0076697

DOSPLUS

DOSPLUS first in quality! First in the industry!

Call Toll Free 1-800-327-8724 ext. 207

MICRO SYSTEMS SOFTWARE, INC.
Specializing in the Tandy Line

5846 Funston Street, Hollywood, FL 33023
Outside of Florida phone toll free 1-800-327-8724 ext. 207
For Visa/MasterCard/C.O.D. Orders
Toll free lines will accept orders only!
For applications and technical information, call (305) 983-3390 or drop us a card. Dealers inquiries invited.

CALL TOLL FREE 800 424-2738

the PROGRAM STORE

STELLAR ESCORT

By Jeff Zinn from Big Five
This new arcade game from Big Five continue their tradition of bringing you the most exciting action in innovative space games. Your fighting spacecraft must run the gauntlet of the attacking alien's weaponry in order to accomplish your defense mission. You'll use all your skill and dexterity just to survive!

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

NEW TRISSTICK JOYSTICK FROM BIG FIVE! Can be used for all Big Five games plus programs you write. Utilizes the Atari joystick with special interface. Plugs right in!

Specify model I or III...$39.95

DEFENSE COMMAND

By Hogue & Konyu from Big Five
You are the lone defender of 10 Krotnium fuel cells essential for the survival of the planet. Aliens swoop down from above to steal the fuel; it's your job to destroy them. You can still save the cells after a raid, but you must shoot the alien and simultaneously move under the cell to catch it. If things look bad you can set off one of your 4 antimatter bombs and destroy all enemies on the screen! Arcade fun with action and sound.

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

MYSTERIOUS ADVENTURE

ARROW OF DEATH part 1

By Brian Howarth from Acorn
Proper English only! This British import is Acorn Software's first adventure and it meets their reputation for high quality. The vocabulary you use in this adventure must conform to proper English standards -- not the barbaric tongue spoken here in the colonies! You and Sorcerer Zarda must restore the kingdom from the engulfing sense of bitterness and ill feeling that forced the ruler to flee the palace.

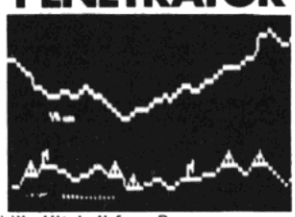
16K tape or disk...$19.95 Hint Sheet...$1.00

VOYAGER I

From Avalon Hill
A graphic science fiction game that puts you on board a spaceship infested with killer robots! Your job is to clear the four level, 144 location ship of the robots and arm it to self-destruct. The high speed graphics are represented in a 3-D perspective that represents your eye's view. Instant switching to floor plan maps aids your movement. The action is fast and furious!

16K tape for model I & III or
Color (Ext. Color Basic)...$19.95

PENETRATOR



By Philip Mitchell from Beam
Armed with missiles and bombs, you must fly your fighter to the enemy's cache of neutron bombs and destroy them. Your mission is in four stages, involving rugged terrain, caverns and manmade obstacles -- not to mention enemy radar, missiles and paratroopers. This new departure in arcade gaming allows you to set up your own terrain and enemy emplacements, then save them for future use. Make your mission as hard or easy as you like!

Tape or disk...$24.95

SPACE CASTLE

From Cornsoft
Ahead of you lies the menacing castle, floating in space amidst its layers of orbiting shields. At intervals, smart mines spin off the shields and head for your ship. Dodging the mines and destroying the shields isn't your only problem, though: once you penetrate the innermost shield, The evil Yugdab will unleash all his fury in an attack! A fast-paced and challenging arcade game, indeed.

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

Write Arcade Games in BASIC with...

ACCEL 3
BASIC COMPILER

By Southern Systems from Algorix
Ever wish your programs would run faster? Assembly language is one answer. But -- even after you spend the time needed to learn Z-80 code -- it is a long, tedious process to write in assembly language, and debugging is very difficult. Fortunately, there is another way -- compiled BASIC.

In our opinion ACCEL3 is the best compiler for the TRS-80. It's flexible: works with tape or disk, model I or III, and requires as little as 16K of memory. It's fast: only a few seconds to compile. It's a memory miser: only a 15-35% increase in program size in most cases.

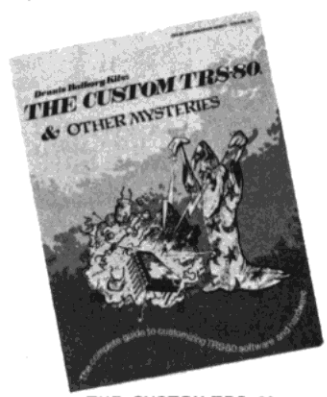
ACCEL3 will work under TRSDOS, NEWDOS or LDOS, and there are very few programming restrictions. To save ACCEL-compiled programs on tape you will need TSAVE (sold separately, $9.95)

Tape or disk...$99.95

Instant Search/Sort DATABASE

By G. Hatton from Acorn
An easy-to-use, yet powerful database management tool, ISS alleviates many of the complications usually found in setting up and maintaining information files. Written in machine language, ISS can provide a multitude of sorts, subsorts, searches and categorizations in seconds. Because the processing is done in memory, you can manipulate the data at will without risk to your database on tape or disk. The number of records is limited only by your file format and the amount of RAM in your system.

16K tape or 32K disk...$49.95



THE CUSTOM TRS-80

By Dennis Kitsz from IJG
The hardware hacker's delight! This book contains many, many useful modifications that could make your TRS-80 model I more reliable, easier to use, faster, and more capable! Clear, concise instructions, along with many useful construction tips, guide your way through the computer's innards. Some unique software ideas are included as well. WARNING: some of the hardware modifications in this book will void any warranties and may cause Radio Shack to refuse repairs on modified equipment.

335 pages...$29.95

Visit our other stores: Seven Corners Center · Falls Church, VA & W. Bell Plaza · 6600 Security Blvd. · Baltimore, MD

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

THE PROGRAM STORE
4200 Wisconsin Avenue NW, Dept. 8 206 Box 9609
Washington, D.C. 20016

MAIL ORDERS: Send check or M.O. for total purchase price, plus $1.00 postage & handling. DC, MD & VA: add sales tax. Charge cards: include all embossed information on the card.

 Prices Subject to Change

Get the most from your micro with software and accessories from one of the world's largest selections.

The Program Store

CALL TOLL FREE 800 424-2738

asylum

From Med Systems
Your eyes are bloodshot as you peer into your computer's screen and cry, "I must be CRAZY!" If this has never happened to you, you've never tried ASYLUM, the 3-D graphics adventure! ASYLUM places you on a cot in a small (padded?) room. Periodically the janitor lobs a hand-grenade through the window. What you do next could mean escape -- or disaster.
16K tape...$19.95 16K disk...$22.95
Advice Sheets...$1.00

Crush, Crumble and Chomp!

From Epyx
It's a monster movie, and you are the monster! You can be The Glob, Kraken, Mantra, Mechismo, Arachnis, or Goshilla -- or even design your own "custom" monster (disk version only). This hilarious action game is loaded with graphics and sound as you practice your villany.

16K tape or 32K disk...$29.95

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

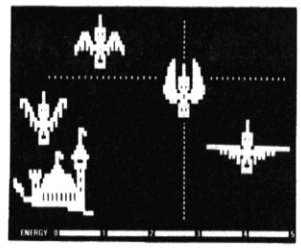
Model I & III, 16K tape...$24.95
Model I disk version.....$29.95

LUNAR LANDER

By Wall & Moncrief from Adventure Int.
You get a vast lunar landscape, graphically depicted in both long range and close up, with many choices for landing sites. Choose a more difficult site and get more points -- if you can land successfully. Great graphics and sound add to the realtime challenge and fun.

16K tape...$14.95

VOYAGE OF THE VALKYRIE

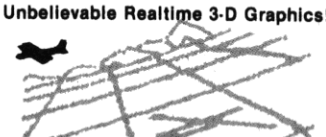


By Leo Christopherson from AOS
Combine the animation and music techniques pioneered by Christopherson with the challenge of his first fast-moving arcade game and you have VOYAGE OF THE VALKYRIE! You speed through a magical maze guarded by ferocious birds that swoop down to attack if you don't get them first. To list all the play and options of this exciting game would take the 16 pages of instruction included.
16K tape...$34.95
16K disk...$39.95

ALSO FROM CHRISTOPHERSON:
DUEL-N-DROIDS
Teach your "animated android" how to wield a laser sword in mortal yet comic battle. Entertainment for all ages.
16K protected tape...$14.95
16K protected disk...$20.95

By John Allen from Acorn
Once you load ASTROBALL into your TRS-80, the arrow keys become flipper buttons, the screen becomes the play board, and you become the "Pinball Wizard!" A flying saucer, spaceships, meteors, and black holes add to the fun as your ball realistically zings around the board. Five skill levels.
16K protected tape...$19.95
16K protected disk...$19.95

Unbelievable Realtime 3-D Graphics!



FLIGHT SIMULATION

From Sub-Logic
The wait is over! If 3-D graphics seem impossible on the low resolution TRS-80, you haven't seen this brilliant program. During FLIGHT SIMULATION, you instantly select instrument flight, radar, or a breathtaking pilot's-eye-view. But be sure to strap yourself in -- you're liable to get dizzy!

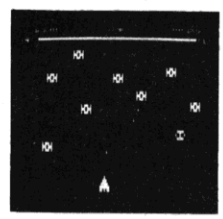
Once you put in some air time learning to fly your TRS-80, head for enemy territory and try to bomb the fuel depot while fighting off five enemy warplanes. Good Luck!

NOW FOR MODELS I & III! (specify which)
16K tape...$25.00
32K disk...$33.50

SCARF-MAN

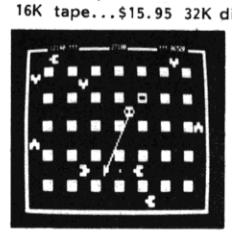
From Cornsoft Group
Race your Scarfman around a maze, gobbling up scoring dots. You are pursued by five monsters: if you eat a "+" they'll lower their eyes and you can eat them, otherwise they'll eat you! SCARFMAN may be played using the keyboard or Alpha Product's Joystick. WARNING: MAY BE HABIT-FORMING!

Tape...16.95
Disk (specify mod. I or III)...$20.95
Alpha Joystick:
mod. I: $39.95, mod. III: $49.95



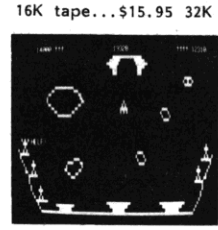
COSMIC FIGHTER

By Hogue & Konyu from Big Five
While fighting off the alien convoys -- each more skillful than the last -- you must keep track of your rocket fuel or risk explosion.
16K tape...$15.95 32K disk...$19.95



ATTACK FORCE!

By Hogue & Konyu from Big Five
Unlike the usual space "shoot-em-ups," Attack Force lets you control both speed and direction as you maneuver all over the screen in search of the alien Ramships and Flagships.
16K tape...$15.95 32K disk...$19.95



METEOR MISSION 2

By Hogue & Konyu from Big Five
Six astronauts are stranded on a desolate planet. You must undock from your command module and maneuver your rescue shuttle through the asteroid field to save them.
16K tape...$15.95 32K disk...$19.95

COMING SOON: MYSTERIOUS ADVENTURES
A new series of adventures from Acorn that is bound to keep you asking for more!

THE PROGRAM STORE · Dept8 206 Box 9606 · 4200 Wisconsin Ave, NW · Washington, D.C. 20016

Item _____ Price _____

_____ _____

_____ _____

Specify model: □ I □ III □ Color

Postage $1.00
Total _____

□ CHECK □ VISA
□ MASTERCARD MC Bank #____ Card # _____ Exp____

name _____
addr _____
city _____ state __ zip _____

Items at random

Is it really summer? Somehow it seems like we went directly from winter to summer without a spring. Maybe your part of the world worked better than ours.

Things around 80-U.S. are back on the upswing. You may note the additional sixteen pages in this issue. That's the first such increase since January '82, and I am happy to see it. There is nothing I like better than to make a lie out of the economic depression you constantly hear about on the news. We aren't exactly ready to declare stock splits and dividends, but it isn't that bad, and we are happy to see the faith and confidence shown by our readers and advertisers.

It's ridiculous to try and project my thoughts two months into the future, but by now (when you are reading this), we should have moved into our expanded offices and Cam Brown should be working full time as an editor. One of the neat things we gain with the expansion (aside from Cam, of course) is a complete darkroom and a new film processor. We are even putting a silver reclaimer on the processor so that we can salvage that precious stuff.

As of this writing (in May) we still haven't seen a Model 16. I understand there is no BASIC or operating system available for it yet, though it will work in assembly language and COBOL. We already have an evaluation of the DT-1 terminal and will be presenting it shortly.

We get all sorts of press releases to print in the New Products column. We are also asked many times what it takes to get a mention in that column. Here are some tips: Write your press release like an announcement. It should be anyway, since that is what press releases are for — to announce a new product or service. Many times we simply get advertising copy, which isn't the same thing. It makes it very hard to pick out the newsworthy portions, and with a stack of ten times more than we can use — well, you can guess which ones get in and which don't. A good glossy photograph will up your chances too. Good press releases are written in short, concise paragraphs. Editors *love* paragraphs! Make your statement in the first paragraph, then follow with other subordinate paragraphs explaining the product or service in more detail. This way, an editor can choose those paragraphs according to the space available without re-writing the whole release. The closing paragraph should always give the price and "where available" information. You have now completed a short course in writing press releases!

Corrections

Steve Cox, who wrote the article @Love in our April '82 issue, wrote the following: "Thank you for the excellent handling of my article @Love. I don't know how you do it but you seem to have 75% less errors than those who think they're number one. I also like the larger type used in programs. I wish to correct one statement I made in the article. The Data I/O program used with the Stringy Floppy does not set below BASIC as I stated. It first loads into the BASIC program area then relocates to the top of memory, but will not overlay any programs already up there. By loading the Data I/O program then running my program which appears in the article, it can be discovered where the data program resides. Just look at where memory is set."

In this issue

Our theme this issue is Law. Richard H. Stern starts things off with an interesting discussion on what should be done about software protection. William M. Amundson follows with a disussion of Westlaw, and John A. Records talks about a small computer in a small law office.

Following these three theme leads, we get into a video save and retrieval program by Tim Daneliuk. Just the thing for your business program, where you need to format a screen with blanks to be filled in.

Measuring the speed of your disk drives is taken up by Darrell A. Bymoen. He presents a Model I and III tachometer program.

Tim Daneliuk is back again with the second part of his article (which started in the Jun '82 issue) on LDOS for Beginners. This time he looks at file handling and the job control language.

Spencer Hall has put together a label maker for all occasions and called it "X-label." It uses the Microline 80 or MX-80 printer and produces a variety of useful labels.

Jerry Latham presents part two of his three-part series on building a checkbook program using NEWDOS80.

A tutorial on the storage and retrieval of data is presented by Terry Hazelett. Along the way, you build a mail list routine which may be useful.

A word processor for the Model III and Line Printer IV is the subject of an article by Leonard Zucker. Yes, it's a complete word processor and it works.

For the color computerists, there is an article by R. N. Roberts concerning the development of a library of frequently used routines which can be merged with others to quickly develop complex programs.

R. Shane Causer tells us how to squeeze the last bit of space out of any memory. No, it's not a hardware addition, but techniques in programming that do it.

There is more, too: on microcomputers for the deaf and blind, an expansion interface evaluation, and an evaluation of the Percom and LNW doublers. There's files and foibles and a disk-mapping program. Then there are about a half dozen reviews.

There should be more than enough to hold you through a rainy vacation. We'll be back in August with an educational issue and more for your TRS-80. In the meantime, tell them all you saw it in 80-U.S., and remember that nice days are made, not had.

Mike

QUALITY SOFTWARE FOR TRS-80 COLOR AND OSI

ADVENTURES!!!
For TRS-80 COLOR and OSI. These Adventures are written in BASIC, are full featured, fast action, full plotted adventures that take 30-50 hours to play. (Adventures are inter-active fantasies. It's like reading a book except that you are the main character as you give the computer commands like "Look in the Coffin" and "Light the torch.")
Adventures require 16K on COLOR-80 and TRS-80. They sell for $14.95 each.

ESCAPE FROM MARS
(by Rodger Olsen)
This ADVENTURE takes place on the RED PLANET. You'll have to explore a Martian city and deal with possibly hostile aliens to survive this one. A good first adventure.

PYRAMID (by Rodger Olsen)
This is our most challenging ADVENTURE. It is a treasure hunt in a pyramid full of problems. Exciting and tough!

TREK ADVENTURE (by Bob Retelle)
This one takes place aboard a familiar starship. The crew has left for good reasons – but they forgot to take you, and now you are in deep trouble.

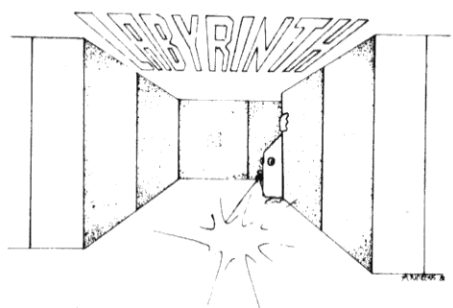
NEW!!
CIRCLE WORLD – We got Kzinti and puppeteers and problems. Our newest and biggest adventure. Requires 12k on OSI and 16K on TRS-80 Color.

NUCLEAR SUB – You are trapped in a nuclear sub at the bottom of the ocean. Escape and even survival is in doubt. Plotted by three of the most sadistic – I mean "creative" minds in adventure programming.

VENTURER!–A fast action all machine code Arcade game that feels like an adventure. Go berserk as you sneak past the DREADED HALL MONSTERS to gather treasure in room after room, killing the NASTIES as you go. Great color, high res graphics, sound and Joystick game for the TRS-80 Color or OSI machines. (black and white and silent on OSI.) Tape only. $19.95.

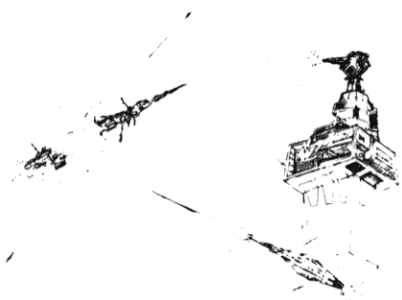
BASIC THAT ZOOOMMS!!
AT LAST AN AFFORDABLE COMPILER FOR OSI AND TRS-80 COLOR MACHINES!!! The compiler allows you to write your programs in easy BASIC and then automatically generates a machine code equivalent that runs 50 to 150 times faster.
It does have some limitations. It takes at least 8K of RAM to run the compiler and it does only support a subset of BASIC–about 20 commands including FOR, NEXT, END, GOSUB, GOTO, RETURN, END, PRINT, STOP, USR(X), PEEK, POKE, *, /, +, -, X, X, =, VARIABLE NAMES A-Z, A SUBSCRIPTED VARIABLE, and INTEGER NUMBERS FROM 0 - 64K.
TINY COMPILER is written in BASIC. It generates native, relocatable 6502 or 6809 code. It comes with a 20 page manual and can be modified or augmented by the user. $24.95 on tape or disk for OSI or TRS-80 Color.

LABYRINTH – 16K EXTENDED COLOR BASIC – With amazing 3D graphics, you fight your way through a maze facing real time monsters. The graphics are real enough to cause claustrophobia. The most realistic game that I have ever seen on either system. $14.95. (8K on OSI)



PROGRAMMERS!
SEE YOUR PROGRAM IN THIS SPACE!! Aardvark traditionally pays the highest commissions in the industry and gives programs the widest possible coverage. Quality is the keyword. If your program is good and you want it presented by the best, send it to Aardvark.

QUEST – A NEW IDEA IN ADVENTURE GAMES! Different from all the others. Quest is played on a computer generated map of Alesia. Your job is to gather men and supplies by combat, bargaining, exploration of ruins and temples and outright banditry. When your force is strong enough, you attack the Citadel of Moorlock in a life or death battle to the finish. Playable in 2 to 5 hours, this one is different every time. 16K COLOR-80 OR TRS-80. ONLY $14.95.



SPACE ZAPPER – Protect your central Star Base from ships that attack from all four sides. Fast reflexes are required as the action speeds up. Great for kids or Dads. This game has high speed high resolution graphics and looks as if it just stepped out of the arcades. – 16K extended or 32K disk. BASIC TRS-80. Color only. $14.95.

Please specify system on all orders

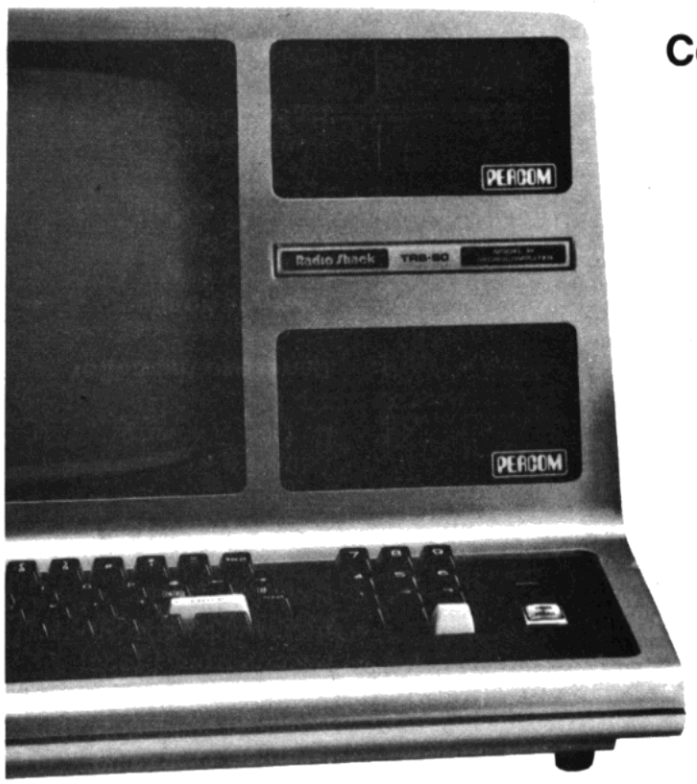
This is only a partial listing of what we have to offer. We have arcade and thinking games, utilities and business programs for the OS1 and TRS-80 Color. We add new programs every week. Send $1.00 for our complete catalog.

AARDVARK - 80
TRS 80 COLOR **2352 S. Commerce, Walled Lake, MI 48088** **OSI**
(313) 669-3110

TRS-80* Computer Owners...

GREATER VALUES

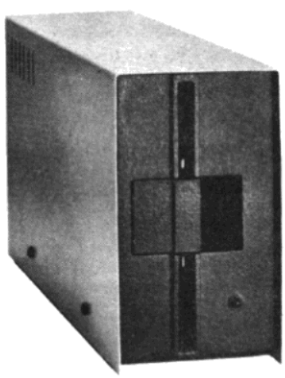
New Low Prices
for Percom Model III Disk Drives!



Complete First-Drive Systems from $449

Internally mounted drives that feature double- or single-density storage, comprehensive pre-delivery testing, 48-hour burn-in and, best of all, Percom's widely acclaimed disk controller. A First-Drive System includes the four-drive disk controller, one or two drives, power supplies, cables and a fully illustrated owners manual. Use Model III TRSDOS*, Percom OS-80+ and all other Model III compatible DOSs.

Note: If you prefer not to make the First-Drive System installation, we'll do it for only $29.95 plus shipping. This includes a 48-hour operating burn-in of your Model III computer, installation of the drive system and final checkout of the expanded system.



Our Own Fully Tested Drives from only $275

- Every Access Floppy Disk drive is electrically tested, mechanically tested and burned-in 48 hours under operating conditions. Look for the signed test list in the shipping carton.
- AFD drives are capable of either single- or double-density operation. Store up to 364 Kbytes (formatted) in double density. Store twice the data on a single diskette using AFD "flippy" drives."
- 15-day free trial offer...comprehensive 90-day limited warranty...unbeatable prices...AFD-100 @ $275.00, -100F @ $329.00, -200 @ $429.95, -200F @ $449.95.

How to order – Order by calling Access Unlimited toll-free on **1-800-527-3475**. Or order by mail. Orders may be charged to a VISA or Master Card account, or paid by cashier's check, certified check or money order. Sorry, we cannot accept personal checks. We pay freight (surface only) and insurance charges for shipments in U.S. on orders over $1,000.00. Add approximate insurance and shipping charges for orders under $1,000.00. If in doubt about these charges, ask when you call in your order. Texas residents include 5% sales tax. Minimum mail order $20.00. Allow 2 to 4 weeks for delivery.

Texas **(214) 340-5366**

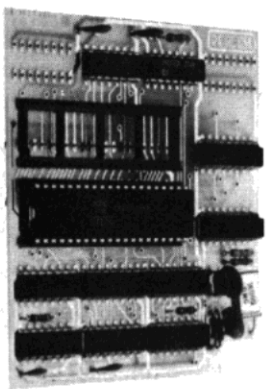
Save! VISA and Master Card charges are not deposited until the day your order is shipped.

PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE.

Everybody's Favorites – for Less!

Percom Data Separator	$23.95
Green Screen I, II, or III	$18.95
Head Cleaning Kit	$24.95
Drive Numbering Tabs (pkg 0-3)	$4.50
5¼" Flip-N-File	$24.95
8" Flip-N-File	$39.95

— LOWER PRICES



Percom's Proven DOUBLER II†
Upgrade Your Model I to Double Density for only $159.95

And not with some unproven hacker's kit, either. But with an assembled and fully tested *Percom DOUBLER-II†*. The DOUBLER II† simply plugs into the disk drive controller socket of your Expansion Interface. Then it lets you store over four times more on one side of a diskette, depending on the drive, than you can store on standard Tandy Model I drives. Other features: • Reads, writes and formats *either* single- or double-density • Includes DBLDOS™, a TRSDOS* compatible disk-operating system • Runs TRSDOS*, NEWDOS††, Percom OS-80† and other single-density software immediately. Change to double-density operation when convenient • Use Move 1-2-3 utility ($29.95) or other appropriate software to read Model III programs on your Model I. • Includes on-card high-performance data separation and write precompensation circuitry.

Clear-Out Sale on Microline-80 Printers!

only **$379**

Low Cost System Furniture

Modular design permits custom arrangements • Furniture styling and quality • TRS-80* colors • OK for UPS shipping • Reassemble without tools

System desks from $115
Printer stands from $144

Arrick Products Quick-Switch‡

Just flick a switch to connect your TRS-80* printer port from device #1 to device #2. Or, "Quick-Switch" a peripheral between computers. Versions for RS-232 or Centronics interfacing. Includes box-mounted 34-pin card-edge plugs. From $89.95. Cables from $29.95.

‡ trademark of Arrick Products Company.

Save! Make ACCESS your one-call shopping center for TRS-80* hardware and software. Call our toll-free order number for free product literature and latest prices. **Ask for our new, free software catalog.**

Toll-Free Order No.

1-800-527-3475◇

(orders & literature only)

ACCESS UNLIMITED

401 N. Central Expressway #600
Richardson, Texas 75080
(214) 340-5366

™Trademark of Access Unlimited
†Trademark of Percom Data Company, Inc.

*Trademark of Tandy Radio Shack Corporation
††Trademark of Apparat Corporation

What should be done about software protection?

An attorney's opinion

Recent Case Comments and articles in this Review have indicated the general state of disarray of intellectual property law as to software protection. In the United States, some courts think that object code[1] can be protected by copyright,[2] some think that it cannot,[3] and some tribunals do not want to decide the question if they can help it.[4] There is virtual unanimity, in the United States at least, that algorithms[5] are 'ideas,' and as such they are unprotectable.[6] The same principle necessarily applies to the concepts embodied in the programs. Protection of software by copyright is, at the very best, problematical. More likely, it is ineffective, except for close copies of source programs and documentation. Patent protection of software is at best limited, outside the field of special purpose computer-operated machine systems.[7]

That effectively leaves proprietors of software only contract and breach of confidence remedies, along with other torts related to breach of confidence, such as tortious interference with contractual or business relations, the trade secret type of unfair competition, and perhaps conversion or misappropriation of property rights. These theories and remedies are also of limited value. For example, third parties are usually insulated from liability to the software proprietor unless it is shown that they knew of and were responsible for the allegedly wrongful appropriation of the software. Breach of confidence and trade secret related remedies are probably not available in the case of mass-marketed software packages. Finally, it is almost certain that trade secret related remedies, and highly probable that contract remedies, are not available to protect mass-marketed object code (for example disks and ROMs) or products embodying them.[8]

The difficulty with contract remedies is closely related to the lack of patent, copyright, or other intellectual property remedies. Software contractual arrangements are usually not enforced against third parties who acquire the software, because the courts do not recognise a statutory or common law 'property right' in software. Unauthorised appropriation of the subject matter of a patent or copyright is deemed 'misappropriation' of a property right, and the courts therefore accord proprietors of patents and copyrights a considerable range of familiar legal remedies — damages, accounting for profits, injunction and so on. Unauthorised appropriation of software is likely to be held 'misappropriation,' and thus a ground for according those remedies, only when legislation is enacted imposing statutory rights and duties corresponding to those of patent and copyright law. To be sure, there is an obvious circularity in reasoning between whether X is a property right and whether the courts will protect against 'infringement' of X. But the ineluctable is that without a legislative declaration that software is intellectual 'property,' the courts will not readily be persuaded to grant remedies to the plaintiff who alleges that he is the proprietor of X.

So far, I have refrained from normative statements. I have said what I think the law is, and also what I would expect from it, rationally interpreted. I have not said what the law should be. I think it clearly would be desirable to protect software better than current law, as summarised above, does. But the arguments are business and economic ones. The points I think it better to make in the pages of this Review concern the legal policy issues, not the business/economic ones, attending software legislation. The premises of this Opinion, however, are that business conditions in the software industry justify and require software protective legislation, and that general economic policies would be furthered by enacting such legislation. If you will assume with me that some sort of software legislation is needed, it would be proper to move on to the main issues that I believe need more discussion at this time, and whose discussion I use this Opinion to stimulate: what kind of software legislation we should have.

At the outset, there is the question whether special software legislation is needed at all, rather than simply an amendment to the patent or copyright laws. The answer to that question of course depends on the structure of national patent and copyright laws. The general structure of United States patent and copyright law is unsuitable to protecting algorithms, for instance, and so too, probably, is that of United Kingdom law.[9] It would be disharmonious with, or even do violence to, the rest of the patent and copyright system to protect algorithms under either law, or to protect under them the concepts or ideas embodied in programs. Object code could probably be protected by explicit amendment of the copyright

law to do so, although any such amendment of the patent law would cause disharmony and tension. In any case, however, there would be serious questions whether the remedies and other procedures of patent or copyright law are appropriate for software. In the past, proponents of software protection in the United States have believed that it was easier, and thus tactically better, to try to persuade Congress to amend the copyright law than it was to persuade it to consider a whole software law. This was, I believe, a grave tactical error, for it resulted in poorly thought-out and ineffective legislation, for example, the 1980 computer software amendment to the United States copyright law, which is of little use to the software industry. The back door route to legal protection of software is short-sighted, disingenuous, and a mistake. A specific law directed to software would be far better. Do the same considerations apply to the United Kingdom and other jurisdictions?

The basic problems that should be faced in considering general legislation governing legal rights and obligations as to software are:

— First, how do we more effectively protect software, while recognising the competing interests of those groups who use software or benefit from its use?

— Second, how do we better deal with all of the different types of acts of appropriation to which the different types of software are susceptible?

Answering these questions requires, initially, definition of the groups with an interest in software and their respective interests.[10] That in turn will lead to a definition of the acts that it is appropriate to prohibit.[11] Then we must consider which possible remedies are appropriate to the different prohibited acts.

The effect that a software protection system has on the interests it touches must depend on a number of legislative 'variables.' By that term I mean factors on which the law could do one of several things, and different types of things or conduct to which the law could be made to apply. The difference between a good and bad system, whether in the United States, the United Kingdom, or anywhere else, is likely to turn on whether the legislative variables are carefully related to one another differently in different factual contexts or are instead handled in terms of gross generalities.

The interrelationship of remedies and prohibited acts is particularly important; yet it has received negligible attention in previous writings on proposed software laws. Many plans propose adoption of copyright law *in toto*, and do not even consider which copyright remedies should apply to which conduct or types of software. Other proposed software statutes simply list a broad range of remedies, and in the most general terms direct that they be applied 'as appropriate to the circumstances of the case.' This is too gross.

There are different types of what we may call 'software infringement,' and they call for quite different remedies. When a mass marketer of software appropriates and competitively markets a competitor's software package, a permanent injunction against that defendant would seem proper. But when the unknowing purchaser of a machine tool or consumer product embodying unlawfully taken object code is the defendant, such an injunction would seem harsh and inappropriate. Punitive damages may well be proper against one who unloads a protected ROM and markets it, but not against a programmer or his employer who reasonably believe, although incorrectly, that a program the programmer writes is not within the scope of protection to which the plaintiff's algorithm is legally entitled. We therefore need to develop a matrix of remedies and wrongs. That is part of the discussion that should precede writing any software law.

If we had such a discussion, and I hope to stimulate one in the pages of this Review, these are some of the questions that deserve, indeed need, ventilation: what should be deemed the infringement of an algorithm? One answer, an unsatisfactory one, is suggested by the patent/copyright doctrine against protection of 'ideas.' Ideas, it is said, are the currency and basic tools of scientific progress, and such progress will be hampered without a rule of free access to ideas.[12] When applied to machines and books, that is a good principle; if applied elsewhere, it may not be. Legal protection of algorithms will hamper scientific progress only if the amount of protection is so great as to have that effect. If a lesser amount of protection can reward and encourage development of algorithms useful to industry and society, the net effect of protection will be beneficial to progress. The question is one of degree and practicability, not an absolute. Nonetheless, it is clear that the principle is valid insofar as it commands that no known or 'obvious' algorithm should be protected. Rather, there should be an 'inventive step' for an algorithm to be protectable. The same principle applies to the concept of a program.

In protecting algorithms and the concepts of programs, it is as important to determine how far to protect them in granting relief as it is to determine how far to protect them in terms of defining the scope of infringement. Indeed, the two variables interact, for they jointly determine how burdensome protection would be and how much of an incentive the protection will afford software creators and proprietors. A question that must also be considered is whether it is so important to technological progress that good new algorithms be widely and rapidly adopted that injunctions and severe relief should almost never be allowed against the unauthorised appropriation of an algorithm or concept. I am inclined to believe that anyone should be allowed to use an algorithm or concept, upon payment of a reasonable royalty, particularly when he enhances it or adapts it to a new use. But this is a matter that deserves the attention and benefit of many views before proper conclusions can be drawn.

Should there be a 'copying' requirement for liability? I believe that absolute, no-fault liability is proper, rather than copyright law's requirement of actual copying. Intellectual property law protection of software is intended to reward

and encourage software creation, and to facilitate its marketing, not primarily to punish copyists. It would make no sense, and conflict with the goals just stated, to make the existence of the software proprietor's rights depend on whether the defendant deliberately copied or else independently invented the algorithm or concept.

Another important question is whether to protect programs[13] that do not, or whose algorithms do not, display inventive steps. A long, debugged, tried program may be of substantial commercial value and costly to perfect, even though all of its parts and concepts are known or obvious. It deserves protection, but not to the same extent as a program based on a new and unobvious algorithm. The scope of infringement is probably the right variable to modify here, rather than scope of relief. But this is a matter of opinion and calls for discussion. My proposal would be to use a standard of infringement, here, close to that of present copyright law, 'substantial similarity,' rather than the broader kind of protection accorded a patent along with the equivalents of what is precisely claimed.

Defining a concept of infringement for object code is important. The principal economic rationale of appropriating object code is the cheapness of doing so as compared with developing, debugging, and then compiling an independent program.[14] This economic rationale is likely to lead to a total duplication, for otherwise some of the economy is lost. In most cases, therefore, there should be only minimal problems in defining how close a 'copy' of object code must be for it to be an infringement. But the question must be asked, first, whether object code should be protected *per se*, and whether the same rationale applies to a game or machine tool ROM or disk as to an applications program for data processing or operating systems software. To determine whether object code has been unlawfully appropriated, when the plaintiff's claim rests solely on the object code, a test of 'substantial identity' would seem more appropriate than copyright law's test of 'substantial similarity.' A patent law infringement standard would clearly be inappropriate to protect object code *per se*.

That should not necessarily be the end of protection of object code. If a program is protectable in source code form,[15] whether because it is tried, debugged, and the product of considerable effort or because it has been derived from a legally protected algorithm, it should also be protectable when it is compiled into object code. I would suggest that software legislation should protect object code in these circumstances, if it is practicable to do so. It must be recognised, however, that severe administrative difficulties could attend trying to determine whether particular object code is compiled from a protected source program, or is derived from a protected program concept or algorithm.

Remedies in the case of object code are particularly troublesome. 'Infringement' of object code can be a concept of wide spectrum. At one extreme is the wretched copyist caught after he has completely 'unloaded' and 'reloaded' his competitor's ROM, disk, or tape into his own commercial product. At the other extreme is the innocent, unknowing consumer or commercial end user. My tentative proposal would be to let off all consumer end users completely, lest there be a chilling effect on the general consumer marketing of software and end products embodying it; to subject commercial end users to reasonable royalty liability, at most; and of course to throw the injunctive and punitive damages book at the wretched copyist-unloader.

A final policy question that must be considered, and on which I would hope to stimulate discussion, is whether the rights, duties, and remedies of the software law should supersede the application of other laws. If there is a comprehensive regulatory system for protecting software, the operation of other laws could tamper with and disturb any carefully struck balances. Uniformity is a further consideration, one very important in the United States, because contract law and trade secret law vary widely from state to state. If national uniformity is important to the software industry, and I believe that it is, a rule of supersession and preemption is essential to bring about uniformity of legal treatment of software. Patent and copyright law are, of course, already uniform in the United States, because they are federal. In their case, and in an already uniform system of law, such as that of the United Kingdom, the question of supersession of other law is more difficult. This is a highly debatable issue, and I simply raise the question as one deserving discussion.

I have hoped to stimulate discussion more than to provide answers to the questions I have raised. The issue of software legislation is of constantly increasing importance, not only because of the growing economic importance of software, but because of the increase in transnational commerce in it. An extensive discussion of the foregoing issues would be an invaluable preliminary step towards proposing rational legislation in the field. ■

Richard H. Stern is a partner in the Washington, D.C., office of a national law firm, Baker & Hostetler. He specializes in litigation and counseling practice involving patents, copyrights, trade secrets, and problems in commercial exploitation of high technology and software. Mr. Stern was formerly chief of the Intellectual Property Section and Patent Section of the U.S. Justice Department's Antitrust Division. He has a B.S.E.E. from Columbia University and a law degree from the Yale Law School, and he was Supreme Court Justice Byron White's first law clerk.

[1] By 'object code' I mean the machine-readable non-human-intelligible form in which a computer program is coded for use by a computer. Conceptually that is a string of 1s and 0s. Physically it is a set of high and low voltage levels in an electronic device, a set of magnetised and unmagnetised domains on a disk or tape, a set of closed and open circuits imprinted on a silicon chip, a set of holes and non-holes on a piece of paper or cardboard, and so on.

[2] See *Tandy Corp. v Personal Micro Computers, Inc.* 524 F. Supp. 171 (N.D. Cal. 1981)

[3] See *Data Cash Systems Inc. v JS&A Group Inc.* 480 F. Supp 1063 (N.D. Ill. 1979), aff'd on other grounds, 628 F.2d 1038)7th Cir. 1980); (1980) EIPR 25. Accord, *Data General Antitrust Litigation* 490 F. Supp 1089, 1113 (N.D. Cal. 1980).

[4] *Certain Coin Operated Audio-Visual Games and Components Thereof* (1981) 10 EIPR 301.

[5] By an 'algorithm' I mean the series of steps embodied in the program, which when they are taken solve the problem or accomplish the task in question. A simple algorithm for computing the length of a hypotenuse, for example, can be developed from the Pythagorean theorem.

[6] See *Synercom Technology Inc. v University Computing Co.* 462 F. Supp 1003 (N.D. Tex 1978). See generally *Gottschalk v Benson* 409 US 63 (1972); *Baker v Selden* 101 US 99 (1879); *L.B. Plastics Ltd v Swish Products Ltd* (1979) FSR 145 (HL).

[7] See *Diamond v Diehr* 101 S. Ct 1048 (1981).

[8] The general rule is that, once a product is placed on the market, anyone is free to 'reverse engineer' it. Trade secret remedies evaporate when secrecy is gone. Contractual reservations by notice or the like are generally ineffective against the public or remote purchasers. See generally *Coke on Littleton*, Section 360.

[9] See Note 6 above.

[10] Proprietors of software, for example, have different interests from users. Indeed, those interests may lie at opposite ends of a spectrum that takes in programmers, software marketers, computer hardware manufacturers, data processors and so on. There is also, of course, the pervasive interest of the public in proper maintenance of a system that encourages technological progress, the spread of knowledge, industrial efficiency and free competition.

[11] The interests affected by using the concept of a program, for example, differ from those affected by duplicating a ROM, and the proper thing, if anything, to do about the one may not be proper for the other.

[12] See *Benson*, Note 6 above.

[13] We are now probably considering programs in their 'source code' form. This means a program in a human-intelligible form, written in some computer programming language such as BASIC, COBOL, FORTRAN or the like, and perhaps also more-or-less intelligible assembly code.

[14] A secondary rationale might be that it is easier to get access to someone's object code than his source code.

[15] See Note 13 above.

100+ Reasons

DOS FEATURES

- A 252+ page technical manual with index and detailed table of contents.
- Commands **SYSTEM** and **PDRIVE** allow the user to configure/customize his/her own DOS
- Depending on installed hardware, **NEWDOS/80,** via the **PDRIVE** command, supports within one system, mixtures of single/double density, single/double head, 5 or 8 inch drives with single/double volume diskettes of up to 7680 standard 256 byte sectors. 80 track drives can read 35/40 track diskettes. Parameters for 10 drives may be pre-specified though only a maximum of 4 are in use at anytime. ADR not provided.
- LNW 5/8 and Omikron mapper boards supported.
- APPARAT, AEROCOMP, AM, LNW, and PERCOM, disk doublers supported.
- Most CPU speed up mods may be used though not specifically supported.
- Model I/III data diskette interchangeability when both computers are operating under **NEWDOS/80** version 2.0.
- Model I 2.3 TRSDOS compatible.
- **COPY** to/from Model I 2.3B and Model III TRSDOS diskettes (no other useage allowed).
- Single drive **COPY** and Copy By File features.
- Depending on installed hardware, the system volume may be single/double density, single/double sided, 5 or 8 inch and up to 7680 sectors.
- **RUN-ONLY** program mode restricts the operator to program defined input only.
- **MINI-DOS** allows the executing program to be interrupted by the operator to perform one or more of the 51 DOS commands executable under MINI-DOS, and then continue the interrupted program's execution.
- **CHAIN** or **DO** commands activate chaining whereby keyboard input comes from the specified disk file, allowing a pre-determined set of commands and/or parameters to be automatically inputted.
- Dump display to printer function.
- Enhanced **DEBUG** facility (14 commands) allows interrupting current program execution, inspecting/altering memory or disk, and resuming execution, continuous or single step, with/without stops.
- DOS vectors defined for Assembly Language programmers.
- DOS-CALL allows user programs and BASIC to execute DOS commands.
- The programmer may create his own resident DOS commands.
- Programs may enable/disable user routines driven off the timer interrupt.
- The programmer may create his/her own resident DOS commands.
- Model I built-in lower case driver, blinking cursor, auto key repeat.
- **ROUTE**ing of keyboard, display, printer and (Model III only) RS232C. May be routed to a user routine in memory, but not to/from disk unless via a user routine.
- Except for the spooler, there are no high memory routines for DOS or BASIC; this includes **ROUTE** and **CHAIN** functions.
- Lower case DOS commands honored.
- Full error messages displayed instead of error codes.
- 31 enhanced **COPY** parameters.
- Copy By File allows 6 criteria for file selection.
- 15 enhanced **FORMAT** parameters.
- Partial diskette re**FORMAT** permitted.
- File **PURGE** by wildcard extents and/or user files.
- **DIR**ectory command allows wildcard extents, user files, short or extended format, dump to printer.
- User may specify diskette's directory location.
- Expanded directory provides for up to 222 file entries.
- Some DOS commands may be aborted without reset.
- **R** command repeats last performed DOS command.
- **CREATE** command to pre-allocate a disk file.
- **ERROR** command displays error message associated with error code.
- **HIMEM** command sets/displays DOS/BASIC high memory address.
- **DATE** command sets/displays computer's date.
- **TIME** command sets/displays computer's time.
- Model III **FORMS** command for printer control.
- Model III **SETCOM** command for RS232 control.
- Enhanced **LIST/PRINT** commands for ASCII files with pause, abort and partial file listing.
- Alter chaining state via the **CHNON** command or commands within the chain file.
- A program or a chaining sequence may display a message with/without pause.
- **CLEAR** command to zero memory and to purge routes, user DOS commands and user timer routines.
- Commands to enable/disable BREAK key, blinking cursor, lower case driver
- **PROT** command to change diskette **NAME/DATE/PASSWORD**
- **ATTRIB** command to change a file's attributes.
- **CLS** command to clear screen.
- **AUTO** specifies the command to execute automatically at reset/power-on.
- **SYSTEM** specifies the default system configuration values (usually enable or disable) which become effective on RESET/POWER UP.

  Diskette/file password checking
  RUN-ONLY mode
  Keyboard debounce (Model I)
  Screen dump to printer (JKL)
  DEBUG 123 entry
  MINI-DOS
  Break key as keyboard key
  Hardware lower case (Model I)
  Assign default drive number for **DIR**
  Assign default drive number for file creation
  Memory protect value
  Clear key as keyboard key
  Disk master password required for full diskette or **CBF COPY**
  Auto Repeat key function
  TIME/DATE question on power-up
  TIME/DATE question on reset
  Display disabled until operator/program reenables
  Manual operator chaining pause/abort
  Manual operator **AUTO** command override
  R = repeat last DOS command performed
  Built-in lower case driver (Model I)
  Lower to upper case toggle
  Blinking cursor
  Number of physical drives on computer
  Number of disk I/O retries
  Time delay for 1st repeat of auto repeat key
  Specify the cursor character
  Specify the write of the directory sector's address mark for Model I single density diskette in Model III format for easy Model I, Model III diskette exchangeability

Why NEWDOS/80™ VER. 2.0 Is the Best DOS for your TRS-80 Model I or III Computer

DISK BASIC FEATURES

- In one statement from DOS READY, BASIC can be brought up, the number of files set, the memory size set and a program LOADed or RUN.
- **RUN-ONLY** prevents the operator from getting to READY or DOS READY, thus giving the program almost total control.
- Via the CMD function, all DOS commands are executable from BASIC, either directly or dynamically.
- **MINI-DOS** is available from BASIC.
- **DEBUG** is available from BASIC.
- **CHAIN**ing is available from BASIC.
- Variable passing between programs via the **V** parameter of RUN.
- Abbreviated commands:
  (A)uto; **(D)**elete; **(E)**dit or comma; **(L)**ist or period
- Accidental text line deletion more difficult
- Text line scrolling foreward or backward.
- Text page scrolling foreward or backward.
- **DI** moves text line to new position
- **DU** duplicates text line to new position
- Built-in **RENUM**ber with line number and limited syntax error check. A portion of text may be moved to another part of the program with all references to that code resolved.
- Built-in **REF**erence function will display/print references to all line numbers, integers and variables. It will display references to a single line number, integer, string, function code (reserved word) or a group of packed or unpacked characters, and then allows displaying of each referencing text line in turn with editing as necessary.
- A program may be loaded into reserved high memory via CMD or MINI-DOS and its execution address extracted from the two bytes at 17411 (4403H).
- **MERGE** functions with packed or ASCII text files.
- Built-in text space eliminator and/or remark deleter.
- Built-in calendar date conversion.
- Dynamic **ERASE** of selected variables, keeping all others.
- Dynamic **KEEP** selected variables, **CLEAR**ing all others.
- After clearing an array via **ERASE** or **KEEP**, the array may be redimensioned via **DIM.**
- Dynamic text line deletion.
- Dynamic text line insertion via **MERGE**, which with dynamic **DELETE**, allows use of overlays.
- **SWAP** contents of 2 variables of the same type.
- Single Stepping starting at specified text line number.
- In-memory sort of up to 9 arrays in either ascending or descending order
- **RENEW** function to reinstate **NEW**ed program.
- Full BASIC error messages, including associate DOS error message, if applicable
- With default start up parameters and no reserved high memory, 48K RAM has 38261 bytes available.
- SUPERZAP, DIRCHECK and other programs using only memory from 5200H to 6FFFH can be executed directly from BASIC without disturbing the program text or variables (if 8K BASIC free memory available, exclusive of string area).

FEATURES of NEWDOS/80 enchanced BASIC disk file I/O.

- In addition to TRSDOS sequential and random file types, **NEWDOS/80** has two new file types (Marked Item and Fixed Item) divided into five subtypes **(MF, MU, MI, FF** and **FI)**
- These five subtypes do not require **LSET, RSET, MKI$, MKS$, MKD$, CVI, CVS** or **CVD;** instead, **GETs** and **PUTs** are done directly to/from the variables named in a list.
- The string separating character sequence ; " , " ; used with PRINT is not used with the new file types; instead only a comma is used as the separator.
- **MU** files are used as an option to the older PRINT/INPUT files.
- **FF** files are used as an option to the older RANDOM files.
- Record lengths up to 4095 bytes supported.
- Records may be all of the same length (**MF** and **FF**), of varying lengths (**MU**) or unknown length (**MI** and **FI**).
- Sequential files may be accessed randomly.
- Files may be accessed by **R**elative **B**yte **A**ddress to allow accessing of variable length or unkown length records.
- Existing files may be extended.

UTILITY PROGRAMS INCLUDED WITH NEWDOS/80

- **SUPERZAP** is a disk/memory display and modification program, also used as the vehicle for installing patches to **NEWDOS/80.**

- **DISASSEM** is a Z80 load module (CMD) disassembler that builds cross reference tables for all location references including those by JR instructions, includes in the disassembly printable characters for all hex bytes to help locate character strings and sends the disassembly to the display, printer or a disk file. The disk file can then be edited and/or assembled using EDTASM, if it is not too large.

- **DIRCHECK** is a program that displays directory contents and checks directory integrity (its primary function), displaying specific error codes to assist user attempts at directory trouble shooting and/or repair. Optionally will zero out unused (dead) file names.

- **EDTASM** is Apparat's enchancement of Radio Shack's 1978 tape editor/assembler program to operate from disk and with disk files. Requires purchase of that Radio Shack program (not a newer one) as a pre-condition of use of Apparat's EDTASM.

- **LMOFFSET** allows load module (CMD) transfer between disk and tape. Displays program start, end and entry addresses. Optionally allows load address relocation (not execution relocation) and subsequent execution as from non-disk BASIC via SYSTEM.

- **CHAINBLD** is a mini-text editor for creating/maintaining chaining files.

- **NEWDOS/80** manual chapter titles and page counts

  1. Introduction (5)
  2. DOS Library Commands (52)
  3. DOS Routines (12)
  4. DOS Features (14)
  5. DOS Modules, Data Structures, and Miscellaneous Information (12)
  6. Additional Programs Supplied on **NEWDOS/80** Diskette (22)
  7. Disk BASIC, non-I/O Enhancements (17)
  8. Disk BASIC I/O Enhancements and Differences (21)
  9. Error Codes and Messages (2)
  10. Glossary (9)
  11. Error Reporting, Incompatibility Handling, and Patching (8)
  12. Conversion Information and Miscellaneous Comments (9)
  13. ZAPs (increasing with time)
  14. Appendix A: Marked and Fixed Item File discussion (47)
  15. Appendix B: Marked and Fixed Item File examples (18)
  16. Index (4)

- Full time support staff
- Information, minor enhancements and corrections to **NEWDOS/80** are issued, at no charge, to registered owners only.

NEWDOS/80 Version 2.0 for the model I is a separate and distinct product from the model III. Each is sold separately.

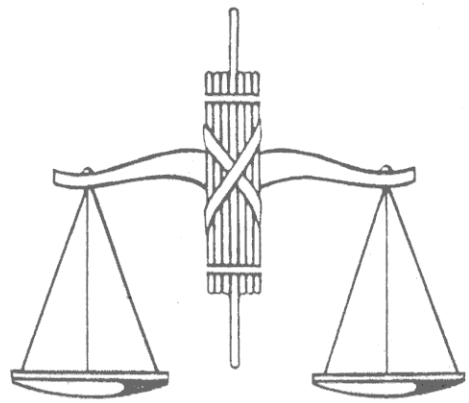
$149.00

"On-going Support for Microcomputers"

VISA master charge

*REGISTERED TRADEMARKS OF APPARAT, INC.

Westlaw



Computer-assisted legal research

William M. Amundson, St. Paul, MN

The TRS-80 Model II is more than a versatile microcomputer to help lawyers manage their law offices. Its compatibility with the WESTLAW Computer System allows the lawyer to perform sophisticated legal research as well. But what, exactly, is WESTLAW . . . and in what ways does it help the lawyer in research?

Not many years ago, computer-assisted legal research seemed an unlikely notion: Law was an art rather than a science; as such, it could not be reduced to numbers and programs. Even if it could be, the results would be poor and prohibitively expensive.

Today, computer-assisted legal research systems exist and are successful. One such system, WESTLAW, was created by West Publishing Company. West applied its 100 years of service to the legal profession to design a system tailored to the needs of lawyers and judges.

The two basic advantages of computer-assisted research are: (1) speed in finding the law and (2) search capabilities not possible with traditional law book research. The first advantage is much needed; about 40,000 new cases are reported each year, making the task of finding cases relevant to a specific problem difficult for the lawyer. WESTLAW gives the lawyer the ability to search thousands of cases and retrieve relevant ones in a matter of seconds.

The second advantage — unique search capabilities — is of equal importance. For example, WESTLAW allows the lawyer to search for anything that may be contained in the text of a court decision — names of judges, witnesses, products, companies, medical terms, and unusual nonlegal terms. Such terms are not normally indexed in law books, which are organized around legal concepts rather than specific details from cases.

The WESTLAW system is an online, realtime system involving geographically dispersed remote terminals linked to a central computer. The central computer in the WESTLAW system is a powerful IBM 3033 located at West headquarters in St. Paul, Minnesota. Visual display terminals located in law firms, courts, government agencies, and other organizations are linked to the computer by normal phone lines. Data is transmitted over the lines at the rate of 1,200 bits per second.

A great deal of analysis preceded the design of the WESTLAW data bases. Currently, the data bases provide access to the full text and headnotes of all opinions of the United States Supreme Court from 1932 to date, the opinions of the United Courts of Appeals from 1950 to date, and the United States District Courts from 1945 to date. (A headnote is a brief explanatory note prefacing a legal case.) The full text and headnotes of all reported opinions of the state courts are available from at least 1967 to date in all states, and in some states coverage is afforded for earlier years. Both state and federal cases are constantly added to the data base as new decisions are reported. In addition to its case-law data base, WESTLAW also offers a specialized federal tax data base and the complete text of the U.S. code.

The TRS-80 Model II is easily adapted for use with WESTLAW

with Radio Shack's special software package, plus a telephone and a "modem" adapter. Paper copies of cases and other legal documents are easily made with Radio Shack's optional printer.

Searches usually use words or phrases that describe a legal issue or fact. Suppose, for example, a researcher wants to find cases involving the issue of whether a doctor must advise a patient about the side effects of a treatment. The researcher might type in certain relevant key words:

DOCTOR PHYSICIAN SURGEON /P INFORM* ADVIS* /P PATIENT /P TREATMENT* OPERATION* SURGERY SURGICAL PROCEDURE*

Synonyms for "doctor," "treatment" and "advise" are given, since a court may use any of several equivalent terms. The symbol * is a root expansion command that will retrieve plurals and other forms of a word. The special connecting symbol /P is used to require that terms will be in the same paragraph and not spread through several pages in a case.

When the ENTER key is pressed, the computer will search for, and retrieve, all cases containing the selected terms. The retrieved cases are ranked by the computer on the basis of the frequency of occurrence of the search terms in the documents. The retrieved documents are then displayed in sequence of apparent relevance.

Another important element of WESTLAW is of special importance to the user. Before actually searching the data base, a lawyer can enter a client or file identification code to identify the client for whom the search is being performed. The total time of the search is recorded in the WESTLAW accounting system and is sent to the user at the end of the month. In this way, the user has an accurate timekeeping system for the purpose of billing clients for WESTLAW searches.

WESTLAW has changed legal research by allowing lawyers to do more research in a shorter time as well as giving them some new capabilities. ■

Birth of a legend.

Epson.

A whole new generation of Epson MX printers has just arrived. And while they share the family traits that made Epson famous — like unequalled reliability and ultra-fine printing — they've got a lot more of what it takes to be a legend.

