80 U.S.

THE TRS-80 USERS JOURNAL

Vol. VI. No. 2 $3 per copy February, 1983

Computer Languages:

"Is there life after BASIC?"

Reports on

Pascal

COBOL

Pilot

Forth

0 60959 02
0 318585

STRIKE GOLD

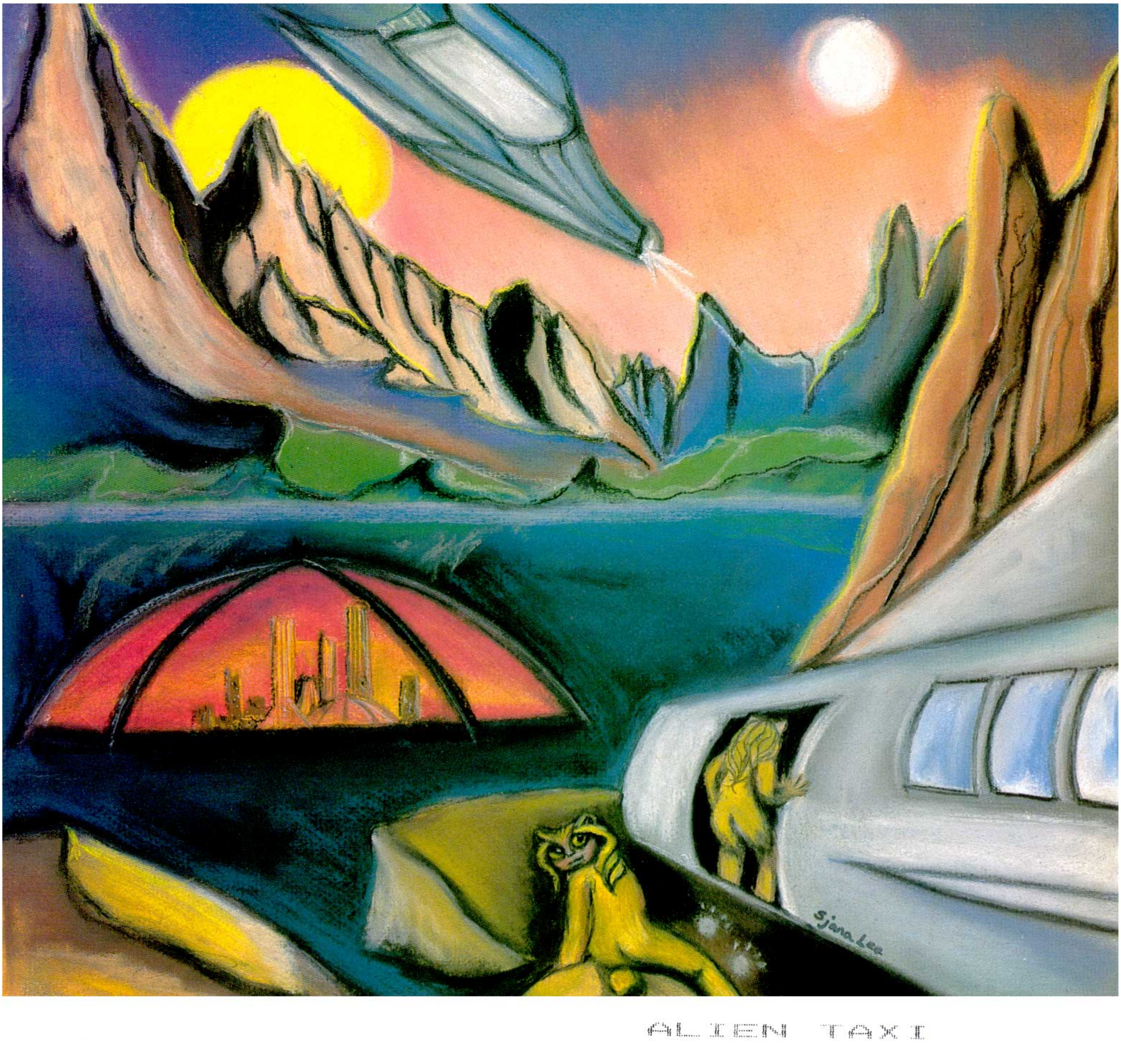
with the most sophisticated TRS-80 operating system available!

LDOS can't really turn your TRS-80 gold, but you'll think you've made the DOS strike of the decade when you turn your micro on to LDOS. You'll find a bonanza of features like full keyboard type-ahead; a true background spooler; file backup by date, class, and between different drive types; hard disk support; data transportability between Model I and III; and a complete communications utility including disk file send and receive. Support for Radio Shack's Doubler and selected others is also provided. With our Job Control Language, you get true "hands off" running of your application programs - give a single command and then walk away. The 400 page manual includes examples of all commands and utilities. The Operator's Guide gives you step by step instructions on how to use LDOS with your applications. Stop running with only "half" a computer! Let LDOS provide the missing features to speed up and simplify your TRS-80 computer system! Visit a dealer or contact LSI for more information on the most popular sophisticated operating system for your TRS-80.

LDOS is available worldwide through thousands of dealers for just $129.

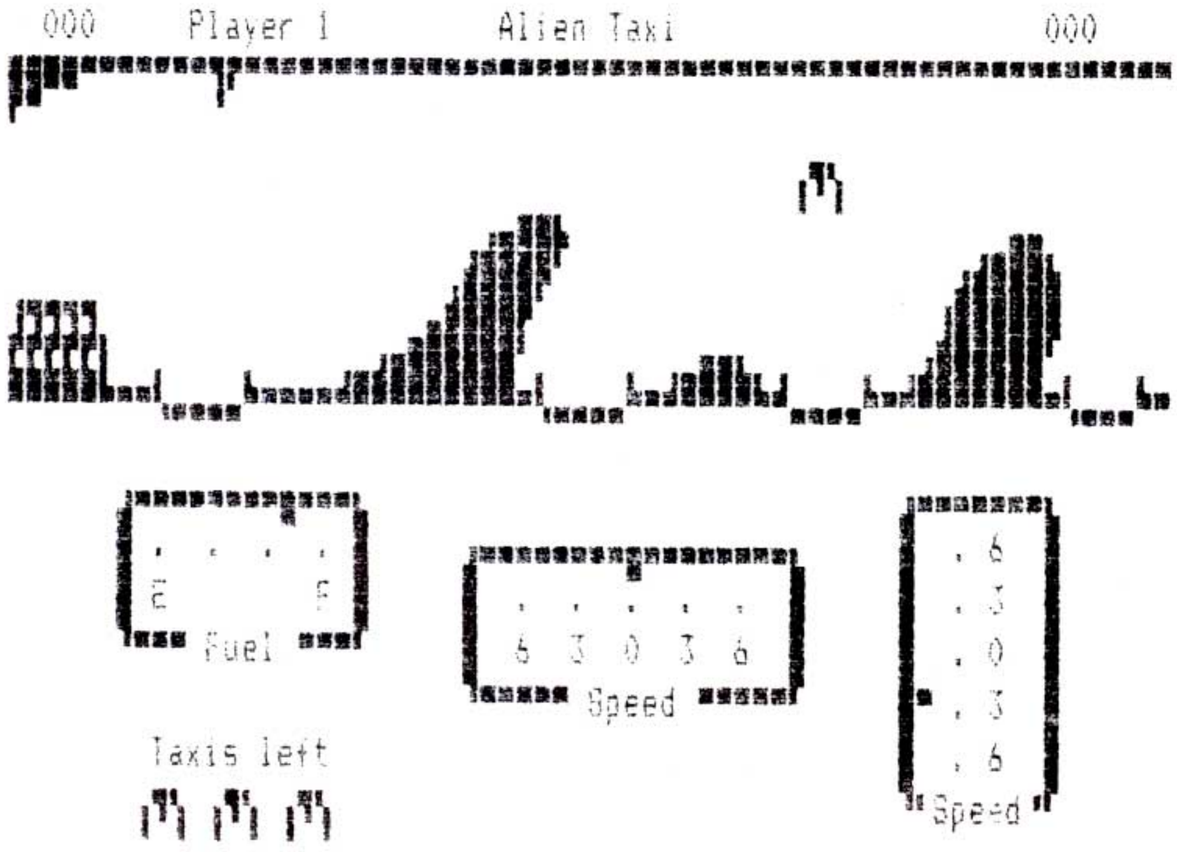
LOGICAL SYSTEMS INC.