For instance, they've got a few extra type styles. Sixty-six, to be exact, including italics, a handy subscript and superscript for scientific notation, and enough international symbols to print most Western languages.

What's more, on the new-generation MX-80, MX-80 F/T and MX-100, you get GRAFTRAX-Plus dot addressable graphics. Standard. So now you can have precision to rival plotters in a reliable Epson printer. Not to mention true backspace, software printer reset, and programmable form length, horizontal tab and right margin.

All in all, they've got the features that make them destined for stardom. But the best part is that beneath this software bonanza beats the

Uh...three legends.

heart of an Epson. So you still get a bidirectional, logical seeking, disposable print head, crisp, clean, correspondence quality printing, and the kind of reliability that has made Epson the best-selling printers in the world.

All of which should come as no surprise, especially when you look at the family tree. After all, Epson *invented* digital printers almost seventeen years ago for the 1964 Tokyo Olympics. We were the first to make printers as reliable as the family stereo. And we introduced the computer world to correspondence quality printing and disposable print heads. And now we've given birth to the finest printers for small computers on the market.

What's next? Wait and see. We're already expecting.

EPSON
EPSON AMERICA, INC.
3415 Kashiwa Street • Torrance, CA 90505 • (213) 539-9140

FEATURE	ORIGINAL MX-80	GRAFTRAX-80*	ORIGINAL MX-100	MX-80 with GRAFTRAX-Plus	MX-80 F/T with GRAFTRAX-Plus	MX-100 with GRAFTRAX-Plus
Bidirectional printing	X	X	X	X	X	X
Logical seeking function	X	X	X	X	X	X
Disposable print head	X	X	X	X	X	X
Speed: 80 CPS	X	X	X	X	X	X
Matrix: 9 x 9	X	X	X	X	X	X
Selectable paper feed			X		X	X
PAPER HANDLING FUNCTIONS						
Line spacing to n/216		X		X	X	X
Programmable form length	X	X	X	X	X	X
Programmable horizontal tabs	X	X	X	X	X	X
Skip over perforation			X	X	X	X
PRINT MODES AND CHARACTER FONTS						
96 ASCII characters	X	X	X	X	X	X
Italics character font		X		X	X	X
Special international symbols				X	X	X
Normal, Emphasized, Double-Strike and Double/Emphasized print modes	X	X	X	X	X	X
Subscript/Superscript print mode				X	X	X
Underline mode				X	X	X
10 CPI	X	X	X	X	X	X
5 CPI	X	X	X	X	X	X
17.16 CPI	X	X	X	X	X	X
8.58 CPI	X	X	X	X	X	X
DOT GRAPHICS MODE						
Line drawing graphics				X	X	X
Bit image 60 D.P.I.		X	X	X	X	X
Bit image 120 D.P.I.		X	X	X	X	X
CONTROL FUNCTIONS						
Software printer reset		X		X	X	X
Adjustable right margin			X	X	X	X
True back space		X		X	X	X
INTERFACES						
Standard — Centronics-style 8-bit parallel	X	X	X	X	X	X
Optional — RS-232C current loop w/2K buffer	X	X	X	X	X	X
RS-232C x-on/x-off w/2K buffer	X	X	X	X	X	X
IEEE-488	X	X	X	X	X	X

*Tandy TRS-80 block graphics only available with GRAFTRAX 80.

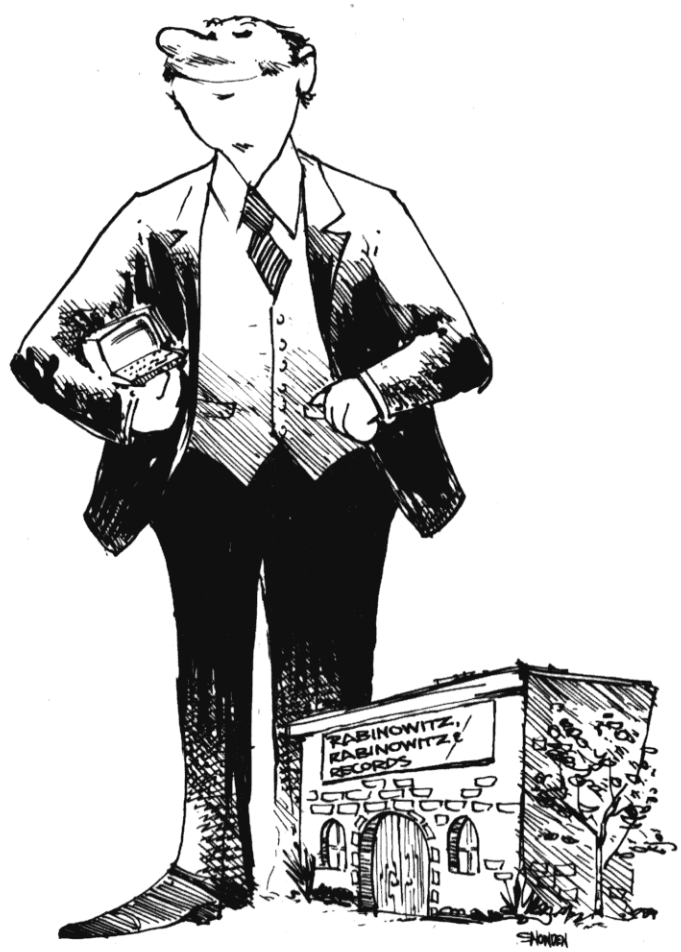
A small computer in a small law office

Model I for efficiency and low overhead

John A. Records, La Grande, OR

My Model I TRS-80 computer by Radio Shack is the heart of my law office: it enables me to run a small, efficient office with a minimum of overhead.

A few words about my background are in order to help better evaluate my comments. I have more interest than ability in electronics, but am fascinated with gadgets. I had one introductory computer programming course in college, and the rest of my computer knowledge has been acquired from reading and experimentation. Although the overall tone of this article is favorable to Radio Shack, I have no financial or other connection to that company.



HARDWARE

My system has the following hardware: a Model I TRS-80 computer with Level 2 BASIC, 16,000 units of memory, and a lower case modification; an expansion interface with 32,000 units of additional memory, a "double density" board and an RS-232 communications board; a Modem I; a video display; two Radio Shack disk drives; a cassette tape recorder; and a Daisy Wheel Printer Model 2. My problems with this hardware have been few, and have been quickly remedied by Radio Shack.

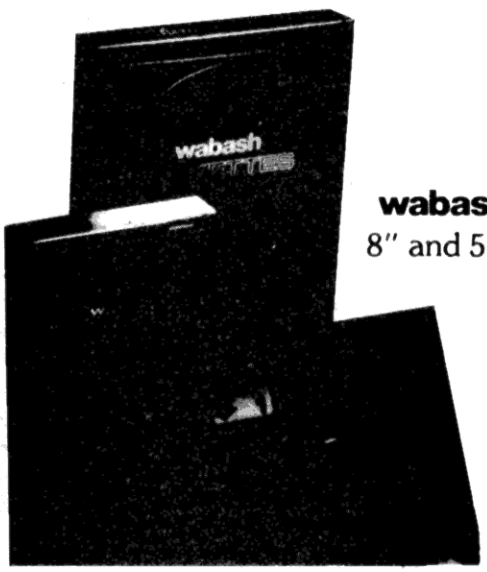
SOFTWARE AND APPLICATIONS

Disk Operating Systems — A disk operating system (DOS) is a kind of master program for the computer: it is the context in which other programs operate, and serves functions which are beyond the scope of this article. The Radio Shack DOS (TRSDOS) has appeared in several versions, and the latest one seems to do the job adequately, if slowly and without frills.

DOSPLUS 3.3D, available for about $100 from Microsystems Software, Inc., 5846 Funston Street, Hollywood, Florida 33023, is now the guts of the software in my computer system.

DOSPLUS has many advantages over TRSDOS. It enables double disk capacity by using double density 40-track operation, as does the Model 3 TRS-80, provided that a doubler board and disk drives rated

Announcing the best Error Free Personal Computer Diskette Money can Buy. For Less.



wabash
8" and 5¼"

- **Error Free**
- 1 year warranty
- Hub ring installed
- Write/Protect notch
- Next day delivery

$19.90/box of 10

- **No** minimum order quantity

If you are a member of a user group or a school district please call for special terms on future offers. **TSS** is the largest specialty supplier of magnetic media in the Midwest. We have the products that you want when you need them. Please take advantage of this introductory offer and call us **now.**

TSS Transaction Storage Systems, Inc.
MAGNETIC MEDIA SPECIALISTS

22255 Greenfield Road
Southfield, MI 48075

CALL TOLL FREE
1-800-521-5700
1-800-482-4770 (Michigan)
313-557-3036 (Detroit)
312-922-0076 (Chicago)
614-221-1788 (Columbus)
513-621-1518 (Cincinnati)
Telex 810-224-4646

EXPECT A MIRACLE

15

YES, **TSS** *is the magnetic media supplier that I have always wanted.*

Please send me _____ (Quantity) , _____ (Description) , @_____ (Price) *ea.*

*I am interested. Please send me more information or call me at( )*_____

For faster order entry call any of our toll free or local numbers

*Name*_____ *Title*_____

Company _____

*Address*_____

City _____ *State* _____ *Zip*_____

Purchase order number _____

Amex/Master Card/Visa orders are accepted *Expiration Date*

_____ _____ _____ *month* ___ *year* ___

for double density are also used. (The disk drives now sold by Radio Shack are so rated, although the earlier ones were not.) DOSPLUS also works much more quickly than does TRSDOS. Information can be pumped in and out of the computer at least twice as quickly.

DOSPLUS is also very "friendly." It is possible to construct a computer software file with DOSPLUS which will take control of the computer from the time it's turned on, and lead a very inexperienced operator through the requisite steps for any given operation. DOSPLUS also includes a utility called COPY1, which helps in the use of a two-disk drive system with only one disk drive in a pinch.

A "spooler" program is also provided with DOSPLUS. The spooler can greatly speedup some applications which require use of a printer. The spooler sends information to the printer in bursts, so that the computer need not wait for the much slower printer to finish a line before the computer can proceed with its work.

Word Processing — Word processing is the predominant use for my computer system. SCRIPSIT is a fine word processing program sold by Radio Shack for about $100. I use it almost every day, and am generally pleased with it. An essential addition to SCRIPSIT is called Super Script, by Acorn Software. Super Script adds several capabilites to SCRIPSIT, all of which are extremely useful to a lawyer: these include underlining and superscripting for footnotes.

A relatively minor limitation of SCRIPSIT is that the length of a document is effectively limited by the size of the computer's memory. On my system, which has maximum memory, the longest document that will fit into the computer is about 20 pages, double-spaced. I don't very often go over this length, and there are tricks to evade the limitation when the time comes.

My word processing work breaks down into boilerplate, custom documents and letters. This is not an exhaustive list, and there is some overlap between boilerplate and custom documents.

Boilerplate is very important to lawyers. My brothers in the legal fraternity may not like me for saying this, but we all use form documents quite a bit. It would hardly pay to reinvent the wheel (or the rack) each time a client needs something. Forms enable an attorney to keep already high legal prices within an acceptable range. However, they must be used intelligently, and a word processor is wonderful for this. It is also possible (and often desirable) to merge some boilerplate into an otherwise custom created document. This would be analogous to buying some off-the-rack slacks to go with your tailored blazer.

Some documents are custom created in their entirety for just one application. A good example of such a document would be an appellate brief, which is made for a particular case and for that case only. An appellate brief might go through several drafts, and word processing makes it much easier to incorporate changes from draft to draft. New cases may arise, or stylistic changes may be necessary. Another example of a custom document is this article.

I also crank out a lot of letters with my computer. I can create ten letters and treat them as just one document. I can enter the date once and have it appear correctly on all ten letters, and there is no need to retype the address for the envelope. When the letters are ready to print, I can have them printed out page after page, followed by their respective envelopes, or with the envelopes appropriately interspersed with the letters. And since the printout is error-free, I am comfortable using carbon copies instead of more expensive photocopies.

Financial Analysis — I find financial analysis to be a necessary adjunct to my practice of law, and much appreciated by my clients. The computer makes it easier to consider and understand the tax implications which appear in almost every commercial transaction. Software is available for preparation of federal and state income tax returns. I have not used this software myself.

Computer-generated mortgage amortization tables, which set forth the amount of principal and interest in each payment, are a nice extra touch in real estate transactions. Other computer programs are useful for comparing investments, estimating damages and evaluating investment real estate.

Visicalc, an electronic spreadsheet, is a very popular program, and many have said that it, alone, would justify the purchase of a microcomputer system. I use Visicalc for budget analysis, estimation of human life values, and for analysis of projected retirement income.

All of this financial analysis, or most of it, could be done on a $50 financial calculator, but it would take much longer, and the printout would not be in as usable a form. I feel that use of computers by lawyers for financial analysis will be more and more prevalent.

Defense of Drunk Driving — Another application of my computer system is blood alcohol analysis. Intoxitron, written by me and available through the Lawtech Company, Box 1523, La Grande, Oregon, estimates blood alcohol content based on such factors as weight, number of drinks imbibed, and time since the first drink. This is a useful check on the blood alcohol content as measured by the police, as well as a means of eliciting the whole truth from clients with regard to drinks taken. See Figure 1.

Corporate Power Allocation, Inc. — Also available through the Lawtech Company, and written by me, is used for the analysis of cumulative voting in corporations. I have found it useful in explaining the concept of cumulative voting to clients, and in demonstrating why a 49% interest in a small corporation is of questionable value, even if a seat on the board of directors is assured.

Billing — I also use my computer system to bill my clients. I haven't yet found a good program for this purpose at a price which I deem fair, so I have written my own, rather simple program.

SERVICE

Computers don't always work. They sometimes break and sometimes develop inexplicable glitches that disappear on their own. This happens to even the most

costly computers. It will happen to your computer, if you get one. A quick fix is essential in a business application. I have found Radio Shack's service to be more than adequate.

It seems to me that with the present state of the art, you don't have to be a computer whiz to use a system like mine, but you are much better off if you know something about it. This will save you time and money, and you'll enjoy it a lot more. I routinely encounter minor problems which I am able to evade because of my broad reading.

If you have any reservations about your ability or desire to deal with such matters, perhaps you should acquire a system from one of the big companies like IBM or Xerox. It'll probably cost you a lot more, but the handholding you receive may well be worth it. Note, however, that Radio Shack has recently begun to support its products more strongly with toll-free inquiry numbers, service contracts for equipment, and stores dedicated solely to computer equipment.

If you live in a large city, there is a good chance that there is a Radio Shack Computer Center near you. These centers have experts available. If you are not near a center, the personal quality and ability of your local Radio Shack personnel should play a very important part in your decision on the purchase of a Radio Shack computer.

If you buy from a Radio Shack company store, you will be charged the list price, unless an item is on sale. The same is true if you buy from a Radio Shack Computer Center. However, if you buy from an authorized dealer, you can often get a discount.

I am delighted with my Radio Shack computer system. I use it daily, and it serves me well.

Figure 1
BLOOD ALCOHOL ANALYSIS

INTOXITRON
PERCENT BLOOD ALCOHOL:
190 POUND MALE
HOURS SINCE FIRST DRINK

DRINKS	1	2	3	4	5	6
2	.02	.00	.00	.00	.00	.00
3	.03	.02	.00	.00	.00	.00
4	.05	.04	.02	.01	.00	.00
5	.07	.05	.04	.02	.01	.00
6	.08	.07	.05	.04	.02	.01
7	.10	.09	.07	.06	.04	.03
8	.12	.10	.09	.07	.06	.04
9	.13	.12	.10	.09	.07	.06
10	.15	.13	.12	.10	.09	.07
11	.17	.15	.14	.12	.11	.09

'Drinks' are one ounce, 86 proof, or 12 ounces of 3.5% beer.

Greater than '.10' is presumptively under the influence in some states.

Calculations are based on the modified Widmark formula. Because of individual variations in metabolism, the actual percentage may be higher or lower than the calculation. ■

We've grown on you

With Acorn Programs, Your TRS–80 Does Things You Wouldn't Think Possible

Our sophisticated programs make sight and sound seem simultaneous—you're never disappointed when you select a program from us. ***And,*** our customer service is unparalleled in excellence.

Acorn Has a Program For Everyone

Whether you want games, adventure programs, word processing enhancement programs, teaching programs, utility programs that help serious programmers improve their programming and get more out of their computers . . . Acorn has them all.

NEW!

Your Mysterious Adventures Begin With Arrow of Death, Part I.

By Brian Howarth

The first in our brand new, exciting Mysterious Adventure Series with all the popular features of TRS-80 Adventures ***plus*** a new, challenging mental approach. In Arrow of Death, Part I, a mysterious blight has fallen on your land. It seems to come from the Golden Baton, once good but now a source of evil incarnate. Rain has destroyed the crops, friends are at each other's throats, despair and hatred fill the hearts of the people. The King implores you to aid his sorcerer, Zardra, in banishing this evil from the land. You agree, and the adventure begins. You'll face constant danger, possibly death —but you'll also have the opportunity to perform heroic tasks.

All Mysterious Adventures have a "save the game" feature. Arrow of Death, Part I is complete in itself . . . but we think you'll find it so challenging, you'll want to continue with Part II. Other Mysterious Adventures will include **Time Machine** and **Golden Baton**—all written to give you many hours of gaming pleasure.

16K Tape
or 32K Disk.
TRS-80 I & III **$19.95**
plus $2.00 shipping and handling

Everest Explorer

By William Godwin & Don Knowlton

Will you be one of the daring few to make it to the pinnacle of Mount Everest? Or will you succumb to the lack of oxygen, the unexpected violence of the storms, the incredibly rough terrain? You, as the leader of the Everest expedition, will have to choose the route, choose the timing, make sure your climbers are well-rested, set up a chain of camps and, if you reach the summit, get your followers back down to base camp. You'll have to manage money, climbers, Sherpas, tents, oxygen, food and fuel. Danger lurks at every step—can you get to the summit and return? "Save the game" feature on disk.

16K Tape
or 32K Disk.
TRS-80 I, III, **$19.95**
Color Computer and Atari
plus $2.00 shipping and handling

Lost Colony

By David Feitelberg

You are the Economic Manager of the world's first space colony. The next support ship from Earth isn't due for another 15 years and you have instructions either to make things go better or get out of office in shame. You must assure the survival of this struggling space colony—it's all up to you. You'll be presented with the human, natural and industrial resources of the planet. You must allocate labor, explore new territories, decide on production quotas, determine pay scales and taxes for the most productivity —you're armed with maps and charts. 10 levels of difficulty. "Save the game" feature on disk.

16K Tape
or 32K Disk.
TRS-80 I & III **$29.95**
and Atari
plus $2.00 shipping and handling

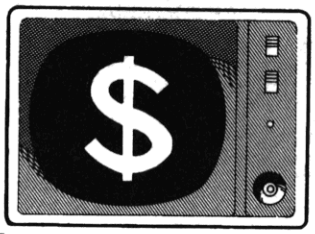
Because Our Programs Deliver What They Promise.

Instant Sort/Search Database

By Gordon Hatton

An easy-to-use fast information manager for home, personal or small business use. Even a novice can make full use of this program and it's extremely "user-friendly" in helping minimize operator errors. ISS comes in 3 parts: (1) lets you create, examine and save your custom-formatted data base; (2) sorts and searches the data base so you can pick out and print just the information you need; (3) provides powerful editing commands so you can modify and update your existing data files. Also includes "DEMO", a 500 record demonstration data file that lets the first time user familiarize himself with the command set. ISS handles both alphabetic and numerical data in a variety of combinations.

16K Tape
or 32K Disk.
TRS-80 I & III **$49.95**
plus $2.00 shipping and handling



Money Manager

By Andrew Bartorillo

In today's economy, we all need to monitor our expenses more closely. **Money Manager** will help you keep track of your income and expenses and give you an easy way to manage your budget. You define the categories according to your needs, including tax-deductible expenses . . . a great help at tax time. You can also reconcile your checkbook with the bank's balance. Full lineprinter capability allows you to print items by each category.

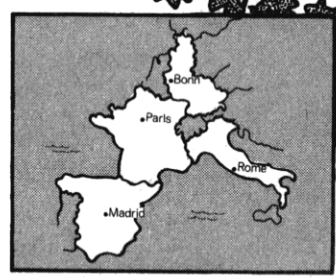
32K (Minimum) Disk.
TRS-80 I & III
and Atari **$39.95**
plus $2.00 shipping and handling

Your Family Tree

By Chuck Acree

Whether you're a professional genealogist or simply interested in your own ancestors, YOUR FAMILY TREE lets you avoid the limitations of paper-based charts. This program sets up your data base quickly and easily—holds name, date and place of birth, marriage and death information for each ancestor, plus a comment line. You can get a complete "pedigree" from any selected family member; or a 3 generation ancestral chart displayed or printed showing the number of known ancestors beyond each branch of the tree. The program will also display an outline map of the U.S. showing migration across the country. Use full or partial information to get full search capabilities by any key field.

16K Tape
or 32K Disk.
TRS-80 I & III **$29.95**
plus $2.00 shipping and handling



Language Teacher

By Cindy & Andrew Bartorillo

Teachers, students, and those who just want to learn a new language—Acorn's Language Teacher gives you your choice of French, Italian, German or Spanish. Use these programs at home or in the classroom. Students like this new way of learning and/or reviewing a previously learned language. Teachers can free themselves from tedious preparation of vocabulary lists and writing quizzes.
Each Language Teacher offers hundreds of word combinations, verb conjugations and phrases. Choose the drill topic and foreign language-to-English or vice versa. Option for multiple-choice answers plus retest on missed items. Gives running percentage of correct answers.
Available: French, Italian, German I & II, Spanish I & II.

32K Disk.
TRS-80 I & III **$29.95**
plus $2.00 shipping and handling

Acorn
Software Products, Inc.

634 North Carolina Avenue, S.E. Washington, D.C. 20003

We've Grown On You.

And, our selection of programs for your TRS-80 I & III expands with a new program monthly.

Shop Acorn by mail! Mail this coupon today for your *FREE* Acorn catalog.

Mail to: ACORN SOFTWARE PRODUCTS
634 North Carolina Avenue, S.E.
Washington, D.C. 20003

NAME_____

ADDRESS_____

CITY_____ STATE____ ZIP_____

B

VIDSAVE /ASM

Create a formatted screen, save it, dump it to video

Model I adaptable to Model III with TRSDOS 2.3 or LDOS

In writing applications programs, it is often desirable to format the screen display in a particular maneuver. Unfortunately, subsequent INPUT statements or data displays may destroy the desired format of the screen. This means that a routine must be written to recreate the desired video. With interpreted languages such as TRS-80 BASIC, this can be a tediously slow process if the screen format is even mildly complicated. VIDSAVE is a machine language program which solves this problem. It has applications in business programs, games and even "real-time" animations.

Note that this program is copyrighted. You may make copies of it for personal use, but the program may not be sold or be used as part of a package which is to be sold.

Features

VIDSAVE allows a screen to be saved in memory or a previously saved screen to be dumped back to the video display. Additionally, the user may specify at which position the cursor is to reappear when the screen save/dump is finished. This allows, for example, a standard business form to be created on screen and the cursor positioned so that each item may be filled out. Because this program is in machine code, it executes with tremendous speed: new screen displays seem to appear "instantly," and the cursor "jumps" from one location to another. This program is completely self-relocating and honors the high memory pointer in the Model I at X'4049-4A'. When the program loads, it moves this pointer down by

Program Listing for VIDSAVE

```
0000            00100          TITLE <VIDSAVE V1.1>
                00110 ; VIDSAVE/ASM - 11/15/81
                00120 ;
                00130 ; By T.A. Daneliuk
                00140 ; Copyright (c) 1981
                00150 ; T&R Communications Associates
                00160 ; Chicago,  IL  60625
                00170 ;
                00180 ;
                00190 ; **********
                00200 ; * This program is used to save or retrieve the entirety
                00210 ; * of video memory.  It is intended to be called via the
                00220 ; * USR command in BASIC.  If a valid variable is passed
                00230 ; * through the USR routine the following is done:
                00240 ; *      VARIABLE                  RESULT
                00250 ; *         0              Save present screen / cursor
                00260 ; *                        Return to present cursor pos'n
                00270 ; *       5000             Dump memorized screen / cursor
                00280 ; *                        Return to memorized cursor pos'n
                00290 ; *      1 to 1024         Save present screen / cursor
                00300 ; *     (offset + 1)       Pos'n cursor @ 3C00H + offset
                00310 ; *    -1 to -1024         Dump memorized screen
                00320 ; *    -(offset + 1)       Pos'n cursor @ 3C00H + offset
                00330 ; * Note that 1 must be added to the actual offset
                00340 ; * desired - i.e. to position the cursor at 3C05H, the
                00350 ; * value 6 must be passed.  Passing the value 1 places
                00360 ; * the cursor at 3C00H.  This was done so that there
                00370 ; * would be unique (non-ambiguous) cursor positioning
```

SAVE / on Software and Hardware for TRS-80*

CHEXTEXT®

Let your TRS-80® do the proofreading on your SCRIPSIT® text files!!

Features of this program include:

- Complete dictionary maintenance including the addition and deletion of words.
- Menu driven for ease of operation.
- Spelling Checker
- FREE expanded dictionaries available, depending on your drive storage capabilities.

NEW LOWER PRICE $59.95

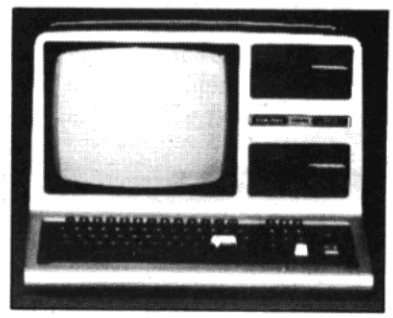
MODEL III PRICE LIST

(All 48K with TANDON drives)

1 single sided 40 track drive	$1695.00
2 single sided 40 track drives	1895.00
1 dual sided 40 track drive	1820.00
2 dual sided 40 track drives	2145.00
2 single sided 80 track drives	2175.00
2 dual sided 80 track drives	2395.00
with RS-232 C Add	115.00

The above computers include NEWDOS 80 Version 2.0.

NOTE: These Model III computers contain Apparat installed disk drives and memory. They are warranted by Apparat, Inc. for 90 days.



MISCELLANEOUS SUPPLIES

DISKETTES

Double density-soft sectored-replacement guaranteed. Spindle/Hub protected. (5¼" only)

Verbatim Datalife 5¼" 40 track	$24.95
Apparat's No Name 5¼" 40 track	$19.95
Verbatim Datalife 8" model II	$39.95

PAPER

9½"x11" blank white, tractor feed paper, full box 15# or 20#	$24.95
14½"x11" green bar, tractor feed paper, full box	$34.95
3½"x15/16" tractor feed mailing labels	$19.95

OTHER

5¼" plastic library case	$ 1.95
8" plastic library case	$ 4.95
5¼" Flip-sort	$18.95
8" Flip-sort	$31.95
16K memory kits	$19.95

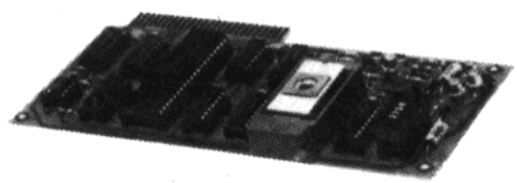
APPARAT'S PROM BLASTER

An eprom programmer for all 25 X X and 27 X X chips. TRS-80*

MOD I & III	**$149.00**
CABLE	**$ 17.95**

BUS EXTENDER

mini version with 2 card slots and no enclosure ..... $ 69.95



SPECIAL - FREE GRAFTRAX

with **EPSON** MX-80 Printers

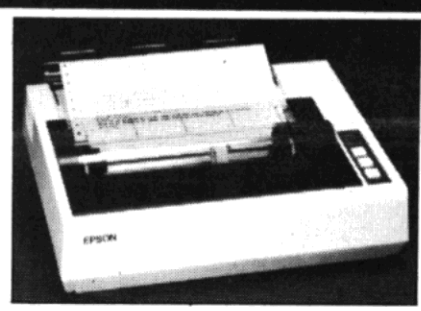
MX-80 **$499.00**

MX-80F/T **$575.00**
with Graftrax, add $79.95

MX-100 **$775.00**

Printer Cables **$24.00**
(Specify Computer Type)

WE ALSO STOCK OKIDATA & NEC PRINTERS CALL FOR PRICES



ASSORTED ITEMS OF INTEREST

MICRO CLINIC, Mod I $24.95, Mod III $29.95
The ultimate in memory & disk diagnostics

MEAL MASTER, Mod I & III Disk $24.95
meal planning & grocery shopping aid

FLEXTEXT/80 (requires Graftrax) **$34.95**
ultilize the additional features of the MX-80 printers under model I & III scripsit

MICRO ACCOUNTING SYSTEM, $479.00
Interactive G/L, A/R, A/P & checkbook manager

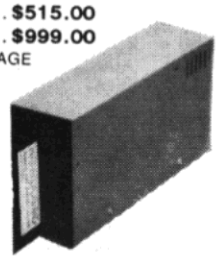
UNI-TERM/80, $89.00
Universal terminal program that takes advantage of the extended NEWDOS/80 commands

BASIC BETTER & FASTER BOOK, $23.95

NEW LOWER PRICES ON TANDON DISK DRIVES

Complete with power supply, chassis & configured for TRS-80® Model I or III

- Single sided 40 track **$285.00**
- Dual 40 or Single 80 track **$415.00**
- Dual sided 80 track **$515.00**
- **Special** - 2 Dual sided 80's **$999.00**
  (ADDS ALMOST 1.5 MEGABYTES OF STORAGE TO A D. D. MOD I OR A MOD III)



MODEL I DOUBLE DENSITY PACKAGE

Converts a standard TRS-80 Model I to Double Density.
Apparat Doubler & NEWDOS/80 V 2.0.
$278.00 value for only **$219.00**
Doubler alone **$129.00**

NEWDOS 80 Version 2.0

The most sophisticated DOS ever produced for the TRS-80® Models I and III. It provides the user with "MAINFRAME" power on a "MICRO".

Some Features available are:

- Jobstream Control Language
- Mod I/Mod III Diskette interchangability
- Double Density Support on Model I
- Pagenation of BASIC listings on the screen
- Basic program single stepping
- Dynamic variable manipulation
- Multiple array sorts with BASIC CMD
- Complete technical support provided

All this plus much more for only

$149.00

InfoWorld
Software Report Card

NEWDOS/80 Version 2.0

	Poor	Fair	Good	Excellent
Performance	☐	☐	☐	☑
Documentation	☐	☐	☐	☑
Ease of Use	☐	☐	☑	☐
Error Handling	☐	☐	☐	☑

VISA

Apparat, Inc.

"On-going Support for Microcomputers"

4401 S. Tamarac Pkwy. • Denver, CO 80237 • (303) 741-1778 • (800) 525-7674

Scripsit & TRS-80 are a registered trademark of Tandy Corporation

Freight F.O.B. Denver call for shipping charges. Foreign Orders shipped Air Freight

its own length and then moves itself up into high memory. VIDSAVE uses no ROM calls, but does use several DOS calls. It has been tested under TRSDOS 2.3 and LDOS on a Model I, but should run on the Model III with minor modification.

Loading and Entering VIDSAVE

VIDSAVE locates itself dynamically in memory when it is first loaded depending on what the high memory pointer value is at the time of loading. For this reason, the actual address at which VIDSAVE begins varies. The easiest way to find this entry address is to load VIDSAVE and immediately look at the value of the pointer at 4049-4A hexadecimal (16457-58 decimal). The entry point will be the value of this pointer plus one.

For those not familiar with the high memory pointer, some explanation is in order. As mentioned above, the TRS-80 maintains a pointer (really just the number of a memory location) which tells BASIC and the operating system which memory locations are available. Anything from this number up to the highest memory location is considered to be "protected" — that is, neither the DOS or BASIC will try to put any data in the memory addresses higher than the value of the pointer. This feature is very convenient for programs like VIDSAVE, which you would like to be operational after BASIC or another program is loaded.

When VIDSAVE loads, it checks to see what the value of HIGH$ is (we'll call the pointer HIGH$ from now on, which is the LDOS terminology for it). The program then decrements the value of HIGH$ by the length of VIDSAVE. VIDSAVE is moved into this new protected area of memory. (This is done in lines 620 to 1470 of the program listing.) Now we can load any other program (as long as it honors HIGH$ and doesn't try to put data above that memory location), and VIDSAVE will be fully operational. HIGH$ is at X'4049-4' in a Model I and X'4411-12' in a Model III.

If all this seems a bit confusing at

```
                        00380 ; * commands for the SAVE and DUMP commands.
                        00390 ; * If no valid value is passed, the program simply
                        00400 ; * returns with no action taken.
                        00410 ; **********
                        00420 ;
                        00430 ;
                        00440 ; **********
                        00450 ; * SYSTEM EQUATES
                        00460 ; **********
                        00470 ;
000A                    00480 LF        EQU     10
000D                    00490 CR        EQU     13
4049                    00500 HIGH$     EQU     4049H
4467                    00510 @DSPLY    EQU     4467H
402D                    00520 @EXIT     EQU     402DH
3C00                    00530 VIDEO     EQU     3C00H
0A7F                    00540 USARG     EQU     0A7FH
4020                    00550 CURSOR    EQU     4020H
                        00560 ;
                        00570 ;
                        00580 ; **********
                        00590 ; * LOAD AND RELOCATE
                        00600 ; **********
                        00610 ;
5200                    00620           ORG     5200H
5200 F5                 00630 VIDSAVE   PUSH    AF              ; SAVE REGISTERS
5201 C5                 00640           PUSH    BC
5202 D5                 00650           PUSH    DE
5203 E5                 00660           PUSH    HL
5204 219C52             00670           LD      HL,MSG1         ; LOG-ON MESSAGE
5207 CD6744             00680           CALL    @DSPLY
520A 21C352             00690           LD      HL,MSG2
520D CD6744             00700           CALL    @DSPLY
5210 21DD52             00710           LD      HL,MSG3
5213 CD6744             00720           CALL    @DSPLY
5216 21FB52             00730           LD      HL,MSG5
5219 CD6744             00740           CALL    @DSPLY
521C 210F53             00750           LD      HL,MSG6
521F CD6744             00760           CALL    @DSPLY
5222 21003C             00770           LD      HL,3C00H        ; INITIALIZE CURPOS
5225 22D657             00780           LD      (CURPOS),HL
5228 2A4940             00790           LD      HL,(HIGH$)      ; GET TOP OF MEMORY
522B 018D04             00800           LD      BC,LAST-START   ; LENGTH OF PROGRAM
522E A7                 00810           AND     A               ; CLEAR CARRY
522F ED42               00820           SBC     HL,BC
5231 224940             00830           LD      (HIGH$),HL      ; RESET TOP OF MEMORY
                        00840 ;
                        00850 ;
                        00860 ; **********
                        00870 ; * RUN-TIME RELOCATION MODULE
                        00880 ; **********
                        00890 ;
5234 C5                 00900           PUSH    BC              ; SAVE REGISTERS
5235 E5                 00910           PUSH    HL
5236 1804               00920           JR      RELO
5238 71                 00930 FORCE     LD      (HL),C          ; LO BYTE
5239 23                 00940           INC     HL
523A 70                 00950           LD      (HL),B          ; HI BYTE
523B C9                 00960           RET
523C 010A00             00970 RELO      LD      BC,KEEP-START+1
523F 09                 00980           ADD     HL,BC
5240 E5                 00990           PUSH    HL
5241 C1                 01000           POP     BC              ; NEW ADD. OF KEEP
5242 219053             01010           LD      HL,KEEP1+1      ; KEEP1
5245 CD3852             01020           CALL    FORCE
5248 21AE53             01030           LD      HL,KEEP2+1      ; KEEP2
524B CD3852             01040           CALL    FORCE
524E 012200             01050           LD      BC,REST-START+1
5251 E1                 01060           POP     HL              ; HIGH$
5252 E5                 01070           PUSH    HL
5253 09                 01080           ADD     HL,BC
5254 E5                 01090           PUSH    HL
5255 C1                 01100           POP     BC              ; NEW ADDRESS OF REST
5256 219F53             01110           LD      HL,REST1+1      ; REST1
5259 CD3852             01120           CALL    FORCE
525C 21C453             01130           LD      HL,REST2+1      ; REST2
525F CD3852             01140           CALL    FORCE
5262 018C00             01150           LD      BC,AREA-START+1
```

first glance, don't worry! It won't prevent you from using this program. The only thing you need to know is that the address found in HIGH$ is BACKWARDS (". . . curiouser and curiouser," said Alice.)! For example, if the value in memory location X'4049' is X'OA' and the value at X'404A' is X'FF', then the highest unprotected memory location is X'FFOA'.

When VIDSAVE has been loaded, its entry point will be the address of the high memory pointer plus one. You may determine the value of the pointer in one of three ways:

1) Use DEBUG and look at X'4049-4A'.

2) PEEK at the two memory locations in BASIC.

3) If you are using LDOS, use the MEMORY command.

It would have been possible to avoid this manual method of finding the entry point to VIDSAVE by putting a pointer to the program somewhere in the memory space of the operating system. The problem is that different operating systems use differing parts of memory below X'5200', and memory which is not used by one DOS may be used by another.

Once the entry point to VIDSAVE is known, it is simple to use. First, use the DEFUSR command to pass the entry point to BASIC. Then when you want to use the routine, enter it with a USR call (an example is given below).

Using VIDSAVE

There are four main commands that VIDSAVE responds to:

1) Save the present screen and return to BASIC with the cursor at the present position.

2) Dump a previously saved screen and return to BASIC at the present cursor position.

3) Save the present screen but return to BASIC with the cursor at a new position.

4) Dump a previously saved screen but return to BASIC at a new cursor position.

The command you want to execute is initiated by passing a variable through the USR command when VIDSAVE is called. Note that anytime a screen is

```
5265 E1       01160         POP     HL              ; HIGH$
5266 E5       01170         PUSH    HL
5267 09       01180         ADD     HL,BC

5268 E5       01190         PUSH    HL
5269 C1       01200         POP     BC              ; NEW ADD. OF AREA
526A 215E53   01210         LD      HL,AREA1+1      ; AREA1
526D CD3852   01220         CALL    FORCE
5270 217053   01230         LD      HL,AREA2+1      ; AREA2
5273 CD3852   01240         CALL    FORCE
5276 018C04   01250         LD      BC,CURPOS-START+1
5279 E1       01260         POP     HL              ; HIGH$
527A E5       01270         PUSH    HL
527B 09       01280         ADD     HL,BC
527C E5       01290         PUSH    HL
527D C1       01300         POP     BC              ; NEW ADD. OF CURPOS
527E 216653   01310         LD      HL,CURPOS1+1    ; CURPOS1
5281 CD3852   01320         CALL    FORCE
5284 217B53   01330         LD      HL,CURPOS2+1    ; CURPOS2
5287 CD3852   01340         CALL    FORCE
528A E1       01350         POP     HL              ; RESTORE REGISTERS
528B C1       01360         POP     BC
528C F3       01370         DI                      ; INTERRUPTS OFF
528D 23       01380         INC     HL              ; NEW START
528E EB       01390         EX      DE,HL           ; DESTINATION
528F 214B53   01400         LD      HL,START        ; SOURCE
5292 EDB0     01410         LDIR                    ; MOVE IT
5294 FB       01420         EI                      ; INTERRUPTS BACK ON
5295 E1       01430         POP     HL              ; RESTORE REGISTERS
5296 D1       01440         POP     DE
5297 C1       01450         POP     BC
5298 F1       01460         POP     AF
5299 C32D40   01470         JP      @EXIT           ; BACK TO DOS
              01480 ;
              01490 ;
              01500 ; **********
              01510 ; * SIGN-ON MESSAGE FIELDS
              01520 ; **********
              01530 ;
529C 0A       01540 MSG1    DM      LF,LF,'VIDSAVE - Screen Save / Dump
                                    Routine',CR
     0A 56 49 44 53 41 56 45
     20 2D 20 53 63 72 65 65
     6E 20 53 61 76 65 20 2F
     20 44 75 6D 70 20 52 6F
     75 74 69 6E 65 0D
52C3 28       01550 MSG2    DM      '(c) 1981 by T.A. Daneliuk',CR
     63 29 20 31 39 38 31 20
     62 79 20 54 2E 41 2E 20
     44 61 6E 65 6C 69 75 6B
     0D
52DD 54       01560 MSG3    DM      'T&R Communications Associates',CR
     26 52 20 43 6F 6D 6D 75
     6E 69 63 61 74 69 6F 6E
     73 20 41 73 73 6F 63 69
     61 74 65 73 0D
52FB 43       01570 MSG5    DM      'Chicago, IL   60625',CR
     68 69 63 61 67 6F 2C 20
     49 4C 20 20 20 36 30 36
     32 35 0D
530F 0A       01580 MSG6    DM      LF,'Copying, except for backup purposes
     43 6F 70 79 69 6E 67 2C
     20 65 78 63 65 70 74 20
     66 6F 72 20 62 61 63 6B
     75 70 20 70 75 72 70 6F
     73 65 73 20 62 79 20 6F
     77 6E 65 72 2C 20 50 52
     4F 48 49 42 49 54 45 44
     21 0A 0D
              01590 ;
              01600 ;
              01610 ; **********
              01620 ; * ACTUAL VIDSAVE ROUTINE
              01630 ; **********
              01640 ;
534B CD7F0A   01650 START   CALL    USARG
534E F5       01660         PUSH    AF              ; SAVE REGISTERS
534F C5       01670         PUSH    BC
```

saved (whether or not you are returning to the present cursor position), the present cursor position is also saved. This allows you to later dump back a screen *and* the previous cursor position. The exact variable to pass to VIDSAVE for each command is as follows:

CMD 1 — Pass a 0 (zero) to VIDSAVE.

CMD 2 — Pass 5000 to VIDSAVE.

CMD 3 — Pass a number between 1 and 1024 to VIDSAVE. The exact number depends on where you want the cursor to be when VIDSAVE returns to BASIC. This value is the value you would use to position the cursor with PRINT@ plus one. For example, the upper left corner of the screen would be 1, the lower right would be 1024.

CMD 4 — Pass a number between -1 and -1024 to VIDSAVE. The value is computed just as for command 3, but its value is negative to signify that you want to dump a previously memorized screen, (not save the present one) and return to a new cursor position.

Two final points of interest: When VIDSAVE loads, the part of memory it uses to store the screen is not initialized. This means that if the first variable passed to VIDSAVE is 5000, it will dump to the screen whatever was previously in this memory area (usually garbage). This was done so that if you accidentally reboot the system, it is possible to reload VIDSAVE into the same location and recover a previously saved screen. This is helpful during the development stage of programming. Also, the cursor position *is* initialized when VIDSAVE is loaded so that if a 5000 is the first variable passed, the cursor will be positioned in the "home" position (upper left corner). If you pass an invalid variable to VIDSAVE, it will just ignore it and nothing will be done.

An Example

Here is a demonstration program which will allow you to experiment with VIDSAVE. It presumes that VIDSAVE has been loaded and the value of HIGH$ is X'FF09' (i.e., VIDSAVE begins at X'FF0A'):

```
10 DEFUSR0=&HFF0A
```

```
5350 D5       01680          PUSH    DE
5351 E5       01690          PUSH    HL
5352 1830     01700          JR      SAVE
              01710 ;
              01720 ;
              01730 ; **********
              01740 ; * SUB-ROUTINE TO SAVE SCREEN AND CURSOR POS'N
              01750 ; **********
              01760 ;
5354 C5       01770 KEEP     PUSH    BC              ; SAVE REGISTERS
5355 D5       01780          PUSH    DE
5356 E5       01790          PUSH    HL
5357 21003C   01800          LD      HL,VIDEO        ; SOURCE
535A 010004   01810          LD      BC,1024         ; # OF BYTES
535D 11D653   01820 AREA1    LD      DE,AREA         ; DESTINATION.
5360 EDB0     01830          LDIR                    ; MOVE IT
5362 2A2040   01840          LD      HL,(CURSOR)     ; CURSOR POS'N
5365 22D657   01850 CURPOS1  LD      (CURPOS),HL     ; SAVE IT
5368 E1       01860          POP     HL              ; RESTORE REGISTERS
5369 D1       01870          POP     DE
536A C1       01880          POP     BC
536B C9       01890          RET
              01900 ;
              01910 ;
              01920 ; **********
              01930 ; * SUB-ROUTINE TO DUMP SCREEN & CURSOR POS'N
              01940 ; **********
              01950 ;
536C C5       01960 REST     PUSH    BC              ; SAVE REGISTERS
536D D5       01970          PUSH    DE
536E E5       01980          PUSH    HL
536F 21D653   01990 AREA2    LD      HL,AREA         ; SOURCE
5372 010004   02000          LD      BC,1024         ; # OF BYTES
5375 11003C   02010          LD      DE,VIDEO        ; DESTINATION
5378 EDB0     02020          LDIR                    ; MOVE IT
537A 2AD657   02030 CURPOS2  LD      HL,(CURPOS)     ; STORED CURSOR POS'N
537D 222040   02040          LD      (CURSOR),HL     ; TO *DO DCB
5380 E1       02050          POP     HL              ; RESTORE REGISTERS
5381 D1       02060          POP     DE
5382 C1       02070          POP     BC
5383 C9       02080          RET
              02090 ;
              02100 ;
              02110 ; **********
              02120 ; * PASSED PARAMETER INTERPRETED HERE
              02130 ; **********
              02140 ;
5384 CB7C     02150 SAVE     BIT     7,H             ; PASSED VARIABLE - ?
5386 2031     02160          JR      NZ,DUMPC
5388 110000   02170          LD      DE,0            ; IS IT 0 ?
538B ED5A     02180          ADC     HL,DE
538D 2005     02190          JR      NZ,DUMP
538F CD5453   02200 KEEP1    CALL    KEEP            ; IF VARIABLE WAS 0
5392 183D     02210          JR      OUT
5394 118813   02220 DUMP     LD      DE,5000         ; IS IT 5000 ?
5397 A7       02230          AND     A               ; CLEAR CARRY
5398 E5       02240          PUSH    HL              ; SAVE VARIABLE
5399 ED52     02250          SBC     HL,DE
539B E1       02260          POP     HL              ; BRING VARIABLE BACK
539C 2005     02270          JR      NZ,SAVEC        ; JUMP IF NOT 5000
539E CD6C53   02280 REST1    CALL    REST            ; DO IF 5000
53A1 182E     02290          JR      OUT
53A3 110104   02300 SAVEC    LD      DE,1025         ; IS HL>1024
53A6 A7       02310          AND     A               ; CLEAR CARRY
53A7 E5       02320          PUSH    HL              ; SAVE VARIABLE
53A8 ED52     02330          SBC     HL,DE
53AA E1       02340          POP     HL              ; BRING IT BACK
53AB 3024     02350          JR      NC,OUT
53AD CD5453   02360 KEEP2    CALL    KEEP
53B0 11FF3B   02370          LD      DE,VIDEO-1
53B3 19       02380          ADD     HL,DE           ; NEW CURSOR POS'N
53B4 222040   02390          LD      (CURSOR),HL     ; INTO *DO DCB
53B7 1818     02400          JR      OUT
53B9 E5       02410 DUMPC    PUSH    HL
53BA A7       02420          AND     A               ; CLR. CARRY
53BB 1100FC   02430          LD      DE,-1024
53BE ED52     02440          SBC     HL,DE
53C0 E1       02450          POP     HL
```

```
20 INPUT "VALUE TO PASS TO
VIDSAVE: ";A

30 N=USR0(A)

40 GOTO 20

50 END
```

Other Possibilities

Since VIDSAVE is self-loading, it is possible to load it several times in succession (noting the entry point to the routine each time). This allows the programmer to save and dump several different screen displays. Using this technique, you can create animation and other moving displays. You are limited only by imagination and memory space!

Acknowledgement

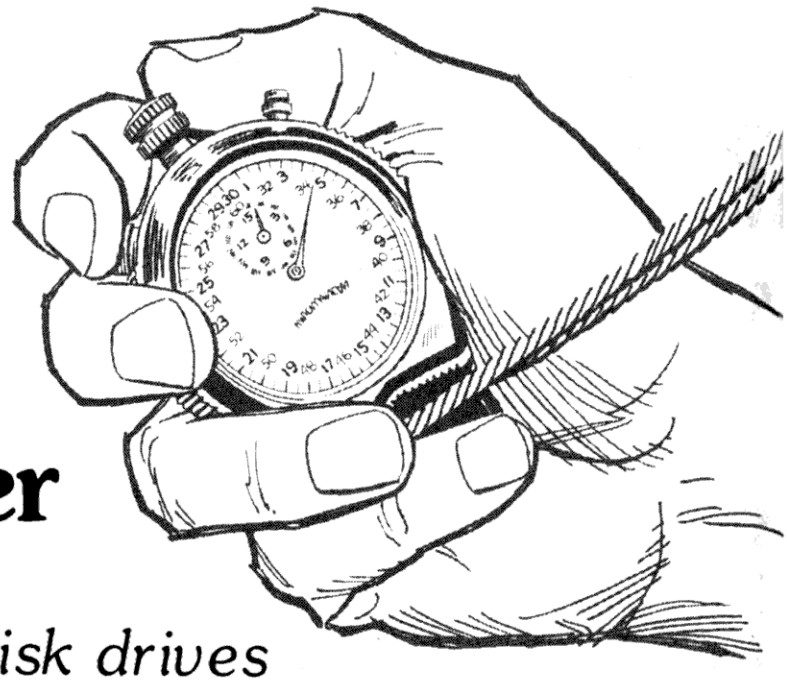
A special word of thanks to Bill Schroeder of Galactic Software, Inc., who suggested this project. His ideas and support have been invaluable.


```
53C1 380E     02460          JR      C,OUT
53C3 CD6C53   02470 REST2    CALL    REST
53C6 E5       02480          PUSH    HL          ; GET -(CURSOR) INTO HL
53C7 D1       02490          POP     DE
53C8 21FF3B   02500          LD      HL,VIDEO-1
53CB A7       02510          AND     A
53CC ED52     02520          SBC     HL,DE       ; NEW CURSOR POS'N
53CE 222040   02530          LD      (CURSOR),HL
53D1 E1       02540 OUT      POP     HL          ; RESTORE REGISTERS
53D2 D1       02550          POP     DE
53D3 C1       02560          POP     BC
53D4 F1       02570          POP     AF
53D5 C9       02580          RET
0400          02590 AREA     DS      1024        ; SPACE FOR VIDEO MEM.
0002          02600 CURPOS   DS      2           ; SPACE FOR CURSOR POS'N
57D8          02610 LAST     EQU     $
5200          02620          END     VIDSAVE
```

```
@DSPLY   4467 @EXIT    402D AREA    53D6
AREA1    535D AREA2    536F CR      000D
CURPOS   57D6 CURPOS1  5365 CURPOS2 537A
CURSOR   4020 DUMP     5394 DUMPC   53B9
FORCE    5238 HIGH$    4049 KEEP    5354
KEEP1    538F KEEP2    53AD LAST    57D8
LF       000A MSG1     529C MSG2    52C3
MSG3     52DD MSG5     52FB MSG6    530F
OUT      53D1 RELO     523C REST    536C
REST1    539E REST2    53C3 SAVE    5384
SAVEC    53A3 START    534B USARG   0A7F
VIDEO    3C00 VIDSAVE  5200
```

■

Floppy disk tachometer



Timing check for disk drives

Model I /III

Darrell A. Bymoen, Ridgecrest, CA

When I purchased my TRS-80 Model III, I was not aware of the many differences between it and its predecessor, the Model I. As time went on, it became obvious that the only way I was going to get the utility software I wanted would be to write much of my own, or modify existing software. When a friend came to me with a suspected problem with his disk drive speed, I decided to write the tachometer program presented here.

The program's main body is written in BASIC (Listing 1) with an embedded machine language routine that does the time/revolution measurement. The machine code is POKEd into RAM via the 'read' and 'data' statements. BASIC is far too slow to accomplish speed measurements with any accuracy. However, BASIC is a very good way to do the number-crunching necessary to convert the count measured by the machine language routine to RPM.

Fifty-two RPM measurements are made and displayed, followed by the average RPM, which is displayed with 'maximum' and 'minimum' RPMs. The average RPM is useful when adjusting the speed setting for the drive, but when speed problems are suspected, the speed variations from average are more likely to cause problems. If a speed problem is encountered, it may be a fluctuation during a short period, even a single revolution. In other words, the average speed may be within tolerance limits, but if the electronic control does not work correctly, the speed may vary far outside specification for a brief period.

It only takes a momentary speed error to cause a read/write problem. Because of this, instantaneous speed is the most desirable to measure. Since this is impossible with software, the best that can be accomplished is to measure the deviation for a single revolution.

A sufficient number of speed measurements need to be made to give a good indication of the speed variance with time. Fifty-two were selected, as this seemed sufficient, and it also formats nicely on the screen. None of the RPM measurements should exceed the manufacturer's specification. The specification on the Radio Shack drives is 300 +/- 4.5 RPM.

This program was written especially for the TRS-80 Model III, and is incompatible, without modification, for the Model I. The Model III has a clock frequency of 2.02752 Mhz, versus 1.774 for the Model I. Another difference is that the Model III control to the disk drives is I/O mapped, versus memory mapped for the Model I. What this means is that the Model III accesses the floppy disk controller with 'IN' and 'OUT' instructions, where the Model I uses 'LD' instructions.

Lines 50 through 110 (Listing 1) load the machine language program into RAM. For reference, Listing 2 is the assembly language listing. Lines 130 through 180 input the drive number and put it in the proper form for the disk drive controller. Listing 1 is for the 2-drive user, and if more or less drives are used, change line 150. For a 3-drive user, line 150 should be 'IF DN < 3 THEN ------ etc'. Line 180 POKEs the drive number into RAM, where the machine language routine will look for it. Lines 190 and 200 make the call to the machine language routine.

The called 'USR' routine (Listing 2) begins by loading the drive number POKEd into 80FFH by BASIC, and

M-ZAL™

Release Two Is Here!

Our Users asked for it, and we made the best TRS-80 disk assembler even better! Now M-ZAL has:

- Recursive MACROS with full symbolic parameter substitution
- Conditional assembly with unlimited nesting
- Text Editor warm-start/recovery option
- Support for all popular source file formats
- Load Module inspect/superzap using LINKER

and the features that made it famous:

- True multi-pass assembly
- Full Screen Option Menus
- ENTRY/EXTERN symbols, relocation, and object module linking
- 8 character labels with sorted symbol table
- Built-in symbol table cross-reference
- Nested *INCLUDES (source file chaining/nesting)
- Lower case support (Model 3 only)
- Over 175 pages of documentation including Z-80 Technical Manual
- Full Screen Text Editing and much, much more!

M-ZAL provides a modular, structured development environment that makes programming your TRS-80 a truly enjoyable experience. Order yours today. (Requires 32k, 2 disk system)

#1050-10	Mod 1	$149.00
#1250-10	Mod 3	$149.00

"We've been using M-ZAL heavily for over two months now, and it's been worth its weight in gold."
—Chuck Tesler
PROSOFT

"...well written software that is long overdue."
—Bruce Douglass
80 Microcomputing
BASIC Editor Review

"During the first 3 days of use, M-ZAL saved more money in programmers time than we had paid for the product."
—William Denman
Author of Asylum
MED SYSTEMS

"The editor has at least doubled my productive capacity...Having spent a good deal of money on so called 'utilities' that don't work, it's a delight to find one that not only lives up to its claims but surpasses the advertising."
—BASIC Editor user

Our choice for the best TRS-80 DOS:		
DOSPLUS 3.4	Mod 1	$149.00
DOSPLUS 3.4	Mod 3	$149.00

CAU Development Software— The Professionals Choice:

Full Screen Text Editor for BASIC: we were the first to give the TRS-80 this indispensable ability! A must for every BASIC programmer:

#1010-20	Mod 1	$24.95
#1210-20	Mod 3	$29.95

T-ZAL: custom designed for the Model 3, this tape based assembler has many of M-ZAL's most desired features.

#1250-20	Mod 3 only	$49.95

XBUG: for machine language analysis and debugging—this tiny (2.5k) but powerful program lets you do it all!

#1020-10	Mod 1	$19.95
#1220-10	Mod 3	$19.95

FILEXFER: transfer any disk file between two TRS-80s. Use direct cable link or RS modems. Mod 1 and Mod 3 can be mixed. RS-232 and disk reqd.

#1040-10	Mod 1 and 3	$49.95

Call or write for more information.
All products require level II.
Mail orders specify catalog #.
Send check, money order, or MC/VISA numbers and expiration date to:

Computer Applications Unlimited
P.O. Box 214, Dept. ABO Rye, New York 10580
(914) 937 6286

N.Y. State residents add applicable sales tax.
Outside USA and Canada add $7.00 shipping.
Dealer Inquiries Invited.