Logical Systems, Inc., 11520 N. Port Washington Rd., Mequon, Wis. 53092 (414) 241-3066



ALIEN TAXI

John R. Olsen Jr.



Program Description & Features

- * Graphics
- * Arcade Style Game
- * Top Ten Scores Display
- * Joy Stick Compatible
- * Machine Language
- * Sound Effects
- * One or Two Players
- * Pause Game Features

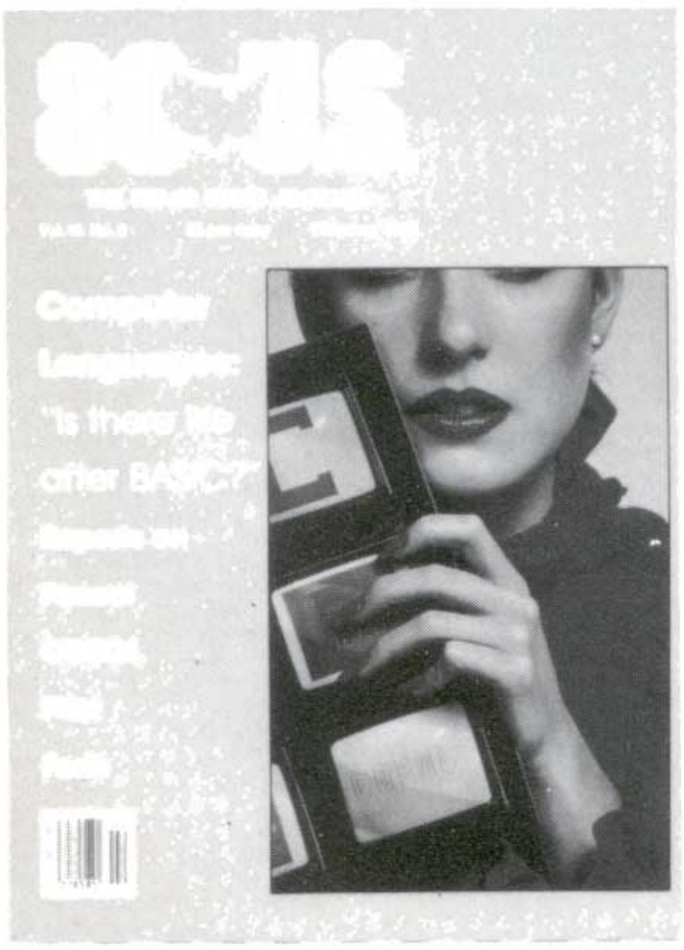
$15.95 tape, $19.95 disc plus $1.50 shipping and handling charges. Master Card & Visa orders accepted. Available at your software dealer.

FANTASTIC SOFTWARE

P.O. Box 27734
Las Vegas, Nevada 89126
(702) 362-1457

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 additional entry points.**



The choice of a new language is being contemplated by our model Lisa Lambert. Art direction was by Stan Shaw and the photographer was Charles E. Taylor. All are from Tacoma, WA.

80-U.S.

THE TRS-80 USERS JOURNAL

TRS-80 is a trademark of the Tandy Corp.

Vol. VI, No. 2 — February, 1983

ARTICLES

A history of languages — **18**
T. R. Dettmann

TRS-80 languages — **22**
For all models — Richard A. Yehle and Cameron C. Brown

Oil tank you not to do that — **26**
Models I/III, PMC-80, LNW80 — Graham Allan
Using defined functions to solve a problem.

Deadstik — **31**
Color Computer — Gary Ludeke
A simplified space shuttle program.

Structured BASIC — **36**
Models I/II/III, PMC-80, LNW80 with disks — T. R. Dettmann
Modern programming techniques in BASIC.

Pascal — **48**
For all models — Mark E. Renne
The advantages of a new language.

Crypto — **56**
Models I/III, PMC-80, LNW80 with disks — Tim Chandler
Have your computer help solve cryptograms.

The BASIC/S Compiler — 62
Pete Carr
Models I/III, PMC-80, LNW80
An evaluation of this package from Powersoft.

CGP-115 printer — 64
Jerry Latham
For all models
A review of this low cost printer-plotter from Radio Shack and a program to help use it.

COBOL — 67
T. R. Dettmann
Model II
An introduction and comparison of three packages.

Basicmon — 74
Ronald Constant
Color Computer
See what is going on inside your Color Computer.

In the chips — 87
Spencer Hall
Models I/III, PMC-80, LNW80
The first in a new series on machine language programming.

Radio Shack double density kit — 89
Harry Avant
Model I

Supervisor calls — 92
T. R. Dettmann
Model II
Part II: Flipping screens on the Model II and the beginning of a disk directory program.

BASIC and Forth — 108
Anthony Scarpelli
For all models
A comparison of the two languages.

80-U.S. interviews John Harding — 110
80-U.S. staff

PILOTPlus — 112
Ranes Chakravorty
Models I/III, PMC-80, LNW80
An evaluation of this CAI language from Radio Shack.

REVIEWS

Color Forth 117
Reviewed by Darrel Wright

Alcor Pascal 118
Reviewed by Paul R. Prescott

Starting Forth 121
Reviewed by Darrel Wright

Bytewriter 123
Reviewed by Steven B. Greene

Penetrator 124
Reviewed by Tim Knight

DEPARTMENTS

Editorial 6
By Cameron C. Brown

Directions 8
By I. Mike Schmidt

Letters to the editor 10

Notes, etc. 17
By Cameron C. Brown

Captain 80 80
By Bob Liddil

Exploring VisiCalc 82
By Timothy K. Bowman

Com 80 84
By Donald L. Stoner

Basically BASIC 97
By James A. Conrad

Tandy topics 103
By Ed Juge

BASIC bits 115
By Thomas L. Quindry

New products 132

Advertiser index 136

Bulletin board 136

80-U.S.

Publisher/Editor-in-Chief
I. Mike Schmidt

Editor
Cameron C. Brown

Associate Editors
Terry R. Dettmann
Spencer Hall
Jim Klaproth

Contributing Editor
Robert W. Liddil

Advertising Sales
Donald Scarberry

Promotion/Circulation
Robert P. Perez

Production
Catherine D. Doud

Editorial Secretary
Eva R. Jones

Accounting
Helen Dalton

RENEWING?

Check your label to be sure it's correct. 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

Cameron C. Brown

Choices, choices, choices. The microcomputerist today has a real problem. No longer can he order any computer as long as it is black. BASIC and its multitude of followers are being barraged by options. It is a confusing and, for many, a frustrating development.

My first programming course involved MAP, Macro Assembly Programming. It was not the best way to start, but it was all the University I went to would offer. From there it was on to FORTRAN (before it was even numbered), PL/I, PL/C, 8080 machine code and finally BASIC on a Hewlitt-Packard 2000E. By the time my journey through the languages reached BASIC I was about ready to pack it in.

BASIC was a savior to me. It finally gave me confidence and ease at the terminal. Now, whenever I program, I use it. For me, it is clear, easy to follow, and makes a lot of sense. But, in many cases it is clearly deficient. I know of many programs I have written that would have benefitted from the strengths of another language.

Years ago, a friend demonstrated the power of APL to me. The ease in which he was able to invert a matrix and transpose its entries was amazing. I can't remember a bit of the code, but his enthusiasm is something I won't ever forget. It was probably my first introduction to a cultist. Later, I noticed the same emotions at a meeting of Forth users at the San Francisco Computer Faire.

It seems to me that anyone who expounds on the virtues of "his" language over all others is really off-track. Languages are developed to meet specific needs. They each approach the same problem, getting a machine to behave, from totally different directions. Perhaps ease of output is desired, or maybe handling extremely precise numbers is needed. Don't expect any one language to be able to perform all functions for all users. It can't be done.

I do wish that some of the offerings were easier to use. Loading an editor, writing in a new language and keeping the editor's syntax in mind, then loading the interpreter so that the commands can be translated into Z80 (or 6809) code and then executed by the ROM is a rather round-about method. But it is better than having no choices at all.

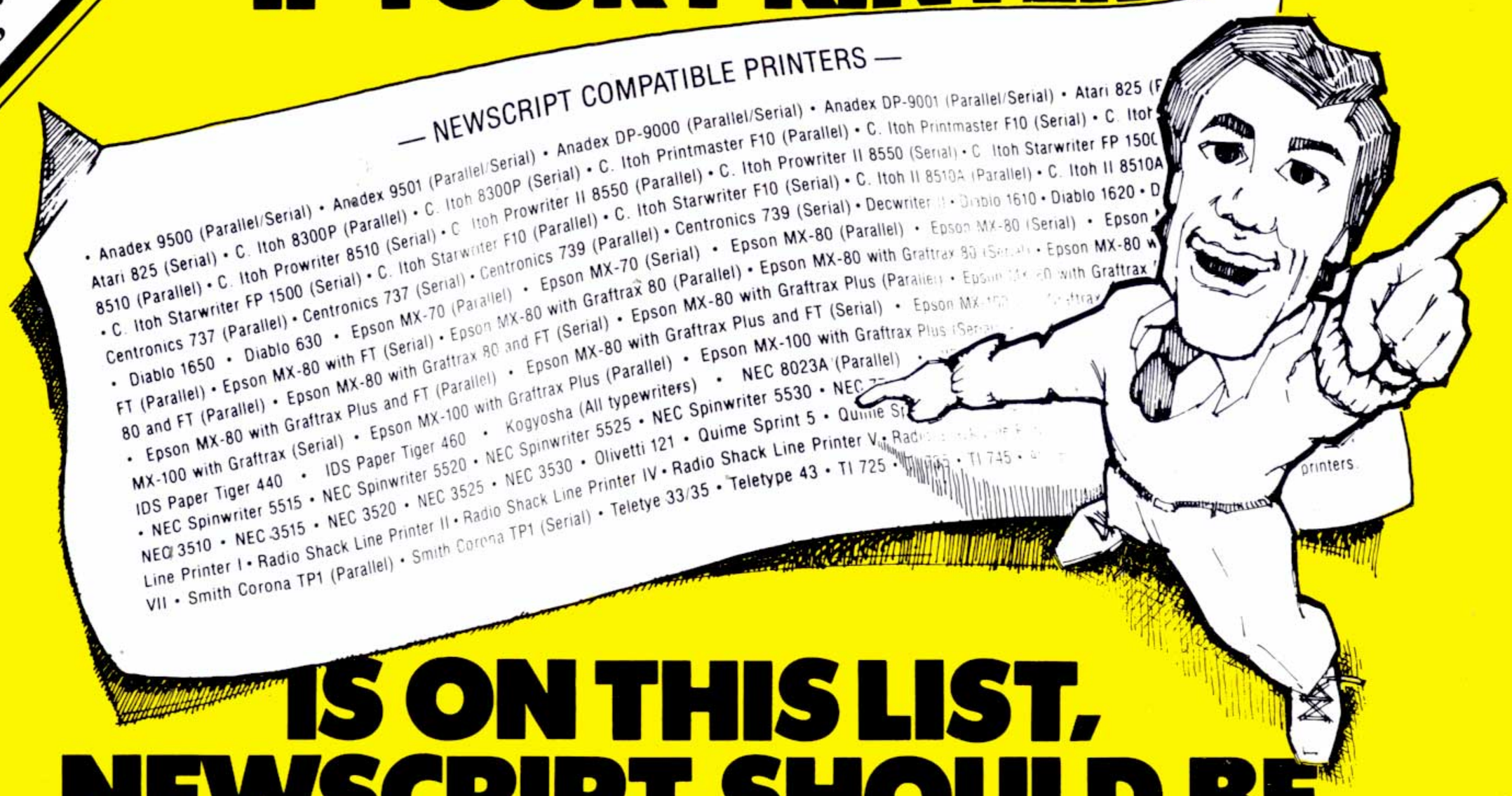
The future is promising. It appears that programming itself may become a lost art. More and more offerings of programs that "write programs" are coming onto the market. In a few years we will be able to tell our machine what to do and it generates the code. At the moment, it is very structured and most program "writers" have such strict rules to follow that they are almost a new "language" in themselves. The attempts bode well for the future, but they aren't quite there yet.

There is also a problem of standards. Be ready to give great attention to detail. Be sure that the language works on your system, look at its compatibility with other versions, other configurations. The decimal points in version numbers can spell the difference between smooth operation and unusability. Does the version you want work with your DOS, with your size disk drives, on tape, drive a printer, format for your video, support all the language commands or just a subset? Caveat emptor is really in effect. Purchase carefully, but I do believe you should give it a try.

The plethora of available languages is a real advancement. Pascal, FORTRAN, COBOL, Pilot, BASIC, Forth, and others open up whole new worlds for us. Each one gives the programmer a different way in which to view a problem — a different way in which to seek a solution. That is exactly what we bought our computer for. Don't be shy, a new language may turn out to give you a new view on an old problem. ■

Special: SAVE $15
NEWSCRIPT + Mailing Labels
$139.95

"IF YOUR PRINTER



IS ON THIS LIST, NEWSCRIPT™ SHOULD BE ON YOUR COMPUTER."

NEWSCRIPT'S exclusive print processor gives you total printer control.

NEWSCRIPT's exclusive print processor takes over where formatting leaves off with over 200 print processing and support features. NEWSCRIPT takes you beyond text formatting — beyond just printing text on paper. With NEWSCRIPT's Print processor you take full advantage of your printer's hardware and software features, creating an unsurpassed printed image. Text editing (the part you see), is only part of the job, printing is the other part — the part others will see.

NEWSCRIPT controls over 80 popular printers.

You don't even have to know how your printer works — only what it can do. NEWSCRIPT will do the rest with such major features as underlining, right justified proportional printing, top and bottom titles, top/bottom left-right-center page numbering, chaining and embedded printing of any length file with disk spanning, fully formatted merging of form letters with selective editing, boldface, sub/super scripts, character substitution/translation, table of contents, indexing, hanging indents, paragraph numbering, line numbering, double width characters, italics, hard and soft hyphens, in-memory spooling, and many other features.*

NEWSCRIPT'S text editor has sophistication to match its print processing.

Buffered key entry rates to 450 characters per second — you never drop characters. Windowing to 240 characters, block move and copy within and between files, definable auto save, "HELP" and "WHOOPS" commands, repeat and query last command, search and replace within column and line limits as well as globally, and an automatic interface to the ELECTRIC WEBSTER spelling checker (sold separately).

NEWSCRIPT'S 277 page manual contains an introductory tutorial with explanations of the beginner's most common needs, a "How to Section" to help when you're stuck, a fully alphabetized description of the commands with literally hundreds of examples, a topical index with over 1,300 entries, and a handy quick-reference card (naturally!).

Start getting the printed results only NEWSCRIPT can give you for $124.95. Requires TRS-80 Model I or III with 48K and 1 disk (minimum — 2 recommended)

Newscript T.M.

is available at computer stores, selected B. Dalton Book Sellers, and selected independent book dealers. If your dealer is out of stock order direct. Include $3.00 (domestic), $6.00 (Canada) for shipping and handling. Foreign residents add $15.00 plus purchase price, in U.S. funds.

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 us.

PROSOFT®
Dept. C, Box 560, No. Hollywood, CA 91603

NEWSCRIPT companion programs (sold separately): MAILING LABELS $29.95, DAISY WHEEL PROPORTIONAL $49.95 (not required for Daisy Wheel II), PENCIL & SCRIPSIT FILE CONVERSION $24.95, ELECTRIC WEBSTER (spelling checker and automatic correction) $149.50, GEAP (TRS-80 graphics — requires Epson MX-80) $49.95, DOTWRITER (Hi-res graphics — requires Epson MX-80/100 with Graftrax) $69.95, GEAP/DOTWRITER combination (requires Epson MX-80/100 with Graftrax) $99.95

Dealers: NEWSCRIPT is distributed by IJG, Inc. (714) 946-5805

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

Directions

I. Mike Schmidt, Publisher

The small computer industry is a particularily curious animal. It has an impact on almost any human endeavor, and is as likely to be found in a beer hall as in a classroom. Last year, I had an interesting conversation with a gentleman who claimed to have written programs for a microcomputer used in a house of ill repute. The programs were not exactly like any other business with accounts receivable, but the computer worked as well there as in any other situation.

Currently, about 3 percent of American homes have a computer. This is expected to rise to about 85 percent in the next five years or so. Are we going to become a nation of programmers? I doubt it. Canned, ready to run programs for any occasion will probably be the main thrust in computing in the future. Scripsit, VisiCalc and Profile are already leading the way. This type of program takes the drudgery away from the end user, and allow (almost) the full flower of the computer to be experienced.

There are even programs on the market which are reputed to be able to write BASIC programs. Talk about pulling oneself up by the bootstraps! Although I am still waiting to see the first of these, I understand that you still have to know a whole lot about programming to use one. No doubt, this will be refined with use and demand from the end user.