MasterCard VISA

COMPUTER APPLICATIONS UNLIMITED™
a div. of CAU, Inc.

TRS-80 is a TM of Tandy Corp. M-ZAL is a TM of CAU, Inc. Z-80 is a TM of Zilog, Inc.

outputing to the drive controller. The drive is told to go to track zero and delay about one-half second to allow the drive speed to stabilize. Register BC becomes the pointer for storage of the count/revolution in RAM and DE is the counter.

LOOP1 continuously checks the disk controller status register until it finds the index hole in the disk. The disk drive has an optical sensor that detects the light passing through the disk index hole. When the controller senses this light, it sets a bit in the disk controller status register. This register bit is read and tested by LOOP1, LOOP2, and LOOP3. LOOP1 just waits for the hole to indicate the index mark. LOOP2 and LOOP3 are the actual counter loops, and are the same, except that one counts the hole interval, and the other counts the no-hole interval. Lines 460 through 510 store the count, and then line 550 determines if all 52 measurements have been made. If not done, it branches back to TEST, and the whole sequence is repeated.

Counter register 'DE' counts the number of combined passes through LOOP2 and LOOP3. One pass through either loop requires approximately 9.25 microseconds at a clock of 4 Mhz. Since the Model III clock is 2.02752 Mhz, the time must be multiplied by the ratio of the clock frequencies (or 4.0/2.02752). The 'NOP' in line 400 is required because the 'JR' statement in line 390 takes less time to drop through than to jump. The 'NOP' compensates for this difference. The above timing information was obtained from a Zilog *Z80 Assembly Language Programming Manual*.

When control returns to the calling program, BASIC will find the data stored in RAM between memory locations A002H and A038H. The data will be in byte pairs representing a number of counts for each revolution of the disk. These data are in the form of lsb,msb, standard for the Z80 cpu. Memory location A000H contains the count for the first measured revolution, but this data is not used, because it may be incorrect. It takes one pass through the routine to guarantee that the drive is synchronized to the software.

Line 240, Listing 1, takes the count per revolution and puts it in a form for BASIC to manipulate. Line 250 converts the count 'CO' to time in seconds. Line 260 divides this time into 60 to get RPM. Lines 290 and 300 then sort the RPM measurements into maximums and minimums. Lines 350 and 360 will flag any measurements that are out of tolerance.

Listing 3 contains the modifications to allow this program to operate with a Model I. The differences in the BASIC listing are in lines 50, 70, 180, 190, 210, 250, and the 'DATA' statements. The 'DATA' statements amount to changing all the 'IN' and 'OUT' statements to 'LD' statements in Listing 2. The following are the changes for the Model I machine language program:

Line	OPCode	Label	Instruction
200	32E137		LD (37E1H),A
220	32EC37		LD (37ECH),A
250	3AEC37	READY	LD A,(37ECH)
290	32E137		LD (37E1H),A
300	3AEC37	LOOP1	LD A,(37ECH)
350	3AEC37	LOOP2	LD A,(37ECH)
400	3AEC37	LOOP3	LD A,(37ECH)
520	32E137		LD (37E1H),A

The Tachometer Program is loaded from BASIC using standard TRS-80 Model III procedures. Protecting memory is not necessary. When the program is RUN, it will show the prompt "TACHOMETER IS LOADING", while the machine language portion is POKEd into RAM.

Next, the user will be asked for the "DRIVE NUMBER". When the drive number is entered, the user must make sure a disk is installed in the drive undergoing test. If a disk has not been installed, or the drive is not available, the system will hang up, and a RESET will be necessary to recover.

After the drive number has been entered, the drives will run for about 30 seconds, and the indicator light on the drive being tested will illuminate. The RPM data will then format on the screen, and the user will again be asked to input another "DRIVE NUMBER". The process can be repeated as long as the user desires. Pressing BREAK takes the user back to BASIC.

It is not known just how great a speed error or fluctuation is necessary to cause a read/write error. If speed excursions do fall outside of the manufacturer's specifications, and you are having some unexplained system crashes, or program load errors, then it may be necessary to give some attention to your drives.

To adjust the Model III drives, it is necessary to remove the case. Remove the screws on the bottom and very carefully lift the case straight up. There are several cables and wires that could get caught and damaged. The cathode ray tube remains attached to the top and can be set to the side.

When the case is off, you will find the drives have two circuit boards each, and the speed adjustment is on the smaller of the boards at the rear of the drive. The potentiometer adjustment will be labeled R4. After the speed adjustments are made, carefully reassemble the system and replace the screws.

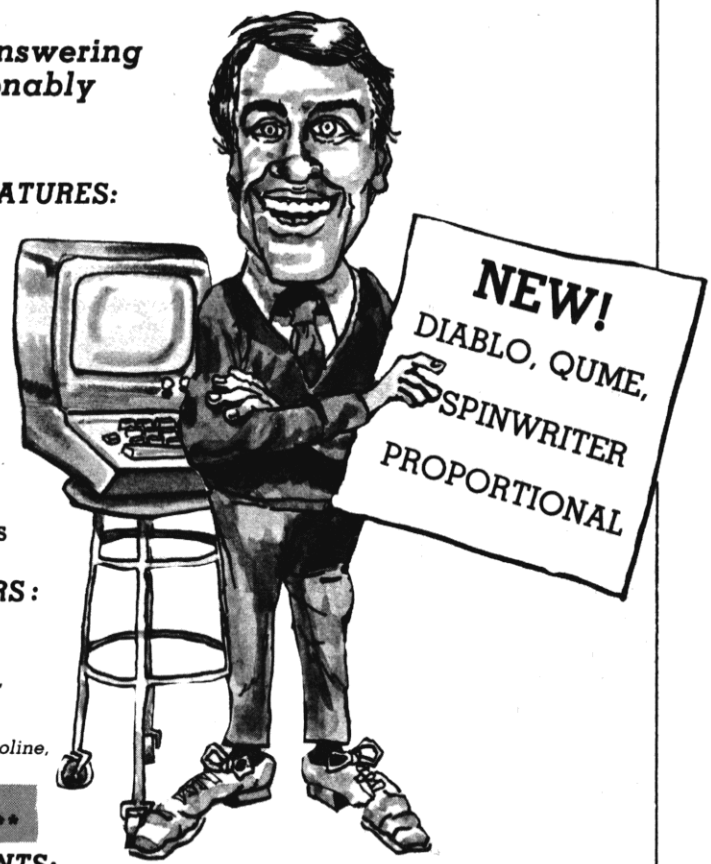
Listing 1
BASIC Listing for Model III

```
10 '*********************************
20 'TRS-80 MODEL III DISK TACHOMETER
30 '*********************************
40 '
50 CLS:R=&HA100
60 PRINT"TACHOMETER IS LOADING"
70 FOR N= 1 TO 71
80 READ D
90 POKE R,D
100 R=R+1
110 NEXT N
120 CLS
130 INPUT"DRIVE NUMBER";DN
140 CLS
```

HAPPINESS IS...NEWSCRIPT™

THE WORD PROCESSOR FOR BUSINESSMEN AND PROFESSIONALS

We offer you ongoing support by answering your questions and providing reasonably priced enhancements.



A FEW OF NEWSCRIPT'S STANDARD FEATURES:

***Form Letters with merging of names and addresses**
***Gives superb appearance to your final documents**
***Comprehensive manual with hundreds of examples**
***Centering, top/bottom titles, indents, pagination**
***Underlining, boldface, double-width, italics†**
***Sub-scripts, super-scripts, proportional pitch†**
***Generates Table of Contents, sorted Index**
***Allows block graphics, special symbols†**

***Search and replace globally or within a range**
***Block move, copy, delete, insert from other file**
***AUTOSAVE, WHOOPS, DIRECTORY, KILL, REPEAT**

***Based on IBM's "EDGAR" and "SCRIPT" systems**
***Supplied ready-to-run on "tiny" DOSPLUS**
***Easily transferred to NEWDOS, NEWDOS/80, LDOS, TRSDOS**
***Includes handy Quick Reference Card**

NEWSCRIPT CONTROLS THESE PRINTERS:

***EPSON MX-80, MX-100: all 12 fonts, plus underlining and block graphics; italics with GRAFTRAX**

***Centronics 737, 739; Radio Shack L.P. IV, L.P. VIII, Prowriter, Daisy Wheel II, NEC PC-8023A, TEC 8500R, C.ITOH 8510: Right-justified true proportional printing**

******Good support for: Diablo, Spinwriter, Starwriter, QUME, Microline, Anadex, modified Selectric, standard line printers*

NOW AVAILABLE: right-justified true proportional support for Diablo, Spinwriter, Starwriter, and Qume (DaisyWheel Option)**

TYPICAL USER AND REVIEWER COMMENTS:

"There is a new word processor on the market that will change the way people think about the capabilities of the TRS-80. Imagine a powerful mainframe text editor running on a TRS-80, with virtual compatibility between the two versions...on-going support second to none, with superb documentation." (Jim Klaproth, 80-U.S. JOURNAL February, 1982)

the manual:

"It definitely rates the first '10' given to any documentation reviewed in this column." (A.A. Wicks, COMPUTRONICS, October, 1981)

the software: "An excellent Word Processor" (D.H.); "Absolutely fantastic" (S.E.S.); "You have features that I cannot duplicate on my $14,000 system" (J.B.)

the support: "Your phone information system and the prompt and courteous staff that you provide to help your clients...are worth the cost of the system." (V.H.H.)

REQUIRED CONFIGURATION: **48K TRS-80 with one disk drive. Specify Model I or Model III.**

† Some features work only if your printer has the mechanical capability.

** DaisyWheel proportional is extra-cost option.

NEWSCRIPT 7.0:	**$124.95**
MAILING LABELS OPTION:	**29.95**
Special: NEWSCRIPT + LABELS:	**$139.95**
Daisywheel Proportional Option: "PENCIL"/"SCRIPTSIT"	**49.95**
File Convertor:	**14.95**
Manual Only (180 pages):	**30.00**
Reference Card Only:	**2.50**
Electric Webster + Correction Feature:	**149.50**
Graphics Editor and Programmer:	**45.95**
DOSPLUS 3.4:	**149.95**

TO ORDER, CALL NOW, TOLL-FREE:

(800) 824-7888, Operator 422
CALIF: (800) 852-7777, Oper. 422
ALASKA/HAWAII: (800) 824-7919
FOR TECHNICAL INFORMATION CALL:
(213) 764-3131, or write to us.

Order from your Software dealer, or from:

PROSOFT™

DEPT. C, BOX 560 • NO. HOLLYWOOD, CA 91603

TERMS: We accept VISA, Mastercard, checks, Money Orders, C.O.D. and even cash. We pay shipping via surface UPS inside U.S.A. Please add $3.00 for Blue Label, 6% tax in California, and 15% outside North America (air shipment).

```
150 IF DN<2 THEN 170
160 PRINT"DRIVE NUMBER TOO LARGE, TRY A
GAIN":GOTO 130
170 DN=2[DN
180 POKE &HA0FF,DN
190 DEFUSR0=&HA100
200 A=USR0(0)
210 M=&HA002
220 RT=0:MAX=0:MIN=500
230 FOR N=1 TO 52
240 CO=PEEK(M)+256*PEEK(M+1)
250 TM=CO*4*9.25/2.02752/1000000
260 RPM=60/TM
270 RT=RT+RPM
280 PRINT RPM,
290 IF MAX<RPM THEN MAX=RPM
300 IF MIN>RPM THEN MIN=RPM
310 M=M+2
320 NEXT N
330 AV=RT/52
340 PRINT:PRINT"AVG RPM =" AV"     MAX R
PM ="MAX"     MIN RPM ="MIN
350 IF MAX > 304.5 THEN PRINT"MAXIMUM S
PEED TOO HIGH"
360 IF MIN < 295.5 THEN PRINT"MINIMUM S
PEED TOO LOW"
370 GOTO 130
380 DATA 243,058,255,160,211,244,062,00
3,211,240
390 DATA 001,000,128,205,096,000,219,24
0,203,071
400 DATA 032,250,058,255,160,211,244,00
1,000,160
410 DATA 017,000,000,219,240,203,079,04
0,250,219
420 DATA 240,019,203,079,032,249,000,21
9,240,019
430 DATA 203,079,040,249,123,002,003,12
2,002,003
440 DATA 058,255,160,211,244,062,106,18
5,032,216
450 DATA 201,000,000,000,000,000,000,00
0,000,000
```

Listing 2
Machine Language Listing for Model III

```
                00100 ;*****************************************************
                00110 ;**     TRS-80  MODEL III  DISK DRIVE TACHOMETER      **
                00120 ;*****************************************************
                00130 ;
0060            00140 DELAY   EQU     60H     ;DELAY ROUTINE
A000            00150         ORG     0A000H
006A            00160 STORE   DEFS    106             ;STORE COUNTS/REV.
A0FF            00170         ORG     0A0FFH
0001            00180 DRIVE   DEFS    1               ;DRIVE NUMBER TO TEST
A100 F3         00190 BEGIN   DI
A101 3AFFA0     00200         LD      A,(DRIVE)       ;GET DRIVE NUMBER
A104 D3F4       00210         OUT     (0F4H),A        ;OUT TO FD CONTROLLER
A106 3E03       00220         LD      A,03
A108 D3F0       00230         OUT     (0F0H),A        ;GO TO TRACK ZERO
A10A 010080     00240         LD      BC,8000H        ;SET UP DELAY COUNT
A10D CD6000     00250         CALL    DELAY           ;DELAY 1/2 SEC.
A110 DBF0       00260 READY   IN      A,(0F0H)        ;GET FD STATUS
A112 CB47       00270         BIT     0,A             ;DRIVE BUSY?
A114 20FA       00280         JR      NZ,READY        ;JP IF FD BUSY
A116 3AFFA0     00290         LD      A,(DRIVE)       ;GET DRIVE NUMBER
A119 D3F4       00300         OUT     (0F4H),A        ;KEEP DRIVE GOING
A11B 010AA0     00310         LD      BC,STORE        ;-BC- POINT TO STORAGE
A11E 110000     00320 TEST    LD      DE,00           ;ZERO INTERVAL COUNTER
A121 DBF0       00330 LOOP1   IN      A,(0F0H)        ;GET STATUS REG
A123 CB4F       00340         BIT     1,A             ;TEST FOR HOLE
A125 28FA       00350         JR      Z,LOOP1         ;WAIT FOR HOLE
A127 DBF0       00360 LOOP2   IN      A,(0F0H)        ;GET FD STATUS
A129 13         00370         INC     DE
A12A CB4F       00380         BIT     1,A             ;TEST FOR HOLE
A12C 20F9       00390         JR      NZ,LOOP2        ;WAIT FOR NO HOLE
A12E 00         00400         NOP                     ;KILL TIME
A12F DBF0       00410 LOOP3   IN      A,(0F0H)        ;GET FD STATUS
A131 13         00420         INC     DE              ;BUMP COUNT
A132 CB4F       00430         BIT     1,A             ;TEST FOR HOLE
A134 28F9       00440         JR      Z,LOOP3         ;WAIT FOR HOLE
                00450 ;TIMED REVOLUTION COMPLETE
A136 7B         00460         LD      A,E
A137 02         00470         LD      (BC),A          ;STORE LOW COUNT
A138 03         00480         INC     BC
A139 7A         00490         LD      A,D
A13A 02         00500         LD      (BC),A          ;STORE HIGH COUNT
A13B 03         00510         INC     BC              ;UPDATE STORAGE POINTER
A13C 3AFFA0     00520         LD      A,(DRIVE)       ;GET DRIVE NUMBER
A13F D3F4       00530         OUT     (0F4H),A        ;KEEP DRIVE GOING
A141 3E6A       00540         LD      A,106
A143 B9         00550         CP      C
A144 20D8       00560         JR      NZ,TEST         ;CONTINUE TILL DONE
A146 C9         00570         RET                     ;RETURN TO CALLER
A100            00580         END     BEGIN
00000 TOTAL ERRORS

BEGIN   A100 00190   00580
DELAY   0060 00140   00250
DRIVE   A0FF 00180   00200 00290 00520
LOOP1   A121 00330   00350
LOOP2   A127 00360   00390
LOOP3   A12F 00410   00440
READY   A110 00260   00280
STORE   A000 00160   00310
TEST    A11E 00320   00560
```

Listing 3
BASIC Listing for Model I

```
10 '*************************************
20 ' TRS-80 MODEL I DISK TACHOMETER
30 '*************************************
40 '
50 CLS:R=-24320
60 PRINT"TACHOMETER IS LOADING"
70 FOR N= 1 TO 79
80 READ D
90 POKE R,D
100 R=R+1
110 NEXT N
120 CLS
130 INPUT"DRIVE NUMBER";DN
140 CLS
150 IF DN<2 THEN 170
160 PRINT"DRIVE NUMBER TOO LARGE, TRY A
GAIN":GOTO 130
170 DN=2[DN
180 POKE -24321,DN
190 DEF USR0=-24320
200 A=USR0(0)
210 M=-24574
220 RT=0:MAX=0:MIN=500
230 FOR N=1 TO 52
240 CO=PEEK(M)+256*PEEK(M+1)
250 TM=CO*4*9.75/1.77408/1000000
260 RPM=60/TM
270 RT=RT+RPM
280 PRINT RPM,
290 IF MAX<RPM THEN MAX=RPM
300 IF MIN>RPM THEN MIN=RPM
310 M=M+2
320 NEXT N
```

```
330 AV=RT/52
340 PRINT:PRINT"AVG RPM =" AV"     MAX R
PM ="MAX"     MIN RPM ="MIN
350 IF MAX > 304.5 THEN PRINT"MAXIMUM S
PEED TOO HIGH"
360 IF MIN < 295.5 THEN PRINT"MINIMUM S
PEED TOO LOW"
370 GOTO 130
380 DATA 243,058,255,160,050,225,055,06
2,003,050
390 DATA 236,055,001,000,128,205,096,00
0,058,236
400 DATA 055,203,071,032,249,058,255,16
0,050,225
410 DATA 055,001,000,160,017,000,000,05
8,236,055
420 DATA 203,079,040,249,058,236,055,01
9,203,079
430 DATA 032,248,000,058,236,055,019,20
3,079,040
440 DATA 248,123,002,003,122,002,003,05
8,255,160
450 DATA 050,225,055,062,106,185,032,21
2,201,000
```

Listing 4
Machine Language Listing for Model I

```
                00100 ;*************************************************
                00110 ;**     TRS-80  MODEL I  DISK DRIVE TACHOMETER     **
                00120 ;*************************************************
                00130 ;
A000            00140 STORE   EQU     0A000H  ;STORAGE FOR COUNTS/REV
0060            00150 DELAY   EQU     60H     ;DELAY ROUTINE
A0FF            00160 DRIVE   EQU     0A0FFH  ;DRIVE NUMBER TO TEST
A100            00170         ORG     0A100H
A100 F3         00180 BEGIN   DI
A101 3AFFA0     00190         LD      A,(DRIVE)       ;GET DRIVE NUMBER
A104 32E137     00200         LD      (37E1H),A       ;OUT TO FD CONTROLLER
A107 3E03       00210         LD      A,03
A109 32EC37     00220         LD      (37ECH),A       ;GO TO TRACK ZERO
A10C 010080     00230         LD      BC,8000H        ;SET UP DELAY COUNT
A10F CD6000     00240         CALL    DELAY           ;DELAY 1/2 SEC.
A112 3AEC37     00250 READY   LD      A,(37ECH)       ;GET FD STATUS
A115 CB47       00260         BIT     0,A             ;DRIVE BUSY?
A117 20F9       00270         JR      NZ,READY        ;JP IF FD BUSY
A119 3AFFA0     00280         LD      A,(DRIVE)       ;GET DRIVE NUMBER
A11C 32E137     00290         LD      (37E1H),A       ;KEEP DRIVE GOING
A11F 0100A0     00300         LD      BC,STORE        ;-BC- POINT TO STORAGE
A122 110000     00310 TEST    LD      DE,00           ;ZERO INTERVAL COUNTER
A125 3AEC37     00320 LOOP1   LD      A,(37ECH)       ;GET STATUS REG
A128 CB4F       00330         BIT     1,A             ;TEST FOR HOLE
A12A 28F9       00340         JR      Z,LOOP1         ;WAIT FOR HOLE
A12C 3AEC37     00350 LOOP2   LD      A,(37ECH)       ;GET FD STATUS
A12F 13         00360         INC     DE
A130 CB4F       00370         BIT     1,A             ;TEST FOR HOLE
A132 20FB       00380         JR      NZ,LOOP2        ;WAIT FOR NO HOLE
A134 00         00390         NOP                     ;KILL TIME
A135 3AEC37     00400 LOOP3   LD      A,(37ECH)       ;GET FD STATUS
A138 13         00410         INC     DE              ;BUMP COUNT
A139 CB4F       00420         BIT     1,A             ;TEST FOR HOLE
A13B 28FB       00430         JR      Z,LOOP3         ;WAIT FOR HOLE
                00440 ;TIMED REVOLUTION COMPLETE
A13D 7B         00450         LD      A,E
A13E 02         00460         LD      (BC),A          ;STORE LOW COUNT
A13F 03         00470         INC     BC
A140 7A         00480         LD      A,D
A141 02         00490         LD      (BC),A          ;STORE HIGH COUNT
A142 03         00500         INC     BC              ;UPDATE STORAGE POINTER
A143 3AFFA0     00510         LD      A,(DRIVE)       ;GET DRIVE NUMBER
A146 32E137     00520         LD      (37E1H),A       ;KEEP DRIVE GOING
A149 3E6A       00530         LD      A,106
A14B B9         00540         CP      C               ;DONE 53 YET?
A14C 20D4       00550         JR      NZ,TEST         ;CONTINUE TILL DONE
A14E C9         00560         RET                     ;RETURN TO CALLER
A100            00570         END     BEGIN
00000 TOTAL ERRORS
```

```
BEGIN   A100 00180   00570
DELAY   0060 00150   00240
DRIVE   A0FF 00160   00190 00280 00510
LOOP1   A125 00320   00340
LOOP2   A12C 00350   00380
LOOP3   A135 00400   00430
READY   A112 00250   00270
STORE   A000 00140   00300
TEST    A122 00310   00550
```

■

Map those disks!

Roadmaps for file rebuilding

Model I with disk

William H. Scott, Jr., Elizabeth, CO

I recently had a horrible experience. Sitting down at my keyboard one day, I booted up the system and entered BASIC. After loading the large program I was currently working on, I proceeded to list it. Much to my horror, it was not the expected program, but a different one I had whipped up hurriedly the previous day. A sinking sensation came over me. You know the feeling . . . something bad has happened, but you are not sure exactly what.

My mind started racing. The day before, I had been working on both programs, alternately loading and saving each one. Suddenly, the light turned on. In my haste the previous day, I must have typed the wrong filespec name for one of the save operations. The program I had spent so much time coding and debugging these last weeks was now gone. Wait a minute . . .

Where did that backup disk get to? I located the disk and inserted it into the drive. After loading the program, another list was done. Drat. The same lousy program! Not only did I destroy the original program, but I backed it up that way.

The thought of retyping page after page of BASIC statements was not very appealing. Maybe there would be enough of the original program left to piece together. Where was that book I bought a few months ago? *TRS-80 Disk and Other Mysteries* had something in it covering this subject . . .

Well, I hope this unfortunate occurrence never happens to you. It would help to be prepared, though. The above mentioned book turned out to be invaluable. I finally did get that program back together, but not without a lot of hard work and much bit manipulation.

Certainly, the Superzap program supplied with NEWDOS/80 also helped greatly. Its ability to copy disk sectors and rebuild them using the Copy Disk Data function was fantastic. But I could have used one more thing — a map of my disk to work from. It is one thing to piece a program back together, and another to be able to locate all of the pieces easily.

Following this exercise, I decided that some sort of disk map program that could be run regularly would provide important information for any file rebuilding operation. The DISKMAP program grew out of this need. This article will describe what information this program can supply to you. Along the way, we will learn where this information comes from, and a little bit about the DOS directory.

DISKMAP outputs two listings. The first listing (Dataset Summary) displays all of the active datasets (files) on the disk.

The second listing, the Track Allocation Table (Figure 2), is an actual pictorial "roadmap" of the disk. It shows, track by track, the locations of each active file, including extent information. All free granules and locked out tracks are also mapped. At the bottom of the map, the hex value of the disk master password is displayed with the number of free/allocated FPDEs and free/allocated granules for the disk.

The program can be used on one to four disk drives of like or different track capacities. Thirty-five-, 40- and 77-track drives can be accommodated by initializing the variables at the start of the program. The only requirement is that the directory format be TRSDOS or NEWDOS/80 compatible. The program can be modified easily to handle the expanded directory capability of the NEWDOS/80 operating system.

Before we take a look at the output provided by this program, let's take a minute to briefly describe the layout of the DOS directory area. In order to rebuild or salvage a crashed file, you will need to become very familiar with it. The standard DOS directory occupies one track of space, at track 17. The directory is made up of 10 sector-records of 256 bytes each. This is how they are ordered on the disk:

Record #— Contents

0 (zero)— Granule Allocation Table, Track Lock Out Table, Disk name, date, password, AUTO command.

1— Hash Index Table.

2-9— File Primary Directory Entries (FPDEs), File Extent Directory Entries (FXDEs).

All of the output produced by this program is taken from the directory. As we take a detailed look at the output listings of this program, I'll attempt to explain the functions of each of these tables and records.

Figure 1 is an example of the Dataset Summary listing. In the first item, data for the file BOOT/SYS will be displayed. To the right of the file name is the heading "HIT". This stands for Hash Index Table.

The Hash Index Table is the second record in the directory area, found at track 17, relative sector 1. It contains one byte file name hash codes which help DOS locate the primary directory entries (FPDEs) for the datasets on the disk. These FPDEs are located in relative sectors 2-9 of the directory. The FPDE is a 32-

byte field that provides DOS with all of the necessary information about a file. It contains such items as: filename, file attributes, passwords, location/length of all extents of the file, and end of file location.

Each directory record contains up to eight active FPDEs. That gives us a grand total of 64 possible directory entries for controlling up to 64 different files. This is plenty for most applications.

Beneath the HIT heading is the field 1/00. This indicates relative directory sector 1, displacement 00 within this sector. That is where the hash code for the BOOT/SYS file is kept. To the right of the equal sign is the actual hash code value for the file. When the file is initially created, DOS takes the filename extension and, using an algorithm, calculates the hash code. This code is then stored at a pre-determined location in the Hash Index Table according to the relative location of the file's directory entry.

The next heading we come to on this line is the dataset attributes. The letters "SIUAP" stand for System, Invisible, Update, Access, and Protection Level. Beneath these letters an "X" marker is placed to indicate that the particular attribute applies to the dataset. If an "S" occurs beneath the Update or Access fields, it indicates that the file has an update or access password which is other than blanks. Beneath the "P" (Protection Level) heading, a number is displayed which is the protection level of the file. The protection level is given in plain English to the very right of this line.

To the right of the SIUAP field are the Update and Access headings. The hexadecimal values of the update and access passwords are displayed below. These are hash values of the actual passwords and are kept in the FPDE. Note that a 9642 password (under the TEST/BAS file) is actually a password of eight blanks, or no password at all.

The next line of the listing shows the file extent information for the dataset. What is an extent? An extent is a single contiguous area of the disk, allocated solely to a file. If a file grows to where it bumps up against another one, a new extent is allocated. The file will then reside in two distinct areas on the disk. A file may grow to any number of extents, until we exhaust the space in the directory.

Each FPDE can contain information about four extents. If a file exceeds four extents, a File Extent Directory Entry (FXDE) is created. This is an entry basically identical to the FPDE. If this overflows, another FXDE is created, and so on. The FXDE is chained to the previous directory entry by a pointer in that entry.

Let's look at the information given for Extent-1. The FPDE for this extent is located in relative directory sector 2 at displacement 00 hex. Moving to the right, the starting location of the extent is given. This particular dataset, BOOT/SYS, starts at track zero, sector zero. BOOT/SYS is the DOS bootstrap system loader. As the listing indicates, this extent is one granule (five sectors) in length. The entire BOOT/SYS file is contained in this one extent.

The next two lines give the end of file information for BOOT/SYS. Two terms need to be defined at this point. FRS refers to the File Relative Sector. It is a relative sector (counting from zero) from the start of the file. DRS is the Disk Relative Sector. It is a relative sector (counting from zero again) from the start of the disk.

For example, track 24, sector 5 would be DRS 245, which is the 245th sector from the start of the disk. This assumes a standard of 10 sectors per track. The end of file for this dataset (actually the last byte of the file) occurs at FRS 4/FF and DRS 4/FF. Since the file starts at the beginning of the disk, these two values coincide. All other files will have different FRS and DRS values.

To the right of the listing, you will see a field termed FPDE EOF=. This is the actual value for the end of file maintained by DOS in the FPDE. Notice that it is one sector and one byte higher than the last byte of the file. Sector 5 is not actually part of this file, nor is it allocated to it. This is just the way that DOS does things. It *can* be a bit confusing. There is no special byte stored to mark the end of file. Its location is indicated solely in the directory.

The last line describing this dataset gives the total granule and sector allocation. BOOT/SYS has a total of one granule allocated. Out of that one granule (one granule equals five sectors), 5/5 or five out of five sectors, are actually used. If you take some time to go through your datasets, you will notice that many times all the sectors allocated are not used. Since DOS allocates on a granule (or five sector) basis, we will often have this condition. If only we could recover some of these unused sectors for other uses!

Following the Dataset Summary, the pictorial "map" (or Track Allocation Table) is produced. This provides perhaps the best visual indication of how your files are laid out on the disk. Figure 2 gives an example of a 40-track disk layout. Each track is divided into two granules, made up of five sectors each.

DOS allocates file space one granule at a time. This allows for some file growth before a new extent must be created. The active file names are indicated next to the track and under the granule at which they are allocated. The number of the file's extent is given to the right of each file name.

Notice the "free" granules displayed. These are, of course, free areas available for the creation of new files or expansion of existing ones. Knowing where these free granules are is extremely helpful in the process of rebuilding a destroyed file. This is usually where the rebuilding takes place.

The free and allocated granule list is kept in a special place called the Granule Allocation Table, located in record zero of the disk directory. DOS uses this table when allocating new space, or when returning old space when a file is killed. It is the key to what is, or is not, available on the disk.

If our disk has any locked out tracks, they will be indicated by the words "LOCKED OUT" in the file name area. Some operating systems support the locked out track feature, and some do not. A track may be locked out during the disk formatting process if an error is encountered while trying to format the track. The track would then become unusable. We would still have the use of the remainder of the disk, however.

A table of these locked out tracks is kept in record zero

Figure 1

```
DISKMAP - WORK1      09/21/81                          DATASET SUMMARY
-------------------------------------------------------------------
BOOT/SYS          HIT      SIUAP  UPDATE  ACCESS     PROTECT-LEVEL
                1/00=A2    XXXX6   607F    1FB2           EXECUTE

            EXTENT-1: FPDE=2/00  TRACK 0 SECTOR 0   GRANS 1
              LAST BYTE OF FILE(FRS)=4/FF   FPDE EOF=5/00
              LAST BYTE OF FILE(DRS)=4/FF
            TOTAL GRANS = 1     SECTORS ALLOCATED = 5/5

TEST/BAS          HIT      SIUAP  UPDATE  ACCESS     PROTECT-LEVEL
                1/20=EA    ....0   9642    9642        NO-RESTRICT

            EXTENT-1: FPDE=2/20  TRACK 26 SECTOR 5  GRANS 1
              LAST BYTE OF FILE(FRS)=2/6F   FPDE EOF=3/70
              LAST BYTE OF FILE(DRS)=267/6F
            TOTAL GRANS = 1     SECTORS ALLOCATED = 3/5

LPRT/ASM          HIT      SIUAP  UPDATE  ACCESS     PROTECT-LEVEL
                1/40=10    ....0   9642    9642        NO-RESTRICT

            EXTENT-1: FPDE=2/40  TRACK 0 SECTOR 5   GRANS 1
              LAST BYTE OF FILE(FRS)=2/86   FPDE EOF=3/87
              LAST BYTE OF FILE(DRS)=7/86
            TOTAL GRANS = 1     SECTORS ALLOCATED = 3/5
```

of the disk directory. This is not a great area of concern when rebuilding a disk file, but you should be aware of its location.

At the bottom of the listing, the number of free granules and free FPDEs are displayed, as well as the number allocated. This gives you an idea of how close you are to filling up your disk. Finally, the disk master password's hex value is listed. This is a hash value of the actual password, and is also kept in record zero.

To get the most use out of this program, you will need a printer. If you use your computer very much, you might want to run this program every few days or so, and keep the listings of your disks in a handy place. The output of the program is also displayed on the monitor, so that it can be used to scan the disk for certain information if you do not desire to print the listing.

Depending on how full your disk is, DISKMAP may take two to three minutes to run to completion. It *is* written in BASIC, which tends to slow things down a bit. There is a lot of string manipulation, so clear as much string area as you can. This will help to lessen the impact of the garbage sort.

Before running the program the first time, be sure to initialize the variable DR and the array variable TA at the start of the program. Variable DR should be initialized to the number of disk drives on your system. Variable TA (zero to three) should be set to the track capacities of these individual drives.

The program listing may be halted at any point by pressing any key on the keyboard. A question mark will then be displayed on the screen. To continue, simply press the ENTER key.

DISKMAP uses the ON ERROR and RESUME NEXT functions of BASIC. When a read is done to a DOS directory record, an error is generated. This is due to the directory being read protected. Even though the error is issued, the directory record is read.

I have purposely not gone into greater detail about the disk directory layout, or the details of rebuilding a disk file. The scope of describing the directory and the rebuilding task would encompass a large article in itself. Basically, rebuilding a file is a seat-of-the-pants operation anyway. The fix pretty much depends on the circumstances which caused the problem.

The DOS disk directory layout is described in several books now available. *TRS-80 Disk and Other Mysteries* is a very good book to own for the would-be file fixer. It describes the DOS directory in detail and gives many good pointers on fixing BASIC and other files. It sure helped me.

If you do not own a general file handling utility like Superzap, you will have to write one. This is really not as tough as it might seem. It could even be done in BASIC, with some careful manual manipulation of the directory.

You should now be on your way to a better understanding of the DOS directory and how files are structured on the disk. With the DISKMAP program, you'll be able to map this out for future reference. Hopefully, it won't be needed. So, get busy, and map those disks! I guarantee that you won't be sorry.

Figure 2

```
DISKMAP - WORK1      09/21/81              TRACK ALLOCATION TABLE
-------------------------------------------------------------------
   TRK#   GRAN-1 (SECTS 0-4) EXT    GRAN-2 (SECTS 5-9) EXT
   ----   ----------------------    ----------------------
    00    BOOT/SYS.............1    LPRT/ASM.............1
    01    STATUS/ASM...........1    PICTURE/BAS..........1
    02    PICTURE/BAS..........1    PDRIVER/ASM..........1
    03    LPRT/CMD.............1    STATUS/CMD...........1
    04    TICTAC/BAS...........1    TICTAC/BAS...........1
    05    TICTAC/BAS...........1    PRINT/BAS............1
    06    PDRIVER/CMD..........1    MACASM/ASM...........1
    07    DISKMAP/BAS..........1    LPRT/BAS.............1
    08    STATUS/BAS...........1    KEYDRV/CMD...........1
    09    ONESTR/CMD...........1    KEYDRV/ASM...........1
    10    LABEL/BAS............1    ONESTR/ASM...........1
    11    ONESTR/ASM...........1    ONESTR/ASM...........1
    12    ONESTR/ASM...........1    FOX1100/CMD..........1
    13    MACASM/ASM...........2    MACASM/CMD...........1
    14    MACASM/ASM...........3    CASSIO/ASM...........1
    15    CASSIO/ASM...........1    CASSIO/ASM...........1
    16    CASSIO/ASM...........1    CASSIO/ASM...........1
    17    DIR/SYS..............1    DIR/SYS..............1
    18    CLEANER0/BAS.........1    LABEL/BAS............2
    19    MACASM/ASM...........4    MACASM/ASM...........4
    20    MACASM/ASM...........4    MACASM/ASM...........4
    21    FASTLIFE/ASM.........1    FASTLIFE/ASM.........1
    22    FASTLIFE/CMD.........1    MACASM/ASM...........5
    23    MEMTEST/ASM..........1    MEMTEST/ASM..........1
    24    XMASTREE/BAS.........1    XMASTREE/BAS.........1
    25    MEMTEST/CMD..........1    DISKMAP/BAS..........2
    26    DISKMAP/BAS..........2    TEST/BAS.............1
    27    MX80/BAS.............1    PERFECT/BAS..........1
    28    DISKSPED/BAS.........1    DEATH/CIM............1
    29    DEATH/CIM............1    DEATH/CIM............1
    30    DEATH/CIM............1    DEATH/CIM............1
    31    DEATH/CIM............1    DEATH/CIM............1
    32    DEATH/CIM............1    DEATH/CIM............1
    33    DEATH/CIM............1    DEATH/CIM............1
    34    DEATH/CIM............1    DEATH/CIM............1
    35    DISKMAP/BAS..........3    COUNTER/ASM..........1
    36    SCRIPLBL/TXT.........1    COUNTER/CMD..........1
    37    COUNTER/BAS..........1    DISKMAP/BAS..........4
    38    DISKMAP/BAS..........4    FREE.................
    39    FREE.................     FREE.................
-------------------------------------------------------------------
        ALLOCATED FDES - 39          ALLOCATED GRANS - 77
          FREE FDES - 25               FREE GRANS - 3
                  MASTER PASSWORD - F023
```

Program Listing for Map Those Disks

```
10 CLS
20 CLEAR6000
30 DR=2          'NBR DRIVES ALLOC TO SYST
   EM
40 TA(0)=40      'DRIVE#0 TRK ALLOC
50 TA(1)=40      'DRIVE#1 TRK ALLOC
60 TA(2)=40      'DRIVE#2 TRK ALLOC
70 TA(3)=40      'DRIVE#3 TRK ALLOC
```

This Hand Is A Winner!

- **MASTER** Assembly/Machine language fast. Save months over other methods! Ideal for beginners. Two books plus disk/tape.
- **EXPERTS** The most powerful monitor/debugger ever written for Mod I/Mod III computers.
- **FREE** Detailed information package with coupon, or **See Your Dealer** for demonstration.
- **DEALER** Inquiries invited. Thousands of obsolete monitors can be easily replaced with this product.
- **GERMAN** Language edition available from:

  Hanspeter Schmid
  Lenaustr 2
  6906 Leimen / W. Germany

the software PLANTATION

THE SOFTWARE PLANTATION, INC.

P.O. BOX 44623 • TACOMA, WA 98444

(206) 531-1506

THE SOFTWARE PLANTATION, INC. — DEPT. 8A
P. O. BOX 44623 • TACOMA, WA 98444

Please **RUSH** information about **BUGOUT/PRO** to:

Name _____

Address _____

City _____ State _____ Zip _____

```
80 ONERRORGOTO1920
90 PRINTSTRING$(24,"-");" D I S K M A P
   ";STRING$(24,"-")
100 PRINT:PRINT@114,"BY BILL SCOTT":PRI
  NT
110 DEFFNMOD(X,Y)=X-INT(X/Y)*Y
120 LV$(0)="NO-RESTRICT"
130 LV$(1)="        KILL"
140 LV$(2)="      RENAME"
150 LV$(4)="       WRITE"
160 LV$(5)="        READ"
170 LV$(6)="     EXECUTE"
180 LV$(7)="  NO-ACCESS"
190 HX$="0123456789ABCDEF"
200 BK$=CHR$(&H96)+CHR$(&H42)
210 ZZ$="  "
220 FG=0:FD=0:RO=0:LI=0
230 PRINT
240 INPUT"OUTPUT TO PRINTER (Y/N) ";A$
250 IFA$="N"THENLP=0:GOTO380
260 IFA$<>"Y"THEN240ELSELP=1
270 PRINT
280 INPUT"PRINT BOTH LISTINGS (Y/N) ";A
  $
290 IFA$="Y"THEN380
300 IFA$<>"N"THEN280
310 PRINT
320 PRINT"SELECT ONE:"
330 PRINT"     (1) DATASET SUMMARY"
340 PRINT"     (2) TRACK ALLOCATION TABL
  E"
350 INPUTA$
360 IFA$="1"THENLI=1:GOTO380
370 IFA$<>"2"THEN320ELSELI=2
380 DA$=LEFT$(TIME$,8)
390 PRINT
400 PRINTDA$
410 INPUT"IS TODAY'S DATE CORRECT (Y/N)
   ";A$
420 IFA$="Y"THEN450
430 IFA$<>"N"THEN400
440 PRINT:PRINT"ENTER TODAY'S DATE (MM/
  DD/YY) FOR REPORT ";:INPUTDA$
450 PRINT
460 PRINT"ENTER DRIVE NUMBER:  0 -";DR-
  1;:INPUTA
470 IF A<0 OR A>=DR THEN PRINT"SYSTEM D
  EFINED FOR ";DR;" DRIVES ONLY";:GOTO4
  60
480 PRINT:PRINT"INSERT DISK IN DRIVE, T
  HEN <ENTER> TO CONTINUE"
490 IFINKEY$=""THEN490
500 CLS
510 DIM HT$(7,7)
520 DIM G$(TA(A)*2-1)
530 DIM GA$(TA(A)-1)
540 DIM LO$(TA(A)-1)
550 A$=RIGHT$(STR$(A),1)
560 DS$="DIR/SYS:"+A$
570 OPEN"R",1,DS$
580 OPEN"R",2,DS$
590 OPEN"R",3,DS$
600 FORI=0TOTA(A)-1
610 FIELD 1, (I*1) AS START$, 1 AS GA$(
  I), 95 AS DUMMY$, 1 AS LO$(I)
620 NEXT
630 FIELD 1, 206 AS DUMMY$, 2 AS PW$, 8
   AS NM$
640 GET 1,1
650 ZZ=ASC(PW$)
660 GOSUB1930
670 P1$=ZZ$
680 ZZ=ASC(RIGHT$(PW$,1))
690 GOSUB1930
700 P2$=ZZ$
710 NM$=NM$
720 PRINT"PROCESSING ...."
730 FORI=0TOTA(A)-1
740 G$(I*2)=STRING$(22,".")
750 G$(I*2+1)=STRING$(22,".")
760 IFASC(LO$(I))=255THENMID$(G$(I*2),1
  )="LOCKED-OUT":MID$(G$(I*2+1),1)="LOC
  KED-OUT":GOTO800
770 IFASC(GA$(I))=254THENMID$(G$(I*2),1
  )="FREE":FG=FG+1:GOTO800
780 IFASC(GA$(I))=253THENMID$(G$(I*2+1)
  ,1)="FREE":FG=FG+1:GOTO800
790 IFASC(GA$(I))=252THENMID$(G$(I*2),1
  )="FREE":MID$(G$(I*2+1),1)="FREE":FG=
  FG+2
800 NEXT
810 FORI=0TO7
820 FIELD 1, (I*32+22) AS START$, 1 AS
  XX$(1,I,0), 1 AS XY$(1,I,0), 1 AS XX$
  (1,I,1), 1 AS XY$(1,I,1), 1 AS XX$(1,
  I,2), 1 AS XY$(1,I,2), 1 AS XX$(1,I,3
  ), 1 AS XY$(1,I,3), 1 AS XX$(1,I,4),
  1 AS XY$(1,I,4)
830 FIELD 3, (I*32) AS START$, 1 AS CN$
  (I), 1 AS SW$(I), 1 AS N1$, 1 AS RP$(
  I), 1 AS N2$, 8 AS FS$(I), 3 AS XT$(I
  ), 2 AS UP$(I), 2 AS AP$(I), 2 AS EF$
  (I)
840 FIELD 3, (I*32+22) AS START$, 1 AS
  XX$(0,I,0), 1 AS XY$(0,I,0), 1 AS XX$
  (0,I,1), 1 AS XY$(0,I,1), 1 AS XX$(0,
  I,2), 1 AS XY$(0,I,2), 1 AS XX$(0,I,3
  ), 1 AS XY$(0,I,3), 1 AS XX$(0,I,4),
  1 AS XY$(0,I,4)
850 FORJ=0TO7
860 FIELD 2, (I*32+J) AS START$, 1 AS H
  T$(I,J)
870 NEXTJ
880 NEXTI
890 GET 2,2
900 GOSUB2010
```

```
910 FORJ=0TO7
920 GET 3,J+3
930 FORI=0TO7
940 IF(ASC(CN$(I)) AND &H10)=0THEN1770
950 FD=FD+1
960 IF(ASC(CN$(I)) AND &H80)=128THEN177
  0
970 IF(ASC(CN$(I)) AND &H40)=0THENS$=".
  "ELSES$="X"
980 IF(ASC(CN$(I)) AND &H08)=0THENI$=".
  "ELSEI$="X"
990 IFUP$(I)=BK$THENU$="."ELSEU$="X"
1000 ZZ=ASC(UP$(I)):GOSUB1930:U1$=ZZ$:Z
  Z=ASC(RIGHT$(UP$(I),1)):GOSUB1930:U2$
  =ZZ$
1010 IFAP$(I)=BK$THENA$="."ELSEA$="X"
1020 ZZ=ASC(AP$(I)):GOSUB1930:A1$=ZZ$:Z
  Z=ASC(RIGHT$(AP$(I),1)):GOSUB1930:A2$
  =ZZ$
1030 CN=ASC(CN$(I)) AND &H07
1040 P$=RIGHT$(STR$(CN),1)
1050 ZZ=I*32+J
1060 GOSUB1930
1070 H1$=ZZ$
1080 ZZ=ASC(HT$(I,J))
1090 GOSUB1930
1100 H2$=ZZ$
1110 K=INSTR(FS$(I)," ")
1120 IFK=0THENK=9
1130 DN$=LEFT$(FS$(I),K-1)
1140 IFXT$(I)="   "THENDE$="   ":GOTO11
  60
1150 DE$="/"+XT$(I)
1160 NS=CVI(EF$(I))
1170 NS$=STR$(NS)
1180 NS$=RIGHT$(NS$,LEN(NS$)-1)
1190 RS$=STR$(NS-1)
1200 RS$=RIGHT$(RS$,LEN(RS$)-1)
1210 IFFNMOD(NS,5)>0THENGR=INT(NS/5)+1E
  LSEGR=INT(NS/5)
1220 SA$=STR$(GR*5)
1230 SA$=RIGHT$(SA$,LEN(SA$)-1)
1240 ZZ=ASC(RP$(I))
1250 GOSUB1930
1260 XP$=ZZ$
1270 IFZZ>0THENZZ=ZZ-1ELSEZZ=255
1280 GOSUB1930
1290 RE$=ZZ$
1300 GOSUB2130
1310 GOSUB1980
1320 EX=1
1330 D$="FPDE="
1340 RO$=RIGHT$(STR$(J+2),1)
1350 M=I
1360 R=0
1370 ZZ=M*32
1380 GOSUB1930
1390 M$=ZZ$
1400 K=0
1410 GOTO1440
1420 IFASC(XX$(R,M,K))=254THEN1600
1430 IFASC(XX$(R,M,K))=255THEN1720
1440 EX$=STR$(EX)
1450 EX$=RIGHT$(EX$,LEN(EX$)-1)
1460 TR=ASC(XX$(R,M,K))
1470 IF(ASC(XY$(R,M,K)) AND &H20)=32THE
  NGG=1ELSEGG=0
1480 XY=ASC(XY$(R,M,K)) AND &H1F
1490 L1=TR*2+GG
1500 L2=TR*2+GG+XY
1510 FORL=L1TOL2
1520 MID$(G$(L),23-LEN(EX$))=EX$
1530 MID$(G$(L),1)=DN$+DE$
1540 NEXTL
1550 GOSUB2230
1560 EX=EX+1
1570 K=K+1
1580 GOSUB1980
1590 GOTO1420
1600 D$="FXDE="
1610 XO=ASC(XY$(R,M,K)) AND &H07
1620 XO=XO+2
1630 M=ASC(XY$(R,M,K)) AND &HE0
1640 M=M/32
1650 IFXO=J+2THEN1360
1660 IFXO=ROTHEN1690
1670 RO=XO
1680 GET 1,RO+1
1690 RO$=RIGHT$(STR$(RO-1),1)
1700 R=1
1710 GOTO1370
1720 DZ=TR*10+XY*5+GG*5+FNMOD(NS-1,5)
1730 DZ$=STR$(DZ)
1740 DZ$=RIGHT$(DZ$,LEN(DZ$)-1)
1750 GOSUB2290
1760 GOSUB1980
1770 NEXTI
1780 NEXTJ
1790 CLOSE
1800 PRINT:PRINT"<ENTER> TO CONTINUE"
1810 IFINKEY$=""THEN1810
1820 GOSUB2410
1830 FORI=0TOTA(A)-1
1840 TR$=STR$(I)
1850 IFI<10THENMID$(TR$,1)="0"
1860 TR$=RIGHT$(TR$,2)
1870 GOSUB2650
1880 GOSUB1980
1890 NEXTI
1900 GOSUB2540
1910 END
1920 RESUMENEXT
1930 ZA=INT(ZZ/16)
1940 ZZ$=MID$(HX$,ZA+1,1)
1950 ZA=ZZ-ZA*16
1960 ZZ$=ZZ$+MID$(HX$,ZA+1,1)
```

```
1970 RETURN
1980 IFINKEY$=""THENRETURN
1990 INPUTB$
2000 RETURN
2010 IFLP=1GOSUB2100
2020 CLS
2030 PRINT"DISKMAP - ";NM$;"  ";DA$;TAB
  (48)"DATASET SUMMARY"
2040 PRINTSTRING$(63,"-")
2050 IFLI=20RLP=0RETURN
2060 LPRINT"DISKMAP - ";NM$;"  ";DA$;TA
  B(48)"DATASET SUMMARY"
2070 LPRINTSTRING$(63,"-")
2080 LC=2
2090 RETURN
2100 PRINT:PRINT"READY PRINTER, THEN <E
  NTER> TO CONTINUE"
2110 IFINKEY$=""THEN2110
2120 RETURN
2130 PRINTDN$;DE$;TAB(17)"HIT      SIUAP
    UPDATE  ACCESS     PROTECT-LEVEL"
2140 PRINTTAB(15)"1/";H1$;"=";H2$;"    "
  ;S$;I$;U$;A$;P$;"    ";U1$;U2$;"     ";
  A1$;A2$;"         ";LV$(CN)
2150 PRINT
2160 IFLI=20RLP=0RETURN
2170 IFLC>57THENFORB=1TO66-LC:LPRINT:NE
  XTB:LC=0
2180 LPRINTDN$;DE$;TAB(17)"HIT      SIUA
  P  UPDATE  ACCESS     PROTECT-LEVEL"
2190 LPRINTTAB(15)"1/";H1$;"=";H2$;"
  ";S$;I$;U$;A$;P$;"    ";U1$;U2$;"     "
  ;A1$;A2$;"         ";LV$(CN)
2200 LPRINT
2210 LC=LC+3
2220 RETURN
2230 PRINTTAB(10)"EXTENT-";EX$;": ";D$;
  RO$;"/";M$;"  TRACK";TR;"SECTOR";GG*5
  ;TAB(50)"GRANS";XY+1
2240 IFLI=20RLP=0RETURN
2250 IFLC>59THENFORB=1TO66-LC:LPRINT:NE
  XTB:LC=0
2260 LPRINTTAB(10)"EXTENT-";EX$;": ";D$
  ;RO$;"/";M$;"  TRACK";TR;"SECTOR";GG*
  5;TAB(50)"GRANS";XY+1
2270 LC=LC+1
2280 RETURN
2290 PRINTTAB(12)"LAST BYTE OF FILE(FRS
  )=";RS$;"/";RE$;"   FPDE EOF=";NS$;"/
  ";XP$
2300 PRINTTAB(12)"LAST BYTE OF FILE(DRS
  )=";DZ$;"/";RE$
2310 PRINTTAB(10)"TOTAL GRANS =";GR;"
   SECTORS ALLOCATED = ";NS$;"/";SA$
2320 PRINT
2330 IFLI=20RLP=0RETURN
2340 IFLC>56THENFORB=1TO66-LC:LPRINT:NE
  XTB:LC=0
2350 LPRINTTAB(12)"LAST BYTE OF FILE(FR
  S)=";RS$;"/";RE$;"   FPDE EOF=";NS$;"
  /";XP$
2360 LPRINTTAB(12)"LAST BYTE OF FILE(DR
  S)=";DZ$;"/";RE$
2370 LPRINTTAB(10)"TOTAL GRANS =";GR;"
   SECTORS ALLOCATED = ";NS$;"/";SA$
2380 LPRINT
2390 LC=LC+4
2400 RETURN
2410 CLS
2420 PRINT"DISKMAP - ";NM$;"  ";DA$;TAB
  (41)"TRACK ALLOCATION TABLE"
2430 PRINTSTRING$(63,"-")
2440 PRINT"    TRK#   GRAN-1 (SECTS 0-4
  ) EXT   GRAN-2 (SECTS 5-9) EXT"
2450 PRINT"    ----   ";STRING$(22,"-")
  ;"   ";STRING$(22,"-")
2460 IFLI=1ORLP=0RETURN
2470 LC=0:IFLI=0THENFORB=1TO66-LC:LPRIN
  T:NEXTB
2480 LPRINT"DISKMAP - ";NM$;"  ";DA$;TA
  B(41)"TRACK ALLOCATION TABLE"
2490 LPRINTSTRING$(63,"-")
2500 LPRINT"    TRK#   GRAN-1 (SECTS 0-
  4) EXT   GRAN-2 (SECTS 5-9) EXT"
2510 LPRINT"    ----   ";STRING$(22,"-"
  );"   ";STRING$(22,"-")
2520 LC=LC+4
2530 RETURN
2540 PRINTSTRING$(63,"-")
2550 PRINTTAB(8)"ALLOCATED FDES -";FD;T
  AB(34)"ALLOCATED GRANS -";TA(A)*2-FG
2560 PRINTTAB(11)"FREE FDES -";64-FD;TA
  B(37)"FREE GRANS -";FG
2570 PRINTTAB(20)"MASTER  PASSWORD - ";
  P1$;P2$
2580 IFLI=1ORLP=0RETURN
2590 IFLC>56THENFORB=1TO66-LC:LPRINT:NE
  XTB:LC=0
2600 LPRINTSTRING$(63,"-")
2610 LPRINTTAB(8)"ALLOCATED FDES -";FD;
  TAB(34)"ALLOCATED GRANS -";TA(A)*2-FG

2620 LPRINTTAB(11)"FREE FDES -";64-FD;T
  AB(37)"FREE GRANS -";FG
2630 LPRINTTAB(20)"MASTER PASSWORD - ";
  P1$;P2$
2640 RETURN
2650 PRINT"     ";TR$;"     ";G$(I*2);"
    ";G$(I*2+1)
2660 IFLI=1ORLP=0RETURN
2670 IFLC>59THENFORB=1TO66-LC:LPRINT:NE
  XTB:LC=0
2680 LPRINT"     ";TR$;"     ";G$(I*2);"
    ";G$(I*2+1)
2690 LC=LC+1
2700 RETURN
```

■

LDOS for beginners

Part II: File handling and Job Control Language

Model I /III with disk

T. A. Daneliuk, Chicago, IL

The heart of any disk system is the disk files themselves. A well designed DOS makes dealing with these files a simple, efficient process. It is important to realize that LDOS was intended, from the very beginning of its design, to be able to deal with hard disks. These disks have so much storage space that literally hundreds of files can be stored on one of them. For this reason, LDOS goes to great lengths to find and manipulate files easily.

File Handling in LDOS

The main file handlers in LDOS are the APPEND, COPY, LIST, KILL, DIR, CREATE and PURGE library commands, as well as the BACKUP utility. All of these have been enhanced significantly over their TRSDOS counterparts, but the commands of particular interest are DIR, PURGE and BACKUP. First, a few general observations are in order.

LDOS dates all files with the date they were created. Furthermore, any time a disk file is modified (i.e., written to), a "modifier flag" is set which indicates the file has been changed since the last BACKUP. This flag appears as a plus sign ("+") next to the file name when a directory is done. This allows you to quickly determine which files need to be archived with a BACKUP. Another feature of interest is that when LDOS displays a directory, it is sorted into alphabetical order. This alone really speeds up finding a particular file on a disk.

The reason DIR, PURGE and BACKUP naturally go together is that they have been enhanced to handle files "by class." As we will see, this allows the user to selectively pick the files on which these three commands operate. One other thing — LDOS treats all disk sizes, types and formats the same way as far as the user is concerned. Once the proper device drivers are loaded into the system, it makes no difference if the file is on a single density, double density, single-sided, double-sided, 5-inch, 8-inch, 5-meg hard drive, 8-meg hard drive... well, you get the idea. All of this hardware is "transparent" to the user.