The other thing which is curious about the small computer industry is its performance on the stock market. There was a time, of course, when only the giants of the industry were traded there. IBM, Digital Equipment, Control Data Corporation and others like them had the market all to themselves. Now, we have the Tandy Corporation, Apple, Commodore and Warner Communications, all traded on the big board.

In December of last year, Warner Communications announced that their expected fourth quarter earnings would not be what they had expected. Not that they were bad - just that they were not what they were expecting. This announcement caused a ripple effect through the rest of the industry. Warner's stock went from 17 in 1980 to 63 early in 1982. Their concern about fourth quarter earnings was due to uncertainty over the durability of income growth based on video games. Even though they were concerned about video games, others felt the impact and Commodore International stock dropped 9 points, Tandy fell 3 and Texas Instruments dropped 5 points. The shocker here was that Digital Equipment also fell three and a half points.

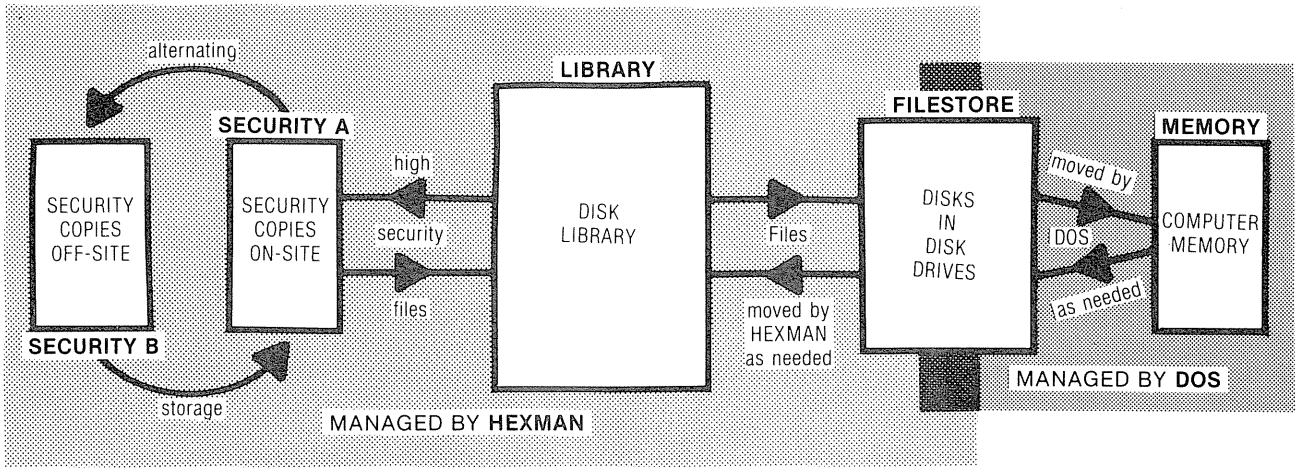
Apparently, the stock buying public still does not know the difference between video games and personal computers. In fact, they probably don't know the difference between personal computers and mini-computers, since Digital Equipment stock also fell, though it may have been coincidence.

It is surprising that during these times of high unemployment and general bad times that consumer electronics in general, and microcomputers in particular, are doing so well. I am sure that the December, 1982 setback was only temporary, and that soon we will see a resumption of the steady climb to even greater heights. If 85 percent of American homes have computers by 1990, we had better get started. ■

HEXMAN D.M.S.

DISK MANAGEMENT SYSTEM

Computerized file storage and retrieval for your whole disk library.



Your Disk Operating System makes a fine job of managing the files that are in your disk drives. No need to worry about where your old files are kept or where to store new files, your DOS takes care of it. Now HEXMAN takes over where your DOS leaves off, and extends worry free file storage and retrieval to your whole disk library.

Storage Management

Only a small portion of your disks can be in your disk drives at any one time. These disks form the "Filestore" to which your computer has immediate access, and the remaining disks form your disk "Library". Ideally the "Filestore" should contain the most active files, while inactive files are kept on "Library" disks stored near the computer.

HEXMAN comes as close to this ideal as it can by monitoring file activity and keeping the most frequently used files in the Filestore. If previously inactive files are needed, they are moved in from the Library. If the Filestore becomes full, the least active files are moved out to the Library. The net result is the files you are most likely to use are ready and waiting.

HEXMAN also performs other storage management chores such as daily backups of modified files, on-site and off-site storage of security copies, and file growth monitoring.

File Retrieval

HEXMAN manages up to 8000 files (2000 originals with up to 4 copies of each). If remembering that many eight letter file names gives you a headache — don't worry. HEXMAN gives you two easy ways to find the files you need. With HEXMAN you find files by function e.g. "Get all the files I need for the end of month Sales Reports" or by description e.g. "Find the letter I sent to Radio Shack about my disk drives". HEXMAN finds your files in seconds. Once you have found them you may Review them, Load them or Delete them as needed.

Requirements

Model III — 48K, 2 drives.

Model I — 48K, 2 drives. Double Density adaptor. Lower case modification.

LDOS operating System. If you do not have LDOS, please add $35 for a copy of the smal-LDOS operating system.

US $169

VISA master charge

90-day trial · just $20.

We are sure HEXMAN will become a "must have" program for all serious TRS-80 users. But check it out for yourself. Purchase HEXMAN now and try it out. If you feel you can live without it (unlikely!), return it for a prompt refund less a $20 evaluation charge.

Offer valid for direct sales only - dealer terms may vary.

WARNING — TIMELOCKED SOFTWARE

This product is timelocked. Unregistered copies will expire at a future date. Registered owners will receive an extension patch when needed.

A NOTE ON CANADIAN MAIL — please allow 5 to 10 days in each direction for mail delivery. For faster service leave your orders or enquiries on our 24-hour answering machine.

HEXAGON SYSTEMS

P.O. Box 397, Station A
Vancouver, B.C. Canada V6C 2N2
Telephone (604) 682-7646
Electronic Mail-Micronet 70235,1376

141

Letters to the editor

This is in response to the article by Ralph Vickers in the November 1982 issue. I have found that POKE 4099H,0 just prior to the GOSUB will also eliminate the problem with the INKEY$. I also enjoy your magazine very much, with a lot of benefit derived.

A. Robert Meyers
San Jose, CA

Very much enjoyed Ron Goodman's article in your October issue on recovering a lost or NEW-ed program. In the same vein, may I offer the following?

A program that has been lost, either through an accidental NEW command, or some machine language bug, can be recovered by a short two step process. First, while in command mode, type POKE 17130,1. Second, still from command mode, type SYSTEM. Answer the prompt with /11395. Now list the program to find it has returned.

Like Mr. Goodman's method, this method has its limitations. DELETEing lines can be hazardous in some circumstances. I have also found that I seem to have better luck if the first line of the program is 10 CLS. This is often a good starting point anyway, so perhaps it isn't that bad. I might suggest that the recovered program be CSAVEd before attempting to clean it up. That way, at least a major portion will be saved, making reloading and debugging easier.

C. Russell Eurich
Pottstown, PA

We found that on our 48K Model I, without disks, that a program would not run and afterwards the first line is turned into garbage. But it does list out quite nicely. Trying to edit also locks up the computer.-Ed.

More on PRINT to LPRINT

In the October 1982 issue you had several short programs under the main heading Tips and Tricks. Where these programs do have potential, one in particular has the ability to be disastrous. The program in question is PRINT to LPRINT by Ray B. Harrill.

This short program will change your PRINTs to LPRINTs as stated but one must be careful when searching through memory in this manner. By starting at the beginning of a BASIC program, 17129 (no disk), and searching each memory location until the end of the program is encountered, you are also checking line pointers and line numbers. This is where the problem exists. The line IF PEEK(X) = 178 AND PEEK(X+1) < > 64 AND PEEK(X+1) < > 35 THEN POKE X,175 could change a line number or totally crash the listing by changing a line pointer. As an example of this, the following line numbers contain the number 178 when stored in memory: 690, 1970, 3250, 4530, 5810, and so on. If your program has GOTOs or GOSUBs this could end up in one big mess. This could also happen with line pointers. For example take 178 + 70*256, this would point to memory location 18098. If this was one of the line pointers and it was changed by the POKE routine it would mean the end of your program. I believe a BASIC routine could be created to eliminate this error, but BASIC programming is not exactly my area.