Now then, what is a "by class" file handling procedure? Well, the basic idea is that a DOS command, or utility, is forced to operate on only a select file, or group of files, in the system. This "class" of files may be determined in several ways. You might want to specify all of the files ending with the extension /CMD. Or, perhaps, you would like to deal only with files created before April 26, 1982. Another possiblility would be to work only with files whose modifier flag is set. DIR, PURGE and BACKUP have certain class handling in common. Each also has unique file handling features which help it do its particular task in the grand scheme of LDOS.

To begin with, all of the three commands allow you to specify a partial filespec (abbreviated as "partspec") in issuing the command. For example, DIR /CMD will show you a directory of all the files on every disk with the extension /CMD, while DIR L will show every file whose filespec begins with "L."

An enhancement of this concept is the idea of a "wildcard" character. The wildcard character is used in a partspec to indicate a "don't care" position in the filespec. For example, PURGE donuts/$$$:1 will purge (if you choose) all files whose names are "DONUTS" regardless of the extension (whose characters are wildcarded). By the way, notice that LDOS doesn't care if the command is in upper or lower case.

Another handy class operation is that of date. LDOS permits DIR, PURGE and BACKUP on the basis of date of file creation. The date may be specified one of several ways. A specific date, a range of dates, everything up to a specific date, or everything after a specific date may be used to define which files are to be operated on. A related procedure is

MORE BYTES FOR YOUR BUCKS

CP/M Compatibility Plus 64K RAM
for your TRS-80

Freedom Technology International

PRESENTS

Freedom Option

CP/M Compatibility for your TRS-80 Model I and III, PMC 80 and 81, and LNW 80 systems.

Freedom Plus

CP/M Compatibility Plus 64K RAM for your TRS-80 Model I and III.

TWO SYSTEMS IN ONE
Run CP/M application programs in addition to TRSDOS

INDUSTRY STANDARD
Enter the world of powerful CP/M application programs; WordStar, dBase II, etc.

OPEN AND SHUT CASE
EASY to install, plug-in boards; no wires or traces to cut; no soldering.

TECHNICAL BACKGROUND
Over 50 years of high technology experience...from Maxi's to Micro's...both Hardware and Software.

FREEDOM OF CHOICE
T8/OS (our powerful Z-80 code operating system) and CP/M 2.2 if you choose.

- Dealer Inquiries Invited
- Six Month Warranty
- 30 Day Money Back Guarantee

Freedom Option $275.00
Freedom Plus $490.00
Manual Only .....$ 25.00

Call us for information about our large selection of CP/M application programs.

To order Call Toll Free
1-800-523-4067
In Pennsylvania call
(215) 569-2381

TRS-80 is a registered trademark of Tandy Corp.
LNW-80 is a registered trademark of LNW Research Corporation
PMC 80,81 is a registered trademark of Personal Microcomputer Inc.
TRSDOS is a registered trademark of Tandy Corp.
CP/M is a registered trademark of Digital Research Inc.
WordStar is a registered trademark of MicroPro Int. Corp.
dBase II is a registered trademark of Ashton-Tate

Freedom Technology International

119 North 18th Street, Philadelphia, PA 19103

found in BACKUP which allows selection on the basis of whether the modifier flag is set. This makes archiving a snap! You just type in BACKUP :0 :1 (mod), and all files on drive 0 (zero) that have been modified will be backed up to drive 1. Similarly, BACKUP allows you to specify whether all files, only old (i.e., existing) files, or only new files are to be backed up from the source disk to the destination disk.

While this is only a brief and incomplete view of LDOS file handling, it should be clear that the system goes to great lengths to simplify user/computer interaction. This makes LDOS ideal for the business environment in which the nontechnical user is the rule, not the exception.

JCL: The LDOS Executive Language

A final major concept found in LDOS is its JCL (Job Control Language). This is *not* simply another version of DO files. The LDOS JCL is indeed a true "executive language" as found on minicomputers and mainframes. First, we'll take a quick look at the idea of an executive language, then we'll examine a practical JCL or two.

An executive language is a computer language whose instructions (or "syntax") are made up of DOS commands. It is also often called a "Job Control Language," hence the name "JCL." The latter is really a more descriptive name, because it hints at what JCL is used for: the control and management of different jobs the computer is to do. The idea is to pre-plan all of the different programs you would like to run in a given session. Determine what you would type in to run those programs. For example, to run an executable file, you just type in its name as in FILENAME/CMD. On the other hand, if it's a BASIC program, you'd have to first type in LBASIC and then RUN "FILE/ NAME/BAS."

Now, instead of actually running each program separately, you simply create an ASCII file with a wordprocessor (or the LDOS BUILD command) of all the commands necessary to run your programs. Then, you tell LDOS to DO that ASCII file. From then on, it's a hands-off operation! JCL reads the ASCII file when it's ready for another command instead of going to the keyboard and waiting for you to type it in.

"So what," you say, "lots of other DOS products can deal with DO files." Well, in the following section, we'll see what makes JCL stand out in a DO crowd! For now though, I'll throw in a simple example of a JCL file which might be used in a small business. Notice that all lines beginning with a period are comment lines and are ignored by LDOS.

```
. DEMO/JCL by T.A. Daneliuk
. Creation Date 03/25/82
.
. First we'll do the bookkeeping
PAYABLES/CMD
RCVABLES/CMD
. Probably should pay everyone too
. But we'll need BASIC for this one
LBASIC
RUN"PAYROLL/BAS"
. Now let's go back to LDOS
CMD"S"
```

Now, let's look at some of JCL's advanced features. To begin with, JCL allows execution "Labels" to be placed in a JCL file. These always start with the AT symbol ("@") and are used to divide one large JCL file into many small procedures. This allows you to create a huge file with every JCL operation you ever want to use in it, and call just the one procedure you need at the moment. For example, if we had started the JCL procedure above with the label "@BOOKS" we could have included it in a library of JCL procedures called PROCLIB/JCL. Whenever we want to do bookkeeping, we would just key in DO PROCLIB/JCL (@BOOKS), and the bookkeeping JCL procedure would be invoked.

Another JCL feature is the TOKEN substitution field. This allows you to put a dummy name (or variable) in the JCL procedure, and define it when you actually DO the procedure. For example, let's say you wrote a general purpose JCL procedure to document your BASIC programs:

```
@DOC
. First load disk BASIC
lbasic
. Then load the program
load "#NAME#/BAS:#S#
. Make a program listing
llist
. Go back to LDOS
cmd "s"
. And make a copy of the file
copy #NAME#/BAS:#S# :#D#
```

There are three TOKENs used in this JCL. #NAME# is the filename, #S# is the source drive on which the file presently exists, and #D# is the destination drive to which you'd like to copy the file. So, to document the file DONUTS/BAS, you key in:

```
DO PROCLIB (@DOC,NAME=
DONUTS, S=0,D=2)
```

This would look for the file on drive 0 (zero) and then later copy it to drive 2 after the listing was made.

JCL also supports certain control features called "MACROS." These are commands which are not part of the DOS Library and are used to control the execution of a JCL procedure. Space does not permit a complete description of each macro, so we'll just touch on a few of them.

First, there are conditional macros. These are similar to the BASIC IF-THEN-ELSE programming constructs. You can make decisions within the JCL as to how it is to execute. A practical application of this is in the use of a JCL to load and assemble an assembly language program which has to run on both a Model I and III. By using if-else JCL macros, you can assemble the program several different ways for the different machines. JCL also supports "OR" and "AND" type logical statements which can be used along with the conditionals to make decisions.

There are also a set of macros used to alert the user in some way. A tone may be generated (via the cassette port) or a message may be flashed on screen to alert the user. JCL has a set of macros which act to delay program execution. You can force a specific time delay anywhere in the program. This would be useful if you used a JCL procedure to arm a burglar alarm. Enough delay would be set at the beginning of the program to allow you to get out of the building before it armed the alarm.

Another macro can cause the JCL to wait for a specific time on the real time clock before continuing. This is handy if you have jobs to schedule which need to be run at a specific time. For example, how about a JCL that turned your furnace off at night after you went to sleep, and on again in the morning an hour before you got up.

There is also a pause macro which will wait until you press ENTER before continuing. This might be used to temporarily halt JCL execution to give the user a chance to put in a new disk, check to see if the printer is on line, etc.

Finally, there are a set of macros used to accept keyboard input in the middle of a JCL procedure. This gives the user the best of both worlds, hands-off operation with the option of keyboard entry when needed.

In summary, JCL provides a way for very complex computer jobs to be scheduled and executed with a minimum of effort on the part of the operator. Again, one of the best applications of JCL will be in small business. The programmer can write a JCL procedure which makes business programs easy to run for the non-technical user.

and Finally . . .

As of this writing, LDOS is the least expensive of all the TRS-80 operating systems available ($129). If you are looking for unparalleled performance and ease of use, this is the system for you! ■

X-Label

Use dot matrix printers to produce labels

Model I/III with or without Exatron Stringy Floppy

Spencer Hall, Contributing editor

This started out as a simple routine to print labels for cassettes and wafers on my Microline-80 dot-matrix printer. Now it is as much fun to run as a game program. X-label brings out the creative artist in almost anybody. Read the formal documentation which follows and you will see why. The samples here were produced using all the features described. You too, can make decorative lines of any symbol, center text or short decorative lines, change your mind at any time, print a sample and then revise the image and finally store the result on a wafer or on tape for later use. Lower case is available at a touch of the CLEAR key whether or not you have a lower case modification.

X-label will run on both Model I and III. (For Model III delete line 260 and change line 270 to read: 270 IF S=1 THEN IF Z<96 AND Z>123 THEN Z$=CHR$(Z—32). It will also drive the Epson MX-80 printer when it is set to the TRS-80 mode. For the Epson, you will have to put a period between the quotes in line 120. Also, if your label has blank lines, you cannot print your label without a border. The MX-80 just has to have something to print on every line, as far as I can tell, unless it is in its native mode. . .which would put the code for X-label in another ball park.

Label SAVE and LOAD routines in this listing are for the Exatron Stringy Floppy. Load the following in succession: (1) ULCBAS, the TRS-80 lower case mod driver, if you have it. (2) ESF I/O routine. (3) XLABEL. Only a driver located above RAM address 32767 (limit of 16K) works on my Model I. The driver located at 28672 causes things to "bomb" when the ESF I/O routine tries to come in. Omit the I/O routine if you don't plan to save labels for future use. You can now use your 16K driver without problems. Also omit the driver. It's only to allow you to see on the screen what you are going to get on the printer.

If you don't have a Stringy Floppy you can omit the SAVE and LOAD routines and still have fun. If you want to save labels on tape, I've included a list of changes you can make as you type the code. Just remember that although you can print them when they are written, you cannot save labels which contain colons, quotes or commas. The tape I/O routine rejects everything following these characters exactly as does the INPUT statement.

If you are using the original listing, omit the ON ERROR GOTO 2170 in line 10. This way you can test-run and find copying errors. It's only there to let you know if you forgot to load the I/O routine when you try to save to wafer. It can be edited in after everything is working well. Bugs which don't generate error messages can be traced because the remark statements tell you pretty much where everything is happening.

If copying code from a magazine isn't your bag and you still want X-label, contact Exatron on their hot line. They will have it available by the time this appears in print.

Save the labels your wife uses on her home-canned fruits, jams, jellies or what have you. Use them next year, changing only the date packed and, if need be, the variety of fruit used. If you are in the business of selling software, save a cover panel for documentation, attractively laid out with your name, address, etc. For each product, change only the program name.

Documentation for X-label

General: Super X-Label is a versatile and extremely easy-to-use program which exploits the

features commonly available on today's budget-priced dot-matrix printers to produce labels of many types. For the user who wishes to design a label, it permits selection of condensed, expanded or standard type sizes. Line lengths are definable up to 54 characters. Label lengths of one through nine lines can be chosen. Either six or eight lines to the inch are available.

Preformatted labels: It is not necessary to specify labels in detail as just described. An option, available on the main menu, displays a supplementary menu of preformatted labels. Fourteen of these are assigned and two remain open to be programmed by the user. These are callable instantly by number. Single digit numbers require ENTER to activate but numbers 10 through 16 will "trigger" instantly. Available labels include several for positions on compact cassettes and cassette

Sample labels produced using X-label

```
==================

   WOOD  SCREWS
   #10 3" LONG

==================
```

```
**********************************************
        NOVA - TELEVISED ON PBS
  Documentary on Mt. St. Helens Eruption
**********************************************
```

```
Washington Jr. High Glee Club
        WINTER CONCERT
Recorded on January 10, 1973
```

```
NEW YORK PHILHARMONIC CONCERT
Conductor: Zubin Mehta  Soloist: Andre Watts
Numbers include:
   WEBER: Overture to Die Freischutz
   BEETHOVEN: Piano Concerto No. 4
   BRAHMS: Tragic Overture
      (some static during the Brahms!)
```

```
FILES IN USE: 1 2 3 4 5 6 7 8 9
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 35 37 38 39
```

```
>><<>><<>><<>><<>><<>><<>><<>><<>><<>><<>><<>><<>><<
          THE BEATLES
>><<>><<>><<>><<>><<>><<>><<>><<>><<>><<>><<>><<>><<
```

View from Yakima Park looking at Emmons Glacier and Mt. Rainier. 11:45 A.M. 8/17/77

```
BACKUP
#1 ESF Monitor
#2 ESF I/O
#3 @FREEZE
#4 X!.ABEL
```

boxes, front and back labels for ESF wafers, two labels commonly used on VHS video cassettes, labels for plastic parts drawers, 2 X 2-inch colored slides, etc.

Input: When format specifications have been chosen, the height and width of the resulting text area is given. The user may now proceed or, if desired, specify a new label if the dimensions reported are not right for the intended purpose. Following either the display of label dimensions or the choice of a preformatted label, a format box appears on the screen with line numbers at the outside left. Below this, a one-line input box is drawn and labeled #1. Lines of the label are typed in the input box and transferred to the corresponding line in the format box using the ENTER key. The input box is cleared each time a line is transferred and its line number label automatically advanced.

Input options: These options, available at all times during the input operation, are the *only* instructions not displayed on the screen. You will want to learn them at once because they stimulate creativity and make the design of labels a fascinating and literally habit-forming experience.

ENTER ← as the first key struck on any line, creates a full blank line in the label.

ONE CHARACTER ONLY — sent to the format box using the ENTER key, causes that character to be replicated across the entire line. This feature simplifies the design of attractive labels using, in particular, the asterisk (*), the hyphen (-), the equal sign (=) and the dollar sign ($).

CENTERING TEXT: Text to be centered on the current line need not be centered in the input box. Type from the start as if it were just another line. Use the right arrow instead of ENTER. Centering will be done automatically.

PSEUDO—LOWER CASE is available at sign-on. This is indicated by a message at the lower left of the display. Each stroke of the CLEAR key toggles into or out of the CAPS ONLY mode. Current mode is always displayed. The CAPS MODE, of course, affects only the alphabet characters. If the Radio Shack (and possibly other) lower case modifications are installed and the driver program is activated, the actual type mode will be visible on the screen.

LINE CORRECTIONS are continuously available during the input operation. This mode is called by hitting the down-arrow, which changes the input box number to a question mark. This converts instantaneously to the desired line when the number is hit. After the corrected line has been entered, the input box offers a new start at the line number from which the change mode was called.

ALL CHARACTERS ON THE KEYBOARD are available for use since the INPUT statement is not used in the program.

ABORT INPUT is available if the user decides that the text or format is not what is wanted. The shifted up-arrow returns control to the main menu.

LINE LIMITER — If the users inadvertently writes a line past the end of the input box, the ENTER key will cut it to fit and clean up the input box.

Options after input: Three choices appear below the now complete box. These are (C) Change, which calls the change mode described above, (P) Print, and (R) Return to main menu. The latter option does not destroy the label in memory. It may be @SAVED to wafer from the main menu and/or recalled to the screen for further manipulation.

Print command: A question, answerable by one keystroke, asks if a wide border (W), a narrow border (N) or no border (ENTER) is desired. Either border will increase the typed area dimensions displayed earlier. The number of copies is requested next. The default (ENTER only) is 1. If the default is not used, the number of spaces to place between copies will be requested. Experimentation with label length and lines between copies will allow use of pin-feed labels of various sizes. Next inquiry is the tab position on the paper where the label or labels are to begin. A default (ENTER only) response to this question will cause the message CENTER? to appear. A response of Y will place the label or labels in the center of the page. This is excellent for title pages of reports. Default on this question places the label or labels at the start of a 64-character line unless expanded characters are being used. In this case, labels are printed at the left edge of the paper to allow space for the 40-character maximum line length permissible.

Perforation walk-over: If 8 1/2 by 11-inch pin-feed paper is being used, set the perforation between sheets at the print head center. A test of remaining space on the page is continuously made and when not enough space remains to allow printing of a label without crossing a perforation, the paper will "walk over" the perforation. Printing will begin again on the next panel.

Options after print: (1) Immediate redisplay of the label for modifications. This is excellent when several labels, such as those for bolts, screws, etc., have common names but variable sizes. (2) Create a new label in the same format. This makes it unnecessary to go through the specification routine again. (3) Return to main menu. This makes all options available. You may @SAVE your label to wafer and immediately return to it for further modification if desired. (4) Redisplay the menu of preformatted labels for another choice.

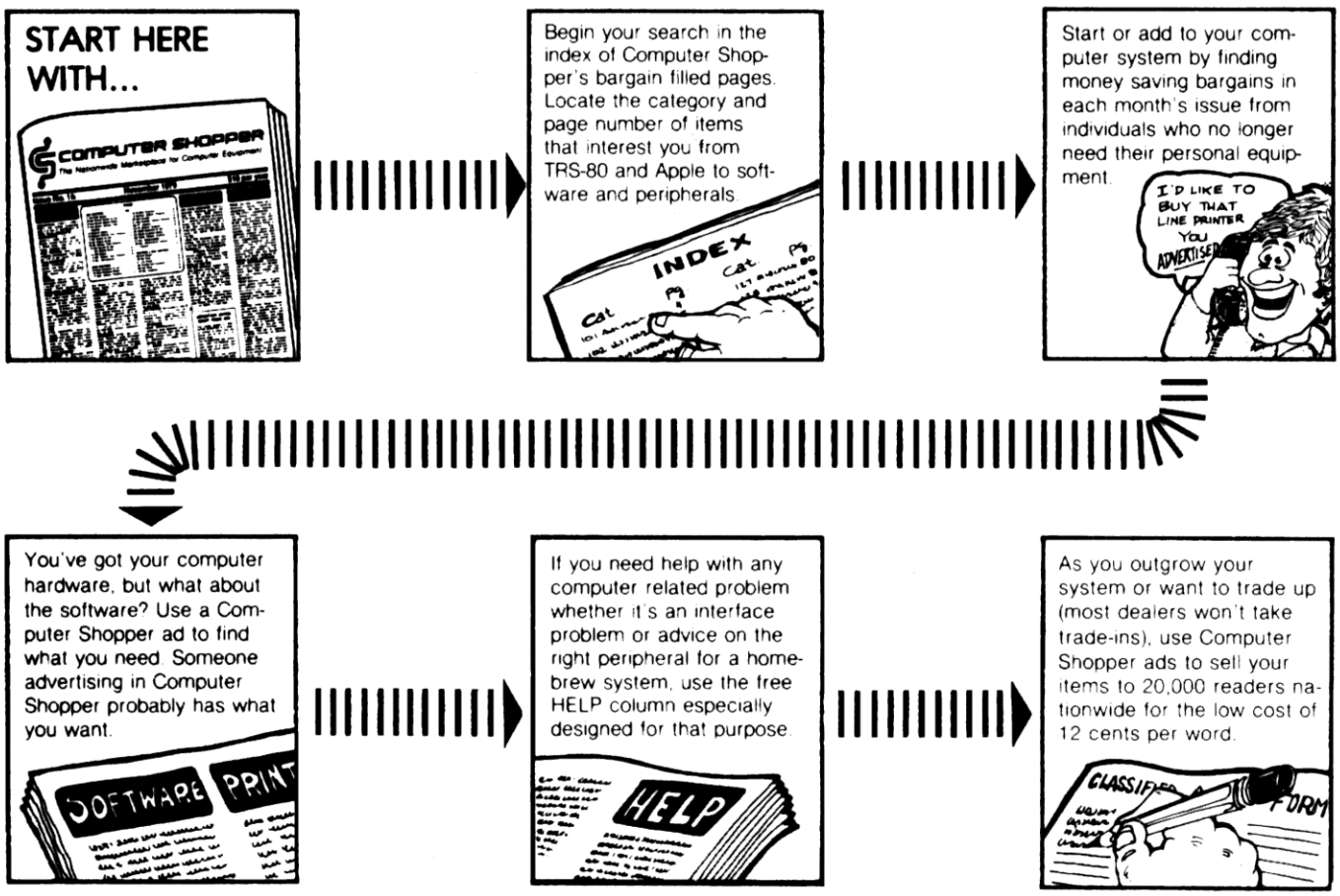
Program Listing for X-Label

```
10 CLEAR 1200:ON ERROR GOTO 2170:DEFINT
   M,Z:GOTO770
20 PRINTTAB((64-LEN(ZT$))/2)ZT$:RETURN
30 Z=960+(64-LEN(ZB$))/2:PRINT@Z,ZB$;:R
   ETURN
40 GOSUB400:PRINTTAB((64-ZN)/2)STRING$(
   ZN,ZC$);:RETURN
```

```
50 GOSUB 390:GOTO 40
60 PRINT@ 64*(ZP+1),;:PRINTCHR$(31):PRI
  NT@ 64*(ZP+1),;:RETURN
70 FOR ZZ=1 TO 345*ZS:NEXT ZZ:RETURN
80 PRINTAB(21) "TO PROCEED HIT ANY KEY
  "

90 Z$=INKEY$:IFZ$=""THEN 90     ELSE ZZ=
  VAL(Z$):RETURN
100 ZO=64+64*ZL-LEN(ZT$)/2-32:PRINT@ ZO
  ,ZT$;:RETURN
110 PRINT@981,"TO PROCEED HIT ANY KEY";
  :GOTO 90
120 FOR Z=1 TO ZN:LPRINT " " CHR$(10):N
  EXT:RETURN
130 PRINT "BE SURE WAFER IN PLACE IS YO
  UR FILE WAFER":PRINT:GOSUB 80     :PRI
  NT:RETURN
140 '-----INPUT ROUTINE-----
150 ZZ$="":J=0
160 PRINTCHR$(143);
170 Z$=INKEY$:IF Z$="" THEN 170 ELSE Z=
  ASC(Z$)
180 IF Z>31 THEN 260
190 IF Z=27 THEN 1720
200 IF Z=8 AND ZZ$="" THEN 170
210 IF Z=8 ZZ$=LEFT$(ZZ$,LEN(ZZ$)-1):PR
  INTCHR$(8);:PRINTCHR$(8);:J=J-1:GOTO
  160
220 IF Z=31 AND S=0 S=1:PRINT@960,STRIN
  G$(10,32);:PRINT@960,"CAPS ONLY";:PRI
  NT@ZE+2+J,;:GOTO 170
230 IF Z=31 AND S=1 S=0:PRINT@960,STRIN
  G$(10,32);:PRINT@960,"U & L CASE";:PR
  INT@ZE+2+J,;:GOTO 170
240 IF Z=10 THEN 690
250 IF Z=9 GOTO 380
260 IF Z>96 AND Z<123 THEN Z$=CHR$(Z-32
  ) ELSE IF S=1 THEN 280
270 IF Z>64 AND Z<91 THEN Z$=CHR$(Z+32)

280 IF Z=13 PRINTCHR$(8);:GOTO 300
290 ZZ$=ZZ$+Z$:J=J+1:PRINTCHR$(8);:PRIN
  T Z$;: GOTO 160
300 IF LEN(ZZ$)=1 THEN ZZ$=STRING$(CS,Z
  Z$)
310 IF ZZ$="" THEN ZZ$=STRING$(CS,128)
320 IF LEN(ZZ$)>CS THEN ZZ$=LEFT$(ZZ$,C
  S)
330 IF RIGHT$(ZZ$,1)=CHR$(10) ZZ$=LEFT$
  (ZZ$,LEN(ZZ$)-1)
340 RETURN
350 '-----TAB SETS FOR TABBED PRINTOUT-
  ----
360 T=(R-CS)/2:GOTO 1490
370 '-----CENTER TEXT ON LINE-----
380 M=(CS-LEN(ZZ$))/2:ZZ$=STRING$(M,128
  )+ZZ$+STRING$(M,128):RETURN
```

```
390 ZC$=CHR$(ZG)
400 IF ZN RETURN ELSE ZN=64:RETURN
410 '-----SUB# 230: TWO LINE SIGNATURE-
    ----
420 CLS:ZN=0:ZT$="EXATRON LABEL WRITER"
    :ZG=131:ZS=5:GOSUB 20:GOSUB 50:RETURN

430 '-----SUB #460: SET SCREEN FOR INPU
    T-----
440 IF LS=0 THEN FOR L=1 TO LS:L$(L)=""
    :NEXT
450 M1$=CHR$(156)+STRING$(CS,140)+CHR$(
    172)
460 M2$=CHR$(149)+STRING$(CS," ")+CHR$(
    170)
470 M3$=CHR$(141)+STRING$(CS,140)+CHR$(
    142)
480 '-----SUB #510: DRAW FORMAT BOX----
    -

490 CLS:GOSUB 30
500 ZT$=M1$:ZL=0:GOSUB100
510 ZT$=M2$:FOR L=1 TO LS:ZL=L:GOSUB 10
    0:PRINT@ZO-4,L;:L(L)=ZO+1:NEXT
520 ZT$=M3$:ZL=LS+1:GOSUB 100
530 '-----SUB #560: DRAW INPUT BOX-----

540 ZL=LS+2:ZT$=M1$:GOSUB 100
550 ZL=ZL+1:ZT$=M2$:GOSUB 100:ZE=ZO
560 ZL=ZL+1:ZT$=M3$:GOSUB 100
570 PRINT@960,STRING$(10,32);:IF S=0 PR
    INT@960,"U & L CASE";
580 IF S=1 PRINT@960,"CAPS ONLY";
590 RETURN
600 '-----SUB #750: WRITE ERASING STRIN
    G-----
610 BL$=STRING$(CS," "):RETURN
620 '-----SUB #790: GO UP AND TRY AGAIN
    -----
630 PRINTCHR$(27);:PRINTCHR$(255):PRINT
    CHR$(27);:RETURN
640 ZB$="":GOSUB 420:GOTO 850
650 '-----SUB 860: CALL FOR WAFER-----
660 IF INP(240)<> 242 PRINT "NEED YOUR
    FILE WAFER":GOSUB 630:GOTO 660
670 RETURN
680 '-----ACCEPT/PROCESS CHANGES (ALTER
    NATE PATH)
690 GOSUB 540:PRINT@ ZE-4," ? ";:GOSUB
    90:CC=VAL(Z$)
700 IF CC<1 THEN 690  ELSE IF CC>LS THE
    N 690
710 PRINT@ ZE-4,CC;:PRINT@ ZE+1,;:GOSUB
     150:L$(CC)=ZZ$
720 PRINT@ L(CC),BL$;:PRINT@ L(CC),L$(C
    C);
730 ZP=LS+1:GOSUB 60:GOSUB 540:IF X=9 G
    OTO 1270 ELSE GOTO 1210
740 '-----TRAP FOR WRONG CALL TO #5----
    -
750 ZS=1.5:GOSUB 60:PRINT@ 470,"NO LABE
    L IN MEMORY":GOSUB 70  :GOTO 810
760 '-----PROGRAM SIGN ON-----
770 GOSUB 420
780 ZL=4:ZT$="CAUTION!":GOSUB 100   :ZL
    =5:ZT$=STRING$(LEN (ZT$),131):GOSUB 1
    00:ZL=8:ZT$="BE SURE TO HAVE PRINTER
    ON LINE":GOSUB 100
790 ZL=10:ZT$="BEFORE PROCEEDING!":GOSU
    B 100:PRINT:PRINT:GOSUB 80
800 '-----MASTER MENU-----
810 ZP=2:GOSUB 60:PRINT"CHOOSE....."
820 PRINT:PRINTTAB(5)"(1) DESIGN A NEW
    LABEL"
830 PRINTTAB(5)"(2) CHOOSE A SPECIAL PU
    RPOSE LABEL"
840 PRINTTAB(5)"(3) INPUT A PRE-RECORDE
    D LABEL"
850 PRINTTAB(5)"(4) RECORD A LABEL"
860 PRINTTAB(5)"(5) REVIEW CURRENT LABE
    L"
870 PRINTTAB(5)"(6) CERTIFY A WAFER"
880 PRINT:PRINTTAB(8);:PRINT "ENTER YOU
    R CHOICE":GOSUB 90
890 IF ZZ<1 THEN 810 ELSE IF ZZ>6 THEN
    810
900 ON ZZ GOTO 920 ,1740 ,2120 ,2020 ,1
    670 ,2350
910 '-----INPUT LABEL SPECIFICATIONS---
    --
920 GOSUB 60
930 PRINT"(S) = STANDARD LETTERS - 10 P
    ER INCH"
940 PRINT"(X) = CONDENSED LETTERS - 16.
    5 PER INCH"
950 PRINT"(W) = EXPANDED LETTERS - 5 PE
    R INCH"
960 PRINT:PRINTTAB(5)"WHAT LETTER SIZE?
    "
970 GOSUB 90:IF Z$<>"S" IF Z$<>"X" IF Z
    $<>"W"PRINTCHR$(27);: GOTO 970
980 GOSUB 60:IF Z$="S" PRINT "USING STA
    NDARD CHARACTERS (10 PER INCH)":CN=54

990 IF Z$="X" PRINT "USING CONDENSED CH
    ARACTERS (16.5 PER INCH)":CN=54
1000 IF Z$="W" PRINT "USING EXPANDED CH
    ARACTERS (5 PER INCH)":CN=40
1010 A$=Z$:PRINT"NUMBER OF CHARACTERS P
    ER LINE ("CN"MAXIMUM )";:INPUT CS
1020 IF CS<1 GOSUB 630:GOTO 1010
1030 IF CS>54 GOSUB 630:GOTO 1010
1040 IF CS>40 AND CN=40 GOSUB 630:GOTO
    1010
1050 INPUT "NUMBER OF LINES PER LABEL (
    9 MAXIMUM)";LS
```

How to Buy or Sell Computer Equipment and Software



Computer Shopper is THE nationwide magazine for buying, selling and trading Micro and Mini-computer equipment and software. Each issue has over 60 pages full of bargains of new and used equipment.

You can save hundreds of dollars by getting the equipment you need from the hundreds of classified ads individuals place in Computer Shopper every month.

Now is the time for you to join over 20,000 other computer users who save time and money with a subscription to Computer Shopper.

Subscribe today and get your first issue and a classified ad absolutely FREE. Type or print your ad on a plain piece of paper and send it along with your subscription.

Just fill in the coupon or MasterCard or VISA holders can phone **TOLL FREE 1-800-327-9920** and start making your computer dollar go further today.

Cut out and mail to: **COMPUTER SHOPPER**
P.O. Box F115 • Titusville, FL 32780

Yes, I'll try Computer Shopper, I understand that if I'm not satisfied with my first issue I can receive a full refund and keep the first issue free.

☐ 1 year $10.00 ($30.00 in Canada)

☐ I have enclosed my free classified ad.

☐ I want to use my free ad later, send me a coupon.

NAME: _____

ADDRESS: _____

CITY: _____

STATE: _____ ZIP: _____

COMPUTER SHOPPER
P.O. Box F115 • Titusville, FL 32780
Telephone: 305-269-3211

30

```
1060 IF LS<1 GOSUB 630:GOTO 1050
1070 IF LS>9 GOSUB 630:GOTO 1050
1080 INPUT "NUMBER OF LINES PER INCH";L
  I
1090 IF LI<>6 IF LI<>8 GOSUB 630:GOTO 1
  080
1100 '----- SET TYPE SIZE AND CALCULATE
   LABEL DIMENSIONS -----
1110 PRINT:IF Z$="X" W=CS/16.5 ELSE IF
  Z$="W" W=CS/5
1120 IF Z$="S" W=CS/10
1130 H=LS/LI:F$="#.###"
1140 PRINT: : PRINT "LABEL WIDTH IS APP
  ROXIMATELY "USINGF$;W;:PRINT" INCHES"
  :PRINT "AND HEIGHT IS APPROXIMATELY "
  USING F$;H;:PRINT" INCHES"
1150 PRINT:PRINT "(P) PROCEED   (R)  RE
  DESIGN LABEL - WHICH?"
1160 GOSUB 90:IF Z$="R" THEN 920
1170 '-----WRITE INPUT DISPLAY-----

1180 GOSUB 610:GOSUB 440
1190 '-----NUMBER INPUT BOX/ACCEPT INPU
  T-----
1200 L=1:X=0
1210 PRINT@ ZE-4,L;:PRINT@ZE+1,;:GOSUB
  150:L$(L)=ZZ$
1220 L$(L)=ZZ$
1230 PRINT@ L(L),L$(L);
1240 L=L+1:IF L=LS+1 X=9:GOTO 1270
1250 ZP=LS+1:GOSUB 60:GOSUB 540:GOSUB 3
  0:GOTO 1210
1260 '-----INPUT PRINT INSTRUCTIONS----
  -
1270 ZN=1:ZP=LS+1:GOSUB 60:PRINT@ ZE-1,
  "C = CORRECTIONS":PRINT@ ZE+63,"P = P
  RINT":PRINT@ ZE+127,"R = RETURN TO MA
  IN MENU"
1280 GOSUB 90
1290 IF Z$<>"C" IF Z$<>"R" IF Z$<>"P" P
  RINTCHR$(27);:GOTO 1270
1300 IF Z$="C" ZP=LS+1:GOSUB 60:GOSUB 5
  40:GOTO 690
1310 IF Z$="R" GOSUB 420:GOTO 810
1320 GOSUB 60:P=ZE-4
1330 PRINT@P,"BORDER: N=NARROW, W=WIDE,
   <ENTER>=NONE":GOSUB 90:Q$=Z$:GOSUB 6
  30
1340 PRINT@ P,;:INPUT "HOW MANY COPIES"
  ;CX:IF CX=0 CX=1
1350 IF CX=1 THEN 1370
1360 GOSUB 630:PRINT@ P,;:INPUT "HOW MA
  NY SPACES BETWEEN COPIES";ZN
1370 GOSUB 630:PRINT@ P,;:INPUT "TABULA
  TE WHERE";T:IF T<>0 GOTO 1490
1380 GOSUB 630:PRINT@ P,;:PRINT"CENTER?
  ";:GOSUB 90:PRINT Z$:IF Z$<>"Y" THEN
  1490
1390 IF A$="W" AND Q$="W" R=36:GOTO 360
1400 IF A$="W" AND Q$="N" R=38:GOTO 360

1410 IF A$="W" R=40:GOTO 360
1420 IF A$="X" AND Q$="W" R=102:GOTO 36
  0
1430 IF A$="X" AND Q$="N" R=104:GOTO 36
  0
1440 IF A$="X" R=106:GOTO 360
1450 IF Q$="W" R=60:GOTO 360
1460 IF Q$="N" R=62:GOTO 360
1470 R=64:GOTO 360
1480 '----- PRINT LABELS -----
1490 IF A$="X" LPRINTCHR$(29) ELSE IF A
  $="W" LPRINTCHR$(31) ELSE LPRINTCHR$(
  30)
1500 IF A$="W" LPRINTCHR$(27)CHR$(65) E
  LSE LPRINTCHR$(27)CHR$(66)
1510 IF LI=8 LPRINTCHR$(27)CHR$(56) ELS
  E LPRINTCHR$(27)CHR$(54)
1520 POKE 16425,1
1530 IF LI=8 POKE 16424,89 ELSE POKE 16
  424,67
1540 IF Q$="N" THEN 2190 ELSE IF Q$="W"
   THEN 2250
1550 FOR C=1 TO CX:FOR L=1 TO LS:LPRINT
  TAB(T)  L$(L):NEXT L:GOSUB 120
1560 IF PEEK(16424)-PEEK(16425)=<LS LPR
  INTCHR$(11)
1570 NEXT C:T=0
1580 '-----OPTIONS AFTER PRINT LABELS--
  ---
1590 CX=0:CLS:GOSUB 420:PRINT:PRINT"AVA
  ILABLE NOW.....":PRINT
1600 PRINTTAB(5)"(1) DISPLAY SAME LABEL
   FOR CHANGES"
1610 PRINTTAB(5)"(2) NEW LABEL IN SAME
  FORMAT"
1620 PRINTTAB(5)"(3) RETURN TO MAIN MEN
  U"
1630 PRINTTAB(5)"(4) CHOOSE ANOTHER SPE
  CIAL LABEL":PRINT
1640 PRINTTAB(10)"CHOOSE....":GOSUB 90

1650 ON ZZ GOTO 1670,1180,1720,1740
1660 '-----WRITE STORED LABEL TO SCREEN
  -----
1670 IF LS=0 THEN 750 ELSE GOSUB 610:GO
  SUB 450
1680 FOR L=1 TO LS:PRINT@ L(L),L$(L);:N
  EXT
1690 X=9:GOTO 1270
1700 '-----SET UP NEW LABEL, SAME FORMA
  T-----
1710 GOSUB 450:GOSUB 610:FOR L=1 TO LS:
  L$(L)=" ":NEXT:GOSUB 490:GOSUB 540:GO
  TO 1200
1720 ZB$="":GOSUB 420:GOTO 810
```

```
1730 '-----PREFORMATTED LABEL MENU-----
1740 ZP=3:GOSUB 60:PRINT"THESE SPECIAL
  LABELS ARE AVAILABLE:":PRINT
1750 PRINT@322,"(1) CASSETTE MAIN LABEL
  ";:PRINT@386,"(2) CASSETTE BOTTOM LAB
  EL";
1760 PRINT@450,"(3) CASSETTE BOX SPINE"
  ;:PRINT@514,"(4) CASSETTE BOX SPINE (
  WIDE)";
1770 PRINT@578,"(5) WAFER FRONT (SQUARE
  )";:PRINT@642,"(6) WAFER BACK (RECTAN
  GLE)";
1780 PRINT@706,"(7) WAFER PROTECTOR (ON
  E LINE)";:PRINT@770,"(8) ADDRESS LABE
  L";
1790 PRINT@354,"(9) VHS CASSETTE SPINE"
  ;:PRINT@418,"(10) VHS CASSETTE SPINE
  (WIDE)";
1800 PRINT@482,"(11) VHS CASSETTE (NORM
  AL)";:PRINT@546,"(12) VHS CASSETTE (C
  OMPRESSED)";
1810 PRINT@610,"(13) PARTS CABINET DRAW
  ER";:PRINT@674,"(14) 2 X 2 COLORED SL
  IDE";
1820 PRINT@738,"(15) ---- NOT YET ASSIG
  NED --- ";:PRINT@802, "(16) ---- NOT
  YET ASSIGNED ---";
1830 PRINT:PRINTTAB(8)"CHOOSE.....":GOS
  UB 90:ZZ$=Z$:GOSUB 90:ZZ$=ZZ$+Z$:Z=VA
  L(ZZ$)
1840 '-----SPECS. FOR PRE-FORMATTED LAB
  ELS-----
1850 IF Z=1 THEN A$="X":CS=54:LS=4:LI=8
  :ZB$="CASSETTE MAIN LABEL":GOTO 1180

1860 IF Z=2 THEN A$="X":CS=54:LS=3:LI=8
  :ZB$="CASSETTE BOTTOM LABEL":GOTO 118
  0
1870 IF Z=3 THEN A$="S":CS=40:LS=3:LI=6
  :ZB$="CASSETTE BOX SPINE":GOTO 1180
1880 IF Z=4 THEN A$="W":CS=20:LS=3:LI=6
  :ZB$="CASSETTE BOX SPINE - WIDE":GOTO
   1180
1890 IF Z=5 THEN A$="X":CS=15:LS=6:LI=8
  :ZB$="WAFER FRONT - SQUARE":GOTO 1180

1900 IF Z=6 THEN A$="X":CS=41:LS=8:LI=8
  :ZB$="WAFER BACK - RECTANGLE":GOTO 11
  80
1910 IF Z=7 THEN A$="S":CS=22:LS=1:LI=6
  :ZB$="WAFER PROTECTOR":GOTO 1180
1920 IF Z=8 THEN A$="S":CS=32:LS=4:LI=6
  :ZB$="ADDRESS LABEL":GOTO 1180
1930 IF Z=9 THEN A$="S":CS=54:LS=4:LI=6
  :ZB$="VHS SPINE": GOTO 1180
1940 IF Z=10 THEN A$="W":CS=28:LS=4:LI=
  6:ZB$="VHS CASSETTE - SPINE (WIDE)":G
  OTO 1180
```

INTRODUCING

THE MULTI-USE CHECK REGISTER ACCOUNTING SYSTEM
(It's completely user-oriented and menu-driven)

CHECK THESE COMPARISONS!

	Acorn Money Manager Version 3.5	Tandy Checkwriter 80	The Business Division MAXI CRAS	Versaledger #0111582
PRINTING				
Print continuous-form checks?	No	Yes	Yes	Yes
Print single checks?	No	Yes	Yes	Yes
Check alignment test?	No	Yes	Yes	No
Print check stubs?	No	No	Yes	Yes with limited information
Print check register statement?	Yes	Yes	Yes	Yes
Print bank statement reconciliation?	Yes	Yes	Yes	Yes
Print income and expense subtotals?	Expense only	No	Yes	No
Print individual account statements?	Yes	No	Yes	Yes
Print check register notes?	No	No	Yes	No
Print account distribution statement?	No	No	Yes	No
Payee Address file (for automatic printing of address on checks)?	No	Yes	Yes	No
Number of payees	N/A	75	40 Mod I 75 Mod III	N/A
Number of lines in payee address	N/A	3	4	N/A
TRANSACTION ENTRY				
Check-In Pay-Out file (for automatic printing of what check is for on checks)?	No	Payee only	Yes	No
Number of payees/payors in file	N/A	75	40 Mod I 75 Mod III	N/A
Distribute transactions over multiple accounts	Difficult	Difficult	Easy	Easy
Flag tax-deductible items in any account?	Yes	No	Yes	Yes
User-friendliness	Good	Excellent	Excellent	Fair
Protection against user error	Good	Excellent	Excellent	Fair
Allow entry of manually written checks?	Yes	Yes	Yes	Yes
Ease of finding checkbook balance	Easy	Difficult	Easy	Easy
MISCELLANEOUS				
Supplied with DOS?	No	TRSDOS	TDOS	Mod I TDOS Mod III No
System	Mod I and Mod III	Mod III only	Mod I and Mod III	Separate Mod I or Mod III
Accept lower-case commands?	No	No	Yes	No
Accept lower-case text?	No	Yes	Yes	Yes
Self-prompting?	Yes	Mostly	Yes	Mostly
Swapping of program diskettes required?	Yes	No	Mod I Yes Mod III No	No
VISICALC(tm)-compatible data?	No	No	Yes	No
Documentation quality	Fair	Excellent	Excellent	Fair
CAPACITIES				
Maximum amount per transaction	99999.99	99999.99	99999.99	Not listed in manufacturer's documentation
Maximum balance	99999.99	99999.99	99999.99	Not listed in manufacturer's documentation
Limit on deposits per day?	No	Yes — 1	No	No
Number of income accounts	1	1	Up to 223 total income and expense	Not listed in manufacturer's documentation
Number of expense accounts	Up to 99	Up to 30	Up to 223 total income and expense	Not listed in manufacturer's documentation
Limits on transactions	Mod I 100/mo Mod III 250/mo	2500/yr	Unlimited	Mod I 300/mo Mod III 2400/mo
Number of bank accounts	Unlimited - no interaction	9 - with interaction	Unlimited no interaction	Unlimited no interaction

MAXI CRAS

The Last Check Register Accounting System You'll Ever Need

MAXI CRAS is the first and last Check Register Accounting System you'll ever need. Strong statement? Check out these features!

- Write checks by hand, or print automatically on single or continuous form checks. Alignment test makes sure each check is printed perfectly!
- Data entry routine second to none — saves time AND eliminates errors. You don't need to know anything about computers to use MAXI CRAS.
- The widest variety of reports available — complete check register, income and expense sub-totals, bank statement reconciliation, list of check register notes, and an account distribution statement.
- For even more detailed analysis, data is readable by VISICALC(TM). Compare!
- Checkbook balance is constantly updated and instantly accessible. No more embarrassing overdrafts!
- Write as many checks per month as you like. No limits like other systems!
- Handle up to 223 separate income **and** expense accounts. Most other systems only allow ONE income account.
- Assign a check or a deposit to a single account, or distribute over multiple accounts. Indispensible if you use charge cards!
- Check address data base stores up to 40 addresses and automatically prints them on checks.
- Fast and easy bank statement reconciliation. Compare!

MAXI CRAS MEANS BUSINESS

MAXI CRAS (short for MAXI Check Register Accounting System) is THE computerized check writing and recording system for small business or personal use. But don't take our word for it! Compare it with any other system available for the TRS-80. Read through our manual, and look at the printed reports that MAXI CRAS produces. Then talk to somebody who uses it. We're sure you'll agree — MAXI CRAS is the most versatile system available!

Don't Know Anything About Computers?

Too many of the computer programs sold today need a computer programmer to understand them. Not MAXI CRAS! Even with all its versatility, MAXI CRAS is so easy to use that we recommend it as an ideal program for new computer owners. And if account books and ledgers scare you off, take heart — our system is so much fun to use that you'll enjoy keeping up to date!

Muzzle A Budget Biter

MAXI CRAS can help you get a muzzle on hidden budget biters. You'll know exactly what you're earning and where it's going, thanks to the remarkable range of reports it can print. A MAXI CRAS exclusive!

The best news of all comes at tax preparation time. MAXI CRAS delivers the information you need in the format required for income tax preparation, uncovering deductible expenditures you might otherwise miss.

MAXI CRAS. An indispensible tool for managing money most effectively.

Use MAXI CRAS with your TRS-80 Model I or III. Requires 48K RAM, two drives, and an 80-column printer.
MAXI CRAS comes with complete operating instructions, sample printouts, and TDOS, a special version of the DOSPLUS operating system.

TRS-80 DISK Model 1&3 **012-0153** **$99.95**

TO ORDER, CALL TOLL FREE 800-327-7172 IN FLORIDA (305) 830-8194

A Division of Scott Adams, Inc.

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

PRICES SUBJECT TO CHANGE WITHOUT NOTICE

If You Don't Have Radio Shack's TRS-80® VIDE(

That's because our exclusive TRS-80 VIDEOTEX keeps you on top of what's happening. Now your personal computer or TV can bring up-to-the-minute news and special information right into your home or office!

Read the News Before It's Printed! With Radio Shack's TRS-80 VIDEOTEX, you can call the CompuServe® Information Service and get late breaking headlines straight from the Associated Press newswire. Or read special "electronic editions" of *The New York Times, The Washington Post* and other major newspapers.

Your Personal Reference Library. You can even access various government publications, magazine articles, and historical information and updates on over 32,000 stocks, bonds and securities!

Enter the Age of Electronic Mail. And keep in touch by exchanging "electronic messages" with other CompuServe subscribers!

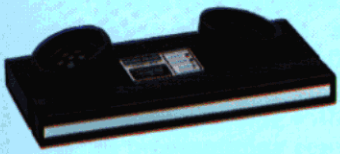
Wall Street Hotline. TRS-80 VIDEOTEX also provides you with a "direct line" to Wall Street. A call to Dow Jones News Retrieval® brings you stock quotes—as recent as 15 minutes—on over 6000 companies listed on the four major U.S. stock exchanges! And there are financial performance statistics—including stock price, volume, and fundamental financial indicators—on over 3200 companies and industries. You can also review articles from *The Wall Street Journal* and *Barron's*, as well as up-to-the-minute news. Stay on top of things with the most authoritative business and financial database in the world!

Only $29.95! TRS-80 VIDEOTEX software works with any TRS-80 desktop computer (and many other systems) equipped with a telephone coupler or modem. And we give you a choice of three affordable Radio Shack modems.

DEOTEX, You're Missing More Than You Know!

Programmable Direct-Connect Modem II. Automatically dials, answers, even hangs up your phone! Connects directly to phone line. #26-1173, $249.

Direct-Connect Modem I. Connect directly to the phone line and your TRS-80 computer for easy access to information. #26-1172, $149.



Telephone Interface II. Dial the number and place phone handset on interface. #26-1171, $199.

Or Use Your TV! The $399 TRS-80 VIDEOTEX terminal (#26-5000), shown above, features a typewriter-style keyboard and lets you access information services using your TV and telephone.

And That's Not All! TRS-80 VIDEOTEX includes a free hour of service on both the CompuServe and Dow Jones networks. After that, you'll be billed at their low hourly rates.

Find Out What's Really New. Get a no obligation TRS-80 VIDEOTEX demonstration at over 6200 Radio Shack stores and dealers, and 225 Radio Shack Computer Centers nationwide.

Radio Shack®
The biggest name in little computers™
A DIVISION OF TANDY CORPORATION

Send me your free TRS-80 Computer Catalog.
Mail To: Radio Shack, Dept. 83-A-38
1300 One Tandy Center, Fort Worth, Texas 76102

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

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

```
1950 IF Z=14 THEN A$="X":CS=30:LS=3:LI=
  8:ZB$="2 X 2 COLORED SLIDE - WIDE": G
  OTO 1180
1960 IF Z=11 THEN A$="S":CS=29:LS=7:LI=
  6:ZB$="VHS CASSETTE (NORMAL)":GOTO 11
  80
1970 IF Z=12 THEN A$="X":CS=48:LS=9:LI=
  8:ZB$="VHS CASSETTE (COMPRESSED)":GOT
  O 1180
1980 IF Z=13 THEN A$="S":CS=18:LS=6:LI=
  6:ZB$="PARTS CABINET DRAWER":GOTO 118
  0
1990 IF Z=15 THEN 2000
2000 GOSUB 60:ZT$="----- NOT ASSIGNED Y
  ET -----":GOSUB 100:ZS=3:GOSUB 70:GOT
  O 1740
2010 '-----SAVE LABEL TO WAFER-----
2020 ZP=2:GOSUB 60:PRINT:GOSUB 660
2030 GOSUB 130:INPUT "SAVE AS WHAT FILE
   NUMBER";F
2040 PRINT:PRINT "LABEL WILL BE SAVED A
  S FILE NO."F"ON NEXT CUE":PRINT:GOSUB
   80
2050 @OPEN F:PRINT "SAVING LABEL FORMAT
  ":PRINT
2060 @PRINT A$,LI,CS,LS,ZB$,ZE
2070 PRINT "SAVING LABEL TEXT"
2080 FOR L=1 TO LS:@PRINT L$(L):NEXT:@C
  LOSE
2090 PRINT:PRINT "THIS LABEL STORED AS
  FILE NO."F
2100 PRINT:GOSUB 80:GOTO 1590
2110 '-----LOAD LABEL FROM WAFER-----
2120 ZP=2:GOSUB 60:PRINT:GOSUB 660:PRIN
  T:GOSUB 130:INPUT"ENTER NUMBER OF FIL
  E DESIRED";F:PRINT:GOSUB 80
2130 @OPEN F:PRINT "LOADING LABEL FORMA
  T"
2140 @INPUT A$,LI,CS,LS,ZB$,ZE
2150 PRINT:PRINT "LOADING LABEL TEXT"
2160 FOR L=1 TO LS:@INPUT L$(L):NEXT:@C
  LOSE:GOTO1670
2170 RESUME 2180
2180 CLS:ZT$="I/O ROUTINE NOT IN PLACE"
  :GOSUB 100:ZS=3:GOSUB70:GOTO 1590
2190 B1$=CHR$(156)+STRING$(CS,140)+CHR$
  (172)
2200 B3$=STRING$(CS+2,131)
2210 FOR L=1 TO LS:F=CS-LEN(L$(L))
2220 B$(L)=CHR$(149)+L$(L)+STRING$(F,"
  ")+CHR$(170):NEXT L
2230 GOTO 2300
2240 '-----FIXINS' FOR WIDE BORDER-----
2250 B1$=CHR$(191)+STRING$(CS+2,143)+CH
  R$(191)
2260 B3$=STRING$(CS+4,143)
2270 FOR L=1 TO LS:F=CS-LEN(L$(L))
2280 B$(L)=CHR$(191)+" "+L$(L)+STRING$(
  F+1," ")+CHR$(191):NEXT L
2290 '-----PRINT LABEL WITH BORDER-----

2300 LPRINT " "
2310 FOR C=1 TO CX:LPRINTTAB(T) B1$
2320 FOR L= 1 TO LS:LPRINTTAB(T)B$(L):N
  EXT L
2330 LPRINTTAB(T) B3$:GOSUB 120:NEXT C:
  T=0:GOTO1590
2340 '-----CERTIFY WAFER-----
2350 @NEW:GOTO 1720
2360 RESUME 2370
2370 CLS:PRINT@460,"NO REFLECTIVE STICK
  ER":ZS=2:GOSUB 70:GOTO 1720
```

Change lines for tape version of X-label

CHANGE LINE NUMBER 10 TO:

```
10 CLEAR 1700:DEFINT M,Z: GOTO 2170
```

OMIT LINE NUMBER 130

CHANGE LINES 650 AND 660 TO:

```
650 '---- SUB #660: TAPE ABOUT TO START
   ----
660 ZP=2:GOSUB 60:PRINT"HAVE TAPE CORRE
  CTLY POSITIONED":PRINT:PRINT"TAPE DRI
  VE WILL START ON NEXT CUE":RETURN
```

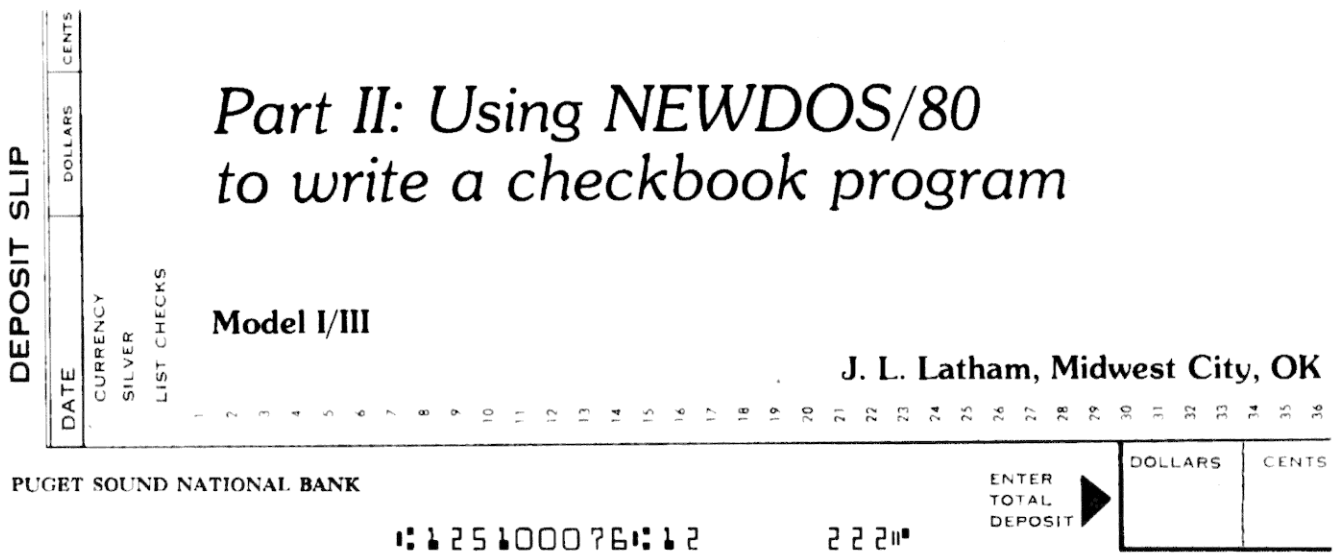
OMIT LINE NUMBER 670

NOW DELETE LINES 2010 THROUGH 2180 AND ADD THESE LINES

```
2010 '-----SAVE LABEL TO TAPE -----
2020 GOSUB 660:PRINT:GOSUB80
2030 PRINT:PRINT"SAVING FORMAT"
2040 PRINT#-1,A$,LI,CS,LS,ZB$,ZE
2050 PRINT:PRINT"SAVING TEXT"
2060 PRINT#-1,L$(1),L$(2),L$(3):PRINT#-
  1,L$(4),L$(5),L$(6):PRINT#-1,L$(7),L$
  (8),L$(9)
2070 PRINT:PRINT"SAVE IS COMPLETE":GOSU
  B70:GOTO1590
2080 '-----LOAD LABEL FROM TAPE-----
2090 GOSUB660:PRINT:GOSUB80:PRINT:PRINT
  "LOADING FORMAT"
2100 INPUT#-1,A$,LI,CS,LS,ZB$,ZE
2110 PRINT:PRINT"LOADING TEXT"
2120 INPUT#-1,L$(1),L$(2),L$(3):INPUT#-
  1,L$(4),L$(5),L$(6):INPUT#-1, L$(7),L
  $(8),L$(9)
2130 GOTO 1670
```

ALSO OMIT LINES 2350 THROUGH 2370 ■

Checkbook



Part II: Using NEWDOS/80 to write a checkbook program

Model I/III

J. L. Latham, Midwest City, OK

In the first part of this article, we discussed how to use the MERGE, DELETE, CMD"O", OPEN "O" and RUN"filename," V commands as they are used in the NEWDOS/80 (version 2.0) Disk Operating System (DOS). You were also shown where to get a page or two of "free" reserved memory, how to store dates as integer values and later retrieve them from memory, how to store dates as integer values and later retrieve them and turn them into readable strings, and even how to keep a nonexistent printer from ruining your week.

Last time out, we also started making plans for, and putting together, some program modules to use with a checkbook management program package. I call it a package rather than a program, because it will actually be a group of programs performing specified functions to obtain a single result. That result will be an easy-to-use package for keeping your checkbook balanced, checking on where your money is going, and perhaps even where it is coming from. I repeat that this program is not the main purpose of this series. That purpose is to provide you with a better understanding of certain commands available with Apparat's NEWDOS/80 version 2.0 which allow more efficient use of disk space and memory allocation.

Before continuing with the discussion of the program, a few programming conventions must be agreed upon. The file that we will keep outstanding transactions (checks, deposits, etc.) in shall always be called OUTSTNDG/DAT. That file, and any other data files, shall be understood to reside on drive one of a two-drive system, while our system diskette with the actual program package will be on drive zero (assuming a Model I TRS-80). To give some order to our other files, we shall call them by the month and year of the bank statement and add the extension /DAT. So, if your statement came out on the 15th of June, 1981, that file would be JUN81/DAT. Our program will supply us with the /DAT extension, and will even point the files to drive one so that we won't have to hassle with all that typing. Readers with only one drive will have to make appropriate changes to the program.

The reason we had to agree beforehand on some of our filenames is that we are going to build another program module that will be used by our package which will contain all of our DATA statements used by several of the main runtime programs. This module will require periodic updating as you add files to your computerized checkbook. This updating is made easier by having only one file to update instead of five. Another part of our data base that we need to think about is the description of our transactions, because the written descriptions are going to be set up during initialization.

By using no more than 12 characters to describe an item, we can get four descriptions per line on the video screen. (With an additional one line for requesting the number that corresponds to the description, we can get up to sixty descriptions on the screen at one time.) That should cover most needs. I would suggest that in your DATA statements for these descriptions, you list descriptions of bank charges first, then describe possible sources of deposits, and list the different categories for your checks. The very first item in this list of data statements is an exception. It should be "BAL FWD." Although the program allows up to 60

descriptions, I suggest that no more than 36 descriptions be used to prevent major "damage" to your screen layout. If you use more, you may want to modify the program to only display part of the array at one time.

You can take a look at Listing 1 now. Notice that the line numbers begin at 830 and run to 980 with a break, and then start again at 1000 and run to 1080. These should all be typed in with the program we called MENU last month. This is the complete initiation routine for our program. It will only occur the first time we run the program, if some disaster happens to our filename in use (AF), or to the machine language routine we stuffed into disk I/O buffer #2. This initiation takes about 60 seconds to complete. By going through this long wait one time, we set up for a faster run in the rest of the program. Generally, the order of the variables has been picked by their rate of use in the program so that the computer can find the most often called variables quickest.

During the initiation, you will see the contents of lines 1010 through 1080 displayed on the screen from the bottom up. These values are read from the screen in lines 930 through 970 and placed into array A(n) for faster printing later. The display will always fill the portion of the screen you see filled during initiation. It will overwrite anything in this area, but won't scroll the page until a response to the input request has been made. By reading this display into four strings, we only have to print four strings to get it onto the screen later instead of having to print up to 61 short strings. The variable PA keeps up with where to start printing the four strings on the screen.

Now, to build the data base for the files we will be using. Take at least a quick look at Listing 2. It is the one with line numbers starting at 12000. Notice that the file named "OUTSTNDG/DAT" is on a line by itself. This is so that we can use one program in the place of two with a judicious choice of the DELETE command later. The data statements in lines 12010 through 12030 are a sample of two years of checkbook-keeping as it might look. The data statement in line 12090 must be "LAST" so that the program will know when to quit looking for files without creating an error.

When you first start your own files, you will probably only have one or two monthly files to include in your own data statements. You could actually run the program with only two files (one containing outstanding transactions to be cancelled each month, and an ever lengthening file that would contain all the transactions that have been cancelled). If you elected to do it that way, I would recommend typing these data statements with filenames directly into the program, rather than placing such a short file on disk. Feel free to be flexible. I prefer my method because it lets me run quick checks against the bank for any given month, and still get overall data from the composite file we have yet to create. Save Listing 2 on disk with the name "FILES."

While we are on the subject of creating things, we can now get into Listing 3. Type this program in and SAVE it as CREATOR. CREATOR allows you to either begin a new file, or to add transactions to the end of an already existing file. Notice that we start by merging in the file saved as DATEIN, and then cause (the now unwanted) lines of code to be thrown out with the DELETE command. Notice also, that the request for a filename only requires the filename itself, not the extension /DAT nor the drive number. These items are furnished by the program after verifying that the filename entered is somewhat valid. The program prompts you for each input and will not put the data out to disk until you tell it that the data is acceptable. There is no way to correct data that you output to disk in this module, but we will take care of that next month. Usually, the only file you will be working with (using this module) will be the OUTSTNDG/DAT file.

Listing 4 should be SAVEd as CANCELS. This module always uses the OUTSTNDG/DAT file for its input. It asks first for the filename to be used for saving the cancelled items, then reads in the entire OUTSTNDG/DAT file, and turns all items into their absolute value, while saving their original status in array SZ(n). All the items are sorted with the CMD"O" function, and finally, each item is presented and you are asked if it was cancelled with the last statement. If your reply is N, no action is taken except to get the next item in the list. If the reply is Y, the item is placed into the output file buffer and marked as removed by filling the array items with zeros. If your reply is E, the presentation is stopped. The items in OUTSTNDG/DAT are resorted (to get rid of the zeroed items) by sorting in inverse order (checking for zero entries), then sorting in ascending order only those entries with values other than zero. Finally, the OUTSTNDG/DAT file is written back to the disk in its updated form. You are presented with the balance in the file and sent back to the MENU program.

At this point in our studies, it is necessary to tell you the format into which our data will be stored on the disk. Last month, I told you that all variable data except for the amount of the transactions would be stored as integers, and that the amount would be stored in double precision format. All that is true, but there is one more step. There are four pieces of data stored: amount, date, transaction description code, and a type code. The type code tells us if the item is a deposit. The type code for a check is simply the check number +1. The type code for a service charge or other bank debit is 32767, and the type code for a credit (deposit) is a 1. To distinguish between credits and debits, all debits have their value, type code, and date converted to a negative value prior to storage. The transaction description code is always stored as a positive number.

The reason that the type code for checks is the check number +1 in negative format, is that quite often we have to convert all numbers back to their positive format to work with them, and check number one would look like a deposit. Rather than trying to keep up with one special case, I just made a general rule for all checks. So, check number 301 would be marked in the files as -302.

The four arrays that hold the data are: DA(n) which holds the amount, ZD(n) which holds the converted date code, WT(n) which holds the transaction description code. Array SZ(n) is used to keep up with the original

sign of a given array item.

Back to Listing 2 now. This module performs the dual purpose of either creating an entirely new file, or extending an already existing file. This is accomplished with the use of the OPEN"E" file opening format of NEWDOS/80. Take a look at line 20 of the listing. There is our first example of MERGE. The module DATEIN is brought off the disk and attached to the file in RAM. Because the DATEIN routine is used in several of the programs, it saves us disk space to have it occupy its own disk file rather than being part of three or four other files.

Line 30 shows the first example of the DELETE command. We get rid of the initialization lines because we no longer need them and may be able to use the extra memory for overhead in our program. If this were a standalone program, you could do your entire initialization, then delete it, to get back the memory before actually starting the main part of the program. Take special notice of the fact that the DELETE command can even delete itself from the program!

Next month, you will get the final program listings. The real heart of this program will be presented there. We will take another look at MERGE and DELETE, and really put CMD"O" to work. If you want to start building up your data files from your old bank statements, feel free to do so. If you get some bad data onto disk, don't worry. Next time, you get the program that does corrections and searching, and the program that builds the composite files I spoke of in the first installment. Wait until you see how DELETE, combined with RUN"filename,"V makes four programs out of two!!

Listing 1
Checkbook

```
830 X=1500:Z1=12:DIMWT(X),DA(X),ZD(X),T
  Y(X),SZ(X),WM(Z1),AM(Z1),A(3),BT(60):
  Z0=1:Z2=31:Z3=30:PM=0:PH=0:PL=0:PC=0:
  PB=0:PT=0:WM=0:WP=0:XP=0:WD=0:WY=0:YR
  =0:ZY=0:WF=0:ZD=0:DB=0:DT=0
840 A="":AD="":AF="":AG="":A9=CHR$(34):
  PRINT"READING DATE CONVERSION DATA"
850 FORX=Z0TOZ1:READWM(X),AM(X):NEXTX:C
  LS:PRINT@960,"";:Y=0
860 READBT(Y):IFBT(Y)="Z"THENY=Y-1:GOTO
  890
870 Y=Y+1:IFY>60THENY=60:GOTO890
880 GOTO860
890 YB=Y:FORX=0TOY:IFBT(X)="NA"THEN910
900 PRINTX;BT(X),
910 NEXTX:IFPEEK(&H4021)*256+PEEK(&H402
  0)<>16320THENPRINT
920 PRINT@960,"ENTER TYPE CODE FOR ITEM
   ";CHR$(191);
930 FORX=&H3C00TO&H3FC0STEP&H40:IFPEEK(
  X+1)=48THENY=X:PA=X-15360:X=&H3FC0
940 NEXTX:PM=0
950 PC=32:FORX=YTOY+254:IFPEEK(X)=191TH
  ENPM=3:X=Y+254:GOTO970
```

```
960 A(PM)=A(PM)+CHR$(PEEK(X))
970 NEXTX:PM=PM+1:Y=X:IFPM<4THEN950
980 ONERRORGOTO0:X=0:Y=0:PC=0:PM=0:GOTO
  20
1000 DATA31,JAN,29,FEB,31,MAR,30,APR,31
  ,MAY,30,JUN,31,JUL,31,AUG,30,SEP,31,O
  CT,30,NOV,31,DEC
1010 DATA"BAL FWD","SVC CHG","DRAFT","S
  PEC CHARGE","**"
1020 DATA"MY PAY","SPOUSE PAY","OTHER I
  NC","**","HOUSE PMT","HOUSE INSUR","H
  OME IMPRV","HOME REPAIR","HOME ELEC",
  "HOME GAS","HOME WATER","HOME PHONE",
  "LIFE INSUR","CAR #1 EXP","CAR #2 EXP
  "
1030 DATA"VISA","MASTER CARD","AMER EXP
  R","SEARS","PENNYS","WARDS","EDUCATIO
  N","FOOD","CLOTHES","VACATION","CASH"

1040 DATA"NA","NA","NA","NA","NA","NA",
  "NA"
1050 DATA"NA","NA","NA","NA","NA","NA",
  "NA"
1060 DATA"NA","NA","NA","NA","NA","NA",
  "NA"
1070 DATA"NA","NA","NA","NA","NA","NA",
  "NA"
1080 DATA"VOID","OTHER","Z"
```

Listing 2
Checkbook

```
12000 PRINT@PA,"";:FORP=0TO3:PRINTA(P);
  :NEXTP:INPUTTY:RETURN
12010 DATA"OUTSTNDG/DAT"
12015 REM LINES 12020 THROUGH 12040 ARE
   FOR EXAMPLE ONLY. INITIALLY YOUR FIL
  ES WILL BE NULL, TO BE ADDED TO AS YO
  U CREATE EACH MONTHLY FILE (USUALLY W
  ITH THE CANCEL MODULE).
12020 DATA"JAN79/DAT","FEB79/DAT","MAR7
  9/DAT","APR79/DAT","MAY79/DAT","JUN79
  /DAT","JUL79/DAT","AUG79/DAT","SEP79/
  DAT","OCT79/DAT","NOV79/DAT","DEC79/D
  AT"
12030 DATA"JAN80/DAT","FEB80/DAT","MAR8
  0/DAT","APR80/DAT","MAY80/DAT","JUN80
  /DAT","JUL80/DAT","AUG80/DAT","SEP80/
  DAT","OCT80/DAT","NOV80/DAT","DEC80/D
  AT"
12040 DATA"JAN81/DAT"
12045 REM
***** NOTICE THE JUMP IN LINE NUMBERS *
  ****
LINES 12000, 12010T, AND 12090 ARE MAND
  ATORY.
LINES BETWEEN 12010 AND 12090 ARE ALL A
  VAILABLE DATA.
12090 DATA"LAST"
```

Listing 3
Checkbook

```
2000 REM        PROPERTY OF:
JERRY L. LATHAM
1409 EVERGREEN CIRCLE
MIDWEST CITY, OK  73110

2010 CLS:PRINTTAB(18)"CHECKBOOK 1.0":PR
  INTTAB(21)"CREATOR":PRINT:PRINT"INITI
  ATING":IFPEEK(&H6818)<>205THENRUN"MEN
  U"
2020 PRINT"MERGING DATEIN":MERGE"DATEIN
  "
2030 PRINT"MERGING FILES":MERGE"FILES"
2040 CMD"F",DELETE12010-12090
2050 CMD"F",DELETE2010-2050
2060 RESTORE
2070 IFAF=""THEN2090ELSEPRINT"FILE IN U
  SE IS ";AF:PRINT"EXTEND THIS FILE OR
  CREATE/EXTEND ANOTHER (Y=THIS N=ANOTH
  ER) ";
2080 A=INKEY$:IFA=""THEN2080ELSEP=USR0(
  VARPTR(A)):IFA<>"Y"ANDA<>"N"THEN2080E
  LSEPRINTA:IFA="Y"THENGOTO2120
2090 AF="":PRINT:INPUT"ENTER FILENAME F
  OR OUTPUT FILE (/DAT:1 EXTENSION WILL
   BE ADDED)
FILENAME ";AF:P=USR0(VARPTR(AF))
2100 IFLEN(AF)<1ORLEN(AF)>8ORLEFT$(AF,1
  )<"A"ORLEFT$(A1,1)>"Z"THENPRINT"INVAL
  ID FILENAME, RE-ENTER.":GOTO2090
2110 AF=AF+"/DAT:1":PRINT"INITIATION CO
  MPLETE. READY FOR PROCESSING.":PRINT
2120 PRINT"OPENING FILE ";AF:CLOSE:OPEN
  "E",1,AF
2130 CLS:INPUT"ENTER CHECK NUMBER OR IT
  EM TYPE (SC/DEP/END)";A1:P=USR0(VARPT
  R(A1)):IFA1="END"THEN2260
2140 IFA1="SC"THEN2190ELSEIFA1="DEP"THE
  N2210ELSE2150
2150 IFVAL(A1)<0THEN2130
2160 WT=(VAL(A1)+1)*-1
2170 INPUT"ENTER DATE OF CHECK";AD:GOSU
  B10000:ZD=ZD*-1
2180 INPUT"ENTER AMOUNT OF CHECK";DA:DA
  =DA*-1:GOTO2230
2190 WT=-32767:INPUT"ENTER DATE OF SERV
  ICE CHARGE ";AD:GOSUB10000:ZD=ZD*-1
2200 INPUT"ENTER AMOUNT OF SERVICE CHAR
  GE";DA:DA=DA*-1:GOTO2230
2210 WT=1:INPUT"ENTER DATE OF DEPOSIT";
  AD:GOSUB10000
2220 INPUT"ENTER AMOUNT OF DEPOSIT";DA
2230 GOSUB12000:PRINT:PRINT"DATA IS AS
  FOLLOWS:":PRINT"ITEM: ";A1:PRINT"DATE
   ";AD:PRINTUSING"AMOUNT   $##,###.##"
  ;ABS(DA):PRINT"DESCRIPTION: ";BT(TY):
  PRINT"IS THIS CORRECT (Y/N)? ";
```

```
2240 A=INKEY$:IFA=""THEN2240ELSEP=USR0(
  VARPTR(A)):IFA<>"Y"ANDA<>"N"THEN2240E
  LSEPRINTA:IFA<>"Y"THENCLS:PRINT"RE-EN
  TER DATA FOR ITEM ";A1:GOTO2130
2250 PRINT#1,WT;DA;ZD;TY;:CLS:GOTO2130
2260 CLOSE1:PRINT"FILE CLOSED. RUN ENDE
  D. RETURNING TO MENU.":RUN"MENU",V
```

Listing 4
Checkbook

```
3000 REM        PROPERTY OF:
JERRY L. LATHAM
1409 EVERGREEN CIRCLE
MIDWEST CITY, OK 73110

3010 CLS:PRINTTAB(18)"CHECKBOOK 1.0":PR
  INTTAB(21)"CANCELS":PRINT:PRINT"INITI
  ATING":IFPEEK(&H6818)<>205THENRUN"MEN
  U"
3020 PRINT"MERGING DATEIN":MERGE"DATEIN
  "
3030 PRINT"MERGING DATEOUT":MERGE"DATEO
  UT"
3040 IFPEEK(&H37E8)=255THENA="N":GOTO30
  90
3050 PRINT"DO YOU REQUIRE A HARDCOPY OF
   THE RESULTS (Y/N)? ";:GOSUB3470:IFA=
  "N"THEN3110
3060 PRINT"CHECKING PRINTER STATUS":MER
  GE"PRINTCK"
3070 GOSUB11000
3080 CMD"F",DELETE11000-11080
3090 IFA="N"THENCMD"F",DELETE3370-3380
3100 IFA="N"THENCMD"F",DELETE3510
3110 CMD"F",DELETE3010-3110
3120 IFAF=""THEN3140ELSEPRINT:PRINT"CUR
  RENT FILE IN USE IS ";AF;"
DO YOU WANT TO CONTINUE USING (EXTEND)
  THIS FILE,
OR DO YOU WANT TO CREATE A NEW FILE ?
Y=EXTEND CURRENT FILE,  N=CREATE A NEW
  FILE ";
3130 GOSUB3470:IFA="Y"THEN3170
3140 AF="":PRINT"INPUT FILE IS OUTSTNDG
  /DAT:1
ENTER NAME FOR OUTPUT FILE...THE /DAT:1
   EXTENSION WILL BE
ADDED AUTOMATICALLY.  ENTER FILENAME NO
  W.. ";:INPUTAF:P=USR0(VARPTR(AF))
3150 IFLEN(AF)<1ORLEN(AF)>8ORLEFT$(AF,1
  )<"A"ORLEFT$(AF,1)>"Z"THENPRINT"INVAL
  ID FILENAME, RE-ENTER.":GOTO3140
```

```
3160 AF=AF+"/DAT:1"
3170 X=0:CLOSE:OPEN"I",1,"OUTSTNDG/DAT:
  1":CLS:PRINT"LOADING DATA ITEM      F
  ROM OUTSTNDG/DAT:1";
3180 IFLOC(1)$=-1THEN3200ELSEPRINT@18,X
  ;
3190 INPUT#1,WT(X),DA(X),ZD(X),TY(X):X=
  X+1:GOTO3180
3200 CLOSE:X=X-1:Y2=X:PRINT:PRINTX+1"IT
  EMS LOADED SUCCESSFULLY. FILE CLOSED.
  ":PRINT"PREPARING DATA FOR GENERAL SE
  ARCH. STAND-BY."
3210 FORY=0TOX:PRINT@62,"**";:SZ(Y)=SGN
  (WT(Y)):WT(Y)=ABS(WT(Y)):DA(Y)=ABS(DA
  (Y)):ZD(Y)=ABS(ZD(Y)):PRINT@62,"* ";:
  NEXTY:CLS:PRINT"PRE-SEARCH ORGANIZATI
  ON COMPLETED."
3220 PRINT"SORTING PRIOR TO BEGINNING P
  RESENTATION.":CMD"O",X+1,ZD(0),WT(0),
  DA(0),TY(0),SZ(0):PRINT"PRE-PRESENTAT
  ION SORT COMPLETED."
3230 CLS:PRINT"YOU WILL BE PRESENTED WI
  TH EACH ITEM OF THE INPUT FILE, ONE A
  T ATIME, AND ASKED IF THAT ITEM WAS C
  ANCELLED ON THE STATEMENT FROMTHE BAN
  K.  YOU MUST REPLY Y OR N (YES OR NO)
   TO ALL ITEMS, EVEN THE SERVICE CHARG
  ES."
3240 PRINT"IF YOU REPLY  Y  FOR AN ITEM
  , IT WILL BE REMOVED FROM THE INPUT F
  ILE AND PLACED INTO THE OUTPUT FILE Y
  OU HAVE SPECIFIED.  THE  INPUT FILE
  IS ALWAYS "A9"OUTSTNDG/DAT:1"A9".  TH
  E OUTPUT FILE IS THAT WHICH YOU SPECI
  FIED EARLIER."
3250 PRINT"YOU MAY ALSO ENTER  E  TO QU
  IT WITHOUT GOING THROUGH THE
ENTIRE FILE.";:GOSUB3530:A="":CLS
3260 CLOSE:OPEN"E",1,AF:PRINT"OUTPUT FI
  LE NOW OPEN.":PRINT
3270 FORY=0TOX
3280 PRINT@64,"TYPE/#";:PRINT@74,"DATE"
  ;:PRINT@90,"DESCRIPTION";:PRINT@110,"
  AMOUNT";
3290 WT=WT(Y):IFWT=1THENAC="DEPOSIT":GO
  TO3320
3300 IFWT=32767THENAC="SVC CHG":GOTO332
  0
3310 WT=WT-1:AC=STR$(WT)
3320 ZD=ZD(Y):ZS=ZD:GOSUB10150:ZD=ZS
3330 PRINT@128,AC;:PRINT@138,AD;:PRINT@
  154,BT(TY(Y));:PRINT@174,"";:PRINTUSI
  NG"$##,###.##";DA(Y):PRINT
3340 PRINT"IS THIS ITEM TO BE CANCELLED
    (Y/N/E)? ";:GOSUB3470:IFA="N"THEN340
  0ELSEIFA="E"THENY=X:GOTO3400
```

```
3350 INPUT"ENTER DATE ITEM CANCELLED ";
  AD:GOSUB10000:ZD=ZD*SZ(Y):DS=DA(Y)*SZ
  (Y):WT=WT(Y)*SZ(Y):TY=TY(Y)
3360 PRINT#1,WT;DS;ZD;TY;:PRINT"ITEM CA
  NCELLED, NOW REMOVING IT FROM INPUT F
  ILE"
3370 LPRINT"CANCELLED:":LPRINT"ITEM";TA
  B(10)"DATE";TAB(25);"DESCRIPTION";TAB
  (40);"AMOUNT"
3380 LPRINTAC;TAB(10);AD;TAB(25);BT(TY(
  Y));TAB(40);"";:LPRINTUSING"$##,###.#
  #";DA(Y):LPRINT
3390 DA(Y)=0:ZD(Y)=0:WT(Y)=0:TY(Y)=0:SZ
  (Y)=0:Y2=Y2-1:PRINT"ITEM REMOVED FROM
   INPUT FILE."
3400 CLS:NEXTY:PRINT"ALL ITEMS READ. NO
  W CLOSING FILES":CLOSE
3410 PRINT"CANCELLED FILE CREATED AND C
  LOSED.  DO YOU WISH TO SAVE THE NOW M
  ODIFIED INPUT FILE IN ITS CHANGED STA
  TE (USUALLY YOU SHOULD!)  REPLY Y O
  R  N ";:GOSUB3470:IFA="N"THEN3460
3420 PRINT"UPDATING STATUS OF INPUT FIL
  E OUTSTNDG/DAT:1":CMD"O",X+1,-WT(0),-
  ZD(0),-DA(0),-TY(0),-SZ(0):CMD"O",Y2+
  1,WT(0),ZD(0),DA(0),TY(0),SZ(0):FORY=
  0TOY2:WT(Y)=WT(Y)*SZ(Y):DA(Y)=DA(Y)*S
  Z(Y):ZD(Y)=ZD(Y)*SZ(Y):NEXTY
3430 PRINT"STATUS UPDATED.":PRINT"SAVIN
  G UPDATED INPUT FILE OUTSTNDG/DAT:1":
  CLOSE:OPEN"O",1,"OUTSTNDG/DAT:1"
3440 FORY=0TOY2:PRINT#1,WT(Y);DA(Y);ZD(
  Y);TY(Y);:NEXTY:CLOSE
3450 PRINT"MODIFIED INPUT FILE NOW SAVE
  D AND CLOSED.":PRINT"COMPUTING BALANC
  E FOR FILE OUTSTNDG/DAT.":AF="OUTSTND
  G/DAT":GOSUB3480:PRINT"THIS IS THE SU
  M OF ALL OUTSTANDING CHECKS & DEPOSIT
  S.":GOSUB3530
3460 PRINT"RETURNING TO MENU":RUN"MENU"
  ,V
3470 A=INKEY$:IFA=""THEN3470ELSEP=USR0(
  VARPTR(A)):IFA<>"Y"ANDA<>"N"ANDA<>"E"
  THEN3470ELSEPRINTA:RETURN
3480 DB=0:FORY=0TOX:IFTY(Y)=0THEN3500
3490 DB=DB+DA(Y)
3500 NEXTY:PRINT"BALANCE FOR THE FILE "
  ;AF;" IS ";:PRINTUSING"$##,###.##";DB

3510 LPRINT"BALANCE FOR THE FILE ";AF;"
   IS ";:LPRINTUSING"$##,###.##";DB
3520 RETURN
3530 PRINT:PRINT"PRESS <ENTER> TO CONTI
  NUE ";
3540 A=INKEY$:IFA<>CHR$(13)THEN3540ELSE
  CLS:RETURN
```

■

Storage and retrieval of data

Programming techniques and a mail list routine

Model I/III tape or disk

Terry Hazelett, Parkersburg, WV

Now that you have a computer, what are you doing with it? By now, you probably are able to find your way around the "Pyramid of Doom" without getting killed, and can play Blackjack with the best of them, but are you doing those things that a "personal computer" can do best? How about writing a spelling program that will allow the computer to flash words on the video display, or writing a program that will store and retrieve coupon data? Maybe you need a program to keep track of the results of your golf league, or any one of a hundred other jobs that require data manipulation.

I do not claim to be an expert in these matters, but I have written the three programs that I mentioned, and maybe I can pass on some tips on storing and retrieving data. First, I'll try to explain how to manipulate data in memory, then I'll show you a way to speed up cassette data flow.

You must decide what your variables will be, and use them in the most efficient manner. When you have over 900 variables to choose from, it is easy to lose track of how each variable is used. In the following examples, I am going to show you how to keep track of a mailing list. It doesn't matter if you want to keep track of your Spiderman comic collection or your checkbook— the same techniques can be used to write any program that stores and retrieves data. All examples can be used on Radio Shack computers as written, and others with very little modification.

In order to start, you need to dimension your arrays for the length of your file. But what if each file is a different length? If you set your DIM statement at 25, and the next time that you run the program you need 45, you have to modify the program before it can be run. My solution is to do the following:

```
10 CLEAR 500
30 INPUT"HOW MANY NAMES ARE ON YOUR
   LIST";M
40 DIM A1$(M), A2$(M), A3$(M), A4$(M)
```

Each time the program is run, the arrays are set at the beginning of the program. This does not mean that you

have to enter (M) names, but it does set the maximum limit of names to be entered.

Now, on to entering the actual data. Each entry must be identified to avoid confusion.

```
50 I=0   'SET COUNTER TO ZERO
60 I=I+1   'INCREMENT COUNTER
70 CLS   'CLEAR SCREEN
80 IF I=M PRINT "THIS IS YOUR LAST ENTRY!!":
   PRINT
90 INPUT"FIRST & LAST NAME"; A1$(I)
100 INPUT"STREET ADDRESS"; A2$(I)
110 INPUT"CITY & STATE (NO COMMAS)"; A3$(I)
120 INPUT"ZIP CODE"; A4$(I)
```

If you are asking yourself, "Why did he make I=0 in line 50, and then, in line 60, add one to it?", I'll tell you why. I use the letter "I" for most of my counters, and it must be set to zero before I start, or it may have a previous value left over from another routine. Also, I have allowed no commas in line 100 because a comma is a delimiter in Radio Shack BASIC. If a comma is used, it will truncate (chop off) anything after the comma. If "Parkersburg, WVa" was used in line 100, you would get an "EXTRA IGNORED" message and A3$(I) would equal "Parkersburg". On with the show.

```
130 Z$=" ": PRINT: INPUT "CORRECT (Y/N)"; Z$
140 IF Z$="Y" OR Z$="N" THEN 150 ELSE 130
150 IF Z$="N"" THEN 70
160 IF I=M THEN 1000
170 Z$=" ": PRINT: INPUT "END SESSION (Y/N)";
    Z$
180 IF Z$="Y" OR Z$="N" THEN 190 ELSE 170
190 IF Z$="N" THEN 60
200 T=I   'TOTAL DATA ITEMS ENTERED
```

Lines 140 and 180 are unique in that they only allow a "Y" or "N" before program execution can proceed. Line 130 will let you enter your information over again if you find that you have entered your information incorrectly. Line 160 will not let you exceed the limits of your arrays. These kinds of error trapping can eliminate a lot of hair pulling later on if used correctly.

Now that you have entered your data, let's do something with it. What can we do with it? We can print it out on the video display, print it out on paper, or save it to tape or disk for later recall. First, let's check out the possibility of printing out and then saving the data.

```
210 CLS
220 Z$=" ": INPUT "PRINTER OR VIDEO (P/V)"; Z$
230 IF Z$="P" OR Z$="V" THEN 240 ELSE 210
240 IF Z$="P" THEN P=1
250 CLS: PRINT "MAILING LIST": PRINT
```

```
255 IF P=1 LPRINT "MAILING LIST": LPRINT" "
260 FOR I=1 TO T
270 CLS
280 PRINT "RECORD NUMBER"; I: PRINT
290 PRINT
300 PRINT "NAME- - - - - -"; A1$(I)
305 IF P=1 LPRINT "NAME - - - - - -"; A1$(I)
310 PRINT "ADDRESS - - - - -"; A2$(I)
315 IF P=1 LPRINT "ADDRESS - - - - -"; A2$(I)
320 PRINT "CITY, STATE - - -"; A3$(I)
325 IF P=1 LPRINT "CITY, STATE - - -"; A3$(I)
330 PRINT "ZIP CODE - - - -"; A4$(I)
335 IF P=1 LPRINT "ZIP CODE - - - -"; A4$(I)
340 IF P=1 LPRINT STRING$ (2,138) 'TWO
    LINEFEEDS
350 INPUT "PRESS <ENTER> FOR NEXT
    RECORD"; Z$
360 NEXT I
```

What's going on here? I have made the program work with two options in line 220. I can send the information to the video display only if I want to, or I can send it to the line printer. Notice that if it does go to the printer, it still appears on the video display. That way, you can follow the flow of information a little easier. The rest of the routine is very easy to follow. It's a FOR...NEXT loop that prints out the data previously entered.

If you make a slight modification, this information could be printed out on labels. Change the following lines to accomplish this:

```
305 IF P=1 LPRINT A1$(I)
315 IF P=1 LPRINT A2$(I)
325 IF P=1 LPRINT A3$(I)
335 IF P=1 LPRINT A4$(I)
```

Now you have a program to print mailing labels. If you just want a tab list, delete lines 305, 315, 325, and 340, and enter the following line.

```
340 IF P=1 LPRINT A1$(I); TAB (18) A2$(I); TAB (36)
    A3$(I); TAB (54) A4$(I)
```

Now, for the final act. If we want to recall this information at a later date, we must save it on tape or disk. Enter the following:

```
400 CLS: PRINT "MINI MENU": PRINT
410 PRINT "1. SAVE TO DISK"
420 PRINT "2. SAVE TO TAPE"
430 PRINT "3. LOAD FROM DISK"
440 PRINT "4. LOAD FROM TAPE"
450 PRINT: INPUT "WHICH ONE"; Z
460 IF Z < 1 OR Z > 4 THEN 400
470 ON Z GOTO 500, 600, 700, 800
500 CLS: PRINT "DISK SAVE ROUTINE": PRINT
510 INPUT "WHAT IS THE FILENAME"; F$
520 OPEN "O", 1, F$
530 PRINT # 1, T
540 FOR I=1 TO T
550 PRINT # 1, A1$(I), A2$(I), A3$(I), A4$(I)
560 NEXT I
570 CLOSE
580 GOTO 1000
600 CLS: PRINT "TAPE SAVE ROUTINE": PRINT
610 PRINT "DATA NOW SAVING . . . ."
620 PRINT # -1, T
630 FOR I=1 TO T
640 PRINT # -1, A1$(I), A2$(I), A3$(I), A4$(I)
650 NEXT I
660 GOTO 1000
700 CLS: PRINT "DISK LOAD ROUTINE": PRINT
710 INPUT "WHAT IS THE FILENAME"; F$
720 OPEN "I", 1, F$
730 INPUT # 1, T
740 FOR I=1 TO T
750 INPUT # 1, A1$(I), A2$(I), A3$(I), A4$(I)
760 NEXT I
770 CLOSE
780 GOTO 1000
800 CLS: PRINT "TAPE LOAD ROUTINE" PRINT
810 PRINT "DATA NOW LOADING . . ."
820 INPUT # -1, T
830 FOR I=1 TO T
840 INPUT # -1, A1$(I), A2$(I), A3$(I), A4$(I)
850 NEXT I
860 GOTO 1000
```

It looks like I have put a lot of programming between line 400 and 860, but I really have only repeated myself four times. Look closely at lines 530, 620, 730, 820, and the three lines that follow each. There you will see that all I have done is this: Make sure that the computer knows how much data to read or write (PRINT # 1, T) and then run a FOR...NEXT loop to do just that. All of the rest is just trimming to make it look better. The disk version must have a filename and a buffer must be opened, but the working part is really no different from the tape version.

Now that you know how to save and load cassette data, let me show you a better way. This was explained to me by a friend and it really speeds up the time it takes to read or write a data file to tape. In the line 640 version, there are probably 60 bytes of information to be stored on tape. According to the Level II manual, you can store up to 255 bytes. This means that each time a PRINT # -1 is executed, 195 bytes of blank space is written to the tape. If you want to speed up data saves and reads by four times, do the following:

```
620 FOR I=1 TO T STEP 4
```

```
630 PRINT # -1, A1$(I), A2$(I), A3$(I), A4$(I),
    A1$(I+1), A2$(I+1), A3$(I+1), A4$(I+1), A1$(I+2),
    A2$(I+2), A3$(I+2), A4$(I+2), A1$(I+3), A2$(I+3)
    A3$(I+3), A4$(I+3)
```

Now you're cooking with gas. To prove my point, run the previous program without this modification and time it. Then change line 620 and 630 (plus 830 and 840) and time it again. You will see that it now takes much less time to accomplish the same task.

I have added several parts to this mailing list program, but it still lacks one important thing: organization! If it is run as is, things could get messy. Look at line 580 and you will see that it is telling you to GOTO a line that is not there. Line 1000 is where I will put the main menu. Enter the following:

```
45 GOTO 1000
205 GOTO 1000
370 GOTO 1000
1000 CLS: PRINT "MAIN MENU": PRINT
1010 PRINT "1. ENTER MAILING LIST"
1020 PRINT "2. PRINTOUT"
1030 PRINT "3. SAVE/LOAD DATA"
1040 PRINT: Z=0: INPUT "WHICH ONE"; Z
1050 IF Z < 1 OR Z > 3 THEN 1000
1060 ON Z GOTO 30, 210, 400
```

After all of that programming, there is only one thing wrong. It won't work. I have left this part until last, so listen up. Each piece of information for this program is entered in string form. With the TRS-80, memory space must be cleared in advance for strings (before it is used). With this program, I would estimate 60 bytes of information are used for each record. That means if I want to enter 100 mailing addresses, I must clear 6000 bytes of string space in the beginning of my program. Enter this:

```
10 CLEAR 6000 '100 Labels limit
```

Another option would be to delete line 10 and enter line 35 as this:

```
35 CLEAR M*60
```

Now it should work. What I would like you to do now, is to change this program. Make it do something different. Fix it so that you can stop part way through your mailing list, check it out on the video display, then jump back to the data entering part and start where you left off. There are a lot of improvements that can be added— more features, better error detecting routines, etc. After all of that, you might want to change it to keep an inventory of household appliances, warranty dates, purchase prices and tax information. As you can see, there is a lot that can be done, so get busy. ■

Basically BASIC

Simple screen control

All models

© 1982 T. R. Dettmann, Associate editor

A frequent question that I get from people is, "How can I control the cursor on the screen and put things where I want them?" Luckily for the TRS-80, this question has a simple answer, the PRINT@ statement.

If you look in your manual, you'll find that there's a screen map somewhere (for the Model III, it's page 235 in the BASIC language manual). This map assigns a number to each character position on the screen. You don't need the map, though. The number is really quite simple. The upper left corner on all machines is position zero. Character positions increase going to the right. When you run out of line, you simply go to the next line and continue.

On the Model I/III, there are 16 lines of 64 characters. The starting print position is at 0 (zero). The first character on the next line is 64, line 3 is 128, then 192, and so forth.

On the Model II, with 80 characters per line, the lines are 0, 80, 160 . . . The Color Computer has 32 characters per line, so the numbering goes 0, 32, 64 . . . If you want to print starting at the first character of the second line, you can simply say:

PRINT @32, "HERE'S THE SECOND LINE"

With this kind of technique, you can come back to the same place every time if you want.

The program listing illustrates still another way of working with PRINT@. This one allows you to print at any place on the screen (upper case, numbers and punctuation only). Using the arrow keys, you can move to any location on the screen.

If you look at the program, line 150 allows printing the characters we want to print and only those characters. Lines 160 through 190 look for the arrow keys and interpret them correctly. If an arrow key is found, the program picks up at line 210, where some additional processing is done to handle special positions and move our marker. If no arrow key is found, we simply ignore the key that was pressed and get another key.

Once we've found an arrow key and updated the current location of the marker, lines 210-240 check to make sure it's on the screen and, if not, to put it back there in a reasonable way. For example, if we press the down arrow key with the marker at the bottom of the screen, we want the marker to stay in the same place. In this case, we subtract out the line length we added in earlier (we could compact this some but it's easier to see what's happening this way).

The comments in the program are intentionally profuse to help you figure out how it works.

Program Listing for Basically BASIC

```
10 REM*******************************
20 REM
30 REM  SCREEN CONTROL
40 REM  (C)1982 BY TERRY R. DETTMANN
50 REM
60 REM  FOR 80-U.S. JOURNAL
61 REM          THIS PROGRAM IS TYPED S
O THAT LINES THAT ARE MULTIPLES OF 10
62 REM          SHOULD BE LEFT IN AND L
INES WITH NUMBERS NOT A MULTIPLE OF
63 REM          10 MAY BE LEFT OUT.
70 REM
80 REM*******************************
90 CLEAR100
91 REM  THE FOLLOWING VARIABLES ARE THE
 SCREEN CONTROL ASCII CODES AND LIMITS
92 REM  MD  MID POINT OF SCREEN      BG
 BEGINNING          ND  END OF SCREEN
93 REM  U   UP ARROW KEY             D
 DOWN ARROW KEY
94 REM  L   LEFT ARROW KEY           R
 RIGHT ARROW KEY
95 REM  LN  LINE LENGTH              CU
R     ASCII CHARACTER TO TURN OFF CURSO
R
98 REM  THESE VARIABLES ARE SET FOR THE
 MODEL II, SIMPLY RESET THEN
99 REM FOR YOUR SYSTEM
100 MD=1000:BG=0:ND=1918:U=30:D=31:L=28
:R=29:LN=80:CUR=2
101 REM FOR MODEL I/III
102 REM MD=511:BG=0:ND=1022:U=91:D=10:L
=8:R=9:LN=64:CUR=15
103 REM FOR COLOR COMPUTER
104 REM MD=256:BG=0:ND=510:U=94:D=10:L=
8:R=9:LN=32:CUR=0
105 REM          CLEAR THE SCREEN, PRINT
 A WELCOME, AND TURN OFF THE
106 REM          CURSOR FROM THE SYSTEM
110 CLS:PRINT"TRY SOME SCREEN HANDLING"
:PRINTCHR$(CUR)
115 REM          INITIALIZE THE PROGRAM
BY SETTING LC (CURRENT LOCATION) TO THE
```

```
116 REM          SCREEN MID POINT, PUTTI
NG OUR OWN MOVING CHARACTER THERE,
117 REM          AND THEN SETTING LL (TH
E LAST PRINT LOCATION) TO WHERE WE PUT
118 REM          THE CHARACTER
120 LC=MD:PRINT@LC,"+";:LL=LC
130 REM - - - - - - - - - - START THE CON
TROL LOOP HERE - - - - - - - - - - - - -
-
135 REM          GET A CHARACTER FROM TH
E KEYBOARD WITHOUT PRINTING IT
136 REM          THE TEST IS TO MAKE IT
WAIT UNTIL SOME CHARACTER IS
137 REM          IS HIT
140 C$=INKEY$:IF C$=""THEN140
145 REM          AS LONG AS THE CHARACTE
R IS BETWEEN A BLANK AND A CAPITAL
146 REM          Z, PRINT IT (LOOK UP TH
E CHARACTERS IN A ASCII CODE TABLE)
150 IF C$>=" " AND C$<="Z" THEN PRINT@L
C,C$;:LC=LC+1:PRINT@LC,"+";:LL=LC:GOTO1
40
155 REM          IF THE CHARACTER IS AN
UP,DOWN,LEFT, OR RIGHT ARROW, THEN
156 REM          MOVE THE CURRENT LOCATI
ON APPROPRIATELY
160 IF ASC(C$)=U THEN LC=LC-LN:GOTO210
170 IF ASC(C$)=D THEN LC=LC+LN:GOTO210
180 IF ASC(C$)=L THEN LC=LC-1:GOTO210
190 IF ASC(C$)=R THEN LC=LC+1:GOTO210
195 REM          IF NO CHARACTER IS MATC
HED, GO GET ANOTHER
200 GOTO140
205 REM          CHECK THE BEGINNING AND
 ENDING CHARACTER POSITION LIMITS
206 REM          AND CORRECT THE POSITIO
N IF IT FALLS OFF THE SCREEN
207 REM          NOTICE LINE 210 AND 230
 WHICH CHECK FOR THE SPECIAL
208 REM          CASE WHERE IT IS ONLY O
NE CHARACTER OFF THE SCREEN
210 IF LC=BG-1 THEN LC=BG
220 IF LC<BG THEN LC=LC+LN
230 IF LC=ND+1 THEN LC=ND
240 IF LC>ND THEN LC=LC-LN
245 REM          BLANK OUT THE POSITION
CHARACTER AT THE LAST LOCATION
246 REM          AND PRINT THE CHARACTER
 AT IT'S NEW POSITION, FINALLY
247 REM          UPDATE THE LAST LOCATIO
N (LL)
250 PRINT@LL," ";
260 PRINT@LC,"+";:LL=LC
265 REM          GO GET ANOTHER CHARACTE
R
270 GOTO140
```

■

Prowrite

Complete word processor for TRS-80 Model III, and LP-IV printer

Model III, 32K with disks

Leonard Zucker, Charleston, SC

Prowrite is easy to use. To take full advantage of its capabilities, read these instructions thoroughly, then **EXPERIMENT!** Throughout this article, letters or combinations of letters that are enclosed between the symbols "< and >" denote special keys to strike, rather than letters to type in. Examples:

<ENTER> — the white ENTER key
<DOWN> — the down-arrow key
<ESCAPE> — the key combination <SHIFT> <UP-ARROW>
<CONTROL> — the key combination <SHIFT> <DOWN>

The latter combination is employed only in combination with a third key. For example, <CONTROL U> denotes the key combination SHIFT, DOWN-ARROW, "U" pressed simultaneously. Notice that the keyboard layout makes it easy to depress SHIFT and DOWN-ARROW with two fingers of the left hand, leaving the right hand free to strike the "U." Actually, the three keys don't have to be struck at exactly the same time. To activate <CONTROL U>, for example, you can press <SHIFT>, then <DOWN> while holding down <SHIFT>, the "U," while still holding down the other two keys. (By the way, Prowrite will interpret <CONTROL U> as a command to begin underlining.)

Typing in text

Type normally to input new text. Prowrite will automatically wrap the text around to a new line when necessary, breaking the text at the last space, so that words aren't divided. Use <ENTER> only to end a paragraph. It is not necessary — nor possible — to use the ENTER key to instruct the computer to begin a new line of text on the screen. When <ENTER> is hit, the paragraph symbol appears on the display to indicate the end of the old paragraph and the beginning of a new one. However, succeeding characters will appear on the screen immediately following the paragraph symbol, not on a new line. Prowrite will normally accept input as fast as you can type. Periodically, however, there will be a few seconds delay while string space is being freed. If you make a mistake while typing, use <BACKSPACE> (the left-arrow key) to erase. The backspace key will continue to erase characters until it is released, or the cursor reaches the beginning of the current line. To correct mistakes on an earlier line, use the editing commands explained later. When you have finished entering your text, use <ESCAPE> to return to the menu.

Practice Session

You are now ready to give Prowrite a try. From BASIC type RUN "Prowrite" or from TRSDOS, hit the orange reset key. When the prompt appears, hit <ENTER> for the menu. Hit "N" to input new text. When the screen clears and the cursor starts blinking, type in a few lines of text. Use <ESCAPE> to return to the menu. Hit "P" to print the text. If you are feeding single sheets of paper to the lineprinter, answer "Y" to the first question asked; otherwise, "N." If you want succeeding pages to be numbered, answer the second question "Y"; otherwise, "N." Since you have entered only a few lines of text for this test, your answer to the second question is immaterial. The prompt, "Width of print in millimeters?" will appear. Answer this prompt

with any reasonable number, say 130. Now sit back and watch the screen and the printer as Prowrite takes over and prints your text the width you specified, centered and right-justified. When printing is complete and the menu reappears, hit "P" again. This time use a different print width, say 80 (millimeters). Watch as Prowrite prints your text with the new line width. When printing is done, hit "Q" to exit Prowrite. To get the most out of Prowrite, read the following sections that will explain how to use control codes for special effects, and how to use the disk and editing commands available with Prowrite.

Control Codes

Prowrite employs three types of control codes to give you complete control over the form of the printed text: paragraph terminators, print-parameter controls, and special feature controls. These codes are entered along with the text, but Prowrite treats them as directions to follow, rather than as text to print. They appear as special symbols on video.

Paragraph Terminators — As discussed previously, <ENTER> is the normal paragraph terminator. On video, <ENTER> appears as the paragraph symbol. When printed on the lineprinter, the current line is terminated and printed *without* right justification. When your text is to be single spaced, try using two <ENTER>'s between paragraphs to give the printed text a crisper appearance.

For special purposes, especially when the paragraph consists of a single line, you may want the last (or only) line in a paragraph to be centered and right-justified. For example, the heading of this section was printed 35 millimeters wide, centered and right-justified. To accomplish this, use <DOWN>, rather than <ENTER>, as the paragraph terminator. On video, this will appear as the infinity symbol (a sideways 8).

Print Parameter Controls — The print parameter controls govern line width, indentation and vertical spacing of the text. As a rule, all print parameter controls should precede the first word of the paragraph in which they appear. They will work at other positions in the paragraph, but since you have no way of determining before print time where individual lines will end, the results can be unpredictable.

Use <CONTROL W> followed by the desired width in millimeters to change print width. The new print width *must* be expressed as a *three* digit number; e.g., 130 for 130, 090 for 90 millimeters. Maximum width is 200 millimeters. Your text will be centered and printed with the width you specify until superceded by a new line width control command. If no print-width control codes are included in the text, the entire text will be printed using the print width specified in answer to the prompt at the beginning of the print routine. On the other hand, if a print-width control command is included at the beginning of the text, you may answer the print-width prompt with <ENTER>.

To control vertical spacing, use <CONTROL S> followed by a single digit number equal to the number of extra *half* vertical spaces to insert between lines. Use zero for single spacing (i.e., no *extra* spaces), 2 for double spacing manuscripts. Use 1 for "1½" spacing, which makes very attractive copy. If no vertical spacing code is included in the text, single spacing is assumed. As with line width, vertical spacing remains the same until a new vertical spacing command is received.

To control horizontal tab or indentation, use <CONTROL T> followed by *three* digits representing the number of millimeters to tab over before printing. This tab is measured from the left margin of the current print field. For example, with a line width of 140mm, use a tab of 090 for the date at the top of a letter and for the closing at the bottom; use 010 or 012 for regular paragraph indentation. *Note:* Unlike line width and vertical spacing commands, horizontal tab reverts to zero after each use. You must specify the amount to tab (if other than zero) at the beginning of each paragraph. Also note that a tab control takes effect at the beginning of the print line that contains it — it is *not* possible to tab in the middle of a print line.

Special Feature Controls — The special feature control codes govern underlining, double-width characters and boldface print. In each case, notice that the control code which activates the feature is the first letter in the name of the feature ("U" for underlining, etc.), while the letter which terminates the feature is the *next* letter of the alphabet.

<CONTROL U> — start underlining
<CONTROL V> — end underlining
<CONTROL D> — start double-width characters
<CONTROL E> — end double-width characters

The above controls can be used anywhere in the text to highlight a single letter or word, a phrase, an entire paragraph, or more.

<CONTROL B> — begin boldface
<CONTROL C> — end boldface

Boldface is possible for whole lines only, and since the makeup of individual lines in a paragraph is not known before print time, boldface would normally be employed only in whole paragraphs. But remember, a paragraph may consist of a single line. For example, the major headings in this manual are single line paragraphs that are printed in boldface. To print a paragraph in bolface, use <CONTROL B> at the beginning of the paragraph, and <CONTROL C> *after* the paragraph terminator (<ENTER> or <DOWN-ARROW>). If <CONTROL C> is used before the paragraph terminator, the last line of the paragraph will not be printed in boldface.

Disk Operations

Prowrite automates four disk operations for you — <S>ave, <R>ead, <M>erge and <K>ill. When using any of these, you will receive the prompt, "Filename?". When saving a file, use any name (eight characters maximum) you choose. When reading, merging or killing a file, of course, use the already chosen name of the file. If you include no extension in your filespec, Prowrite will add the extension /TXT.

After <R>eading a file, Prowrite will "remember" its name unless you use <N>ew to enter new text or <M>erge another disk file. If you subsequently <S>ave the text, the current filename will appear in the filename prompt. If this is the name you want to use, simply hit <ENTER>. Otherwise, type in the name you desire. Use <K>ill to kill a text file you no longer need. After

entering the filename, Prowrite will prompt you again with "Kill -->filename?, where 'filename' is the name of the file you specified. Type "Y" if you do want to kill the file; otherwise "N" will return you to the menu without killing the file. Use <M>erge to merge a disk file with the text currently in memory — whether entered manually or from disk. The merged file will be appended to the *end* of the current text.

Editing Commands

Use <E>dit to edit or simply view the text currently in memory. The first seven lines of text will appear, with the first line outlined in the "view window." Each line will end in a vertical line segment so that any trailing spaces will be plainly visible. Use the up- and down-arrow keys to scroll the text up or down, and the left- and right-arrow keys to move the cursor within the view window. Use <ENTER>, except when in change or insert mode, to return to the menu. Note: the Prowrite editing commands are similar to, but *not* identical to, the corresponding Model III BASIC editing commands.

nL (List) — to list line number n of the current text. If n is greater than the current number of lines, Prowrite will list the last line. When listing a line, or while scrolling the text, the "current" line appears in the view window, and the six lines above and below the current line are also displayed. Use "nL" to get close to the portion of text you want to view or edit, then use the arrow keys to find the line desired. A quick way to list the last line of text is use "99L", if the current text is less than 100 lines long, otherwise use "999L."

nC (Change) — to change n characters beginning at the cursor position. If n is not specified, *one* will be assumed, and if n exceeds the number of characters from the cursor position to the end of the current line, Prowrite will use just that number of characters. After "nC," type in the n correct values, and Prowrite will replace the wrong characters with the correct ones. Once n characters have been typed in, typing additional characters will have no effect, unless, of course, one of the characters typed initiates another edit subcommand. If you decide to change fewer than n characters, use <ESCAPE> to escape the change subcommand while remaining in edit mode.

nD (Delete) — to delete n characters beginning at the cursor position. As above, if n is not specified, *one* is assumed, and if n exceeds the number of characters between the cursor position and the end of the current line, Prowrite will delete the balance of the current line only. Another method of deleting multiple characters is simply to depress "D" and leave it down. Prowrite will continously delete characters until "D" is released.

K (Kill) — to kill the current line beginning at the cursor position. If the cursor is at the first character in the line, the entire line will be deleted and all succeeding lines will scroll up one line.

I (Insert) — to insert characters beginning at the cursor position. Prowrite will move text down to the following line as necessary to accomodate the characters entered. Use <ESCAPE> to end the insertion and remain in edit/view mode.

H (Hack) — to kill the remainder of the line and insert at the current cursor position. This has the same effect as "K" followed by "I."

Limitations

Prowrite will more than adequately fulfill your everyday wordprocessing needs. The control it gives you over the special features of the Lineprinter IV will add a touch of professionalism to your letters or reports. However, Prowrite does have its limitations. The maximum number of video lines of text allowed is 250 — enough for about five or six pages of single-spaced text (depending on print width).

The paperfeed and pagination controls assume 8½ x 11 inch paper. If you use odd-sized paper and your text is contained in one page, this limitation will not affect you. However, if your printed text will occupy more than one page, you will have to use the ONLINE/LOCAL switch manually to stop printing at the bottom of the page and to feed the paper manually. In addition, you will have to disable the paper-feed control built into Prowrite by changing the "48" in line 560 of the program to "999."

One further limitation of Prowrite is its slow print speed, but that is a minor inconvenience for the quality of the print it provides you. (By the way, if you ever need to stop the printing process and return to the menu *without* destroying the text in memory, hit <BREAK> then type "GOTO1000".)

Prowrite Variables

All variables without type declaration tags are INTEGER variables. Listed in approximate line-number order of first appearance.