If you are going to use this routine I suggest that you have the program you are about to change saved on tape and check the finished product after you use this program. It might work (probably will) but it might bomb everything. A machine language routine would be much better for this operation. If anyone is interested they may send me a SASE and twenty-five cents and I will mail them such a routine.

Truly yours,

Theodore J. LeSarge
6027 W. Decker Rd.
Ludington, MI 49431

The BASIC patch you refer to was given in the December 1982 Notes section, page 14. Your offer of supplying source code to our readers is most generous, thank you.-Ed.

Editor: I am planning to convert my TRS-80 Model III to operate under CP/M. I would like to see articles on conversion, use of CP/M and programs that are available.

G.A. Downsbrough
State College, PA

Take a look at Harry Avant's article on the MTI hard disk system and its use of CP/M in our December 1982 issue. Terry Dettmann's article on CP/M in our May/June 1981 issue should also be useful even though it was oriented toward the Model II. We would like to review more CP/M packages for our readers, but very few come our way that are not configured for just the Model II. We have over twenty-five reviewers on a regular basis, and only two have an expressed interest and expertise in CP/M. -Ed.

In the September, October and November issues there have been somewhat glowing reviews of the accounting packages by Plus Computer Technology for the Model

III. We have been using these packages since May of 1982 in our office and feel that your readers are being somewhat misled by the reviews in your magazine.

The manuals for these packages seem impressive and complete when they are initially read, but upon actually using the system they are very incomplete and leave large gaps in the documentation. Your article referred to the manuals as being each almost two inches thick in a three ring binder, but upon actually measuring them they vary from three-eighths to five-eighths inches.

As you stated in these articles, these accounting packages are very comprehensive, which is true. However, they are full of program errors which prevent them from running smoothly. The "excellent" support from Plus Computer Technology consists of their admitting when you call them on the telephone that there is an error in the program and their statement that a correction or revision will be forthcoming in "approximately two weeks." We have been told this on numerous occasions, but have yet to see even one correction or revision. In May when we wrote our first payroll we found significant errors in the program involving the state tax and we immediately, at their request, mailed them a copy of our payroll data disk and a copy of our state tax booklet. We were to receive some changes "within two weeks or so" and now, five and a half months later we have continued to receive excuses and promises of "two weeks or so" but we have no changes.

The payroll and accounts payable modules are supposed to be able to use either NEBS or Wilson Jones checks, but if you use NEBS (as we have) the name is in the wrong place on the check and it is impossible to use a window envelope. I understand that Plus Computer Technology is now part of the Wilson Jones Company.

We purchased four packages from Plus Computer Technology (payroll, accounts payable, fixed assets, and general ledger) and after our sixty days of free support ran out we sent them $400 for continued support for these modules. We have never received a technical bulletin or newsletter as is promised in all of their advertising.

In summary, these financial packages would be what your articles purport them to be if Plus Computer Technology would "get their act together" and correct their many program errors and support packages. The basic design of the packages is excellent, but the documentation and support is basically non-existent.

James O. Shaw, M.D.
Ashland, WI

Your letter has raised numerous issues. We called Plus Computer Technology for some extra information and spoke to Mr. Gary Simon, President. He was familiar with your problems and some of them have been solved. The package is being fixed to fit the NEBS forms. They did work when originally

released but it is possible that the form has changed. A second newsletter is being readied and should arrive anytime. They are expecting to produce a newsletter about every other month.

A new release was scheduled for December 1982. As with most sophisticated packages, it is better to fix a few bugs all at once rather than inundate the customer with a new version for each one. In no case has a bug ever been found that has made the package inoperative. The manuals were upgraded in the summer of 1982 and it is quite possible that the documentation in your version was not the same as the one we evaluated. According to Mr. Simon the tax computations were derived from an IBM mainframe system and any error you are encountering should also have been reported by the IBM users. At the time of our conversation he was not aware of any such error, but his staff is very carefully investigating your data.

The packages are quite comprehensive and it is unreasonable to expect any reviewer to investigate all possible options such as the tax table of each state or every possible form that could be used. In any review we have to take some information from the manufacturer at face value. In the time we have to do a review it is not possible to verify every option in every program. We seek to give you timely and accurate reviews and will report honestly what we discover.-Ed.

Dear "Mike",

RE: Nov 1982 issue...Duck! Here it comes.

Page 12: Killer must be a chicken? Page 26: Table 1 is upside down. I'll excuse you for this. You may not have known on which page the table would actually appear, but your printer is getting paid enough to know his top from his bottom. Page 39: Brilliant recovery by Don Scarberry - "Whom are you going to call?" Only owls go around calling, "To Who, To Who!"

I'm glad I'm writing this, for you are just about taking over from the Computronics group as my favorite magazine; and if you work very hard, you might out rank "BYTE". Well, that will keep you-all busy for a while.

Also, here is my pet peeve. Why after twenty years of education in this country cannot Americans speak, read, and write English? And, why do they tell me that they can write BASIC after only two weeks? If you cannot write the English comment, do not write the BASIC code!

And, I liked your magazine better when you did not have the four-color glossy center spread. Does four-color glossy advertising discolor the text? I hope not. All we need are the facts, and fairy tales can be found in dandy Tandy advertising. I know a couple of RS store managers I would gladly boot over the Rockies if you want them. Just tell it to us straight. Tandy's programs cannot be perfect, for even mine are not! - always! - sometimes! - damn it!

And one more thing: Just who is

writing "Basically BASIC" anyway? In any case, please thank James A. Conrad; and with this letter, you should have no trouble at all convincing him that I am one of his "Grizzled Old Pro"grammers.

F. L. Eskholm
Nutley, NJ

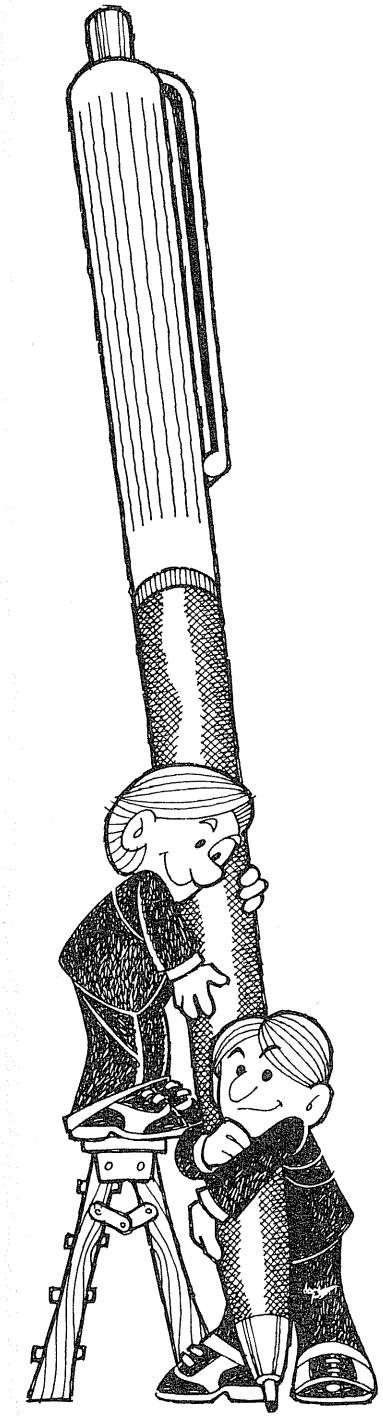
Our printer was correct. We meant for you to read it while standing on your head. According to Mike, Killer is a combination of German Shepherd, Husky and JD (just dog). —Ed.

I had heard about your magazine for quite a while but I was unable to find it. Recently I found it in a newsstand and was very happy with it. I decided to get a subscription right away because it is the first magazine where I really found information and programs for the Model II.

I own a TRS-80 Model II and use it mainly for word processing and VisiCalc applications... I have been told that the RS hard disk will not work with the present version of Scripsit 2.0. I have been told that a new version will work only with the Radio Shack hard disk and no other one. Do you think that another hard disk could support Scripsit? Has anyone ever had a positive experience with Scripsit and a non-Radio Shack hard disk? If yes, please get in touch with me...

M. Gutelman
393 West Broadway
New York, NY 10012

Our information is that Radio Shack has a hard disk patch for Scripsit 2.0 that is available to owners for no or minimal fee (catalog number 26-2831). The problem is due to the fact that the hard disk drivers and Scripsit both overlay in the same region of the ROM. The problem is really with the current version of Scripsit 2.0 and the operating system you use, not the hard disk. Even the DOSPLUS II people have given up trying to get out a Scripsit patch for the Model II (in either floppy or hard disk configuration). We are convinced that this problem will be solved very soon, probably with the release of a completely rewritten Scripsit. Word processing, the Model II, and a hard disk are a natural combination and the current situation has to change.- Ed.