I, J, II, JJ, WW = COUNTERS
L$ = TEMPORARY STORAGE OF VARIOUS STRING VARIABLES
K$ = INPUT STRING (CHARACTER)
IN$ = INPUT STRING (LINE)
**EF$** = END-OF-FILE STRING = <ESCAPE> = CHR$ (27)
NL = # OF VIDEO LINES OF TEXT
NC = # OF CHARACTERS
NC = (IN EDIT MODE) LINE # TO DISPLAY OR # CHAR TO DELETE OR CHANGE
A = ASCII CODE OF VARIOUS CHARACTERS
LS = LAST SPACE (OR <ENTER> — USED TO DETERMINE WHERE TO BREAK LINE
X(I) = POINTER TO I'th LINE OF TEXT
LN$( ) = VIDEO TEXT LINES
D = # OF DELETED LINES
CF = CHANGE FLAG (= 1 IF CHANGE OR NEW ENTRY MADE, ZERO (0) OTHERWISE)

PL = PRINT-LINE NUMBER

PP = 1 IF LAST PRINT LINE ENDED IN DOUBLE-WIDTH MODE, ZERO (0) OTHERWISE

QQ = 1 IF LAST PRINT LINE ENDED IN UNDERLINE MODE, ZERO (0) OTHERWISE

VS = # VERTICAL HALF-SPACES TO SPACE BETWEEN LINES

DF = DOUBLE-WIDTH FLAG = 1 IF CURRENTLY DBL WIDTH, ZERO (0) OTHERWISE

BF = BOLDFACE FLAG

UF = UNDERLINE FLAG

LN! = # LINES AND HALF-LINES PRINTED (FOR PAGING CONTROL)

PG = PAGE NUMBER

DP = DOT-SPACES TO PRINT

TA = AMOUNT TO TAB TO REACH LEFT MARGIN OF CURRENT PRINT WIDTH

TF = TAB FLAG = AMOUNT TO TAB BEYOND LEFT MARGIN FOR INDENTING

ND NUMBER OF DOT-SPACES SO FAR IN LINE

ND TEMPORARY VARIABLE IN LINES 10060 - 10065

P$(J) = J'th PRINT CHARACTER

L(I) = DOT-WIDTH OF CHR$(I)

MX = MAXIMUM NUMBER OF DOTS IN LINE

**D$** = SINGLE DOT SPACE = CHR$(27)+CHR$(1)

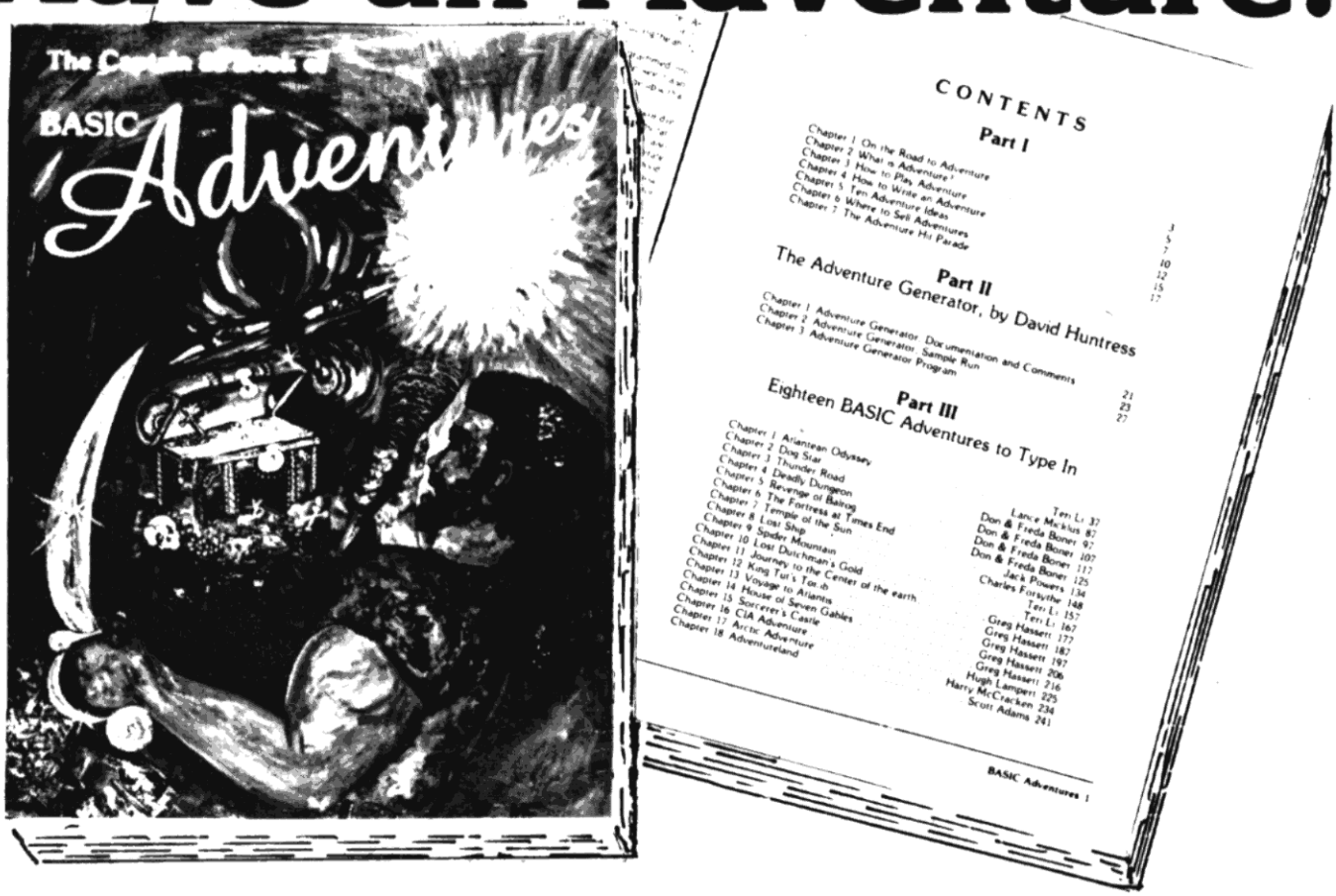
**D3$** = TRIPLE DOT SPACE = CHR$(27)+CHR$(3)

TP = TYPE OF PAPER = 1 IF AUTO FEED, ZERO (0) IF MANUAL FEED

AP = AUTO-PAGE FLAG = 1 IF PAGES ARE TO BE NUMBERED, ZERO (0) OTHERWISE

MM = WIDTH OF PRINT IN MILLIMETERS

Have an Adventure!



- **The first book of its kind!**
- **TRS-80® Model I & III, adaptable to other computers using Microsoft® Basic.**
- **18 Famous adventures by Famous authors.**
- **Contains a unique adventure generator!**
- **Just $19.95 (+ $2.05 S/H).**
- **Buy through your local dealer or order direct from:**

80-U.S.
3838 South Warner Street
Tacoma, Washington 98409
(206) 475-2219

"Well done, highly readable, easy to use. The book is a delightful adventure." —**George Blank**, Editorial Director, **Creative Computing.**

Please send _____ copies of the **ADVENTURE BOOK** for only $19.95 each, plus $2.05 shipping and handling. Check, M.O., VISA/MC # enclosed. C.O.D. orders accepted.

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

VISA/MC _____ EXP. DATE _____

SIGNATURE _____

SEND TO:
80-Northwest Publishing, Inc.
3838 South Warner Street, Tacoma, WA 98409
DEALER INQUIRIES INVITED

F$ = FILENAME FOR DISK OPERATIONS

A$(I) = I'th SCREEN LINE OF TEXT IN EDIT/-VIEW MODE

ML = MAXIMUM NUMBER OF VIDEO LINES OF TEXT = 250

CN = NUMBER OF CONTROL CODES IN MODE = 11

CD(J) = J'th EDIT MODE CONTROL CODE

LC = LINE COUNTER = CURRENT LINE DISPLAYED IN VIEW WINDOW

TB = TAB POSITION OF CURSOR IN EDIT MODE

**SL$** = CHR$(2) = VERTICAL LINE SEGMENT (AFTER LAST CHAR IN EDIT LINE)

ZZ = ERROR NUMBER

ER$ = ERROR MANUAL (MODEL III OR DISK BASIC)

Program Listing for Prowrite

```
1 ' *** PROWRITE ***
2 GOTO10000
3 JJ=JJ+1:IFJJ<=LTHENK$=MID$(L$,JJ,1):R
ETURN
4 JJ=1:I=I+1:IFI<=NLTHENL$=LN$(X(I)):L=
LEN(L$):K$=MID$(L$,JJ,1):RETURNELSEK$=E
F$:RETURN
5 K$=INKEY$:IFK$=""THEN5ELSERETURN
7 CLS:FORI=NL-9TONL-1:IFI>0THENPRINTLN$
(X(I))
8 NEXT:L$=LN$(X(NL)):L=LEN(L$):NL=NL-1:
D=D+1:LS=0:FORJJ=1TOL:IFMID$(L$,JJ,1)="
 "ORMID$(L$,JJ,1)=CHR$(241)THENLS=JJ:NE
XTELSENEXT
9 POKE16409,0
10 K$="":IN$=L$:NC=L:PRINTL$;CHR$(14);
20 GOSUB5:A=ASC(K$):IFA=8THEN90ELSEIFA=
27THEN60ELSEIFA<32THENGOSUB9000
30 NC=NC+1:IN$=IN$+K$:PRINTK$;:IFA=32OR
A=13THENLS=NC
40 IFNC<63THEN20ELSEIFLS=0THENLS=NC
50 L$=MID$(IN$,LS+1,63):IN$=LEFT$(IN$,L
S):L=LEN(L$):PRINTSTRING$(L,8)
60 NL=NL+1:LN$(NL+D)=IN$:X(NL)=NL+D:IFK
$<>EF$ANDNL+D<MLTHENLS=0:GOTO10
70 CF=1:PRINTCHR$(15);:IFIN$=""THENNL=N
L-1
80 RETURN
90 IFNC=0THEN20ELSENC=NC-1:IN$=LEFT$(IN
$,NC):PRINTCHR$(24);:IF(PEEK(16418)=320
RPEEK(16418)=241)ANDIN$<>""THENLS=0:FOR
I=1TOLEN(IN$):IFASC(MID$(IN$,I,1))=32TH
ENLS=I:NEXTELSENEXT
```

```
100 PRINT" "CHR$(8);:FORJJ=1TO90:NEXT:I
FPEEK(14400)=32THEN90ELSE20
300 GOSUB700:GOSUB8010:IN$="":JJ=0:I=1:
L$=LN$(X(1)):L=LEN(L$):PL=0:PP=0:QQ=0:V
S=0:DF=0:BF=0:UF=0:LN!=-2:PG=1:LPRINTEF
$CHR$(17)
310 PRINT@320,CHR$(31)IN$;:DP=0:TF=0:ND
=0:LS=0:NC=LEN(IN$):IFNC>0THENFORK=1TON
C:A=ASC(MID$(IN$,K,1)):P$(K)=CHR$(A):IF
A<127THENND=ND+L(A):NEXTELSENEXT
320 GOSUB3:IFK$<>EF$THENA=ASC(K$)ELSENC
=NC+1:A=241:P$(NC)=CHR$(241):GOTO9800
330 NC=NC+1:PRINTK$;:P$(NC)=K$:IFA>191T
HEN9800
340 ND=ND+L(A):IFA=32ORA=45THENLS=NC:DP
=ND
350 IFND<MXTHEN320ELSEIFLS=0THENLS=NC:D
P=ND
360 IN$="":IFLS<NCTHENFORJ=LS+1TONC:IN$
=IN$+P$(J):NEXT
370 PRINTSTRING$(LEN(IN$),8);
380 PL=PL+1:IFP$(1)=" "THENLS=LS-1:FORJ
=1TOLS:P$(J)=P$(J+1):NEXT:IFPP=1THENDP=
DP-14ELSEDP=DP-7
390 A=ASC(P$(LS)):IFA<>32THEN420ELSELS=
LS-1
400 IFINSTR(IN$,CHR$(203))>0THENDP=DP-7
ELSEIFINSTR(IN$,CHR$(204))>0THENDP=DP-1
4ELSEDP=DP-L(32)
```



```
410 GOTO390
420 'Insert dot spaces
430 DP=MX+18-DP:PRINT@207,"Line #"PL,@2
22,"Dots to insert:"DP;:FORJ=1TOLS:IFAS
C(P$(J))>191THENGOSUB9500ELSENC=J
440 NEXTJ
450 FORJ=1TONC-1:IFP$(J)=" "THENIFDP>2T
HENP$(J)=P$(J)+D3$:DP=DP-3:PRINT@237,DP
;CHR$(30);
460 NEXT
470 IFDP>0ANDA<>241THENFORJ=1TONC-1:IFP
$(J)>EF$THENP$(J)=P$(J)+D$:DP=DP-1:PRIN
T@237,DP;CHR$(30);:IFDP>0THENNEXT:GOTO4
70ELSEELSENEXT:GOTO470
480 LPRINTTAB(TA+TF);:IFPP=1THENLPRINTC
HR$(27)CHR$(14);
490 IFQQ=1THENLPRINTCHR$(15);

500 FORJ=1TOLS:LPRINTP$(J);:NEXT:LPRINT
CHR$(14);:IFPEEK(16426)=0THENLPRINT" "E
LSELPRINT
510 IFBF=1THENGOSUB8030
520 IFVS>0THENFORWW=1TOVS:LPRINTCHR$(27
)CHR$(28):NEXT
530 IFDF=1THENPP=1ELSEPP=0
540 IFUF=1THENQQ=1ELSEQQ=0
550 LN!=LN!+.5*VS+1:IFK$=EF$THEN580
560 IFLN!>48THENGOSUB800
570 GOTO310
580 IFLN!<>INT(LN!)THENLPRINTEF$CHR$(28
):LN!=LN!+.5
590 IFTP=1THENLPRINTSTRING$(63-LN!,138)

600 FORJ=1TO150:P$(J)="":NEXT:RETURN
700 CLS:PRINT@128,"Manual paper feed (Y
/N)? "CHR$(14);:GOSUB5:PRINTK$CHR$(15):
IFK$="Y"THENTP=0ELSEIFK$="N"THENTP=1ELS
E700
710 PRINT@192,"Numbered pages (Y/N)? "C
HR$(31)CHR$(14);:GOSUB5:PRINTK$CHR$(15)
:IFK$="Y"THENAP=1ELSEIFK$="N"THENAP=0EL
SE710
720 CLS:PRINT@143,"Width of print in mi
llimeters";:INPUTMM:POKE15532,ASC(":"):
IFMM>200THENMM=200
730 MX=7*MM/1.2-18:TA=(200-MM)/2.4:PRIN
T@173,CHR$(30)MM;@256,STRING$(64,"=");:
RETURN

800 IFTP=0THENPRINT@320,CHR$(31)"Change
 paper. <ENTER> when ready. "CHR$(14);:
GOSUB5:PRINTCHR$(15):IFASC(K$)<>13THEN8
00ELSEFORWW=1TO2:LPRINTEF$CHR$(138):NEX
T:GOTO830
810 IFLN!<>INT(LN!)THENLPRINTEF$CHR$(28
):LN!=LN!+.5
820 LPRINTSTRING$(61-LN!,138)
830 PG=PG+1:IFAP=1THENLPRINTTAB(79)"-"P
G"-"ELSELPRINT
```

```
840 LPRINT:LPRINT:LPRINT:LPRINT:LNI=0:R
ETURN
1000 POKE16409,1:CLS:PRINT@21,"<< P R O
 W R I T E >>"
1010 PRINTTAB(21)"==================
"

1020 PRINTTAB(21)"<N>ew text"
1030 PRINTTAB(21)"<X>tend text"
1040 PRINTTAB(21)"<E>dit or view text"
1050 PRINTTAB(21)"<P>rint text"
1060 PRINTTAB(21)"<S>ave text on disk"
1070 PRINTTAB(21)"<R>ead text from disk
"
1080 PRINTTAB(21)"<M>erge text from dis
k"
1090 PRINTTAB(21)"<K>ill text file"
1100 PRINTTAB(21)"<Q>uit"
1110 POKE16419,252:PRINTTAB(21)"=======
==========":PRINTTAB(23)"Enter choi
ce < >"CHR$(24)CHR$(24)CHR$(14);
1120 K$=INKEY$:GOSUB5:POKE16419,176:PRI
NTCHR$(15);
1130 IFK$="N"THEND=0:F$="":CLS:L$="":L=
0:NC=0:FORI=1TOML:LN$(I)="":NEXT:NL=0:G
OSUB9:GOTO1000ELSEIFK$="X"THENGOSUB7:GO
TO1000
1140 IFK$="E"THENGOSUB2000:GOTO1000ELSE
IFK$="P"THENGOSUB300:GOTO1000

1150 IFK$="S"THENGOSUB4000:GOTO1000ELSE
IFK$="R"THENGOSUB5000:GOTO1000
1160 IFK$="M"THENGOSUB5500:GOTO1000ELSE
IFK$="K"THENGOSUB7000:GOTO1000
1170 IFK$="Q"THENGOSUB6000
1180 GOTO1000
2000 'Edit/View
2010 CLS:IFNL=0THENRETURNELSEPOKE16409,
0

2020 NC=1:PRINT@384,STRING$(64,140);@51
2,STRING$(64,131);:GOSUB2500:NC=0
2030 PRINTCHR$(14);:GOSUB5:A=ASC(K$):IF
A=13THENPRINTCHR$(15);:RETURNELSEIFA>47
ANDA<58THENNC=VAL(K$)+10*NC:IFNC>MLTHEN
NC=ML

2040 IFA>96THENA=A-32
2050 FORJ=1TOCN:IFA<>CD(J)THENNEXT:GOTO
2030
2060 ONJGOSUB2080,2120,2160,2190,2220,2
260,2370,2340,2070,2500,2360:NC=0:GOTO2
030
2070 RETURN'Reset NC=0
2080 'Scroll up
2090 IFLC=NLTHENRETURNELSEPRINTCHR$(15)
;
2100 FORI=1TO12:LSETA$(I)=A$(I+1):NEXT:
LC=LC+1:IFLC<NL-5THENLSETA$(13)=LN$(X(L
C+6))+SL$ELSELSETA$(13)=""
```

```
2110 TB=INSTR(A$(7),SL$):PRINT@447+TB,"
";:IFPEEK(14400)=8THEN2090ELSERETURN
2120 'Scroll down
2130 IFLC=1THENRETURNELSEPRINTCHR$(15);
2140 LC=LC-1:FORI=13TO2STEP-1:LSETA$(I)
=A$(I-1):NEXT:IFLC>6THENLSETA$(1)=LN$(X
(LC-6))+SL$ELSELSETA$(1)=""
2150 TB=INSTR(A$(7),SL$):PRINT@447+TB,"
";:IFPEEK(14400)=16THEN2130ELSERETURN
2160 'Tab right
2170 IFTB<INSTR(A$(7),CHR$(2))THENTB=TB
+1:PRINTCHR$(25);
2180 IFPEEK(14400)=64THEN2170ELSERETURN
2190 'Tab left
2200 IFTB>1THENTB=TB-1:PRINTCHR$(24);
2210 IFPEEK(14400)=32THEN2200ELSERETURN
2220 'Change
2230 L=LEN(LN$(X(LC))):IFTB+NC>LTHENNC=
L-TB+1
2240 FORJ=1TONC:GOSUB5:PRINTCHR$(15);:I
FK$=EF$THENRETURNELSECF=1:A=ASC(K$):IFA
<32THENGOSUB9000
2250 POKE15807+TB,ASC(K$):MID$(LN$(X(LC
)),TB,1)=K$:GOSUB2170:NEXT:RETURN
2260 'Delete
2270 PRINTCHR$(15);:CF=1:IFNC<1THENNC=1
2280 L$=LN$(X(LC)):L=LEN(L$)
2290 IFTB+NC>LTHENNC=L-TB+1
2300 IFNC=LTHENNL=NL-1:D=D+1:FORI=LCTON
L:X(I)=X(I+1):NEXT:NC=LC:GOTO2500
2310 LN$(X(LC))=LEFT$(L$,TB-1)+RIGHT$(L
$,L-TB-NC+1)
2320 LSETA$(7)=LN$(X(LC))+SL$:NC=1
2330 IFPEEK(14377)=16THEN2270ELSERETURN
2340 'Kill to end of line
2350 NC=64:GOTO2260
2360 GOSUB2340
2370 'Insert
2380 L$=LN$(X(LC)):L=LEN(L$)
2390 IFA=8ANDPEEK(14400)=32THEN2490
2400 GOSUB5:IFK$=EF$THENCF=1:RETURNELSE
A=ASC(K$):IFA=8THEN2490ELSEIFA<32THENGO
SUB9000:IFK$=""THEN2400
2410 IFL>62THENIFNL+D<MLTHEN2440ELSECF=
1:RETURN
2420 L$=LEFT$(L$,TB-1)+K$+MID$(L$,TB,L)
:TB=TB+1
2430 PRINTCHR$(15);:LSETA$(7)=L$+SL$:LN
$(X(LC))=L$:PRINT@447+TB,CHR$(14);:GOTO
2380
2440 IFTB<LTHEN2470ELSEIN$="":FORI=LTO1
STEP-1:P$(0)=MID$(L$,I,1):IFP$(0)<>" "A
NDASC(P$(0))<>241THENIN$=P$(0)+IN$:NEXT
I:IFI=0THENIN$=""
2450 LN$(X(LC))=LEFT$(LN$(X(LC)),L-LEN(
IN$)):FORI=NLTOLCSTEP-1:X(I+1)=X(I):NEX
T
2460 NL=NL+1:LC=LC+1:X(LC)=NL+D:LN$(X(L
C))=IN$:NC=LC:GOSUB2500:L$=IN$:L=LEN(L$
):GOTO2410
2470 FORI=NLTOLC+1STEP-1:X(I+1)=X(I):NE
XT:NL=NL+1:X(LC+1)=NL+D:LN$(NL+D)=MID$(
LN$(X(LC)),TB,99):LN$(X(LC))=LEFT$(LN$(
X(LC)),TB-1)
2480 NC=LC:GOSUB2500:L$=LN$(X(LC)):L=LE
N(L$):GOTO2410
2490 IFTB=1THEN2400ELSEL$=LEFT$(L$,TB-2
)+MID$(L$,TB,L):TB=TB-1:GOTO2430
2500 'List line NC
2510 IFNC=0THENRETURNELSEIFNC>NLTHENNC=
NL
2520 PRINTCHR$(15);:FORI=1TO13:L=NC+I-7
:LSETA$(I)=""
2530 IFL>0ANDL<=NLTHENLSETA$(I)=LN$(X(L
))+SL$
2540 NEXT:LC=NC:TB=INSTR(A$(7),SL$):PRI
NT@447+TB,"";:RETURN
4000 CLS:PRINT"Filename";:IFF$<>""THENP
RINT" '"F$"'";
4010 INPUTF$:IFINSTR(F$,"/")=0THENF$=F$
+"/TXT"
4020 PRINT"Saving text as "F$:OPEN"O",1
,F$
4030 PRINTNL"lines":PRINT#1,NL
4040 FORI=1TONL:PRINTLN$(X(I)):PRINT#1,
LN$(X(I)):NEXT:CLOSE:CF=0:RETURN
5000 CMD"D:0":PRINT:F$="":INPUT"Filenam
e";F$:IFINSTR(F$,"/")=0THENF$=F$+"/TXT"
5010 PRINT"Reading ===> "F$:OPEN"I",1,F
$
5020 FORI=1TOML:LN$(I)="":NEXT
5030 INPUT#1,NL:PRINTNL"lines"
5040 FORI=1TONL:LINEINPUT#1,LN$(I):PRIN
TLN$(I):NEXT:CLOSE:FORI=1TONL:X(I)=I:NE
XT:D=0:CF=0:RETURN
5500 'Merge disk file
5510 CMD"D:0":PRINT:K$="":INPUT"Filenam
e";K$:IFINSTR(K$,"/")=0THENK$=K$+"/TXT"
5520 PRINT"Merging ===> "K$:OPEN"I",1,K
$
5530 INPUT#1,L:PRINTL"lines":IFL+NL+D>M
LTHENPRINTL+NL+D-ML"lines too long":CLO
SE:FORI=1TO999:NEXT:RETURN
5540 FORI=NL+1TONL+L:X(I)=I+D:LINEINPUT
#1,LN$(I+D)
5550 PRINTLN$(I+D):NEXT:CLOSE:NL=NL+L:C
F=1:RETURN
6000 CLS:IFCF=1THENPRINT"You have enter
ed new text.  Do you wish to save the c
hanges? "CHR$(14);:GOSUB5:PRINTCHR$(15)
:IFK$="Y"THENGOSUB4000
6010 CLS:END
7000 'Kill file
```

```
7010 CMD"D:0":PRINT:IN$="":INPUT"Filena
me";IN$:IFINSTR(IN$,"/")=0THENIN$=IN$+"
/TXT"
7020 PRINT"Kill ===> "IN$"? ";
7030 PRINTCHR$(14);:GOSUB5:PRINTCHR$(15
);:IFK$="N"THENRETURNELSEIFK$<>"Y"THEN7
030
7040 PRINTK$:PRINT"Killing ===> "IN$:KI
LLIN$:RETURN
8000 'Double-width and boldface routine
s
8010 RESTORE:FORII=32TO126:READL(II):NE
XT:RETURN
8020 FORII=32TO126:L(II)=2*L(II):NEXT:R
ETURN
8030 LPRINTCHR$(27)CHR$(138)TAB(TA+TF+2
)D$;
8040 IFPP=1THENLPRINTCHR$(27)CHR$(14);
8050 FORJ=1TOLS:LPRINTP$(J);:NEXT:LPRIN
TCHR$(14);:IFPEEK(16426)=0THENLPRINT" "
ELSELPRINT
8060 RETURN
9000 'Control code routine
9010 IFA=13THENK$=CHR$(241):RETURN
9020 IFA=21THENK$=CHR$(232):RETURN
9030 IFA=22THENK$=CHR$(230):RETURN
9040 IFA=23THENK$=CHR$(223):RETURN
9050 IFA=2THENK$=CHR$(201):RETURN
9060 IFA=3THENK$=CHR$(202):RETURN
9070 IFA=4THENK$=CHR$(203):RETURN
9080 IFA=5THENK$=CHR$(204):RETURN
9090 IFA=10THENK$=CHR$(235):RETURN
9100 IFA=20THENK$=CHR$(218):RETURN
9110 IFA=19THENK$=CHR$(227):RETURN
9120 NC=NC-1:K$="":RETURN
9500 'Inverse control code routines
9510 K=ASC(P$(J))
9520 IFK=241ORK=235THENP$(J)="":RETURN
9530 IFK=232THENP$(J)=CHR$(15):UF=1:RET
URN
9540 IFK=230THENP$(J)=CHR$(14):UF=0:RET
URN
9550 IFK=201THENP$(J)="":BF=1:RETURN
9560 IFK=202THENP$(J)="":BF=0:RETURN
9570 IFK=203THENP$(J)=CHR$(27)+CHR$(14)
:DF=1:RETURN
9580 IFK=204THENP$(J)=CHR$(27)+CHR$(15)
:DF=0:RETURN
9590 RETURN
9800 IFA=241ORA=235THENLS=NC:DP=ND:GOTO
360
9810 IFA=223THENMM=0:FORWW=1TO3:GOSUB3:
PRINTK$;:MM=10*MM+VAL(K$):NEXT:GOSUB730
:NC=NC-1:GOTO320
9820 IFA=218THENTF=0:FORWW=1TO3:GOSUB3:
PRINTK$;:TF=10*TF+VAL(K$):NEXT:PRINT:NC
=NC-1:TF=TF/1.2:ND=ND+7*TF:GOTO320
9830 IFA=203THENGOSUB8020:GOTO320
9840 IFA=204THENGOSUB8010:GOTO320
9850 IFA=227THENGOSUB3:PRINTK$:VS=VAL(K
$):NC=NC-1:GOTO320
9860 GOTO320
10000 'Initialization
10010 CLS:PRINT@21,"<< P R O W R I T E
>>":PRINTTAB(21)"=================="
10020 K$=CHR$(191):PRINT:PRINTTAB(17)CH
R$(188)STRING$(26,140)CHR$(188)
10030 PRINTTAB(17)K$"   LEONARD ZUCKER"
TAB(44)K$
10040 PRINTTAB(17)K$"   10 Hobonny Lane
"TAB(44)K$
10050 PRINTTAB(17)K$"   Charleston, SC
29407"TAB(44)K$
10060 PRINTTAB(17)K$"   Telephone:  571
-6677"TAB(44)K$
10070 PRINTTAB(17)STRING$(28,131)
10080 CLEAR26000:DEFINTA-Z:K$="":IN$=""
:JJ=0:I=0:J=0:A=0:ND=0:MX=0:L=0:L$="":E
F$=CHR$(27):D$=CHR$(27)+CHR$(1):D3$=CHR
$(27)+CHR$(3):ML=251:DIMX(ML),L(126),LN
$(ML),P$(150)
10090 GOSUB8010:POKE16420,1:SL$=CHR$(2)
:DIMA$(13):FORI=1TO13:A$(I)=STRING$(64,
32)
10100 IFI<7THENND=15360+64*(I-1)ELSEIFI
>7THENND=15360+64*(I+1)ELSEND=15808
10110 L=ND/256:K=ND-256*L:J=VARPTR(A$(I
)):POKEJ+1,K:POKEJ+2,L:NEXT
10120 CN=11:DIMCD(CN):FORI=1TOCN:READCD
(I):NEXT:ONERRORGOTO15000
10130 PRINT@660,"Hit <ENTER> for menu":
K$=INKEY$:GOSUB5:IFASC(K$)<>13THEN10130
ELSE1000
15000 IFERR=106THENPRINT:PRINT"File not
 found":FORI=1TO1000:NEXT:RESUME1000
15010 ZZ=ERR:IFZZ<99THENZZ=ZZ/2+1:ER$="
 Model III "ELSEZZ=ERR:ER$=" Disk Owner
's "
15020 PRINT:PRINT"Error #"ZZ"in line #"
ERL
15030 PRINT:PRINT"Consult your"ER$"manu
al."
15040 PRINT:PRINT"<ENTER> for menu."
15050 GOSUB5:IFASC(K$)<>13THEN15050ELSE
CLOSE:RESUME1000
20000 DATA7,7,10,15,12,16,14,7,7,7,12,1
2,7,12,7,12,12,12,12,12,12,12,12,12,12,
12,7,7,12,12,12,12,14
20001 DATA16,15,14,16,14,14,16,16,10,14
,16,14,18,16,16,14,14,15,12,14,16,16,18
,16,16,10,12,12,12,12
20002 DATA12,7,12,12,10,12,12,10,12,12,
8,6,12,8,16,12,12,12,12,10,12,10,12,12,
16,12,12,10,10,7,10,12
20003 DATA91,10,9,8,67,68,73,75,27,76,7
2
```

Togetherness

Tape merge for the Color Computer

R. N. Roberts, Lancaster, OH

Want to save programming time? Tired of keying in the same material over and over? Then, what you need is a technique to add a program on tape to a program in your computer. With the following program, you can pick out routines from the programs you've already written and merge them with your current effort. You can develop a library of frequently used routines which can be merged with others to quickly develop complex programs. Examples of useful routines you might want in your library would be a set of graphics letters and numbers, your favorite "victory" tune for games, or perhaps a day and date routine.

Here's how it's done. In the Color Computer, the address for the start of BASIC programs is stored in memory locations 25 and 26 (decimal). When the CLOAD command is executed, the computer begins loading the incoming program at the address in locations 25 and 26. Locations 27 and 28 hold the address of the first byte after the end of a BASIC program.

To merge programs, use the PEEK statement to get the end-of-program address from locations 27 and 28, subtract 2 from it, and POKE the revised address into locations 25 and 26 (the start-of-program address). When the CLOAD command is executed, the incoming program will begin loading at the end of the program in memory.

Why is it necessary to subtract two from the end-of-program address? The Color Computer puts two zeroes at the end of a program. By subtracting two from the end-of-program address, the incoming program writes over the zeroes. If this were not done, the BASIC interpreter would stop when it reached the two zeroes and the added material would not be processed.

The address of a memory location in an eight bit computer, such as the Color Computer, takes two bytes. The 6809 microprocessor used in the Color Computer stores the most significant byte (MSB) of an address before the least significant byte (LSB). This is contrary to the practice of the Z-80 microprocessor used in Model Is, IIs, and IIIs, which stores addresses in the sequence of LSB before MSB.

If the LSB (location 28) is greater than, or equal to, two, subtract two and POKE the result into location 26. Then the MSB can be POKEd into location 25. However, if the LSB is less than two, it will be necessary to "borrow" from the MSB. When subtracting in decimal arithmetic, we borrow in units of 10. In hex arithmetic, the amount borrowed from the MSB is 256. Consequently, if the LSB is less than two, subtract one from the MSB before POKEing it into location 25. Then add 254 (256-2) to the LSB and POKE the result into location 26.

The merge program is shown in Listing 1. Before using this program, make sure that the line numbers of the program in the computer are lower than those of the incoming program. Note that when the CLOAD command is executed from a program, the "S" and flashing "F" will not appear on the screen. However, since the audio is turned on, you will be able to hear the program loading.

Once the merge is complete, the original starting

address must be restored to locations 25 and 26. This can't be done by the program itself, because the CLOAD *command returns control to the command mode.* The original starting address was saved in variables C and D. The screen displays the information which must be keyed in exactly as shown in the command mode to complete the merge.

The tape merge can also be done entirely in the command mode if desired. My merge program is on a utility tape which I load before keying in new programs. The merge utility may be used repeatedly during the development of a new program and then deleted when the program is finished.

Enjoy this utility. It's a real timesaver. You'll never again have to key in data twice.

Listing 1 for Tape Merge

The number of multi-statement lines has been minimized so that the logic is easy to follow.

```
5 PCLS
10 A=PEEK(27):B=PEEK(28)
20 C=PEEK(25):D=PEEK(26)
30 IF B>1 THEN 70
40 POKE 25,A-1
50 POKE 26,B+254
60 GOTO 90
70 POKE 25,A
80 POKE 26,B-2
90 PRINT"POSITION TAPE AND START RECORD
ER. THEN HIT ENTER.":PRINT
100 PRINT"WHEN COMPUTER DISPLAYS OK ENT
ER THEN FOLLOWING:"
110 PRINT"POKE25,";C;":";"POKE26,";D
120 INPUTA$:AUDIO ON
130 CLOAD
140 PMODE 2,1:PCLS:SCREEN1,1
150 GOSUB260
170 PMODE4,1:PCLS:SCREEN1,1
180 GOSUB260
220 PMODE 2,1:PCLS
240 GOSUB260
250 GOTO 250
260 CIRCLE(128,96),90,1
270 FOR T=1 TO 2500:NEXTT
280 RETURN
```

Listing 2 for Tape Merge

Tape merge program written in three lines to conserve memory.

```
10 A=PEEK(27):B=PEEK(28):C=PEEK(25):D=P
EEK(26):IF B>1 THEN 20 ELSE POKE25,A-1:
POKE26,B+254:GOTO30
20 POKE25,A:POKE26,B-2
30 PRINT"POSITION TAPE & START RECORDER
. THEN HIT ENTER.":PRINT:PRINT"WHEN COM
PUTER DISPLAYS OK ENTER THE FOLLOWING:"
:PRINT"POKE25,";C;":";"POKE26,";D:INPUT
A$:AUDIO ON:CLOAD
```

■

Captain 80

The adventures of a software secret agent

I moved from my little log cabin in the woods to a new Software Secret Headquarters in the old Victorian three-story house owned by the (missing) hardware genius, Professor Megabyte. Getting the phone company to install my teakettle and popcorn popper phones was somewhat of a hassle, but as the days wore on, things settled down somewhat. I eased into the routine of my new role as Joe Software Reviewer.

"Good Morning, Max," says I, turning on the lights to the computer room.

"Morning, boss." he replied. (I wonder if I'll ever get used to having casual conversation with a computer program.)

I found the morning's work on the TRS-80 desk. It consisted of a new arcade release from Adventure International entitled Armored Patrol.

The trend toward arcade games which began with Super Nova from Big Five, has produced some mighty fine TRS-80 graphics displays. Duel-N-Droids from Acorn (Leo Christopherson), proved that stick men were no longer valid. The Med-System Adventures, with their rats-eye mazes, added depth of field. Atlantean Oddysey, from Interpro, added static illustration. Fasteroids, from Greg Hassett, proved age was no barrier to quality, though it died prematurely from lack of distribution. The Adventure International arcade line compresses all these factors into a single thrust. That's where Armored Patrol comes in.

Aside from flirting with Atari, who has sworn to rid the world of all versions of their games except those they have fees coming from, Armored Patrol is a *gem* of an arcade game. It has everything that has advanced the state of the art for the last two years.

Here's the scenario. You are the commander of an advanced laser tank trying to rid the planet of bad guy tanks. The view on your screen is the limited vision of the forward TV camera. That view is spectacular for a TRS-80. The 3-D display is very realistic (no pun intended). The mountains in the background are stationary, but the bunkers in the foreground move toward or away from your tank with perfect perspective.

There are robots which appear from time to time to destroy your

tank. Simply zero in on them, fire, and watch them evaporate.

"HEY," protested Max.

Sorry, forgot he was a cybernaut of sorts. Rephrase. By pressing the space bar, you can dispatch them to somewhere that they can't disintegrate your tank.

The enemy tanks are a different matter. They can roll out from behind the bunkers and fire before you can find the right controls to zero in on them. When you hit them with your laser, they fly into pieces just like in the coin-drop version. If you miss, don't expect mercy — they let you have it. The classic shatter cracks appear across your screen. Off you go to tank commander's heaven.

The controls are a little hard to remember at first. They have the same kind of hand crunching configuration that confounded Fasteroids players at first. But it's easy to get used to, and after a couple of games you're popping back and forth across the board like a champ.

The two-player ability of the game is a good feature, as is the little "enemy tank finder" that tells you which way to turn to get those guys before they get you. What stands out more than any other single factor is the authenticity of Armored Patrol. The U.S. Government has been looking into the possibility of having Atari set up their coin-op as an actual tank trainer. I've got a better idea.Try this. Simply get a few thousand TRS-80s, set them up, and call Scott Adams. Make the world safe and save bread at the same time.

Max snickered a little after scanning my Pencil file.

"What's with you," I asked.

"You know what a hard time you have reviewing non-game programs," he intoned.

It's true, utilities and DOSs give me a fit, but how did *he* know? I've only been here three weeks.

"Well, your next item for review is the new Electric Pencil."

No perspiration there, kiddo. I *love* my Electric Pencil and don't care who knows it.

Michael Shrayer's Electric Pencil was the first of the TRS-80 word processors. Those who used it found an almost immediate love-hate relationship. Pencil, in its early forms, was like a cranky, dependable adolescent, which worked beautifully when it worked. What has been lost in the metamorphosis will not be missed.

The new, trim, lean Electric Pencil has many desirable additions. For example, the WORD and RECORD command, which gave out information about the number of words in the file, is no more. In its place is a fixed display in the menu, WORD RECORD FREE MEMORY, and the appropriate numbers. Very handy. Now, when I need to know how much memory is available, or how many words I'm giving my editor, it is right there.

The system menu accessed by control K is a cornucopia of handy displays and commands. This new Pencil accesses tape (CSAVE, CLOAD, CLOAD?) as if it were in BASIC. It accesses Exatron Stringy Floppy (@LOAD, @SAVE, @NEW), and of course, the disk, which seems to work on 80-track and double density without complaint. All this from the same program — no more separate programs or awkward patches. In fact, a first-time word processor user can readily handle this program like a veteran after only a few moments of scanning book and menus.

For Model III owners, and those who'd rather not hack up a keyboard to make a control key, the control key is the CLEAR key, which is useless in a word processor anyhow.

"Don't forget the PRINT routines," Max prompted.

The printer menu is yet another separate modular file. All of the familiar commands for formatting the printing in the printer are back with slightly different codes. The default values here are also displayed in the menu, making it very convenient to do the kind of special things I have to do with my Daisy Wheel Two at times.

Also present in this extra menu, are complete RS232 setting controls. Only in Omniterm, the smart terminal program, have I seen such user friendly mechanisms in software.

Although the genius of Michael Shrayer is unquestionable, the fine influence of Harv Pennington is evident in this Pencil. The manual is readable, coherent, and actually usable. The attractive cover art (which is IJG's hallmark) professionalizes this product — something that was long overdue. Where we find the most evidence of Harv, a fanatic Pencil user himself, is in user friendly streamlining. The smoothness of operation of this word processor has resurrected it from premature burial, and allowed it to take its rightful place as a challenger to Scripsit supreme and the rest of the gang.

Welcome back Electric Pencil — welcome back old friend. There are a lot of Scripsit fans out there who never knew you. More's the pity, for you are easy to use, eccentric no more, and above all, affordable (a mere $89.95). For those of us who've known and loved you all along, welcome back, we say. Here's looking at YOU, kid.

"Don't you think you're being unnecessarily sentimental, Captain?" queried Max sardonicly.

"Not so!" I defended. "There are certain programs that are timeless. Taipan, for example, which is scheduled for a comeback — and Original Adventure, which pops up in a different form each year. I get excited about revisiting programs which have given me pleasure in the past. That's what worries me about the trend toward arcade games. Once mastered, they burn out, like a fireworks display. Boom, spark and they're gone."

Substantiality is an important consideration in commercial software. Producers take note. We need programs with plot, story, challenge and character. Without these factors to balance the current run of entertainment software, we'll be deadended with nothing on the shelf but Porky Pig and Looney Tunes.

Max is going to download himself onto some bulletin boards and do some research in that area while I comb the industry for more interesting programs to review. Hey, this Joe Software producer disguise is working out pretty well.

"Don't let it go to your head," said Max, as he dug Omniterm off drive two. "Remember, you can be replaced by press releases."

I can dig it. ■

Tandy topics

Ed Juge, Director, Computer Merchandising, Tandy Corp.
1500 One Tandy Center, Fort Worth, TX 76102

The view from my 15th floor window this morning is green! Summer is coming (finally), and along with it, we are starting to put together the RSC-8 catalog you will see next September. No, I'm not going to tell you about all the exciting new listings. Rather, I'm making an excuse for what may be a short and disjointed column this month. Getting through the months when we work on the catalog is rather like an endurance test. The rest of the world, unfortunately, does not stop moving. So, our days are long, and the weeks run together with no noticeable weekends.

By now, the new Pocket Computer PC-2 and its four-color Plotter/Printer should be in great supply everywhere. If you haven't seen one, go look. The BASIC compares more favorably to Model III's language than to the previous PC. There are exciting things in the future for PC-2 owners, too. Our plan is to offer a communications interface late this year or early '83—as soon as we can get it out.

For those Model I owners who were convinced we'd forgotten them, our Double Density Disk Kit should be readily available. It's a great way to increase disk capacity, and it works on 35- or 40-track drives. Capacity with a 35-track drive goes to 152K, and 184K with 40-track. Installation doesn't foul up your current single density operation. The kit allows both modes of operation—in fact, we recommend that you continue to run current single density (SD) software in that mode.

Of course, there's a COPY capability to move SD software to the double density (DD) system. Our customer service folks will only be able to support our software in the mode in which it was distributed. *Note: Model III software is not usable with this system!*

The DD system adds lots of advanced features to TRSDOS, many of which are taken from the Model II operating system. But then, you can find details in our current computer catalog, RSC-7, so I won't dwell on them here.

New Programs Really ARE New!

In late June, we expect to start delivering "Super Scripsit" for Model III, and in late July, "Profile III Plus." We already have a Scripsit and Profile for Model III, but don't write these two off—they're so totally different, we probably should have used new names. Although they're also in RSC-7, I suspect you could overlook them because of the name similarities, so I'd like to tell you a little about them here.

Super Scripsit

Super Scripsit began as an attempt to move the very popular Model II word processing (WP) system to the Model I and III. As usually happens, though, creative programmers manage to do things even better the second time around. People who've used both, tell me "Super" is tough competition for Model II's 2.0!

Super Scripsit is for disk systems only (minimum of two disks required on Model I). It allows any document to be as large as available disk space. You're no longer limited to what will fit in memory. Super Scripsit is our first (and only) WP system to support true proportional spaced printing. You also get multiple line spacing, double and single underlining, boldface, super and subscript, a "Help" command, and more.

Model III allows ten "user-defined" keys, allowing up to 127 predetermined keystrokes. You can set up a form report heading, the signature lines on your letters, or whatever you like. I use this system quite often on Model II to make changes to a form I have stored and need to update or change. It's a very versatile and powerful feature. As with Model II, Super Scripsit interfaces with Profile III Plus (on Model III only), and it *(or your current Model I/III Scripsit)* works with the new Model III Scripsit spelling dictionary.

Profile III Plus

This is another "Model II lookalike," and in fact, is almost identical to Profile Plus for the Model II. It allows specified record lengths for maximum disk space usage, multiple screens so that you

can look at your data in different ways (and you can instantly switch between screens), multiple reports, and more. When selecting files, you can use up to 16 selection criteria at a time, and data can be passed to Super Scripsit and Visicalc™ on Model III.

If you're familiar with Profile Plus on Model II, here are some new features on the Model III version:

Record length increased from 853 to 1020.

Segment containing keys on which you search can contain fields totaling 255 characters (Model II allowed only 85 characters).

Mass hardcopy, purge and delete functions.

The 16-field search ability also works with label and report programs.

Of course, Profile III Plus also allows creation of custom user menus, indexing your files in logical order (rather than the order in which items were keyed in), math calculations, two line reports, and much more.

A very nice "Plus" feature in search/select, is the ability to designate "associated fields." Then, if the keyword or data is found in any of the associated fields, the record is selected.

If you need an excellent multi-purpose, free-form electronic filing system for your Model III, by all means, look at Profile III Plus!

Model III Games on Disk

Our first three disk games are (or are about to become) available. "Zork," with separate versions for Models I and III, is a super adventure into an exciting fantasy world. Then there are Model III only games, "Monty Plays Monopoly" and "Monty Plays Scrabble," all from VisiCorp. If you like them, we will consider more disk-based games.

Okay, that's a lot about production, but I've tried to tell you some things that might not have been readily apparent, rather than just "talking up the goodies." If I'd wanted to do that, I'd have raved about our current Color Computer madness . . . which I won't do.

A couple of weeks ago, Bill Walters was in the neighborhood, so he stopped by Tacoma, and the *80-U.S.* offices. He tells me the folks are at least as friendly, enthusiastic and positive in person as they are in print. He was pleased to see they are all having fun, and very devoted to what they're doing. I've received many "Attaboys" on moving my column to this publication too, so they're doing it right!

Since I spent some time explaining ARCNET last month, I better give you the not-so-good news this month. ARCNET availability date has slipped into the first part of calendar 1983. The project has turned out to be a bit larger than first expected. With Model 16, the DT-1 terminal, related software, and other projects I can't discuss yet, we just don't have enough people in-house to get everything done at once. Slips like this in development of hardware, and even more with software, are not too uncommon. But I'll bet you can see now why we don't like to announce "coming attractions" until they're far enough along to be sure things. Until next month. . . ■

17K of RAM

Squeeze extra space from memory

Model I, II, III and Color Computer

R. Shane Causer, Brunswick, GA

Whoa! Wait a second. Go back and read the title of this article. No, this is not an article for a massive 1K memory expansion for the TRS-80 computer. Instead, this article deals with space used by a computer program and how to get more of it. By the way, the title comes from the fact that I managed to squeeze one solid K of memory out of a program that already took 16K of good old RAM.

I bought my TRS-80 about six months after they hit the market. Not being filthy rich at that time (dear IRS: I am not rich now, either), I only bought a 4K machine. Having little memory to spare, I had to cut corners any way I could. After three months of byte-pinching, I broke down and upgraded to 16K. After I got my computer back, RAM saving was forgotten. After writing my first "whopper" of a program, all those tricks and techniques came to mind very quickly.

A note before I go on: If you're into structured BASIC programming, with its elegance and simply followed GOTOs and GOSUBs, some of these tricks may not be for you. Structuring wastes memory!

First, let's look at an awful memory-wasting program. This program (Listing 1) simply draws several circles, one inside the other, on the monitor of the computer. Listing 1 eats memory like a camel drinks water! Now look at Listing 2. It accomplishes the same thing as Listing 1, but with a substantial decrease in the amount of memory used. The difference is easy to see: Listing 1 is spread out over 13 program lines. Listing 2 (in its proper computer-printed form) is crunched and crammed into only two. Considering that (according to the Level II manuals) each program line uses five bytes as overhead, you have enough memory saved to write a sequel to *War and Peace*. Okay, so I exaggerated a little. But you *do* save memory.

Another painless way to save memory is to simply delete all spaces that aren't vital to the program's execution. That means no spaces between colons and program statements, the parameters of a function statement and the statement itself, or any such thing as that. Sure, it sounds simple, but you would be surprised how many people have not done it, myself included.

Also, trash unnecessary program lines. The most used unnecessary program line is that good ol' REMark line. It does nothing more than tell you something about the program and waste a good bit of memory. Leave REMs in, however, if you plan to change parts of the program. That way, you can make the desired changes without straining the old gray brain cells.

A good thing to do is eliminate variables after a NEXT statement. Level II BASIC allows NEXTs to be followed by several variables (separated by a comma), or left to stand alone. If your program has two NEXT statements, one right after the other, join the variables at the end of one NEXT. If you only have one NEXT, leave off the variable at the end. This trick is always worth at least two bytes of memory.

If you're really pressed for memory, rewrite the program like you're making tax cuts in the nation's budget. That is, if you have a loop that decrements variable X from 127 to zero (perhaps for graphics), you would create a loop that looks something like this: FOR X=127 TO 0 STEP-1. Unless this loop (and the variable X) is being used for some complex formula where the order of the loop *must* remain the way it is, the loop could be written as FOR X=0 TO 127, saving six bytes of RAM. If you were using the loop to draw a line using SET graphics, you would have to make a decision whether good looks or memory space was more important. In most cases, I go for the good looks (call me a romantic)!

My next point has been stated time and time again, but surprisingly enough it is very rarely followed. If a part of the program is used more than once, it is a good memory saver to write it as a subroutine and call it with a GOSUB from the places that require it. If you follow this point, though, be aware that it will slow down execution time of the program. Unless your program is a real time arcade graphics game, this is really no problem. If your program is a real time arcade graphics game, give a little extra effort and learn assembly language programming.

If RAM gets scarcer than hen's teeth, two little hints to help save space are simple. One, be careful not to over-dimension arrays. If you

need an array of 11 elements, you shouldn't have to dimension anything. That's because, if you remember the Level II manual, an array is automatically dimensioned to 11 elements. Also, remember that this "automatic DIM" can waste memory by dimensioning 11 when only four elements are needed.

Two, CLEARing can waste memory partly due to human laziness. If you had a choice of counting *every* string character and CLEARing the exact amount of string space, or just taking a guess and hoping that you don't get the dreaded "OS" error, which would you choose? Wrong answer. You would just be general about the whole thing and take a wild guess. It's alright to guess (it's your computer and your program—do what you want). If saving memory is your game, counting string characters isn't too bad an idea. It's just slow and boring.

I'm not terribly mathematically minded (I turn white at the mention of calculus), but I do know enough to save memory. If the variables you use are integers (-32768 to +32767 inclusive), the computer only uses two bytes to store it. Single and double precision numbers use four and eight bytes, respectively. (See the Level II manual for the limits to these number types. They're too darn big to type!) Therefore, you save memory by classifying variables as integers using the DEFINT statement. To save a nice bit of memory, add the DEFINT A-Z statement to your program. If you're into games, this should do the job. If you're into redefining orbital patterns of multidimensional objects, that's another story. Define variables to whatever you think your program requires.

Other things to know for memory saving: Use a question mark (?) as the abbreviation for the PRINT statement and the comma (,) for the THEN statement. When printing long lines of characters to the screen, such as 27 blank spaces, use the STRING$ command. This saves memory *and* time. Anything that saves both of them can't be all bad. While I'm on the subject of printing, try to use the ASCII codes 192 to 255. These codes print from zero to 63 blank spaces on the screen *as if it were a TAB function.*

While we're on the subject of branching within a program (no, you didn't miss anything—I just wanted to change the subject), use the ON nn GOTO or ON nn GOSUB statements. These statements are a lot easier to use and cost a lot less (memory-wise) than the alternative of testing the variable nn several times and branching off to specified lines if the variable meets the conditions. Who hasn't written a program that has a statement like: IF X=1 THEN 100 ELSE IF X=2 THEN 200 ELSE . . . Well, you get the idea.

That brings me to the end of this article. All of the above suggestions are just that: suggestions. Use the ones that you feel good about using. If you are cramming for every last byte, you can yank out of your program, use all of the suggestions to gain the utmost from RAM. If you don't feel like counting, crunching, hacking, deleting and . . . oops, I got carried away. If you don't feel like doing all of this work, there are some good machine language programs out that can really cut a program up, squeezing every byte out of the program. Get one of those and use it. If you just want some extra space, all of these suggestions are for you. Happy hacking!

Listing 1

```
10 CLS
20 E=2
30 FOR R=E TO 22 STEP 4
40 FOR A=-R TO R
50 X=R*R - A*A
60 Y=SQR(X)
70 Y=INT(Y-.5)
80 SET(2*(A+30),23-Y)
90 SET(2*(A+30),23+Y)
100 NEXT A
110 NEXT R
120 E=E+1
130 GOTO30
```

Listing 2

```
10 CLS:E=2
20 FOR R=E TO 22 STEP4: FO
RA=R TO R: X=R*R-A*A:Y=SQR
(X):Y=INT(Y-.5): Z=2*(A+30
):SET(Z,23-Y): SET(Z,23+Y)
: NEXTA,R: E=E+1: GOTO 20■
```

These listings only serve to demonstrate the author's coding techniques.—Ed.

@ News

Spencer Hall, Contributing editor

Cast of Characters

The American Standard Code for Information Interchange, known familiarly as ASCII, is a widely-used system of letting eight bit bytes (the numbers between zero and 255), stand for what is generated when the keys of a typewriter are struck. The Arabic numerals are represented by ASCII 48 through 57, English (Roman) capitals by ASCII 65 through 90, and lower case by ASCII 97 through 122. Bytes zero through 31 are used to represent the fancy stuff that special keys generate, such as backspace, break, line feed, carriage return, etc.

Outside the alphabet, standardization is less common. For a programmer, this can become a jungle in which the various peripherals just don't communicate.

Inside the alphabet range, the TRS-80 Model I has a few surprises too. POKE 15872,65 gives you a capital "A" in the middle of the screen, because address 15872 is in the middle of the video RAM. PRINT@512, "A" gets you the same thing, plus a READY prompt.

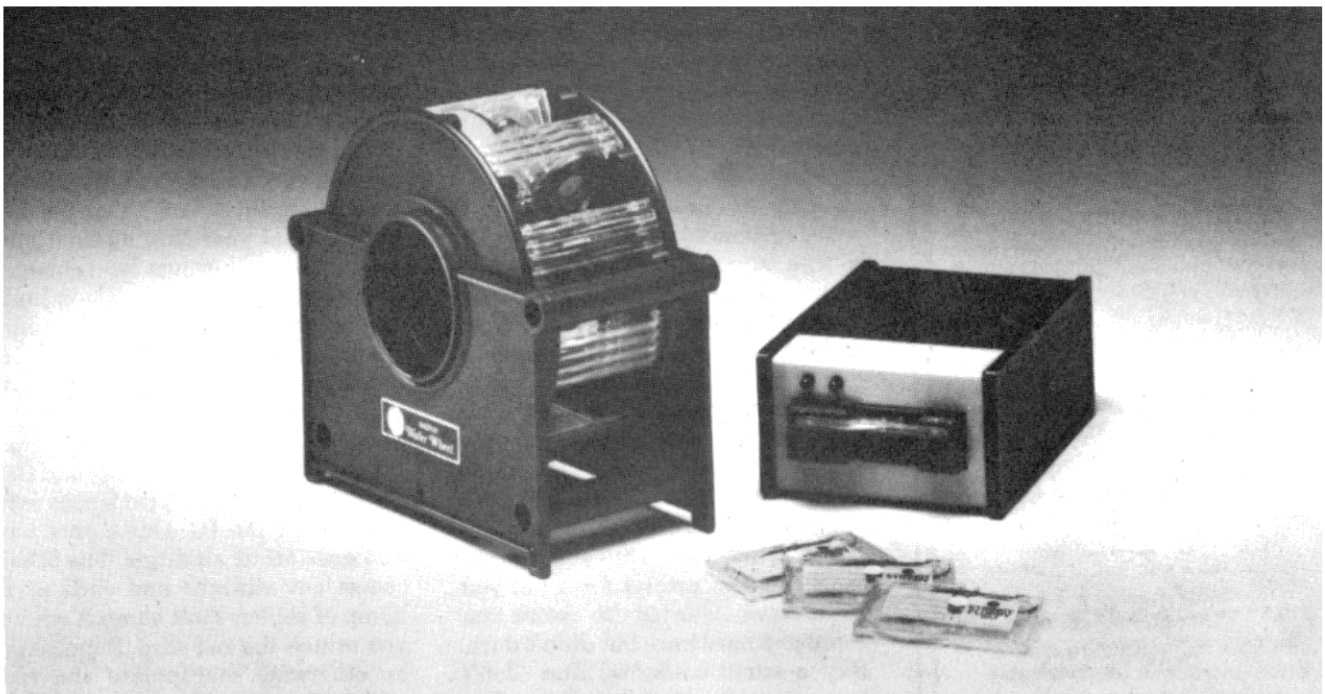
Now try a PRINTPEEK(15872). Surprise! You get a "1." "1" is an unprintable ASCII standing for the BREAK key. So why does it produce a capital "A" on the screen? Simply because Model I, without the lower case modification, has no chip for bit number 5 (2 raised to the sixth power = 64) installed in the video RAM. To compensate for this, Model I hardware automatically alters all bytes destined for the screen up to 31 by adding 64. Thus, "1" becomes "65," "2" becomes "66," etc.

In this same manner, ASCII capital letter codes are produced. How does it know that the number is no larger than 31? Simple! It checks to see if bit number 5, (2 raised to the fifth power = 32) is "on."

Any lower case modification starts by adding a chip for bit number 6 in the video RAM. If you have had a modification (a chip for bit number 6 in video RAM), you can install Exatron's new character generator with no trouble. It costs $19.95 plus $3.00 handling, and simply plugs into IC29 on your logic board. Wiring and mounting a switch to access its power is very simple. You will have reverse video (black letters on a white square), brackets, the caret, and a reverse slash available, as well as underlined capitals. You will be pleasantly surprised by the shape of the characters. The exclamation point is "fatter" than the stringbean your original chip produced.

Wafer Storage

Wafers take up a lot less space



An ingenious device for computerists: a wafer wheel!

than cassettes. But, let's face it — they are a mite harder to store than diskettes. Actually, a wafer occupies only about one and an eighth cubic inches versus one and seven-eighths cubic inches for a mini-floppy diskette. If you don't believe me, measure them for yourself. The problem with wafers isn't that they are larger than diskettes, but that they are . . . well . . . lumpier.

There are many options for storing wafers and ESFers all have different ideas about which is best. Bill Stokes, an eighth grade math teacher in Sunnyvale, has come up with what has to be the most elegant (if not the cheapest) solution of all. Doing business as SPEC (Specialty Products for Computers), he'll sell you an Exatron wafer wheel for $24.95 plus $2.00 shipping. Holding forty wafers, each accessible with a turn of the wheel, it has to be the most ingenious non-electronic device ever designed for a computer workplace.

Use quarter-inch plastic binder bars as wafer protectors and either write on the flat edge with an indelible marker, or paste on a label. Program names are always visible as you turn the wheel. Molded of chocolate brown ABS (that's Acrylonitrile Butadiene Styrene . . . now you know!), which is the best

plastic material available, it looks and works like a thoroughbred. If the price is too steep, just pass the word to that close relative who always asks you what you want for your birthday.

Beeper Blooper

My editorship of @NEWS is off to a fine start with (let's call it a misunderstanding) in my first issue. Exatron is not planning to sell or retrofit the EOT/BOT beeper I mentioned in June. Seems there is no need. Said beeper is a common item in most electronic supply stores. At Radio Shack, it's stock number 273-060 and sells for (grab your seats) $1.99.

Installing the beeper is an excellent first project for all of you who have wanted to work on computer hardware but didn't dare. Buy a small soldering iron (don't pay more than $6.00, or get one that uses more than 25-30 watts) and some resin core solder. Before you go to work, get some double-sided pressure-sensitive tape, and plastic or rubber pads.

Remove the two hex-head sheet metal screws at the top rear of the Floppy. Use a screwdriver that doesn't slip and damage the slot. Better yet, use a 3/16" socket driver. Put the screws where you can find them later! Remove the hex-head screw on the left side of the rear. Now, remove the screw on the left front side of the bottom, very near the rubber "foot." You can ease off the left plastic panel and slide out the top panel. You'll be looking down on the Floppy circuit board.

Page F-2 of your ESF Manual is a picture of what you see. Orient yourself by the words "Exatron T.R.S. 80 ADD-ON," which are printed on the board exactly as they appear in the book. Use the double backed stuff to mount your beeper on the "floor" under the PC board... There's plenty of room.

Next, "tin" the ends of the black and red wires from the beeper. What's "tin?" Hold the hot iron under the bare wire end for a moment to heat it. Keeping the iron in place, touch the wire from above with the end of the solder. It will flow and coat the bare wire. Now we solder!

The black wire is attached to the outside lead of the yellow moon-shaped condenser, near item (IC) 5. It's marked with the number 104, and outside means nearest the edge of the board. Crimp that bare end so it can be loosely attached to the condenser lead right at the board surface (chain-nosed pliers please, nurse). Hold your iron against one side of these wires just long enough to make the solder which you hold against the other side flow around the joint. Remove the iron, and in two or three seconds, jiggle the black wire to make certain it is firmly attached. Now attach the red wire.

Look at the board just beyond the inside end of IC 11. Two traces exit from under the IC. One comes out and goes off at an angle. The other comes out straight and ends at a bump of solder. That bump is where you attach the red wire. Bend, tape or otherwise manipulate the red wire so that its end stays in contact with said bump of solder. As a last resort, ask someone to hold it in place! Hold your iron against the point of contact. Between the tinned lead and the lump of solder, you may not need your own solder at all.

When the solder looks melted, withdraw the iron. In a moment, jiggle the wire to be sure it's firmly attached. Put the Floppy back together, top plate first. Replace the two screws which hold the top plate, being sure to get the front end under the gray edge of the main housing. You'll have to squeeze and cajole the body to fit the side plate squarely onto it.

If all this sounds hard, it isn't. I just made it sound hard by giving such detailed instructions. Your Floppy will now tell you audibly whenever it passes the EOT/BOT sticker.

So-Where's-the-Mailbag-Department

We promised to open the mailbag this time. That's the worst of sitting up here in Washington and writing @NEWS. We got into the June issue by the skin of our teeth... way past the deadline. Mike Schmidt said "never again," so the next deadline was only a few days off. The file of letters got sent fourth class, which equals parcel post, which in turn equals "slow boat to China." This time, I won't promise anything! ■

The computer revolution

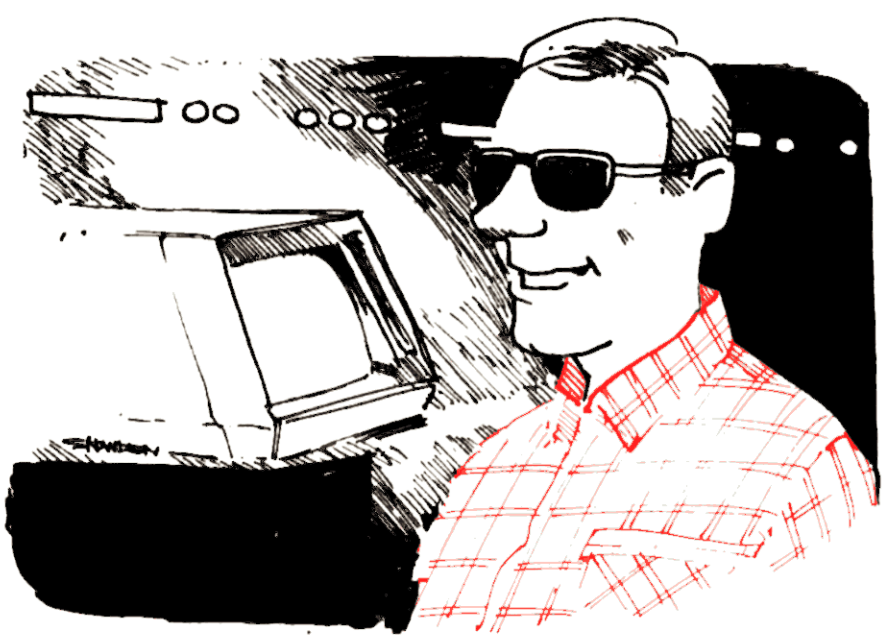
Who are the beneficiaries?

How microcomputerists can help the deaf/blind

—author's name withheld

If you are reading this, you are probably already one of the beneficiaries. You have discovered the intellectual excitement and satisfaction which the personal computer generates and provides. You have foregone nights at the opera, evenings at the club, trips to the country in autumn, and, yes, even television, to sit in front of your home computer getting it to work for you, play with you, discovering how it works, helping you to make a living, etc.

One of the things that my friend Ray hates the most is television. When he is with people who are watching TV, they do not want to bother with him, but Ray can't hear the sound coming from the television, nor can he see what is happening on the screen, for he is deaf and blind. Ray's sensory world is largely the world of touch, and when Ray happens to touch a TV, he quickly turns away from it as holding nothing of promise for him. If this article is successful, Ray's aversion to the TV will be a thing of the past.



I have a wonderful idea on how to turn just about any home computer into a computer for the deaf/blind. When you understand the simplicity of it, I want to challenge *you* to implement it on your home computer, with the result that the whole world of home computers will be opened to the deaf/blind programmer.

Ray has told me that most of the deaf/blind he has met are socially backward, preferring not to communicate as they pass through life almost in a stupor. Ray was fortunate to have a teacher, and fortunate to have parents that didn't give up on him and forced him to stay in school even though it would have been a lot easier on them if Ray was just content to sit in a corner for the rest of his life.

I am an amateur radio operator and I met Ray one day by chance many years ago. He surprised me by saying that he had been listening to about twenty other conversations (in morse code) and he decided that he wanted to talk to me. This certainly was an unusual statement. He quickly followed this up by telling me he was deaf and blind. I couldn't figure out how he was "hearing" me, so I asked him to explain. He said he had taken the grille off the loudspeaker, and could feel the vibrations of the cone with his fingers, and that he could copy morse code up to about 50 words per minute, which was a lot faster than I could. I had to apologize for sending so slow.

That was many years ago, and over the years I have met Ray in person many times. Ray is one of the few deaf/blind who is able to speak (like Helen Keller) and to understand your speech. He does this by reading your lips with his thumb and by feeling the vibrations in your throat with his index finger. It is amazing — and so difficult that not many deaf/blind are lucky enough to be able to learn it.

I want to reproduce the "Address of Miss Helen Keller" for you, because if you take up the challenge of implementing the idea of turning your home computer into a programmable machine for the deaf/blind, you will need a lot of staying power. The hardware, though simple — a morse code oscillator driven from an output port — must be supported by a program which is somewhat less simple. Here, then, is the address which Helen Keller delivered to the Association of Lions Clubs at Cedar Point, Ohio, USA, in 1925:

Address of Miss Helen Keller

Dear Lions and Ladies. I suppose you have heard the legend that represents opportunity as a capricious lady, who knocks at every door but once, and if the door isn't opened quickly, she passes on, never to return. And that is as it should be. Lovely, desirable ladies won't wait. You have to go out and grab 'em.

I am your opportunity. I am knocking at your door. I want to be adopted. The legend doesn't say what you are to do when several beautiful opportunities present themselves at the same door. I guess you have to choose the one you love best. I hope you will adopt me. I am the youngest here, and what I offer you is full of splendid opportunities for service.

The American Foundation for the Blind is only four years old. It grew out of the imperative needs of the blind, and was called into existence by the sightless themselves. It is national and international in scope and in importance. It represents the best and most enlightened thought on our subject that has been reached so far. Its object is to make the lives of the blind more worthwhile everywhere by increasing their economic value and giving them the joy of normal activity.

Try to imagine how you would feel if you were suddenly stricken blind today. Picture yourself stumbling and groping at noonday as in the night, your work, your independence, gone. In that dark world wouldn't you be glad if a friend took you by the hand and said, "come with me and I will teach you how to do some of the things you used to do when you could see?" That is just the kind of friend the American Foundation is going to be to all the blind in this country if seeing people will give it the support it must have.

You have heard how through a little word dropped from the fingers of another, a ray of light from another soul touched the darkness

of my mind and I found myself, found the world, found God. It is because my teacher learned about me and broke through the dark, silent imprisonment which held me that I am able to work for myself and for others. It is the caring we want more than the money. The gift without the sympathy and interest of the giver is empty. If you care, if we can make the people of this great country care, the blind will indeed triumph over blindness.

The opportunity I bring to you, Lions, is this: To foster and sponsor the work of the American Foundation for the Blind. Will you not help me hasten the day when there shall be no preventable blindness; no little deaf, blind child untaught; no blind man or woman unaided? I appeal to you Lions, you who have your sight, your hearing, you who are strong and brave and kind. Will you not constitute yourselves Knights of the Blind in this crusade against darkness?

I thank you.

The description of the Personal Computer for the deaf/blind is at the end of this article. I just wish to emphasize that your implementation should always keep in mind the communication between the deaf/blind and the sighted.

I see the implementation of this idea as opening up a new *inexpensive* world to the deaf/blind. Will not some of you programmers take up this challenge and become — to paraphrase the words of Hellen Keller — Knights of the Deaf/Blind?

Description of Personal Computer for the Deaf/Blind:

Output — A port with a switch. The switch being turned on and off at the rate to send Morse code characters. The deaf/blind programmer will connect his own code oscillator and touch device to the output of our switch.

Input — The keyboard of the Personal Computer.

Communication between the deaf/blind and sighted — The Video Display.

Program — The basic idea is this: When your machine is in the command mode in BASIC, you figure out how to patch a "preprocessor" into BASIC which allows your program to look for a special character, such as CONTROL T (=decimal 20 for the Radio Shack Model I). If the key hit is not CONTROL T, you pass the character along to the BASIC command interpreter as usual. If the character is CONTROL T, you take control away from BASIC and wait for the operator to hit NN "Enter" where NN is a LINE number — the number of the line displayed on the video screen, and "Enter" is the carriage return usually found in BASIC. When you get the NN "Enter," you convert the text found in RAM associated with the line number requested to Morse code. You make an extension to the Morse code for an unprintable character: Unprintable Character = . . - -, a sort of backward Z. Try to keep extensions to the Morse to an absolute minimum. Once you know you have an unprintable character, you can PEEK memory for its decimal representation and send out the morse for the numerical value.

Also, do not (under any circumstances) redefine the Morse code — since this is common ground to many people who are *not* deaf/blind. Try to keep in mind that the communication link between the sighted and the deaf/blind is what is shown on the video screen. The deaf/blind "sees" the screen using the simple protocol described above, and the sighted person "sees" the same thing by looking at the screen. This communication link is very important because when a sighted person hires a deaf/blind programmer, he must be able to communicate effectively with him; and when you are interviewed by a deaf/blind entrepreneur, you have a better chance of getting the job if you can communicate with him.

To accommodate different skill levels of deaf/blind reading the Morse code, you will have to include a command in the preprocessor to change the rate of sending out the Morse characters.

You should incorporate a Morse "feedback" sounding out every character hit on the keyboard.

You should probably incorporate a "stop sending" command to allow the operator to make notes in Braille. ■

MDX-2

Expansion interface evaluation

Model I

Mel Patrick, Surrey, B. C., Canada

Almost everyone who owns a TRS-80 Model I dreams of owning some kind of expansion interface to increase the versatility of their system. Cost, as well as availability, has hampered many from justifying the Radio Shack expansion interface. However, there are now economical alternatives to the Radio Shack expansion.

Many companies now offer their own version of the Radio Shack expansion interface. Some of these are assembled and tested units, while others are either available in kit, or bare P.C. (printed circuit) board form. By doing a little careful shopping, the builder can accumulate all the parts to assemble a bare P.C. board for less than an assembled version.

I was in the same position as many others who were upgrading their systems. Since cost, as well as on-board options, were important factors, I decided on the bare P.C. board. This, in turn, forced me to make another choice, because there were three different companies that offered their expansion as a bare board. First was the DISK 80 from MicroMint, second was the LNW, and last was the MDX-2. Each of these boards had much to offer, but the MDX-2 had the most, so I decided to order it.

I placed an order with Computex (321 El Dorado, Webster, Texas 77598) because they offered prepackaged parts for various sections of the board. All parts for the power supply and floppy disk section were ordered, along with the bare P.C. board and manual.

MDX-2

The MDX-2 has the standard 32K additional memory, floppy disk controller, dual cassette port, real time clock, and line printer port on the board. In addition, there is an on-board RS 232, 2K or 4K EPROM option and a modem.

Should you build an MDX-2? If you can solder neatly, read and follow instructions carefully, and have a desire to save yourself some money, I would have to give a definite yes.

The EPROM option allows the builder to disable the top 2K or 4K of RAM and replace it with either a 2716 (2K) or 2732 (4K) EPROM. Jumpers are provided on the board to configure the system.

The modem is a standard 300 baud, direct connect, and switch selectable to either Answer or Originate modes. It is connected to the RS 232 so that all standard software programs will work with it.

The board is accompanied by a 43-page manual to guide the builder through assembly, check out, and theory of operation. The board itself measures 10" by 12," has edge card connectors on three of its sides, and an RS 232 connector on the fourth side. Parts layout is not crowded and the board is clearly silk screened to provide visual reference to components.

Assembly

The builder has the option of constructing only the sections of the board that they require. There are some sections that require others for proper operation, i.e., the modem requires the RS 232, and the floppy disk section requires the real time clock and extra memory. Apart from these two, all others offer standalone operation.

The first pages of the manual are a system overview with all sections of the board described in detail. Following this are the lists and quantities of all components required. There is a sectional list should only certain options be desired.

I obtained most of the parts from local suppliers. They were the standard integrated circuits, resistors and capacitors. There are some 1% precision components that are required for the modem, and these are available direct from Micro-Design. In the manual, Micro-Design has thoughtfully written the names of four companies who are

able to supply components for the board should you be unable to obtain them locally.

Considering the complexity of the circuitry on the board, assembly progresses smoothly through all stages. To begin construction, all of the sockets for the ICs are soldered on the board. The reason for sockets is simple: it makes any repairs a simple chore of removing the suspected IC and replacing it with another. There are 77 sockets of various sizes to be mounted before the resistors and capacitors can be installed. The last section to be constructed is the power supply.

Upon completion of the board, the manual has a test to be performed on each section of the board. A voltmeter is required to verify power supply operation and proper voltages. There is also a program that can be entered using T-BUG to test the RS 232.

I completed construction of my board in about 6 hours.

Power Up

I connected a short 40-pin ribbon cable from the keyboard buss to the MDX-2 and powered up the system. The screen filled with garbage and locked up. Right. I had forgotten to hold down the BREAK key. I pressed the reset button and was rewarded with the memory size prompt. I pressed ENTER and then typed in PRINT MEM. Although I had all of the additional RAM installed, memory size was only 15570. What happened? Probably a very common error when anyone builds a board like this. I had missed one solder connection. After soldering it and rechecking all of my other connections, I found another omission in the floppy disk circuit. This was soldered as well, and I proceeded with the tests outlined in the manual.

All of the remaining sections checked out except the floppy disk option. The disks would power up, but DOS would not load in. The manual states that a 74504 must be used to generate the 1 MHZ clock for the disk controller chip. Since this took place on a Sunday, I waited until the next day to contact Micro-Design about the problem.

I feel that I must commend Micro-Design on their eagerness to help anyone who has problems, or just needs information. When I contacted them it was 8:00 p.m., and instead of receiving a recording telling me to call during office hours, I was put directly in contact with Mike Shapiro, president and designer of the MDX series boards. He suggested I replace the 74504 with a 74LS04. I did that and the disk worked perfectly.

Mr. Shapiro stated that Micro-Design does not repair their own boards but, rather, has three service centers in the United States and one in Canada. Should you be unable to repair your own board, they will supply you with the address of the nearest repair center. He also went on to mention that Micro-Design tries to keep one of three technicians available from the time the main office closes until about 9:00 p.m., should anyone require help after business hours.

Pros and Cons

Since the MDX-2 board has so many valuable options, it is quite possible to overlook its two minor weak points. First, Micro-Design does not offer any kind of case to house the board. However, there is another company who advertises a "demi-case" at a relatively low cost.

Second, is the fact that there are connectors on all four sides of the board. This can prove to be awkward if you are connecting and disconnecting cables. These are minor items and the boards options make up for this.

Summary

Should you build an MDX-2? If you can solder neatly, read and follow instructions carefully, and have a desire to save yourself some money, I would have to give a definite *yes*. After I had calculated how much I spent on this project, I could have afforded to send it to a repair center and would have still come out ahead. With the support that they offer, I feel that Micro-Design is one step ahead of the others. ■

MDX-2 bare printed circuit board $74.95, complete kit $269.00.

Don't throw your Model I away— double it!

Percom II and LNW Doubler evaluation

Model I

John Strader, CPA, Lacey, WA

I was in need of more disk space. It became very evident that I would either have to purchase a new computer, or stay with my Model I system. I am sure many of you have had to make the same decision. The Model I Radio Shack computer still has a lot of life in it yet.

In reviewing the Model III and the Model II, it became evident that the Model I was still the most cost effective to use in my business. I needed at least two systems and I already had a Model I and a significant number of programs. However, I needed additional storage capability and decided to obtain 40-track disk drives and a double density board to give me approximately 170,000 bytes per disk for my Model I.

The two double density boards available at the time were the Percom Doubler II from Percom and the LNW board. Each of the boards came with a different disk operating system. Percom uses DOUBLEDOS and has several patches that are for sale for the older NEWDOS operating systems. LNW comes with DOSPLUS. Of the two operating systems, I prefer DOSPLUS because it has more commands and expanded operating features in it. These are especially nice for programmers. Both software systems worked for us with very little trouble.

Some of the commands and features that DOSPLUS has, are as follows:

1. A spooler — I haven't used it very much; however, it appears to work.

2. CMD"M" — This feature displays all of the variables and the values that they are set to. This is an excellent feature of DOSPLUS for programmers.

3. The trace function steps you through each command one at a time which is very useful in debugging a program.

4. CMD"SR" — This will search for and/or replace any variable, string or command.

5. It has an extended directory and map feature.

6. It has a route function which allows you to do such things as change the screen output to the printer. For instance, if you changed the screen output to the printer and letter was typed on the keyboard , that letter would be printed by the printer.

7. It has a tiny BASIC which will allow 40K of available user memory in the machine.

8. It also has a "restore" function to restore killed disk files.

The DOSPLUS system has the majority of features that NEWDOS 80 Version 2.0 has — a renumber command, duplicate commands, upper and lower case driver built in, and a purge command. About the only things it doesn't have are some of the system commands and the assembler sort that NEWDOS has built into it.

The Percom DOUBLEDOS doesn't have any significant features other than being able to read both double and single density disks. However, it appears to work very well. The Percom installation documentation was much better in explaining the operation of the doubler. Both systems install in the same way. They are very easy to install — you just plug them in; no soldering or cutting wires. If you need to get your expansion interface fixed, just pull the board. However, you should be aware that once the seals are broken, your warranty may be altered.

After you have installed one of the double density boards and have over 135,000 bytes of storage on a DOS disk, and 170,000 bytes on a formatted data disk, it is like going from a compact car to a Cadillac. The additional space makes the machine much more useful and reduces the number of disks necessary to handle large data files.

We have used the Model I systems with the boards for over three months and have had no trouble with either system. Almost all of our programs have operated on them (with the exception of the Electric Pencil which needs to be patched. DOUBLEDOS comes with a patch for Electric Pencil). Scripsit must be patched and both software systems have a patch for it. DOSPLUS also has a patch for Profile and Visicalc on Model III.

The Model I doesn't need to be patched when operating in double density mode. If you are using assembly language programs, it appears that they may have to be patched if they access files through a non-standard DOS call. Before buying a double density board, I suggest that you contact the manufacturer, the DOS supplier, or the assembly language programmer, concerning the need for patches if you have a favorite assembly language program. BASIC programs work fine unless they have a non-standard assembly language call which saves data to disk.

Another advantage is that the double density makes the Model I more compatible with the Model III. A friend who has a NEWDOS 80 Version 2.0 operating system, and a double density board, was able to read double density disks from a Radio Shack Model III system with no problems after following the NEWDOS 80 manual's instructions. I was also pleasantly surprised to find out my older Radio Shack disks would operate in double density mode although they were only 35 tracks. However, some of the older disks will only handle single density. You can check with a Radio Shack repairman to determine if your disks are double density. The new Radio Shack disk drives are 40-track double density.

After using both boards, my personal choice would be to buy the LNW if I didn't have a double density operating system. If you already have a double density DOS which you like and are using, then I would buy strictly on price since all the disk operating systems we tried work on either system. We have had no problems with either of the double density boards.

One final caution — I purchased my LNW board from an outside supplier. It took me two telephone calls and one letter to get delivery a month and a half after my check cleared. I suggest that you try to check out the supplier, if you don't order from the factory, since the cost for each board will vary between $149 and $179.

LNW has also come out with a double density board that will handle five- and eight-inch floppies for $219. I haven't used that board, however. Aerocomp also has come out with a double density board which we didn't test. In ads, Aerocomp claims that the double density boards of both LNW and Percom are not as reliable under a "worst case" situation. In everyday use, at least so far, we didn't have any read or write problems. Percom stated in their documentation that their Doubler I was unreliable after a year or so of operation because of the type of electronic system design they were using. The Doubler II uses a different type of system. It is my feeling that under normal conditions the boards are similar and should work with little difficulty. ■

Files and foibles

Stacks

Model I, II, III

In order to proceed with binary tree structures for files, we have to learn a few other programming tricks. These are useful for more than just file operations, so the demonstration this month doesn't deal with files at all.

We are going to go back to the subject of *stacks*. We've used them already in this series, dealing with ways of recovering space in a file when we delete records. We've also covered them in other articles, mostly those dealing with FORTH as a programming language. A stack is a simple "data structure" that organizes the information you put into it in a way that makes it convenient to get it back in the order opposite to that in which we put it in.

If we look at a stack as a black box for the moment, we can define two "operations" that affect the stack. The first is PUSH. When we PUSH something, we are putting it on the "top" of the stack. The second operation is called POP. When we POP the stack, we take the top item off and return it for use. That's all we can do, PUSH and POP, but it's enough.

Let's look at a simple stack and see what we can do with it. The table below shows some of the effects:

Operation	Stack			
	V-top			
Beginning	Stack empty			
PUSH 25	25			
PUSH 50	50	25		
PUSH 75	75	50	25	
PUSH 75	75	75	50	25
POP	75	50	25	
POP	50	25		
PUSH 10	10	50	25	
POP	50	25		
POP	25			
PUSH 10	10	25		
POP	25			
POP	Stack empty			

Each operation (PUSH or POP) affected only the top of the stack. In other words, we dealt with numbers on a Last In-First Out basis (LIFO). Hewlett-Packard (HP) calculators do everything this way. For those of you who are devotees of the HP, there's no need to say more. If you have just never worked with an HP, you may still be skeptical.

All other things aside, stacks find frequent use. Our own routine several months ago to recover deleted disk space is one example. Compilers frequently use stacks to understand algebraic equations. Stacks can be used for problems that have to deal with backtracking. Stacks are used by compilers and interpreters to remember subroutine call points (that's how you get back to where you were when you do a GOSUB).

Most important to us at the moment—a stack will be necessary to be able to print out a binary tree in alphabetic order. This happens because we'll be programming what is known as a recursive procedure (a subroutine that can call itself) in BASIC, which doesn't have that capability.

Stacks in BASIC

In order to get a stack in BASIC, we can use an array and a special index for the array called the "stack pointer" (SP). When we start out (initialize), we set the stack pointer to zero.

In order to PUSH something onto the stack, we add one (1) to the stack pointer and then put the item we want to add into that position in the array. When the array is full, we say the "stack is full."

POPping off the stack involves taking the item from the array at the index equal to the stack pointer, and then decreasing the stack pointer by one. When the stack pointer is zero, we say the "stack is empty," and can't go any further. There's really nothing more to tell. All operations off the stack are done in terms of PUSHes and POPs.

The program which follows, allows you to play with a stack interactively and even programs some simple operations (+, -, *, /) so that you can see how things are done in terms of a stack. The operations provided are PUSH and POP, of course, but we also have PRINT. When PRINT appears as a command, the stack pointer is printed and then the whole stack is printed, with the top at the left.

The program works by accepting commands in a free format (very similar to FORTH). You can even have more than one command per line. By trying out commands with PRINT interspersed between them, you can see the effect of various commands on the stack. (Note: PUSH is just a dummy command, since all numbers are automatically PUSHed).

Try this one:

```
25 35 45 PRINT + PRINT POP PRINT
```

The whole idea is to *play* with the stack concept and see how it's programmed. In particular, you want to know how numbers are put into the stack, and what order they come back out. If you don't like HP

calculators and FORTH when you start, you may become addicted before you're through.

Actual Run

```
>>25 35 45 PRINT + POP PRINT
STACK POINTER ==> 3
TOP OF STACK ===> 45  35  25
POP'ED OFF  80
STACK POINTER ==> 1
TOP OF STACK ===> 25
>>POP
POP'ED OFF  25
>>POP
STACK EMPTY
>>POP
STACK EMPTY
>>HELP
ERROR - INVALID COMMAND: HELP
>>25 35 45 HELP
ERROR - INVALID COMMAND: HELP
>>PRINT
STACK EMPTY
>>25 35 45 PRINT
STACK POINTER ==> 3
TOP OF STACK ===> 45  35  25
>>+ - PRINT
STACK POINTER ==> 1
TOP OF STACK ===> 55
>>25 35 45 PRINT
STACK POINTER ==> 4
TOP OF STACK ===> 45  35  25  55
>>+ PRINT - PRINT
STACK POINTER ==> 3
TOP OF STACK ===> 80  25  55
STACK POINTER ==> 2
TOP OF STACK ===> 55  55
>>1 2 3 4 5 6 7
>>+ PRINT
STACK POINTER ==> 8
TOP OF STACK ===> 13  5  4  3  2  1  55  55
>>- PRINT
STACK POINTER ==> 7
TOP OF STACK ===> 8  4  3  2  1  55  55
>>/ PRINT
STACK POINTER ==> 6
TOP OF STACK ===> 2  3  2  1  55  55
>>* PRINT * PRINT + + PRINT
STACK POINTER ==> 5
TOP OF STACK ===> 6  2  1  55  55
STACK POINTER ==> 4
TOP OF STACK ===> 12  1  55  55
STACK POINTER ==> 2
TOP OF STACK ===> 68  55
>>END
```

Files

BASIC Listing
Files & Foibles — Stacks

```
10 REM**********************************
20 REM
30 REM          STACK OPERATION DEMO
40 REM          BY TERRY R. DETTMANN
50 REM
51 REM          THIS PROGRAM IS DESIGNE
D SO THAT ALL LINES NUMBERED IN
52 REM          TENS ARE REQUIRED. LINE
S NOT NUMBERED IN TENS CAN BE
53 REM          ELIMINATED WHEN TYPING
IN THE PROGRAM
60 REM
70 REM**********************************
75 REM          CLEAR SOME STRING SPACE
 AND SET THE STACK MAXIMUM SIZE
80 CLEAR1000:MX=50
85 REM          PC$ IS THE LIST OF POSS
IBLE COMMANDS. THE SPACES BETWEEN
86 REM          COMMANDS ARE REQUIRED!
FOR EACH COMMAND THERE SHOULD BE
87 REM          A TOTAL OF 5 SPACES INC
LUDING THOSE TAKEN UP BY THE
88 REM          COMMAND, SEE THE PARSER
 AT LINE 600 TO SEE HOW THIS FACT
89 REM          IS USED TO TELL WHICH C
OMMAND IT IS
90 PC$="+    -    *    /    PUSH POP  P
RINTEND  "
95 REM          INITIALIZE THE STACK
100 GOSUB1000
190 CLS:PRINT"STACK OPERATION DEMONSTRA
TION"
200 REM - - - - - - - - COMMAND LOOP - -
- - - - - - - - - - - - - - - - - - -
205 REM         PRINT A PROMPT AND INPU
T A COMMAND LINE
210 PRINT">>";:GOSUB400:PRINT
215 REM         GET THE NEXT COMMAND WO
RD
220 GOSUB600
225 REM         EXECUTE THE NEXT COMMAN
D WORD
230 ON CD GOSUB 2000,2100,2200,2300,240
0,2500,2600,2700,2800
235 REM         IF A POP WAS DONE, THEN
 PRINT WHAT CAME OFF THE STACK
236 REM         UNLESS THERE WAS A STAC
K ERROR
240 IF CD=7 AND EF=0 THEN PRINT"POP'ED
OFF ";X
245 REM         IF THERE WAS A STACK ER
ROR, THEN DISREGARD THE REST OF
246 REM         THE LINE AND ZERO THE S
TACK POINTER
```

```
250 IF EF=1 THEN LN$="":SP=0
255 REM .          WHEN THE WHOLE LINE IS
COMPLETE, THEN GET ANOTHER
256 REM            OTHERWISE GET THE NEXT
WORD
260 IF LEN(LN$)<>0 THEN 220
270 GOTO200
400 REM - - - - - - - - GET A COMMAND LIN
E - - - - - - - - - - - - - - - - - -
405 REM            BLANK OUT THE LINE
410 LN$=""
415 REM            GET A CHARACTER, IF IT
IS <ENTER> THEN WE'VE GOT IT ALL
420 GOSUB500:IF ASC(C$)=13 THEN RETURN
425 REM            IF IT WAS A BACKSPACE,
THEN BACKSPACE ON THE SCREEN
426 REM            AND TAKE THE LAST CHARA
CTER OUT OF THE LINE BUFFER
430 IF ASC(C$)=8 THEN IF LEN(LN$)>0 THE
N PRINTC$;:LN$=MID$(LN$,1,LEN(LN$)-1):G
OTO420
435 REM            OTHER CONTROL CHARACTER
S AREN'T ALLOWED
440 IF ASC(C$)<=31 THEN 420
445 REM            IF IT MADE IT TO HERE,
ADD IT TO THE LINE BUFFER AND
446 REM            PRINT IT ON THE SCREEN,
 THEN GO GET ANOTHER
450 LN$=LN$+C$:PRINTC$;:GOTO420

500 REM - - - - - - - - GET A CHARACTER -
 - - - - - - - - - - - - - - - - - - -
505 REM            GET A SINGLE CHARACTER
FROM THE KEYBOARD
506 REM            WAITS HERE UNTIL IT GET
S ONE
510 C$=INKEY$:IFC$=""THEN510 ELSE RETUR
N
600 REM - - - - - - - - PARSE NEXT WORD F
ROM LINE - - - - - - - - - - - - - - -
605 REM            REMOVE BLANKS FROM THE
BEGINNING OF THE LINE
610 IF MID$(LN$,1,1)=" " THEN LN$=MID$(
LN$,2):GOTO610
615 REM            IF THERE'S NOTHING LEFT
, THEN DONE
620 IF LN$="" THEN RETURN
625 REM            LOOK FOR A BLANK, IF TH
ERE IS ONE, GET THE WORD UP TO THE
626 REM            BLANK, OTHERWISE WHAT'S
 LEFT IS THE WORD WE WANT
630 L=INSTR(LN$," "):IF L<>0 THEN CD$=M
ID$(LN$,1,L-1):LN$=MID$(LN$,L+1) ELSE C
D$=LN$:LN$=""
635 REM            LOOK FOR THE WORD AMONG
 THE COMMANDS, IF IT'S THERE THEN
636 REM            SET CD TO INDEX TO THE
SUBROUTINE, OTHERWISE ASSUME IT'S
```

Files

```
637 REM          A NUMBER TO START WITH
640 L=INSTR(PC$,CD$):IF L=0 THEN CD=1 E
LSE CD=INT(L/5+1)+1
645 REM          PUSH IS A LEGAL COMMAND
 THAT WE SIMPLY IGNORE BECAUSE WE
646 REM          AUTOMATICALLY PUSH ALL
NUMBERS ON THE STACK
650 IF CD=6 THEN CD=0
660 RETURN
1000 REM - - - - - - - - INITIALIZE STACK
 - - - - - - - - - - - - - - - - - -
1005 REM                 SET UP THE STAC
K ARRAY AND ZERO THE STACK POINTER
1006 REM                 THE STACK WILL
ADD NUMBERS ABOVE THE CURRENT ONE
1007 REM                 POINTED TO BY T
HE STACK POINTER
1010 DIM STK(MX):SP=0
1020 RETURN
2000 REM - - - - - - - - NUMBER - - - - -
 - - - - - - - - - - - - - - - - - - -
2005 REM                 IF WE CAN INTER
PRET THE WORD AS A NUMBER (STARTS WITH
2006 REM                 A DIGIT, +, -,
OR .) THEN PUSH IT TO THE STACK
2007 REM                 OTHERWISE PRINT
 AN ERROR MESSAGE SAYING THAT IT
2008 REM                 AN ILLEGAL COMM
AND AND SET THE ERROR FLAG
2010 EF=0:C$=MID$(CD$,1,1):IF (C$>="0"
AND C$<="9") OR C$="+" OR C$="-" OR C$=
"." THEN X=VAL(CD$):GOSUB2510 ELSE EF=1
:PRINT"ERROR - INVALID COMMAND: ";CD$
2020 RETURN
2100 REM - - - - - - - - PLUS - - - - - -
 - - - - - - - - - - - - - - - - - - -
2105 REM                 GET THE TOP TWO
 NUMBERS OFF THE STACK, AS LONG AS
2106 REM                 THE STACK IS NO
T EMPTY, EVERYTHINGS OK, OTHERWISE
2107 REM                 WE'RE DONE
2110 GOSUB2600:X1=X:IF EF=1 THEN RETURN
2120 GOSUB2600:X2=X:IF EF=1 THEN RETURN
2125 REM                 ADD THE NUMBERS
 TOGETHER AND PUT THE RESULT ON
2126 REM                 THE STACK
2130 X=X1+X2:GOSUB2500:RETURN
2200 REM - - - - - - - - MINUS - - - - -
- - - - - - - - - - - - - - - - - - - -
2205 REM                 AGAIN, 2 NUMBER
S OFF THE STACK, RETURN IF STACK
2206 REM                 EMPTY
2210 GOSUB2600:X1=X:IF EF=1 THEN RETURN
2220 GOSUB2600:X2=X:IF EF=1 THEN RETURN
2225 REM                 SUBTRACT THE SE
COND NUMBER FROM THE TOP OF
2226 REM                 THE STACK AND P
USH THE RESULT BACK TO THE STACK
```

```
2230 X=X1-X2:GOSUB2500:RETURN
2300 REM - - - - - - - MULTIPLY - - - -
 - - - - - - - - - - - - - - - - - - -
2305 REM                    GET TWO NUMBERS
, RETURN IF STACK EMPTY
2310 GOSUB2600:X1=X:IF EF=1 THEN RETURN
2320 GOSUB2600:X2=X:IF EF=1 THEN RETURN
2325 REM                    MULTIPLY THEM A
ND PUSH RESULT ON THE STACK
2330 X=X1*X2:GOSUB2500:RETURN
2400 REM - - - - - - - DIVIDE - - - - -
 - - - - - - - - - - - - - - - - - - -
2405 REM                    GET TWO NUMBERS
, RETURN IF STACK EMPTY
2410 GOSUB2600:X1=X:IF EF=1 THEN RETURN
2420 GOSUB2600:X2=X:IF EF=1 THEN RETURN
2425 REM                    DIVIDE THE TOP
BY THE SECOND, RESULT TO THE STACK
2430 X=X1/X2:GOSUB2500:RETURN
2500 REM - - - - - - - PUSH - - - - - -
 - - - - - - - - - - - - - - - - - - -
2505 REM                    CHECK FOR FULL
STACK, SET ERROR FLAG IS FULL
2510 EF=0:IF SP>=MX THEN PRINT"STACK FU
LL":EF=1:RETURN
2515 REM                    INCREMENT THE S
TACK POINTER AND PUT THE NUMBER
2516 REM                    IN THE NEW PLAC
E
2520 SP=SP+1:STK(SP)=X:RETURN
2600 REM - - - - - - - POP - - - - - -
- - - - - - - - - - - - - - - - - - -
2605 REM                    CHECK FOR STACK
 EMPTY, SET ERROR FLAG IF IT IS
2610 EF=0:IF SP<=0 THEN PRINT"STACK EMP
TY":EF=1:RETURN
2615 REM                    GET THE TOP AND
 DECREMENT THE STACK POINTER
2620 X=STK(SP):SP=SP-1:RETURN
2700 REM - - - - - - - PRINT - - - - -
- - - - - - - - - - - - - - - - - - -
2705 REM                    CHECK FOR STACK
 EMPTY CONDITION
2710 IF SP=0 THEN PRINT"STACK EMPTY":RE
TURN
2715 REM                    PRINT STACK POI
NTER AND THEN STACK
2716 REM                    STACK IS PRINTE
D WITH TOP TO THE LEFT BY POINTER
2720 PRINT"STACK POINTER ==>";SP
2730 PRINT"TOP OF STACK ==>";
2740 FOR I=SP TO 1 STEP-1:PRINTSTK(I);:
NEXTI:PRINT
2750 RETURN
2800 REM - - - - - - - END OF PROGRAM -
 - - - - - - - - - - - - - - - - - - -

2810 CLS:PRINT"BYE NOW":END
```

■

MMSFORTH Version 2.0
Model I/III, 32K with disk
Miller Microcomputer Services
61 Lake Shore Road
Natick, MA 01760
(617) 653-6136
$129.95

Like a lot of other people I know, I was introduced to the FORTH language by the August, 1980 issue of *Byte* magazine. At that time, I was not very impressed with it. After all, who wanted that much control over their computer? I put it aside as 'just another computer language.'

Over time, however, I kept finding articles on FORTH and, much to my surprise, found myself reading every one and even rereading several. Eventually, I had to face the truth: FORTH had reached out and grabbed me. Just the thought of having that much power over my computer was thrilling.

Finally, I could take it no longer, I had to have FORTH on my Model I. With credit card in hand, I boldly called Miller Microcomputer Services and confessed my obsession. The nice lady who had answered the phone was understanding and, after taking my credit card number, assured me that a copy of MMSFORTH would be on its way to me as soon as possible.

Four days later, it arrived. (Luckily for me, it was a Friday. I don't think I would have been much good at work the next day!)

Unlike most people with a new toy, I actually read the documentation before playing, so I settled into my reading chair and went to it. On the table of contents page was written this heart breaking notice: "** These Appendices are provided upon return of your properly completed MMSFORTH License Agreement." Sure enough, what I would consider the most interesting Appendices were missing. There was still more than enough information in the manual for me to get started, but I had this feeling of incompleteness.

I immediately filled out the single user License Agreement, put a stamp on it, and walked down the block to the nearest post box and mailed it. (I received the missing Appendices the next Friday.)

The manual is one of the best third party pieces of documentation I have seen for the TRS-80. Similar to the TRSDOS manual, the first part of the manual is a tutorial of how to use the language, while the second part (the Appendices) is an excellent reference manual. The first thing the manual has the new MMSFORTHer do is back up the master disks. Smart thinking! With MMSFORTH, the programmer is *king* and what the programmer orders, the computer does — even if it means destroying the master disks! As it says in the manual: "IBM says 'THINK!', MMSFORTH says 'BACKUP!'."

Once the neophyte MMSFORTH programmer (me, in this case) has working copies of the master disks and the masters are safely tucked away, the manual proceeds to teach the potential MMSFORTHer how to "go FORTH." In a short time, I was doing fairly complicated programming with very little pain. And when the other Appendices arrived, I could begin programming in earnest.

The two supplied disks, labeled "SYSTEM" and "PROGRAM," are almost completely full. One of the Appendices (and not one of the missing ones) contains an index to their contents. The "SYSTEM" disk contains well over half of the source code for the MMSFORTH language. It is through the use of this source code that the MMSFORTHer can customize a version of MMSFORTH to the exact needs required. Do you have a serial printer and the Electric Pencil lowercase mod? Just select the right options on the option select block, and you can generate a new version of MMSFORTH which supports them both. Do you want to be able to type in a line before the computer is ready, like when the disks are being accessed? Again, just select the option and regenerate your MMSFORTH!

I should point out that this source contains an 8080 assembler, a full screen editor for source code, printer drivers, keyboard drivers, and interrupt routines — many things that a typical user wouldn't need, but a programmer quickly develops a need for.

The "PROGRAM" disk contains several examples and games in MMSFORTH (again, with the source code). Do you have a single disk system? Don't worry, MMSFORTH does not use overlays like TRSDOS does. Once MMSFORTH is loaded, you can take your system disk out of the drive and put a data disk in! (Warning: MMSFORTH, as it comes, does not read TRSDOS files. However, if you are clever, and knowledgeable about the TRSDOS file formats, you can transfer the data from TRSDOS to MMSFORTH.)

Also available from Miller Microcomputer Services are several utilities and packages written in MMSFORTH. One package is a personal database management system called "THE DATA-HANDLER," which I understand is very good, but I have not seen it. Another package is a "communications module" called "FORTH-COM," which I have seen and is excellent. (I use a slightly modified version of it to talk to an HP-3000 at 1200 baud without any problems.) Some of the utilities available are a full Z80 assembler, floating point numbers, and a word cross reference generator that can come in handy, especially during the testing of application systems.

If you have a TRS-80 Model I or III and have picked up an itch for FORTH, I have no qualms about recommending MMSFORTH to you. The manual is excellent, the language is excellent, and you will enjoy the feeling of power you have over your computer.

Anthony F. Pepin

Bug +
Model III, $14.95
Soft Sector Marketing
6250 Middlebelt
Garden City, MI 48135
(800) 521-6504

After the Model III microcomputer came out some time ago, many software developers saw a large new market. This market consisted of the people who moved up to a Model III computer, but soon found that many of their programs simply wouldn't work on their Model III as they did on the Model I, (such as Radio Shack's Editor/Assembler, Micro Movie, and of course, T-BUG).

Up to this point, there have been several solutions to the Model III user's problem of Radio Shack's old

T-BUG not working on his system. Among these are the program "TASMON," several modifications for the old T-BUG published in various magazines, "DEBUG" from Radio Shack, and what I'll be reviewing here: "Bug+."

Bug+ is a relatively new, inexpensive program for the machine language buffs of the world who do not yet have a monitor for their Model III. It is made by Soft Sector Marketing, a Michigan firm famous for its high quality games and utilities. Bug+ is certainly a fine addition to the family of software from SSM.

Some of the new features that set Bug+ apart from Radio Shack's old T-BUG (besides the fact that it will work on the Model III) are truly quite helpful. For instance, the program can be relocated anywhere in memory with the aid of a program which is actually a part of Bug+. With this, the user may "make way" for any programs whose entry points may overlap that of Bug+. In addition, a registration sheet is included with the 11 pages of documentation so that any purchaser of the program may register that he owns the program and will be notified of any modifications or enhancements.

The commands are fairly complete and simple to use. To begin with, there are the loading and saving commands. Pressing "L", for "Load", will take the first machine language program found on tape and load it into memory. In addition to this, it will display the filename, starting address, ending address, and the entry point of the machine language program. There are even special error messages displayed if there are any problems encountered in loading.

Second, there is the "V" command, for "Verify." As one might assume, this will just check a program, byte for byte, to make sure that the saved program is okay. The "P", for "Punch" (save), will save the program in memory to tape. Any of the commands can be aborted by pressing "X" (very handy if you've ever accidentally hit the wrong key while in a monitor program). Bug+ even has its own routine to save and load programs instead of accessing a ROM address. In fact, very few ROM addresses are used, making for a larger program.

Controlling execution is another integral part of any monitor and Bug+ has several commands for that. Pressing "B" will establish a break point for the program in memory, giving the user the power to temporarily stop a program at a particular point. The "F" command will remove the break point ("Fix" it) and "C" will continue executing any program after a break has been encountered. The "N" will set up a "Next break point," and "E" will end the execution of the machine language program in memory.

Executing certain parts of memory (jumping to parts) can be accomplished with Bug+, also. To jump to a section of memory, the user need only to press "J" and a hexadecimal number to go to, (which I think is rather limiting, since many people work with

decimal also, and would appreciate having the capability to enter decimal instead of hexadecimal). The "G" will go directly to the memory location that the PC register (program counter) is set to.

Viewing and changing the memory (the main reason for having a monitor in the first place) is accomodated with two commands. The "M" will display the contents of any memory location specified, and if the up or down arrow is pressed, the next higher or lower memory location can be seen also. Hitting "R" will display all of the main, alternate and special registers.

I believe that for the price, Bug+ is an adequate program for the medium-serious machine language programmer. It is neither a "bare bones" nor "full scale" monitor, but because of its very moderate price, I can still recommend it. (Incidentally, a demonstration program is included for the user of Bug+ to try out his new monitor!)

Tim Knight

GRBASIC
(Graphic BASIC)
Model I/III, $69.00
Med. Systems Software
P.O. Box 3558 Chapel Hill, NC
27514, (800) 334-5470

I've had a Model III computer for quite awhile now, and it has worked great for me. It has been much more reliable and versatile than the first Model I that I bought a few years ago. The only thing I always wished that the Model III had, was better graphics, or at least faster access to what it does have, and in a more direct manner. I finally broke down and splurged for the TRS-80 Color Computer a few months ago, and am very impressed with it. The Extended BASIC 16K version I have has great BASIC graphic commands like CIRCLE, DRAW, SOUND, etc. These (and other similar commands) make programming the Color Computer very enjoyable.

Don't get me wrong. I still have (and love) my Model III. Matter of fact, I use it much more than I do my Color Computer because of its being more business oriented (64 character lines versus 32) and having a lot more software available at this point. After a while of programming graphics on the Color Computer, I do miss the Extended BASIC graphic commands when I get back to the Model III. Granted, the Model III wouldn't perform as well as the Color Computer, even if it had some of the Extended BASIC commands, because of the low resolution and non-color capabilities (unless it was modified by one of the high-res boards now available), but just having some of the commands at my disposal would still be nice.

Just yesterday, I received a package in the mail from Med Systems Software. It was a program called GRBASIC that I had ordered. I wanted a version that would be compatible with the Model III and the DOSs I use.

The disk writing said, "GRBASIC 3.0 — Model I. Runs on Model III with Convert." It looked like I was in business. I inserted a copy of DOS-PLUS into drive zero, put the

GRBASIC disk into drive one, and converted the program over. (There are three versions on the disk for 16K, 32K and 48K). I renamed GRBAS48/CMD to GRBAS/CMD, since my machine was a 48K model.

Reading the manual as I went along, I typed GRBAS/CMD. This started the disk drive, and a few seconds later BASIC appeared on my video. As stated in the manual, GRBASIC integrates itself into BASIC and protects itself in high memory. Nothing strange so far.

I started going through some of the examples and it looked really nice. The first command I tried was LDRAW(S) 0,0 to 127,47. Immediately, a line streaked across my screen. Great! Then I tried the LDRAW(R) 0,0 to 127,47 variation. By replacing the (S) with a (R), it erases instead of drawing the line. Good. The next command was CIRCLE(S),x,y,z.

This is where I ran into a problem. What was supposed to be a circle was not. I tried it a few more times with no luck. I finally surmised that I should try it using TBASIC. TBASIC is DOSPLUS's tiny BASIC and is supposed to be compatible with TRSDOS BASIC. Regular DOSPLUS BASIC is a larger BASIC which, on rare occasions, seems in conflict with certain machine language programs. I think it has something to do with the DOSPLUS BASIC being based at a higher point in memory than other TRS-80 BASICs.

Anyway, since GRBASIC automatically loads BASIC, I either had to rename TBASIC to BASIC or change GRBAS/CMD to auto load TBASIC. I went for the last option to save future confusion. It's a small change using DISKDUMP. If you have DOSPLUS and GRBASIC (and have the above problem), here's what you do. Enter DISKDUMP GRBAS/CMD. Go to sector zero, offset 72. Here you should see the number 00 followed by the word BASIC. Change this 00 to 54, which inserts the letter T. Move the cursor to offset 79. It should contain the number 66. Change it to 65. Now press ENTER. That's it. Now when you type GRBAS/CMD from DOS, it will load TBASIC instead of BASIC. This allows GRBASIC to run just fine with DOSPLUS.

After I did this, I resumed with the CIRCLE command. I entered CIRCLE(s) 64,23,15. It drew a circle nearly as fast as the Color Computer could. Of course, the circle was more jagged-looking than one drawn with the Color Computer, but at least the command was there.

After running a few of the demos, I came to SDRAW(s) x,y. This command will draw predefined shapes. This command was a little more involved to use, but very versatile. You have to define the shape you want in a series of DATA statements. The manual showed a graph of arrows and numbers. If you wanted to go up, you used the number 3, left 5, down 7, etc. You use even numbers if you want to use angles. The number 8 would be go right-down.

This procedure is close in logic to the DRAW"BM128,96;U10 R10" command that the Color Computer uses. You are allowed to define more

than one shape at a time. At the end of each shape, the DATA statement 99 is used to tell the computer that the shape is finished. The following is a short example program for loading two shapes into memory:

```
10 SZ = -128 ' 48K version
20 FOR X=SZ TO SZ+27
30 READ Y: POKE X,Y
40 NEXT
50 DATA 1,7,7,7,7,5,5,5,5,3,3,3,1,1,
1,99
60 DATA 1,4,4,6,6,8,8,2,2,1,99
```

By using SDRAW(s) 64,24 USING 1, a small box is printed on the video. By changing the USING 1 to USING 2, a real small angled box is printed. The USING command allows you to choose which predefined shape you want to draw. This is close to the PUT and GET array feature of the Color Computer. Real nice.

There are other ways to define these shape tables. Included on the disk, is a shape edit program called SEDIT/BAS. This allows you to sketch your shape on the screen, then gives you a number table that represents that shape.

Here's a procedure I found useful. After the shape tables are poked into memory, I go to DOS READY. I type DUMP TBL1/GRB (START=FF80 H, END=FFFH). This dumps the table from memory to a disk file. Whenever I want to use this predefined table, I just type, from DOS ready, "LOAD TBL1/GRB". I type GRBAS/CMD, and I'm in GRBASIC with the shape table TBL1/GRB ready to use. This saves having to use POKE statements in your runtime programs, and also allows you to store a library of different shape tables on disk for different purposes if you wish.

The next two commands allow us to manipulate the shapes. They are called TURN x and SIZE x. TURN is used to rotate a shape in 45-degree increments. SIZE is used to magnify or enlarge a shape. It can also be used to restore the original shape. I find these commands very nice. They remind me very much of the Color Computer's SCALE commands.

The final command is called AUDIO. I think in ways it is better than the Color Computer's SOUND command. Its syntax is AUDIO n1, n2,f,(r). N1 and n2 are tones. If they are the same number, only one tone will be heard. If they are different, a tone is generated starting at n1 and ending at n2. The f parameter specifies the number of times each tone in the range will be executed. This allows stretching or compressing the sound. The fourth parameter is a repeat option. It specifies how many times to repeat the first three parameters. It's like a built in FOR...NEXT LOOP.

I think this package is very nice. It does add some of the graphic and sound commands of the Color Computer to a Model I or Model III. Usefulness will certainly be determined by how much you like to use graphics and sound in your programs. Also, I think Med Systems offers an add-on package for this business graphing purpose. I did have a problem with it working with different versions of DISK BASIC. I had to use TBASIC with DOSPLUS for it to work properly. NEWDOS/80 version 2 BASIC worked with no problem. TRSDOS BASIC works fine, but I couldn't get LDOS LBASIC to work with it. I'm sure it's something simple that can be fixed with a small patch to allow use with LBASIC. Med Systems will probably have a patch if you call them.

After using GRBASIC for a while, I started thinking, "It sure would be nice to have a high-res mode on the Model III."

Well, I've got to go. I'm going to finish my Laser Blaster program I've been working on with GRBASIC.

Pete Carr

REMDISK-1
Joseph E. Willis
REMsort, Inc.
571 E. 185th St.
Euclid, OH 44119
(216) 531-1338

REMDISK-1 is a system for independent study of assembly language disk I/O programming for the student with some experience in assembly language programming. It is recommended by the author that the student complete the REMASSEM-1 course also by J.E. Willis. I believe that this is a good idea, not only because it gives you the necessary background the author assumes you have, but also because it will acquaint you with the format of the REM-course, thus allowing you to concentrate on the material rather than worrying about the mechanics of course material.

The course includes the following material to aid your study of disk I/O and to turn your TRS-80 into a digital blackboard:

1.) Two 45-minute lectures on audio cassette.

2.) A double-sided diskette with:

a.) Machine readable source code for all the programs in both standard EDTASM format and Macro format.

b.) The driver program which turns your TRS-80 into a digital blackboard.

c.) The display program which contains the illustrations.

d.) Routines to convert from one assembler format to the other.

3.) A 45-page source booklet, which contains program listings

and useful information.

One minor complaint I had was that a table of programs or a library of commands was not included with the course material, although all programs are correctly identified in the booklet. In any case here is a list of the programs on the flip side:

Z80TOMAC/CMD
MACTOZ80/MAC
Z80TOMAC/MAC
MACTOZ80/Z80
NOFILE
TRKSECIO/Z80
Z80TOMAC/Z80
ASMLIB/MAC
ASETUP/Z80
TRKSECIO/MAC
ASMLIB/Z80
TRKSECIO/CMD
MACTOZ80/CMD
ASORT/Z80

The author also recommends that the following source material be obtained and studied:

1.) EDITOR/ASSEMBLER — Since the course is disk based either one of the following disk based assemblers requires that you purchase the RS tape based EDTASM; NEWDOS disk EDTASM from Apparat or DISK*MOD from MISOSYS.

2.) MACRO ASSEMBLERS — Any one of the following, if you plan to write A.L. (assembly language) programs of greater than 30K; Microsoft's M80, EDIT or L80 programs or RADIO SHACK's Macro Assembler package. Again, machine readable source code is included to interconvert the formats.

3.) DISK OPERATING SYSTEM — TRSDOS and NEWDOS have been checked out.

4.) TRSDOS and DISK BASIC REFERENCE MANUAL both from R.S.

5.) EXPANSION INTERFACE SERVICE MANUAL from R.S.

6.) and TRS-80 and OTHER MYSTERIES by Harvard Pennington.

All system compatibilities are discussed and many problems have been anticipated by Mr. Willis. One of the programs included on the disk is called ASMLIB and is in itself worth the price of the package to most full time A.L. programmers. It includes a collection of subroutines commonly needed during everyday A.L. programming. As with all the programs in this course, full listings are provided to the student in the booklet provided with the course.

The following is a brief discussion of the contents of the rest of the course because I wouldn't want to tell you all about it. Mr. Willis tells us about two secret single character disk I/O routines which are hiding in ROM. Next, a discussion of assembler format interconversion routines follows. From here, you are led into the wonders of ramdom access I/O in the author's easy-going, often humorous, lectures. Finally, we are told about the dangerous subject of track and sector I/O, and then are told how to command the floppy disk controller to do our

bidding.

We are again led through example programs step-by-step just as in the REMASSEM-1 course. Detailed explanations of almost every line are a major part of the lectures with the actual program scrolling at your command.