The current issue of Radio Shack's TRS-80 Microcomputer News *(December, 1982 -Ed.)* announces that you are offering a sample copy to their readers. I have never seen your magazine, as the few stores in my area do not carry it. I would appreciate your sending me a sample.

I have a Model II and am naturally interested in material related to that machine. I am delighted, as I am sure you are, that Radio Shack has finally shown the good sense to cooperate with the independent specialist magazines. This change in their attitude will work to their good, to the publisher's and software writer's good, and certainly to the good of those of us who use Radio Shack computers.

Thank you for this assistance.

James F. Waters
Humboldt State University
Arcata, CA

Your sample copy is on the way. We agree with your assessment of the recommendation by Radio Shack and share your view of its impact on TRS-80 owners. By the way, more and more B. Dalton stores now carry 80-U.S. Journal, you can look for us there. —Ed.

The Mr. Computerhead game by Kenneth Gibbs in the November, 1982 issue is a hit with my 18 month old son. However, he is not too keen on pressing only the number keys 1 through 5. To take care of this, I suggest the following modification that allows any key to be pressed. Delete lines 150 to 190. Add the new line 150 Z1 = (ASC(IN) + 1) - INT((ASC(IN) + 1)/5)*5 +1 and the line 160 ON Z1 GOSUB 210, 400, 530, 660, 800. The new line 150 uses modular arithmetic to calculate from the ASCII code a value from 1 to 5, regardless of the key pressed. Note that the ASCII value is still in the 1 to 5 range as Mr. Gibbs specified. Now, any key or combination of keys (except BREAK and SHIFT) will result in a change of the display. As a reminder to TRSDOS 2.3 disk users, POKE 23886,0 disables the BREAK key.

Brandon C. Nuttall
Frankfort, KY

And who said kids are too young to start computing? —Ed.

Concerning your Puzzler #3, the origin of the word "debug" is as follows: Mario Minestrone, on September 7, 1623 was frightened by a huge crawling insect crawling across his map of downtown Sicily, and cried out to his brother Clyde, "Debug! Debug!" Naturally, Clyde wrote this down in his diary, which was found in 1944 by a doctor named Doctor Schlemmenhauserkinderfelter, in the Advanced Watercolor basement of M.I.T. The good doctor (his friends called him "Schlemmy") upon reading of this incident, promptly forgot it, except for the word "debug." (He couldn't read Italian anyway.)

Later that day, Schlemmy was walking down the hall past The Machine, and heard a poor operator crying over his latest programming failure. Schlemmy was lost in his own world, and when the operator moaned, "What am I to do?" Doctor S. mumbled, "Debug." The bright boy said, "Of course! Debug! Why didn't I think of it before?"

Doctor Schlemmenhauserkinderfelter did not notice this, and walked on by. However, the young programmer (Stanley Weber) remembered this word and told all of his friends (there were two). So, from that day, we have been using the word "debug."

Gary Teter, Paradise, CA

P.S. This story is true, as told to me by Don Juan Minestrone-Schlemmer-Martinez (some guy that was rolled a block from our offices here at W.I. (Weber Industries)). Honest!

We have another winner! —Ed.

DOS Users Respond

December, 1982 was a fine issue, as always. No complaints, merely some comments relative to disk drives and their operating systems.

For the Model I, I have TRSDOS 2.1, 2.3, 2.3B, 2.7DD, NEWDOS 2.1, NEWDOS/80 Version 2.0, DBL-DOS, and SOS (from the IJG book *Machine Language Disk I/O and Other Mysteries* by Michael J. Wagner). Actually, I guess you could say I am a collector of operating systems. I also have available two systems, one with a Percom Doubler II and the other with a Tandy Double Density Adapter. Aside from operating systems, I can swap disk between them with a minimum of problems.

...I have yet to find anything, other than NEWDOS/80's SYSTEM and PDRIVE commands that will let the operator specify his drive parameters and maintain them. They are usually either preset or run under automatic density recognition, with the tracks limited to

SOFTWARE

IT'S JUST GREAT

ON SALE NOW

IJG

EXTRA! EXTRA!

50,000 Word Dictionary and Spelling Correction for TRS-80 Model I & III.
Electric Pencil 2.0z' Integrated Word Processing System!

Dateline: January 1983.

IJG has done it again.

First, IJG brought the world the Electric Pencil 2.0z Word Processing System. The easy to learn, easy to use word processing system considered by many to be one of the finest available.

Now, IJG brings to the Electric Pencil system . . .

Blue Pencil, the Expandable 50,000 Word Dictionary.

That's right. A 50,000 word expandable dictionary that becomes an integral part of your Electric Pencil.

A virtually unlimited number of new words can be added to your Blue Pencil, making Blue Pencil especially useful for proper names, acronyms and professional terms.

No need to exit the operating system or run any other programs, just a simple two key command and Blue Pencil is quickly checking your text.

It's so simple.

Spelling and corrections are then easily accomplished by using Electric Pencil's search-and-replace feature. Or, if you would like your system to do all the work, there also is . . .

Red Pencil, Easy Spelling Correction.

Red Pencil brings together all the features of Electric Pencil and Blue Pencil, with a flexible spelling correction program that makes copy proofing a breeze.

Instead of dropping you off to edit and correct your own text, your enhanced Pencil will now go down the list of misspelled words with you, word by word, and offer you several options from a menu.

You can type in the correct spelling, leave the word alone, display the word in the sentence and then make your correction, display the dictionary at the nearest location and scroll up or down to find the correct spelling, or add or delete the word in the dictionary.

Then Red Pencil returns you to your Electric Pencil file with all your corrections present in the text, and your new file saved under a special name on the data disk. Your original file is untouched.

Fast and Easy Proofing.

Both Red Pencil and Blue Pencil are available (disk only) for $89.95 each at computer stores, B. Dalton Booksellers and independent book dealers around the world.

Along with the Electric Pencil 2.0z Word Processing System for $89.95 (disk version).

If IJG products are not available from your local dealer, order direct from IJG.

Include $4.00 for shipping and handling. Foreign residents add $11.00 plus purchase price. U.S. funds only please.

IJG, Inc. 1953 West 11th Street
Upland, California 91786
Phone: 714/946-5805

IJG

Helping You Help Yourself.

#8

accepted standards. ...My drive 0 is running 91 tracks, double density, with 43 tracks on drives 1, 2, and 3. Strictly for developmental and personal work. Under both NEWDOS/80 with the PDRIVE specs set and TRSDOS 2.7DD (as I have zapped it), the systems work fine. I do use 35 or 40 tracks on finished products when I turn it over to someone with a "normal" system.

I knew that most disk drives had some breathing room, so, after careful testing, I got the 91 tracks on a Micropolis 77 track drive, the 43 tracks on a Tandy/Tandon 40 track drive, 44 tracks on a regular Tandon 40, and even 36 tracks on a Tandy/Shugart 35 track unit. The track 0 on the Micropolis is a single density boot track, but that still leaves 90 double density tracks. I have also specified a four Gran directory, which I haven't seen any DOS do, except NEWDOS/80.

Obviously, I am more than partial to Apparat, even if they did leave TRACE out of the NEWDOS/80 library. So far, I have found nothing that will beat it unless I buy a hard disk. Even then, I am not sure. After all, it's very hard to beat the flexibility, and a Meg of floppy storage. I guess it boils down to: Use whatever DOS you need to do what you want.

William E. Allen
Metairie, LA

The Radio Shack Double Density Board is a well built upgrade to any Model I TRS-80. There is one small problem, the TRSDOS 2.7 (I hear it is soon to be 2.8). This DOS is extremely slow to run. I researched the DOSs for one disk systems. While LDOS and NEWDOS/80 are super for programmers, and DOSPLUS is very friendly to use, the winner for single drive systems in terms of size and speed was MULTIDOS.

With version 1.5 for the Model I doing automatic recognition of which doubler is installed, and its Super BASIC, it is now my double density DOS choice. I utilize many different communication programs and use the power of the BASIC in MULTIDOS to make a menu program that can get me from, and to, different programs with minimal hassle.

The ability to pass variables and nest programs has proven to be a handy feature. Instead of my having to go to DOS, type the program name (e.g. XMODEM), the function (e.g. send), and then the file I am sending, all I have to do is boot my communications disk and follow the prompts that I put in my BASIC menu program.