The insights the author provides on file handling are worth the cost of the course even if you do not intend to program in A.L., *ever*. Besides, it is neat to know what is happening every time you open/close a file or write/read data from a file. This part (of what I hope will be a continuing series) is even more professionally done that the first part, and most of my trivial complaints were rectified. The audio is excellent, except for one small blank spot of about five seconds.

The disk booted the first time and the first order of business is to make a backup. As Mr. Willis so aptly puts it, ". . . there are few things in God's universe more fragile than a unique diskette."

Although I'm not firmly convinced that this course is the ultimate goal in Computer Aided Instruction (CAI), I do believe that this is the best first step which could have come along. I firmly recommend the course to the diehard A.L. programmer (and even the not so diehard) because this course will give you a good idea of what your DOS is doing for you when you are manipulating data into and out of files.

Frank D. Gunseor, Jr.

Cosmic Fighter
By Bill Hogue & Jeff Konyu
Big Five Software
P.O. Box 9078-185
Van Nuys, CA 91409
Model I or III, 16K, Tape $15.95
Model I or III, 32K, Disk $19.95

As with all Big Five games, Cosmic Fighter comes in a very nice four-color package. It includes concise instructions for loading, playing and transferring to disk. But, best of all, what's inside is better than the outside.

Mission: destroy aliens and refuel! The idea of the game is simple enough. Aliens are coming from the top of the screen in waves of nine ships. You are at the bottom of the screen moving left and right firing laser blasts up toward the oncoming ships.

One or two people may play, with ample time to switch chairs between players. You use arrow keys, or the "<" and ">" keys, for direction and the "F" key, or spacebar, to fire. Screen information includes current players' scores, current top score, fuel remaining and ships remaining. Sound effects are produced through the auxiliary plug and the switch from one player to the next is as clever as I've ever seen.

The game begins with a laser at the bottom of the screen "spelling" out COSMIC FIGHTER with graphics. You are prompted to receive instructions or to start playing the game.

In the first sector, one hit will destroy the ensuing alien. The first wave is always easy to eliminate, the second more difficult, the third is most difficult and the fourth consists of the nine aliens coming down in three columns of three each. The last wave can sometimes be tricky because you must fire directly from underneath them and they usually fire directly down. This of course calls for the old "stick and run" method of alien fighting. The other waves move in a random pattern and fire toward wherever your last shots came from.

To add to the excitement, there is a wandering alien that appears from time to time. This one never misses, and if he hits you six times, you lose your ship. If, by chance, any of the aliens should get past your ship at the bottom of the screen, your ship is moved an inch closer to the aliens and a brand new wave of aliens comes from the top. Is there no end to these evil invaders?

Assuming that you somehow manage to get through 36 assorted aliens and some mystery ships, it's now time to refuel. The mother ship rolls ominously from the right hand side of the screen sounding much like a 1922 Ford pickup. There is a docking hangar in the middle of this ship which is just about the size of your ship. (To be precise it's one graphic location wider on each side.)

Of course, while you're docking, that little mystery invader has a

habit of showing up. If you should take a shot at him, miss and hit the mother ship, Mom will simply destroy you with great haste. The moral here seems to be let the little guy shoot at the mother ship since it doesn't seem to hurt. If he should happen to be shooting at the hangar while you're docking, your ship will be destroyed.

If you should get through all four waves and dock, it now requires three shots to destroy them and they score double. After these next four waves and docking it will require four shots and double points will be awarded. I believe that's the limit or at least that's as far as I can get.

This game runs smoothly and plays well. It is a nice change from the "space invaders" type of game. I find it challenging but not overwhelming. I always enjoy receiving a new game from "Big Five" because they seem to always come up with winners, and Cosmic Fighter is another winner!

Mark E. Renne

The Universal Elixir and Other Computing Projects Which Failed
Tales of Computing Folk
The Power of Peonage
The Second Coming: More Computing Projects Which Failed
(A Set of Four Books)
Robert L. Glass
(Some with Sue De Nim)
Computing Trends
6925 — 56th Ave. S.
Seattle, WA 98118
$8.95 ea.

We've all read stories about how some computer project has done wonders for a company, or a technological break-through has given us new wonders. But what happened to the failures? We never hear about the programmer who doesn't document, the computer that didn't work, the program that was so fast it got its company in trouble. Just wait though, you can hear about those and more now.

Robert Glass, a man who has been around programming and computers since the 1950s, when computing was a shiny new field, has been writing about failure for *Computerworld* for a number of years. These books are collections of stories.

Think failure can't be interesting? Well, enter the world of Balderdash Steel, or Mill Run (just another run of the mill government agency), or Wings Aloft Aviation. Meet Samual Smoothdog, whose programs are so neat and well organized that one was published in a family magazine as an article. How about Byron Iconoclast, whose assembly language code was known for its brilliance and unreadability?

There are a whole load of others who appear here, living again, through some of the worst disasters of computer lore. All this for your interest and education.

I've had one of these books for several years. I find that I come back to it about twice a year — partly for the fun of it. Each story is entertaining, and each tells a story that leads to some very important conclusions. Each story, you might say, has a moral.

The stories are all aimed at experience with larger computers. No TRS-80s show up (though maybe that's the Little Wonder 80???). Every story is educational — fun, as well as funny.

I highly recommend these to *anyone* interested in computers. This is probably the best collection of computing stories available from anywhere.

T.R. Dettmann

Space Intruders
Adventure International
Box 3435
Longwood, FL 32750
(800) 327-7172
$19.95

While glancing through one of my computer magazines recently, I saw that yet another "space invaders" type game had hit the market. The first thing that popped into my mind, naturally, was "Oh my gosh, not another space invaders game! When will something original come out?" Well, something has. Though this new, exciting game from Adventure International is based on a game that has been imitated by many software authors, Space Intruders by Doug Kennedy presents new challenges, making the game something almost new in itself.

Of course, I did not purchase Space Intruders without being convinced in some way that I should buy it! I noticed in the advertisement that if any customer was not satisfied with the game, he could return Space Intruders to Adventure International for a full refund. In addition to this, the ad boasted that the program was the best of its kind, and even bore the "Scott Adam's Seal of Approval," an honor not bestowed upon every program published by the well known and reputable software marketing company, Adventure International. Therefore, with all of these advantages, I figured I couldn't lose, so I ordered the new game.

The features and rules of the game are such that it is rightly deemed "the spectacular arcade version." The game follows most of the original patterns and rules of the original game, of course, in which

there are several rows of evil creatures swarming down on a laser base. In addition to their constant motion, the hideous creatures fire missiles, destroy shelters and eventually destroy the laser base! There are also mystery saucers at the top of the screen which may blink on and off, or stay solid, but are always worth a lot of points!

In addition to these creature features, there are quite a few new goodies. The invaders actually split apart in some portions of the game, and consequently make shooting a lot more difficult! There are also special sets of scoring rules which may be followed during the game for a real challenge and a lot of extra points, such as: if the player shoots 22 times, and on the 23rd shot hits a mystery ship, an extra 300 points is given! With the constantly moving graphics and action sounds accompanying the game, that type of "fun" is probably only for experts. Mystery ships also may start dropping extra aliens into the already strong rows of aliens, just to make the going a little tougher for the underdog player.

The controls are as simple as 1-2-3. In fact, they are 1-2-3! The one and three are used for moving the laser base, while two fires. I find these controls the very easiest that I've ever worked with. More time can be devoted to saving the earth and less to whether this finger goes here or that finger goes there.

The documentation, as is the habit of Adventure International, is excellent. A light-hearted, but thorough, explanation is given of the entire game, including all of the special scoring techniques. The instructions mention a demonstration mode which exists in the program that is really quite attractive. In the true spirit of space invaders, a tiny invader moves about the screen in the demonstration mode (which displays the title), occasionally deleting or replacing letters on the screen (and with some of the spelling I've seen in programs recently, we need more of these little guys!).

The sound isn't as good as expected for such a high quality program. There really aren't a large variety of sounds, and there are absolutely no sounds outside of the game itself (no sound in the demonstration mode). I assume that with graphics this fine and logic this complete, sound had to be reduced because of memory requirements. However, it would have added a lot to the program to have more and higher quality sound effects.

The program is fantastic. The Space Intruder graphics are excellent. The machine language programming and logic actually make for a variety of games because of the different methods of scoring. The programming is extremely challenging. After a few minutes, a player will not have mastered the game, but will still be trying to get a respectable score!

The game, Space Intruders, though, is number one material, and is at least worth a try since you can't lose anything. If you're dissatisfied, just send the program back for a refund. I recommend inviting the space intruders into your computer, and taking another crack at saving the human race.

Tim Knight

Graphic Trek 2000
Game Pack
Simutek Computer Products
4877 E. Speedway
Tucson, AZ 85712
(800) 528-1149
16K Model I or III $14.95

Graphic Trek 2000 is part of a five-game package. The other games in the package are Invasion Worg, Star Wars, Space Target and Saucers. I'll talk a little about each of the other games later. I was unable to load my first tape from either of the dumps. The tape was replaced quickly, and without question, by Simutek.

The game has three levels of play. Commands are all one letter. They are entered using the INKEY$ function so no ENTER is required. Warp Drive is "W."

After entering warp drive, you steer using "<" for left, ">" for right, "Z" for up and "X" for down. Impulse engines, "I," use the same

directional controls, but you move one CHR$ at a time. "E" or exit is used to stop warp or impulse drive.

"S" activates your shields. Using a number from zero to nine, you transfer that percentage of your power (times ten) to your shields (i.e. "4" transfers 40% of your power to the shields). The galaxy map, "G," shows the history of all quadrants in the universe viewed by your long range scanners. Phasers, "P," are locked in automatically on any Klingon in the quadrant by the ship's computer.

The computer will give a damage report, "R," indicating any damage to any ship function. Damages are repaired at starbases. To dock with a starbase, you use impulse drive to maneuver to a blinking dot located in the center of the starbase.

Photon torpedoes are launched using the spacebar. They are put in position using the direction keys and then detonated by pressing the spacebar again. They must be dead center to score a kill.

The game is played in real time, so things are always happening, even if you're just sitting there. I'm not sure why the author used "Z" and "X" for directions, so I changed them to arrow keys to avoid confusion. Torpedoes are very difficult to control, and often never contact the middle of the Klingon. Most of the time, I use the phasers for simplicity. Impulse drive is unbelievably *slow*! I usually use warp drive to get within a few spaces of the space station and then use impulse.

Since I've played over 30 Star Trek games on a variety of computers, I would rate this one as an okay version. There's nothing special here, except maybe the docking, which has a cute graphic for the starbase. Also, considering the price works out to about three dollars a game, you can't really expect machine language — high speed graphic — super duper results.

Invasion Worg is a strategy game involving androids, neutrino blasters, various planets and invading bad guys. I've never found a strategy game I enjoyed on a computer. I don't know why, because I enjoy them off the computer. This one is no exception.

Saucers is simply a waste of bits, bytes and disk space. You are trying to destroy the enemy by getting them in your crosshairs.

Space target is not too bad. Ships are moving across the top of your screen. You fire a missile and guide it with keys as above. If you destroy the ship, its pilot bails out, thus allowing another target for the bloodthirsty space cadet. This one even has sound!

Star Wars is a take-off of the famous epic motion picture. You do battle with several hyper fighters, Darth Vader, and finally drop a bomb into the chute to destroy the Deathstar. I enjoy this one for a change of pace also.

If you own a Star Trek game, I wouldn't recommend adding this package to your collection. However, if you're just starting out, this game package is a way to acquire five games at a reasonable price.

Mark E. Renne

Escape from Traam
Adventure International
Box 3435
Longwood, FL 32750
(800) 327-7172
$19.95 cassette
$20.95 disk
Model I & III

Scott Adams — famous for his adventure games — now has adventures from other authors featured in his catalog. This series (called Other-Venture) features an infinitely sadistic author: Jyym Pearson.

Other-Venture #3 (Escape From Traam) is one of the most challenging and complex situation adventure games I have ever encountered in my years of adventure playing.

The relative simplicity of the beginning of the game is an excellent camouflage of the events yet to occur.

It begins with your space cruiser crash landing on the planet Traam. Your obvious mission is to escape from Traam and rejoin your comrades in space. However, you have amnesia and there is no way that you know what you will encounter on this strange planet.

This game is so difficult that the author has available hint sheets. However, hint sheets are *not* answer sheets, so don't expect to read the hint and be able to immediately solve your problem! In fact, you may *never* get the hint!

Don't expect to play this game alone and solve it overnight, in a week, or even a month! I suggest that you play it with a friend — that way, with *both* of you involved, you *may* have a chance to solve the adventure! It will take a while, though — but don't give up!

Some of the most useful commands are LOOK, LISTEN, TALK, FEEL, CLIMB and OPEN. That sounds easy enough, but let me use the command LOOK as an example. You can LOOK, LOOK UP LOOK (OBJECT), LOOK DOWN, again LOOK at whatever response you get from LOOK (OBJECT), and again repeat the process. You must use the exact wording needed — and you are *not* limited to one- or two-word commands!

Another example of complex situations would be the point in the game where all you are told is that you are in a field on the outskirts of Traam city. You can LOOK to your hearts content and not find what is needed. You have to LISTEN and you hear a tinkling sound. Then, only after you have listened, can you LOOK UNDER BUSH — and you subsequently find a drainpipe that leads into the city. Then you have to CLIMB DRAINPIPE in order to enter the pipe!

If you die during the game, you are immediately started over at the beginning location with all of the inventory you have accumulated so far in the game. In fact, there is one object needed, a helmet, that (as far as I can tell) you must die for, from poison fumes in the drainpipe, in order to get it at a later point in the adventure!

Solving Escape From Traam is extremely difficult — but exceptionally satisfying! You will feel elated and happy that you were clever enough to make it through the game. But, be warned — Jyym Pearson's adventure games are addicting. *If* you make it through one of his games, the *first* thing you will do is order one of his other games!

On a scale of one to ten, I give this game a very solid ten! Somehow, though, I feel that the author is currently working on an eleven!

Bob Krotts

MAGAZINE SAMPLES
FREE listing of over 150 magazines offering a sample copy - 50¢ per sample. Send stamped self-addressed #10 envelope to:

PUBLISHERS EXCHANGE
PO Box 1368, Dept 278
Plainfield, New Jersey
07060

77

YOU CAN PAY MORE BUT YOU CAN'T GET MORE!

Model III 16K
$839
Model III 48K
2 disc & RS232C
$2059

Color Computer 4K
$310
w/16K Ext. Basic
$459
w/32K Ext. Basic
$525

BUY DIRECT. These are just a few of our great offers which include Printers, Modems, Computers, Peripherals, Disc Drives, Software and more. call TOLL FREE 1-800-343-8124

We have the lowest possible fully warranteed prices and a full complement of Radio Shack Software.

computer plus
Write for your free catalog.

245A Great Road
Littleton, MA 01460
617 • 486 • 3193

78

MICRO MOONLIGHTER NEWSLETTER

The **ONLY** publication devoted exclusively to helping you create, build, and maintain a home-based business using your micro-system. **SUBSCRIBE NOW** to what may be the most important publication in YOUR FUTURE !

1 Year (12 issues) only $25 U.S., $29 Canada, $35 World Wide

J. Norman Goode, Publisher
Micro Moonlighter Newsletter
2115-J Bernard Avenue
Nashville, TN 37212

Visa and Mastercard welcome. Send account number and expiration date.

79

DID YOU BUY YOUR COMPUTER TO HELP YOU MAKE OR SAVE MONEY?

BISPLAN

Is for the person wanting to start a small business. Or for the person who is operating a small business. You enter all the financial information pertaining to your business plans and BISPLAN will produce PROFIT & LOSS STATEMENTS, CASH FLOW ANALYSIS and BALANCE SHEETS. Each projected for up to 5 years. BISPLAN will also indicate financing needs. Printed output suitable for loan applications, business proposals, etc. "What if" ideas tested. Entries easily changed. Ideal for small business proposals as well as for projecting existing business expectations.
INTERACTIVE. COMPREHENSIVE.
NEW - Store data on tape or disk.

On cassette: Specify Mod I or III
32,48K - $24.50 +$2.00 shp, hndl.
16K - $19.50 MD res add 3 5/8%

Mariah Computing, P.O. Box 513
Columbia, MD 65205

80

Micro-Grip

FRICTION FEED FOR YOUR EPSON

MX-70™ or *MX-80*™

™'s and ®'s are Trade Marks of EPSON INC.

- Converts your printer for friction feed of SINGLE SHEETS or ROLL PAPER.
- Snap-on Installation (all you need is a screwdriver, no soldering)
- Tractor feed remains undisturbed.
- Proven reliability.
- Available NOW!
- Only $39.95 (add $1.50 for postage)

VISA & Master Card Accepted
(804) 826-8808

MICRO-GRIP, Ltd.
P.O. BOX 873
LAFB, VA 23665

81

MICRO-DESIGN PRESENTS THE MDX-4
PHONE MODEM
P.C. BOARD & USER MANUAL
$29.95

- ★ 300 BAUD
- ★ DIRECT CONNECT
- ★ ANSWER AND ORIGINATE MODES
- ★ SOLDER-MASKED & SILK-SCREENED
- ★ ON BOARD POWER SUPPLY

MDX-4 PC BOARD & USERS MANUAL 29.95
USERS MANUAL 8.95
(Texas Residents Add 5% Tax)
(Add $3.00 Shipping for Board)

CALL OR WRITE:

MD
MICRO DESIGN
P.O. BOX 748
MANCHACA, TX 78652
(512) 282-0225

DEALER INQUIRIES INVITED

82

COLOR COMPUTER PROGRAMS

:COLOR TREK...$34.95
:BERSERK.....$35.95
:PAC ATTACK...$35.95
:TELEWRITER...$94.95
:E-BOARD UPGRADE
4K-32K ONLY $109.95
AND LOTS MORE.....

COLOR PRODUCTS UNALIKE
#211 990 EAST 8TH
VANCOUVER B.C.
(CANADA) V5T-1T8
PH.(604) 873-2372

83

FREE

GAME - BREAKOUT - FOR COLOR COMPUTER
SEND S.A.S.E. FOR A FREE LISTING.
WRITTEN IN MACHINE LANGUAGE, VERY FAST

THE SOLUTION -$12.95

THIS PROGRAM SOLVES THE PROBLEM OF THE COLOR COMPUTER'S SMALL SCREEN SIZE.
FEATURES INCLUDE:
1- WRITTEN IN MACHINE LANGUAGE
2- FULL ASCII CHARACTER SET
3- LOWER CASE WITH DECENDERS
4- 42 CHARACTERS X 21 LINE DISPLAY
5- USER MAY INTERMIX TEXT & GRAPHICS
6- INTERFACES DIRECTLY WITH BASIC
7- WORKS WITH PRINT AND LIST COMMANDS
8- FAST-PRINTS AT OVER 500 CHARACTERS PER SECOND
9- CAN PROVIDE 4 LINES OF TEXT AT THE BOTTOM OF A GRAPHIC SCREEN -JUST LIKE SOME OTHER MACHINES

PROGRAM ON TAPE - 16K OR 32K MACHINES
TERMS-CHECK OR MONEY ORDER ONLY-INCLUDE $1.00 FOR SHIPPING (U.S. ONLY)

SNAKE MOUNTAIN SOFTWARE
P.O. BOX 5722
RALEIGH, NC 27650

84

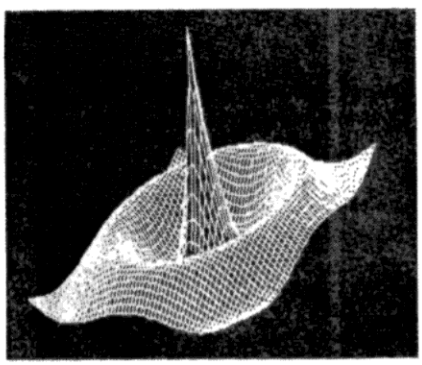
SUPER BASIC V3.1

Adds these features to Level II Basic:
- RAM Printer Spooler
- Programable Shift Key Entries
- Keyboard Debounce
- Auto Repeating Keys
- Spelled Out Error Messages
- Blinking Graphics Cursor
- Graphics at Machine Language Speeds
- Change Line Numbers
- LPRINTTABs > 63
- JKL-TO-PRINTER Function
- Unlimited USER Routines
- Sound & Much More!!!

Requires 16K Level II
Only $19.95 from:
Maine Data & Financial Systems Inc.
S.R. 79, Box 103
Orland, ME 04472

85

New products



Model II/16 High Res Add-on

A Model II Graphics Option (26-4104) including hardware, user's manual and 8-inch diskette with BASIC and assembly language graphics subroutine library, is available for $499 (plus installation, required) at Radio Shack outlets. With this option, a Model II can be used to create sophisticated tables, charts, graphs, illustrations, maps and geometric patterns. The option organizes Model II display into 640 horizontal and 240 vertical picture elements (pixels) and adds eleven new commands to facilitate screen graphics. The graphics option adds 32,000 characters of independent, dedicated graphics memory. Graphics can overlay text in the regular video memory and automatic text reversal is provided when needed.

#130

Data Ace

Data Ace is available for Model I/III at a price of $250 (or, including MMSFORTH, for $375). Version 2.0 of MMSFORTH (from Miller Microcomputer Services) is available and is a prerequisite to the use of the Model I/III versions of Data Ace. Features include a relational database manager and operating system, conversational data definition language, interactive query language, block structured programming language, full screen text editor, HELP and application package start-up kits. Data Ace runs on Model I, II and III requiring the maximum memory, two disk drives and a printer. Plug compatible (DEC 11/23) users should describe their configuration — most can be accommodated. A demo disk is included with the User Guides for the Model II version. The two User Guides and demo disk cost $100 plus shipping from Computer Software Design, Inc., 1904 Wright Circle, Anaheim, CA 92806, (714) 634-9012.

#131

Model I/III Disk/Tape Utility

TRSDUM (for disk system users) and TRSTUM (for tape users), allow you to protect your disks and tapes by facilitating the transfer between disk and tape, disk and disk, and tape and tape. The Model III versions of these programs provide for rewriting existing tapes in high speed mode. TRSDUM and TRSTUM will read any tape or disk file as long as the file is in standard TRS-80 tape or disk protocol. These utilities automatically identify SYSTEM, BASIC, and EDTASM formats, and reproduce them on either disk or tape.

Both utilities are high speed machine language, menu driven, and provide full status displays at all times. They come with a 6-page user's guide and covered by a limited 90-day warranty. TRSDUM, for 16K and up disk-based systems with TRSDOS 2.3 or 1.3, is $17.95 on diskette. TRSTUM, for 16K tape systems is $16.95 on cassette. Available from CRB Microtools, 14835 N. First Ave., Phoenix, AZ 85023, (602) 993-3999.

#132

Color Computer Expansion

The Color Computer Expansion Unit is a powerful add-on developed for the TRS-80 Color Computer (CC). RAM is increased to 64K. A Z-80A microprocessor is included for running CP/M, FLEX, and OX-9 programs while maintaining full compatibility with CC software. Circuitry for an 80x25 alphanumeric display is provided with reverse video, dual intensity, blinking/blanking, inverted and protected characters.

A dual density 5¼" disk controller will support 4 drives up to 800K per drive. The 2" high expansion unit acts as a pedestal for the Color Computer. It plugs directly into the CC cartridge port with no modification necessary and unplugs for standalone use of the Color Computer. Contact George Associates, P.O. Box 960, Berkeley, CA 94701, (415) 843-3587.

#133

Model II Data Manager

IDM-X is the first member of a series of interactive data base managers. It has many advanced features, including: built-in sort/merge, key access, support string, double-precision floating decimal, integer, and date, formatted numeric fields, and an extensive report writer. IMD-X is interactive—no coding, no syntax and commands to remember. Its basic components are a data base initialization program, data base manipulation program, report writer and report generator. A dual disk system with 64K memory and TRSDOS is required. It is available for $399 from Micro Architect, Inc., 96 Dothan St., Arlington, MA 02174, (617) 643-4713.

#134

Cassette Operating Systems

KWIK Software has released KOX3, a Model III version of the KWICOS Cassette Operating System. A software-only enhancement to Level II, KOS3 adds disk-like commands and an improved 2200 baud tape storage format to supplement CSAVE, CLOAD and SYSTEM routines. No add-ons, equipment modifications, or special programming skills are needed.

KWIK also released KWIKIT, a miniature version of KWICOS and KOS3, to improve cassette storage of BASIC programs. KWIKIT takes 880 bytes from user memory, saves and loads BASIC programs at 1000 baud for Model I, and 2200 baud for Model III, with a broad playback volume tolerance. KWIKIT is available on cassette for Model I/III for $9.50 ($11.50 with debounce and

LIST speed control option). KOS3 is available on cassette for $24.00, postage paid, from KWIK Software Dept. E, Bolivar, MO 65613, (417) 326-7154.

#135

CP/M Conversion Boards

The Freedom Option and Freedom Plus boards allow TRS-80 Model I/III, PMC 80 and 81, and LNW 80 to run CP/M applications software as well as TRSDOS software. The Freedom Option is the basic CP/M conversion board, while Freedom Plus provides CP/M conversion and adds 16K of RAM.

The Freedom boards include the CP/M compatible operating system, T8/OS, written in Z80 code. T8/OS can run any combination of 5¼- and 8-inch drives, single or double density, single- or double-sided.

The majority of CP/M applications software is compatible with these boards as well as the higher level language compilers that are available for CP/M. The price of Freedom Option is $275 and Freedom Plus is $490. Contact Freedom Technology International, 119 North 18th St., Philadelphia, PA 19103, (215) 569-2381.

#136

Inventory Management

Inventory Management for Small Computers, by Chuck Atkinson, prepares purchase orders, does physical inventory, and features "quick register," which controls the inventory while printing the sales ticket. Designed for systems that use CBASIC, this book is available for $19.95 (120 pages) from Chuck Atkinson Programs, Route 5 Box 277C, Benbrook, TX 76126 (817) 654-2011 or (800) 547-1842. *Inventory Management* is also available on 5¼" disk for Model III ($250.00), and on 8" disk for CP/M-CBASIC computers.

#137

Model III Expansion Board

The MDX-3 Interface Expansion board from Micro Design, is designed to be mounted inside the Model III computer. It provides a double density floppy disk controller and data port, and a 0-600 baud direct connect "answer" and "originate" mode phone modem. MDX-3 is sold as a bare board and Users Manual ($79.95), partially assembled board and Users Manual ($199.95), or fully assembled and tested board with Users Manual ($299.95). All that is required on the partially assembled board is the insertion of user-supplied integrated circuits, and installation. The Users Manual may be purchased separately for evaluation purposes. The floppy disk controller uses a WD1793 chip and digital data separator. The MDX-3 will handle up to four 40- or 80-track, single or double head drives and supports both single and double density operation. Contact Micro-Design, P.O. Box 748, Manchaca, TX 78652, (512) 282-0225.

#138

Spooler

Compulink Corporation introduced SooperSpooler™, an intelligent printer interface designed to free microcomputers from the task of printing. In addition to being a hardware buffer, many software selectable formatting features are accessable with a simple BASIC program.

The base model includes a 16K byte memory and Centronics compatible I/O ports. The standalone unit ($349.00) has a self-contained power supply and two digit L.E.D. display.

Options include memory expansion ($159.00) to 62K, and RS232 serial I/O ports ($95.00) that can also be used for modem transmission and serial to parallel translation. Contact Compulink Corporation, 1840 Industrial Circle, Longmont, CO 80501, (800) 525-6705.

#139

Pocket Computer Book

Problem Solving on the TRS-80 Pocket Computer is a self-paced book for people with a working knowledge of TRS-80 Pocket BASIC. It provides practical exercises, illustrations and diagrams that present and explain solutions to a variety of practical application problems. The book covers advanced functions including trigonometric, logic, and print functions; random numbers; storing, sorting and searching. The book, by Don Inman and Jim Conlan, is available for $9.95 at Radio Shack stores and outlets.

#140

Architectural Engineering

1. Structural Timber Design— eight loading conditions for any timber species. Also plybeam design.

2. Structural concrete— beams, slabs, retaining walls, etc.

3. Structural steel— eight loading conditions for WF beams.

4. Heat loss and SLR— for passive solar design.

For TRS-80 Model I or III
Cassette or ESF
$35 ea. program. Total pkg. $95.

ERIC CLOUGH
Winlaw, B.C.
Canada V0G 2J0

95

DISKETTES CASSETTES

Error-Free 5¼-inch Diskettes (MD-5) single-sided, soft sector, single or double density, reinforced hub.

Item	Qty 10	Qty 50
MD-5	$25.00	$110.00
C-10	$ 7.50	$ 32.50
C-20	9.00	39.00
C-60	11.50	50.00
C-90	15.00	70.00

UPS SHIPPING INCLUDED
In Continental USA
CA Customers add taxes

MICROSETTE
475 Ellis St., Mt. View,
CA 94043 (415) 968-1604

96

TOTAL DISKETTE BACKUP

INCLUDING UNLIMITED BACKUP OF MOST PROTECTED DISKS

If your diskette software library is not completely backed up, or, if you are wasting diskettes by making backup copies of all your diskettes, your problem is solved! Using DUMPLOAD, the total contents of your diskettes can be safely dumped to tape. The hi-speed tape option allows six 35 or 40 track diskettes to fit on one C60 casette.

- TRS-80* Model 1 16K - 48K
- TRSDOS or NEWDOS80 Compatible
- May be used to back-up TRSDOS, VTOS 4.0, NEWDOS, MICRODOS, or data disks. (Single Density)

— $16.95 on Cassette — $19.95 on Diskette
Plus $1.00 handling (Indiana Res. add 4% tax)
TOLL FREE 24 HR. ORDERING SERVICE
1-800-525-9391 (Ext. 509) (Except Colo.)
1-800-332-9259 (Ext. 509) (Colo. Residents)
or Mail Check or Money Order To:
COMPLETE COMPUTER SERVICES
8188 HEATHER DRIVE
NEWBURGH, INDIANA 47630

97

PORTFOLIO ANALYSIS

Track and Analyze your stock portfolio with these programs for the TRS-80* Mod 1 & 3.

The Stock Analysis Package consists of four programs which allow Database creation & maintenance; Graphing of prices, moving avg. & volume; and Generation of a Portfolio Analysis Report to evaluate your market position.

Programs are written in BASIC and are MENU driven with Error Trapping for ease of use.

TRS-80* Mod 1 or 3, 48K & one disk drive required. Printer required for Portfolio Analysis report. Price for the entire package is $74.95. To order or to receive more information, fill out coupon below or call (214) 867-5656. VISA & MC are accepted.

P.Tree Enterprises Suite 269
2701-C West 15th Street Plano, Texas 75075

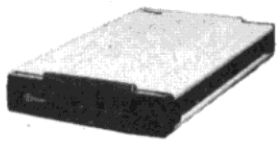
☐ Payment enclosed or charge my ☐ VISA ☐ MC.
☐ Please send more information.

Name_____
Address_____
City_____ State_____ Zip_____
Account #_____ Exp. Date_____
*TRS-80 is a trademark of Tandy Corporation.

98

HAYES SMARTMODEM

$229.88
DELIVERED



High performance data communications system for small computers. **SMARTMODEM** is a stand-alone 300 baud modem system including the **SMARTMODEM** unit, power pack, and modular telephone cable. Completely programmable, the **SMARTMODEM** can be controlled using any programming language.

- ☐ FCC approved direct-connect
- ☐ Auto-answer/Auto-dial
- ☐ Full/half duplex
- ☐ Touch-tone/Pulse-dialing
- ☐ Audio monitor
- ☐ 7 status LED lights

HIGH TECHNOLOGY AT AFFORDABLE PRICES

THE BOTTOM LINE

Orders & Information
(603)-673-8857

Orders Only
(800)-343-0726

12 Johnson Street, Milford NH 03055

99

BARCLAY WHYTE ASSOCIATES.

"THE BEAR WITH THE SOFTWARE." TM

Presents Mel Patrick's

WordProc I/III.

VER 2.0 - BASIC+MACHINE LANGUAGE WORD PROCESSOR.

TRS-80 Model I or III

DOCUMENTATION INCLUDES COMPLETE PROGRAM LISTINGS

JUSTIFIES-CHAINS FILES-MERGES FILES-AUTO CENTERS
TITLES-HEADINGS-SUBHEADINGS-PAGE NUMBERS-MORE

48K Mod I/III disk.
EASILY MODIFIED FOR OTHER CONFIGURATIONS

$14.95 With disk.
$10.95 Documentation only.

BARCLAY WHYTE ASSOCIATES. DEPT D.
549 WEST COLUMBIA STREET.
P.O. BOX 948, NEW WESTMINSTER.
B.C. CANADA. V3L 5C3.

ADD $1.00 FOR POSTAGE & PACKAGING. ADBOUSI
B.C. residents add Provincial Social Service Tax.

TRS-80 Trade Mark of Tandy Corporation.

100

MICRO MINI ADS

are our fastest-growing advertising classification.

TRY ONE ON FOR SIZE!

101

— Professional —

REAL ESTATE SOFTWARE

for APPLE, TRS-80 & CPM SYSTEMS

- **PROPERTY MANAGEMENT SYSTEM: $375**
  - Tenant History
  - Late Rent Report
  - Vacancy Report
  - Income Report
  - Auto Late Charge
  - Returned Checks
  - Operating Stmt
  - Building Reports
  - Utilities Report
  - Tax Expense Report
  - Prints Checks
  - Prints Receipts
- **PROPERTY LISTINGS/COMPARABLES: $325**
  - SCREEN BY —
    - 22 Items/Listing
    - 1000 Listing/Disk
    - Listing Memo Field
    - Max/Min Price
    - Units/Zone/City
    - Max Price/Income
    - Max Price/Sq Foot
    - Min Cashflow
- **REAL ESTATE ANALYSIS MODULES: $50/Module**
  - Home Purchase
  - Income Prop Analysis
  - Property Sales
  - Construction Cost/Profit
  - Tax Deferred Exchange
  - APR Loan Analysis
  - Loan Amortization
  - Depreciation/ACRS Analysis
- **WORD PROCESSOR – MAGIC WAND: $265**

Software Realty Company™

At Computer Stores Everywhere
or Order COD Direct
Cal Residents add 6% Sales Tax

(213) 372-9419

Suite E, 1116-8th Street, Manhattan Beach, CA 90266

102

ABS Suppliers
3352 Chelsea Circle
Ann Arbor, MI 48104
(313) 971-1404

B17 TAPE OPERATING SYSTEM will let you
- Save & Load programs 6X faster than Model-1.
- Save & Load Data Arrays over 800X faster.
- Backup standard System tapes that load-n-go.
- Certify cassette tapes for B17 use.

$24.95 Specify Model 1/3 16K/32K/48K RAM. Inexpensive Upgrades available. 3rd year of sales.

B17 DISK/BAS is a Disk version. Lets you save *any* disk file to cassette at 3000 baud. Inexpensive way to backup your files. Works with *any* DOS. Includes tape certifier. Requires 32K RAM-up.
$19.95 Specify Model 1/3.

WORDSMITH Word-processing program in BASIC. A 'Trainer' for more complex W/P programs. Easy-to-use. Specify Tape/Disk version. Tape version requires B17 Tape Operating System (above). Disk version requires 32K RAM-up. Uses MX-80 Printer.
$19.95 Specify Model 1/3.

Add $1.50 postage/handling per item.

Unclassified

UNCLASSIFIED ANNOUNCEMENT

This unclassified announcement space is available free to individuals with single or unusual items for sale or trade. 80-U.S. Journal reserves the right to reject any commercial advertising in this section and suggests using our display advertising for that purpose.

These notices are free of charge and will be printed one time only on a space available basis. Notices will be accepted from individuals or bona fide computer user clubs only. All these unclassified announcements must be typed, contain 75 words or less and include complete name and address information.

TWO (2) CENTRONICS 700 PRINTERS. Only 2 months old. 60 CPS, parallel interface, friction and tractor feed. $500 each. Will sell separately. List price over $1400 new. Call or write Glenn Collura, 9615 Seminole Trail, Streetsboro, Ohio 44240, (216) 626-3783.

TRS-80 GOODIES! — (1) January-December issues of 80-Microcomputing — $12. (2) Lynx auto answer Model I/III modem (new condition) with $150 worth of terminal software — all $250. (3) Software from Allen Gelder, Southern Software — various prices . . . for information contact Tim Knight, 10 Fieldbrook Place, Moraga, CA 94556 or call (415) 376-8471.

TRS-80 MODEL I EXPANSION INTERFACE, DISK DRIVES, PRINTER, ETC., wanted. Must be in good working order. Please send list and price to: Fred B. Kurtz, 52 Glenn Oaks Drive, Lancaster, PA 17603.

FOR SALE: TRS-80 software, brand new on original disk or cassette with original documentation. R.S. Communications Package (26-1149) $20, Astrology (26-1605) $15. Basic Programming Assistant $8, Pensa-Write & Pensa Mail word processor $8, System-to-Basic Utility $10, RSM-2D $15, TLDIS $8, DUPLIK & RENUM $8, CFETCH & CWRITE $5, Scarfman $10, Big 5 Game — Attack Force $10. George Kwascha, 8007 Mahogany Drive, Charlotte, N.C. 28212, (704) 535-7575.

Coming in the

AUGUST ISSUE

- ★ Emphasis on Education
- ★ The computer revolution in public schools
- ★ Computerize your educational programs
- ★ Color Computer gradebook
- ★ Results of the 80-U.S. reader survey
- ★ AND MORE!

Advertiser Index

★ **Contact these advertisers directly.**

A Sweet Deal...

Buy an Alpha Joystick and games together and deduct up to $14.00 from your order.

(Model I version shown)

The Alpha Joystick adds arcade-style control to TRS-80 action games. Simply plug it in and begin playing joystick compatible games. No modification, wiring or batteries are required, and the Alpha Joystick is compatible with other TRS-80 accessories. The instructions are clear and complete. We even show how easy it is to experiment in BASIC (A=INP(0) reads stick) and convert BASIC programs to Joystick control.

MODEL I - Plugs directly into any Level II Keyboard (card edge on rear) or expansion interface (left side, next to printer port).

MODEL III - Works with any "Model III BASIC" system and plugs into the 50 pin I/O bus (largest edge connector underneath).

THE ALPHA JOYSTICK

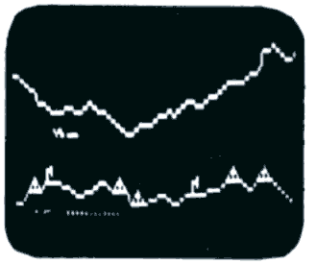
Joystick + 1 game Deduct $ 6.00
Joystick + 2 games Deduct $12.00
Joystick + 3 games Deduct $14.00

ONLY $39.95

MODEL I OR III. SPECIFY WHEN ORDERING. PRICE INCLUDES ATARI JOYSTICK + ALPHA INTERFACE + INSTRUCTIONS + DEMO PROGRAM LISTING.

14 DAY MONEY BACK GUARANTEE - if you are not delighted, return it within 14 days for a prompt and courteous refund.

Choose from any of the Joystick Compatible games below. Be sure to mention the "Sweet Deal" discount when ordering.



PENETRATOR

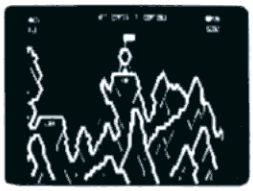
Soar swiftly over jagged landscape, swooping high and low to avoid obstacles and enemy missile attacks. With miles of wild terrain and tunnels to penetrate, you're well armed with bombs and multiple forward missile capability. From Melbourne House. Features sound, trainer mode and customizing program.

• • SCARFMAN

This incredibly popular game craze now runs on your TRS-80! It's eat or be eaten. You control Scarfman around the maze, gobbling up everything in your path. Try to eat it all before nasty monsters devour you. Excellent high speed machine language action game from the Cornsoft Group. With sound.

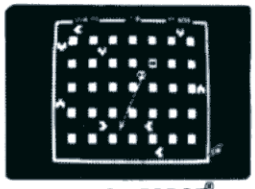
SUPER NOVA®

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



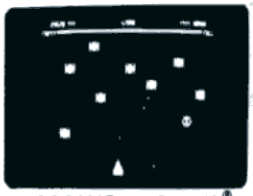
LUNAR LANDER

As a vast panorama moonscape scrolls by, select one of many landing sights. The more perilous the spot, the more points scored... if you can land safely. You control REAL main engines and side thrusters. Absolutely the best use of TRS-80 graphics we have ever seen! From Adventure International. With sound.



ATTACK FORCE®

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



COSMIC FIGHTER®

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

GALAXY INVASION®

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

DEFENSE COMMAND

The invaders are back! Alone, you defend the all important nuclear fuel canisters from thieving aliens who attack repeatedly. An alien passes your guard, snatches up a cannister and flys straight off. Quick! You have one last chance to blast him out of the sky! With sound and voice.

TALKING ROBOT ATTACK

This game TALKS without a voice synthesizer, through the cassette port. With just a hand laser in a remote section of the space station, you encounter armed robots. Some march towards you, more wait around corners. Careful, the wallas are electrified. Zap as many robots as you dare before escaping to a new section. More robots await you.

ARMORED PATROL

A realistic tank battle simulation. Your view is a 3—D perspective of an alien landscape. Manuever your T-36 tank to locate and destroy enemy tanks and robots that lay hidden, ready to assault you. Clever graphics create the illusion of movement and dimension. From Adventure International. With sound.

111

GAME PRICES

"Penetrator" Disk or 2 Cassette Package	$24.95
"Armored Patrol" Cassette $19.95 . . Disk	$24.95
ALL OTHER GAMES:	
16K Level 2, Model 1 + Model 3 Cassette	$15.95
32K Level 2, Model 1 + Model 3 Diskette	$19.95

These games are "Alpha Joystick Compatible." They may be played with or without Joystick (using arrow keys).

Toll Free Order Line

800-221-0916

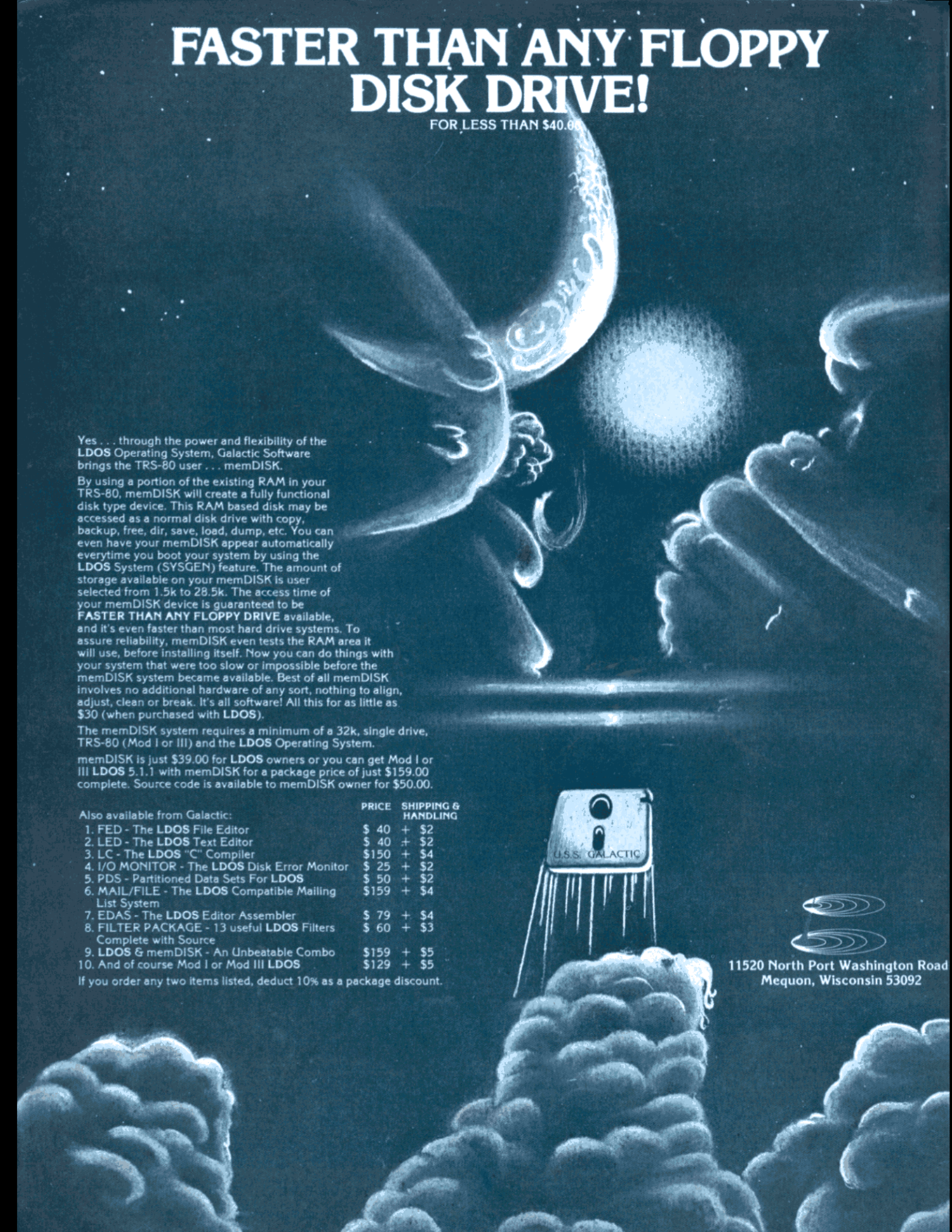
ORDERS ONLY HOURS 9 AM - 5:30 PM E.S.T.
FOR INFO CALL (212) 296-5916

ADD $2.00 PER ORDER FOR SHIPPING AND HANDLING
WE ACCEPT VISA, MASTERCARD, CHECKS, M.O.
C.O.D. ADD $3.00 EXTRA.
NY RESIDENTS ADD SALES TAX
OVERSEAS, FPO, APO, ADD 10%
DEALER DISCOUNTS AVAILABLE

ALPHA Products
79-04 Jamaica Ave., Woodhaven, NY 11421 (212) 296-5916

FASTER THAN ANY FLOPPY DISK DRIVE!

FOR LESS THAN $40.00



Yes . . . through the power and flexibility of the **LDOS** Operating System, Galactic Software brings the TRS-80 user . . . memDISK.

By using a portion of the existing RAM in your TRS-80, memDISK will create a fully functional disk type device. This RAM based disk may be accessed as a normal disk drive with copy, backup, free, dir, save, load, dump, etc. You can even have your memDISK appear automatically everytime you boot your system by using the **LDOS** System (SYSGEN) feature. The amount of storage available on your memDISK is user selected from 1.5k to 28.5k. The access time of your memDISK device is guaranteed to be **FASTER THAN ANY FLOPPY DRIVE** available, and it's even faster than most hard drive systems. To assure reliability, memDISK even tests the RAM area it will use, before installing itself. Now you can do things with your system that were too slow or impossible before the memDISK system became available. Best of all memDISK involves no additional hardware of any sort, nothing to align, adjust, clean or break. It's all software! All this for as little as $30 (when purchased with **LDOS**).

The memDISK system requires a minimum of a 32k, single drive, TRS-80 (Mod I or III) and the **LDOS** Operating System.

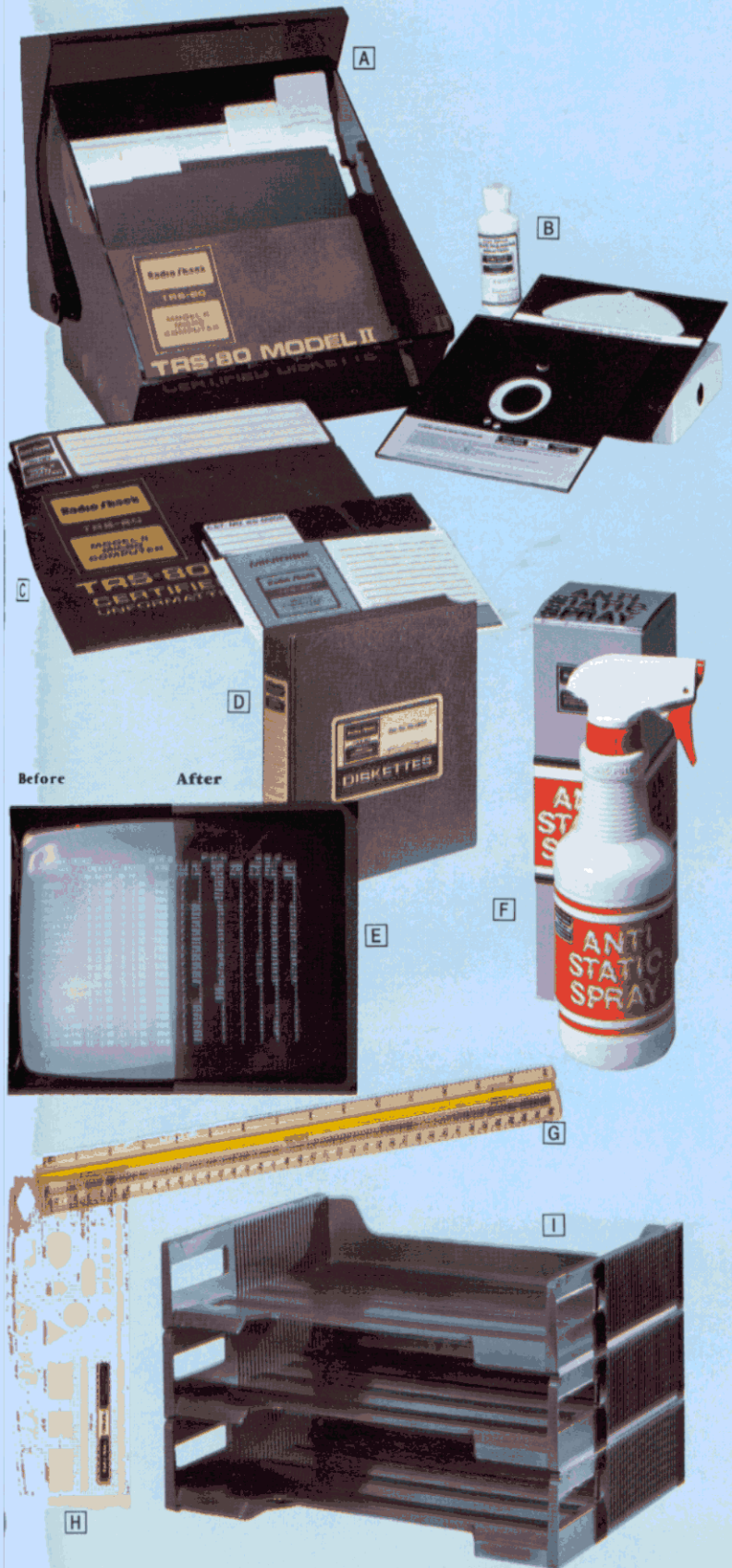
memDISK is just $39.00 for **LDOS** owners or you can get Mod I or III **LDOS** 5.1.1 with memDISK for a package price of just $159.00 complete. Source code is available to memDISK owner for $50.00.

Also available from Galactic:	PRICE	SHIPPING & HANDLING
1. FED - The **LDOS** File Editor	$ 40	+ $2
2. LED - The **LDOS** Text Editor	$ 40	+ $2
3. LC - The **LDOS** "C" Compiler	$150	+ $4
4. I/O MONITOR - The **LDOS** Disk Error Monitor	$ 25	+ $2
5. PDS - Partitioned Data Sets For **LDOS**	$ 50	+ $2
6. MAIL/FILE - The **LDOS** Compatible Mailing List System	$159	+ $4
7. EDAS - The **LDOS** Editor Assembler	$ 79	+ $4
8. FILTER PACKAGE - 13 useful **LDOS** Filters Complete with Source	$ 60	+ $3
9. **LDOS** & memDISK - An Unbeatable Combo	$159	+ $5
10. And of course Mod I or Mod III **LDOS**	$129	+ $5

If you order any two items listed, deduct 10% as a package discount.

11520 North Port Washington Road
Mequon, Wisconsin 53092

Radio Shack Has Everything You Need for Your TRS-80®



A **File Boxes for Your Diskettes.** Two open and shut cases for protecting 5¼" or 8" floppies. Holds 50-60, with index dividers. Durable acrylic.
For 5¼" Diskettes. #26-1453, **$29.95**
For 8" Diskettes. #26-4953, **$39.95**

B **Disk Drive Head Cleaning Kit.** Helps prevent read/write errors. Easy and safe — just apply solution to a cleaning diskette (two included) and run it in drive for 30 seconds.
For 5¼" Drives. #26-407, **$29.95**
For 8" Drives. #26-4909, **$29.95**

C **Genuine TRS-80 Diskettes.** Best you can buy. Certified error-free. Extra-long life, double density, soft sector.
5¼" Single Diskette. #26-305, **$4.95**
5¼" Three-Pack. #26-405, **$13.95**
5¼" Ten-Pack. #26-406, **$39.95**
8" Single Diskette. #26-4905, **$7.50**
8" Ten-Pack. #26-4906, **$59.95**

D **Diskette Library Cases.** Protection against dust and curling. An easel insert pops-up for easy selection. Holds ten floppies. Polystyrene.
For 5¼" Diskettes. Gray. #26-1452, **$3.95**
For 8" Diskettes. Brown. #26-4952, **$4.95**

E **Anti-Glare Mask.** Solves reflection problems. No loss of resolution. For TRS-80 Models I, II and III. Not for color monitors. #26-511, **$16.95**

F **Anti-Static Spray.** Prevents static electricity build-up that can attract dust and damage equipment. Apply to accessories, video display and carpet. 16 ounces.
#26-515, **$5.95**

G **Printviewer.** Magnifies and highlights a single line of your printout. Reduces errors. Clear plastic with yellow "line window." Includes 10 and 12 cpi scales.
#26-1313, **$2.95**

H **Logic Flowchart Template.** This durable, clear-plastic "stencil" with ANSI flowchart symbols and 1/10" lay-out grid makes it easy to draw pro-quality charts.
#26-1312, **$3.95**

I **Stacking Data Trays.** For standard 14⅞"-wide printouts. Eliminates damage to paper caused by using an undersized tray. #26-1309, **each $6.95**

Radio Shack®
The biggest name in little computers™
A DIVISION OF TANDY CORPORATION

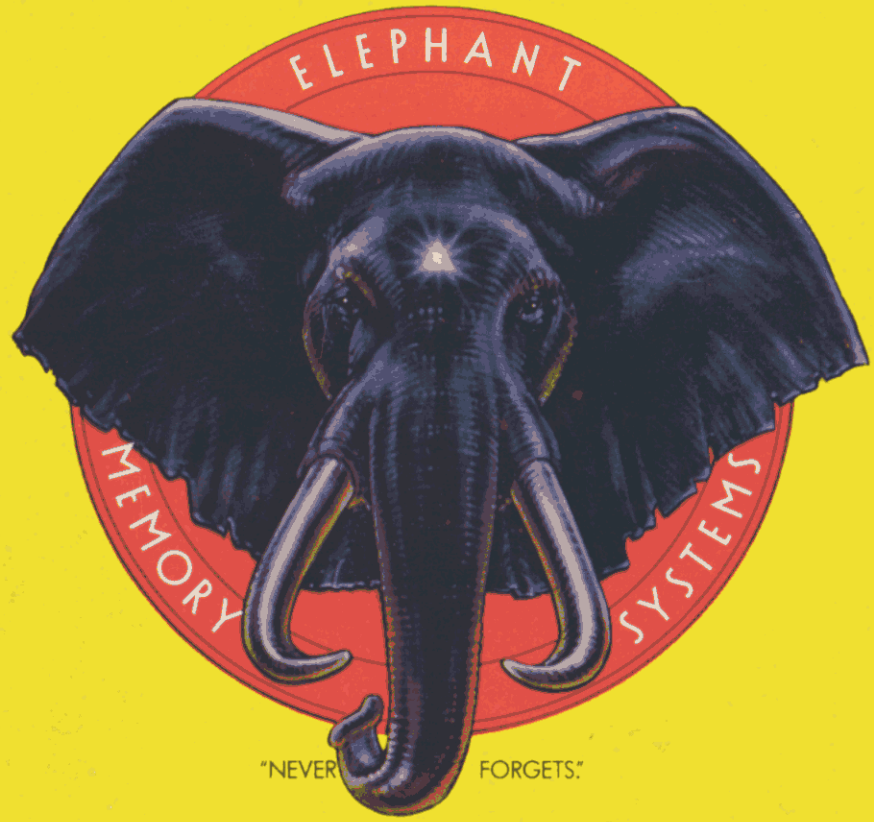
Prices may vary at individual stores and dealers. Some items may require special order.

And there's much more! Send for your free Radio Shack TRS-80 Computer Catalog today!

Mail To: Radio Shack, Dept. 83-A-36
1300 One Tandy Center, Fort Worth, Texas 76102

NAME _____
ADDRESS _____
CITY _____ STATE _____ ZIP _____

REMEMBER:



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.

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.

114