Mike Stark
San Diego, CA

I thoroughly enjoyed Bob Bowker's article on Driver Education in the November, 1982 issue. The only use he pointed out for changing the driver pointers in the device control blocks was to write your own driver programs. By poking the video DCB values into the printer DCB you can send everything that was to go to the printer to the video. This is a simple way to delete LPRINTs without changing the program. To do it, just type in the code FOR A = 16413 TO 16420: POKE A+8, PEEK(A): NEXT. You may insert this line in a program or use it in command mode.

John M. Havercroft
Newcastle, WY

Thank you. After three issues with PRINT to LPRINT it is nice to have it the other way. —Ed.

It was refreshing to read your December editorial about the Tandy Corporation and computers. In general, I find that there is an inverse relationship to the amount a person bad mouths Radio Shack and their overall knowledge of microcomputers. Too bad the bad mouths don't read *80-U.S.*! To these people, when a TRS-80 breaks down it is called trash. When another brand breaks down (as they often do), or the local dealer goes out of business (as they often do), nothing is said.

I find Radio Shack a better than average computer company. Sometimes they cut corners too much, but often they are a leader in new ideas. They are certainly fallible, but the other micro companies seem to have major problems too. Speaking of fallible, what happened to all those Color Computer articles promised earlier?

Brian James
Willamette Valley Color Computer Users
Eugene, OR

We always try to put at least two substantial Color Computer programs in each issue. Quality is not easy, and we prefer to give readers programs of note rather than pack each issue with Color Computer programs of questionable worth. As we acquire more programs they will be included. —Ed.

I smiled broadly as I read the editorial by Lawrence I. Charters in the December, 1982 issue. "What is odd is the lack of loyalty by Radio Shack computer owners and the TRS-80 press." I heartily agree with his statement, but it does not apply to this TRS-80 owner.

Orville Potter
Cocoa Beach, FL

I must applaud your December editorial. I have an Apple computer associate who refers to my TRS-80 as the trash-80. All he ever talks about is how much better the Apple is than other computers — especially the TRS-80. With your permission, I would like to make a copy of the editorial and send it to him.

Also, I must include some objective criticism. In reference to the remark "Compared to IBM, no one is very impressive", you seem to be among the very large circle of people who are unaware of the American Telephone and Telegraph Company. Accomplishments of AT&T include creation of the transistor, the UNIX operating system, the C programming language, bubble memory, the 1E, 1A and MAC8 processors. Also they are the largest private employer, have sales of $45.4 billion, and profits of $5.7 billion. Need I go on? IBM does not impress me!

Gorden Gibson
San Jose, CA

Permission granted. The editorial by Mr. Charters well expressed our own views. Judging from the response, he said what many TRS-80 owners feel. To our knowledge, Mr. Charters is not a TRS-80 employee or in any way connected with the Tandy Corporation. —Ed.

Notes, etc.

Cameron C. Brown, Editor

Puzzler

Your response to this has been tremendous. We are pleased that you find the problems enjoyable. It is getting harder and harder to pick the winner, many excellent submissions have been sent in. Please don't bother sending diskettes or tapes, a listing of your code is sufficient. We are trying to publish the winning code two months after the issue in which the problem appears, so if you are after the $10 and free tour of *80-U.S.* be sure to send in your solution as soon as possible.

This month our problem is derived from one that cropped up when using an early version of VisiCalc. On the early release there is no IF command. So, how can you do the following without an IF statement? When X is positive, add 20 to Y but when X is negative, subtract 30 from Y.

The first correct solution to our December Puzzler regarding the etymology of the word Debug was submitted by James A. Freeman of New Hartford, CT. He correctly identified Capt. Grace Hopper as the creator of the word. She was working in 1945 at the Computation Laboratory as a research fellow on the Harvard faculty. Machines in those days consisted of numerous tubes and mechanical relays. One machine quit working and inside it was "a moth that had been beaten to death." The bug was duly noted in the log-book and the term has been in use ever since. The responses were interesting—we were told that it was a fly, a moth, a beetle-like insect and a cockroach. Capt. Hopper, one of the world's first programmers, did much of the early work on the ENIAC and UNIVAC systems. She published the first paper ever on compilers. Her contributions to computing are well worth reading about.

Be sure to send your answer to Puzzler, c/o *80-U.S. Journal*, 3838 South Warner, Tacoma, WA 98409.

Corrections

In the November 1982 issue we incorrectly listed some of the features of the terminal program, Modem 80. The package can send and receive binary files (through its HEX/CMD program), and it does have a prompted send mode as well as a single-line send mode. Also note that the Radio Shack communication package is available on cassette not just on diskette.

Our Listings

We are now typesetting the listings so that they are crisper and easier to read. Since they are sent directly from the computer to the typesetter, code will not be altered in any way from the working program. But, the typesetter will do some minor changes in spacing. Since typesetters justify spacing and typewriters do not, be careful when entering PRINT USING commands or setting strings to be a specific number of characters. We will note the spacing when it is not obvious from the context in which it is used. Also, the asterisk (*) is supershifted on a typesetter, whereas in BASIC it is meant to be on the program line. Don't worry, our asterisk still means multiply. For very long program lines, there may be a linefeed directly after the line number. That happens because the typesetter has too much to try to print and the only break it can find is after the linenumber. Don't bother entering the linefeed (with a downarrow ↓) but even if you do, it won't affect the running of the program. This problem will usually occur only on long DATA lines.

In This Issue

Our theme is non-BASIC languages or, "Is there life after BASIC?" We have reports and reviews on Pascal, PILOT, Forth, COBOL, and comparisons between them and BASIC. Mark Renne introduces us to Pascal. Terry Dettmann discusses COBOL and gives an excellent overview of where all these languages came from. Anthony Scarpelli shows us how Forth works and Darrell Wright reviews it on the Color Computer.

We are very pleased to announce the beginning of two new columns.

One is entitled "In the chips" by Spencer Hall. Many of you have asked that we help you get started in machine language. If you have ever had any fear or trepidation about learning it, your worries are over. Spencer is under orders to make it so understandable that even the editors can follow it. By the time he finishes with the second installment you will be able to enter and run any of those wonderful machine language utilities you have skipped over.

To take full advantage of the series, be sure to obtain a copy of the DEBUG program from Radio Shack. It is catalog number 26-2000 and sells for $19.95. If you have disks, it is already on TRSDOS, so you don't have to spend a dime.

Our second new column is due to the fine response to our November issue. Such a strong interest was shown in communications, we have arranged for Donald L. Stoner to help keep us informed. His series is called Com 80 and will appear frequently. The technology and the needs in this area are changing rapidly and Don is just the man to keep us on top of it all.

Color computerists should be sure to look at Basicmon, a utility that lets you see what is going on inside your computer. Gary Ludeke gets you off the ground in his space shuttle simulation. Model II owners have their second part on SVC calls, This time Terry Dettmann shows how to have two displays available at any time. For those of you looking for fast Model II graphics it should be quite useful.

No matter which model you own, don't pass up our regular columns on Basically BASIC, BASIC bits, Tandy topics, or the humorous Captain 80. ■

A history of Languages

T. R. Dettmann, Associate editor

Programming languages have become the modern Tower of Babel. As of 1973 there had been over 200 languages produced with no end in sight. I'm not sure anyone even has a count today.

If we add in all the dialects and flavors of the various languages, we have an amazing list. Among the best known and most frequently used are ones like FORTRAN, COBOL, BASIC, and PASCAL. Almost everyone who's played with computers has heard of those.

There are many special languages that are widely known yet may not be familiar to the average person: LISP, FORTH, PL/I, and APL. There are others that are known even less, CLIP, COGO, MAD, COMIT, and many others. New languages are also coming along such as ADA, designed for the Department of Defense. It's amazing to think that all of this has happened since the early 1950's, yet it has.

In the early days of programming, programs were written directly in machine code. There was a number that was recognized by the computer as an instruction to add. That number was actually used by the programmer. What's worse, it started out in binary.Can you imagine writing the BASIC Interpreter for a TRS-80 wholly in binary numbers? Can you imagine the chance of there being a single number incorrect? Can you IMAGINE finding that error?

The earliest 'languages' for computers didn't reduce the number of instructions that had to be written, but it made them easier to work with by assigning mnemonics for each numbered instruction and letting the computer translate those mnemonics into the numbers needed by the computer as instructions. This is an assembler.

Assemblers and assembly languages are still around today because for some things there simply is no substitute. But assembly language programming is time consuming and very computer oriented. For some people that's great, the computer is their toy. But for most people who want to use a computer as a tool, assembly language forces their attention away from what they want to do.

To make the computer more usable, even for computer freaks, languages that deal with the computer on a higher level were invented. The essence of the higher level language is to let some other program worry about the computer details, let the programmer write his program in a form close to the problem he's solving.

APT (Automatically Programmed Tools) in 1956 was one of the very first languages for a specialized area, specifically to help program punched tapes for numerically controlled machine tools.

FORTRAN (FORmula TRANslator) followed in the same year and became the very first widely used language.FORTRAN was designed for and has continued to be used by scientists and engineers. It made it possible for such people to concentrate on their science or engineering and still make good use of a computer.

In the same year, FLOWMATIC was introduced for business data processing. It had a strong emphasis on "English-like" form so that a program read very much like an English description of the procedure.

COMIT, in 1957, was introduced to work with strings or characters and pattern matching. It introduced many powerful features which have come down to us in other languages whenever strings are processed.

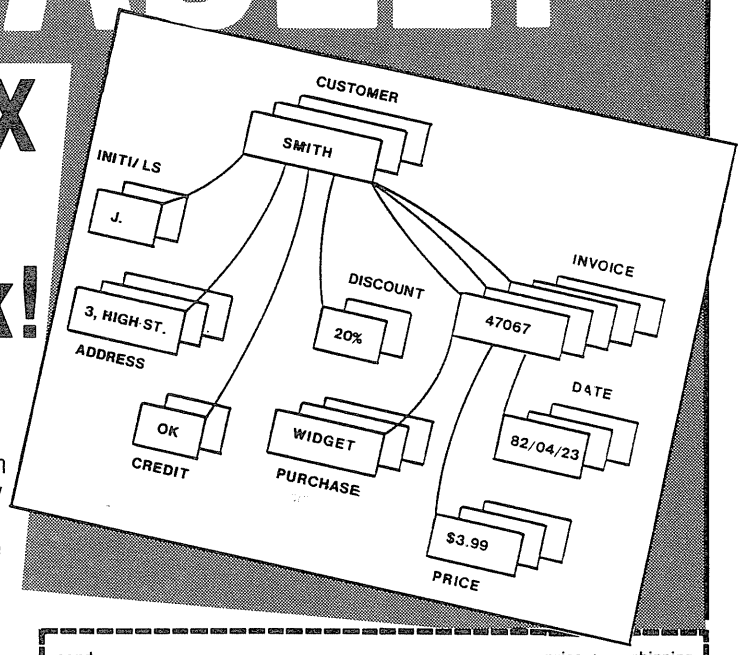
In 1958, IPL-V (Information Processing Language V) was introduced for processing lists. In a general way, you can look at a sentence in English as simply a list of words put together according to certain rules. List processing languages have often been used for working on problems of this sort.

DYANA (DYnamic ANAlyzer) was developed in 1958 by General Motors as an extension of FORTRAN to help analyze vibrational and other dynamic problems associated with automotive design.

COBOL (COmmon Business-Oriented Language) hit the scene in 1960. Most people agree that at least when it comes to business data processing, no language is used

YOUR DATA JUST GOT MORE VALUABLE.

Because ALGORIX Just Got The Electric NoteBook!



Announcing a better kind of database manager for TRS-80 Mod I/III from Southern Software (British developers of ACCEL3/4 compiler). The Electric NoteBook is a *TRUE* relational database manager; your data is held in ordered sets with many internal links between sets. Every set is a pre-sorted key field that can be related to any (or all) other sets, ready for instant selection on any complex of attributes. You may create and link new sets whenever you wish (*NO* reblocking or reprogramming). Inbuilt reports completely document current structure. So your database can grow wider as it grows longer.

Includes: Integrated data dictionary, recursive menu manager, complete BASIC interface, Scripsit interface, Visicalc interface, Utility programs, sample database, 125 page manual with tutorial and BASIC interface documentation. Single database can span 4 disk drives or hard disk.
Requirements: TRS-80 Mod I/III, 48K, at least 1 disk, any DOS.

$140

ALGORIX
(415) 681-9371
Box 11721 San Francisco CA 94101

send	price +	shipping
☐ **Electric NoteBook — Relational database manager**	**$140**	**$3.00**
☐ ACCEL 3/4 Compiler for TRS-80 BASIC	$99.95	$2.00
☐ EDIT Full-Screen Editor for BASIC	$40.00	$1.50
☐ TSAVE System Tape Writer	$ 9.95	$ .75
☐ STRETCH SuperStep Z80 Simulator	$39.95	$2.00
☐ EMU02 6502 Simulator	$29.95	$1.50
☐ BAL80A Basic Assembler Language on Mod I	$79.00	$3.00
☐ Send more information		

$ enclosed (CA add 6%) ..... Model I ..... III .....
VISA/MC # ..... Expiration Date .....
Name .....
Address .....
City ..... State ..... Zip .....

Signature .....

TRS-80 SCRIPSIT[tm] Radio Shack. VISICALC[tm] Visicorp. ELECTRIC NOTEBOOK, ACCEL3/4, EDIT, TSAVE © Southern Software. STRETCH, EMU02 © Algorix Software; BAL80A © Balcode Software.

more than COBOL. Look in the want ads in your paper. If you find ads for programmers, it's likely knowledge of COBOL will be a requirement. COBOL grew out of FLOWMATIC, it really tried hard to make English a programming language and to make it independent of the computer you were using. To a great extent it succeeded.

ALGOL 60 (ALGOrithmic Language), introduced in 1960, was developed to specify numerical procedures (known as algorithms). It was widely used in Europe but never really made it in the United States. But it was the basis for much significant research and became a standard among computer scientists for expressing ideas about programming.

LISP (LISt Processing) was also introduced in 1960 to make list processing much more direct and easy. While many people think it is a very obscure language (some say that LISP really stands for Lots of Irritating Silly Parentheses), it is an extremely powerful one. Devotees claim it can do anything. They may be proving their point since most work in Artificial Intelligence is done in LISP.

JOVIAL (Jules Own Version of IAL) is another 1960 language which concentrated on scientific work. It included greatly expanded facilities for manipulation and storage of information and logical decision making.

COGO (COordinate GeOmetry) was developed at MIT around 1960 to work out surveying problems for civil engineers. It is now widely available and still in use.

GPSS (General Purpose Systems Simulator) was introduced in 1961 to eliminate the need for complicated programming in building simulations. GPSS allows a simulation to be designed and run very quickly without specialized computer knowledge.

JOSS (JOHNNIAC Open-Shop System) was the first interactive programming language. It was introduced in 1964 to meet the need to make the computer more responsive to humans.

FORMAC (FORmula MAnipulation Compiler) in 1964 was the first widely used language for work on algebraic problems. Mathematical symbol manipulation was its forte.

PL/I (Programming Language One) was introduced by IBM in 1965 as a replacement for COBOL and FORTRAN with some of the flavor of ALGOL. The intention was to provide everything everyone wanted in a single programming language. The general feeling among programmers was that there was so much to it that it became nearly impossible to learn. Even worse, many of the original compilers were slow and inefficient and that kept many people away from using it. While its power hasn't been denied, it never replaced either FORTRAN or COBOL. PL/I is used by many computer installations and it is seeing some new interest due to the introduction of a subset by Digital Research for CP/M.

It was felt that there was no language available that was simple enough to teach programming without causing problems that obscured the techniques a student was trying to learn.

APL/360 (A Programming Language) became important in 1967 as an extremely compact and powerful mathematical programming language. APL programmers tend to be messianic about its ability to do anything, but many people are put off by the extremely compact way programs are written. In fact, it has sometimes been almost a contest between APL programmers to find a tighter more compact (many would way less comprehensible) way of writing an APL program.

In 1967, BASIC (Beginner's All-purpose Symbolic Instruction Code) was developed at Dartmouth College. It was felt that there was no language available that was simple enough to teach programming without causing problems that obscured the techniques a student was trying to learn. Because it is an easy, interactive way of working with computers, it was chosen for microcomputer implementation and from that point has grown into a major language.

In 1969, Charles Moore introduced Forth. It grew out of his need to provide powerful programming capabilities on very limited systems. He built a very tight, stack oriented language (like an HP calculator) that can fit in a tiny space, run like greased lightning, and solve every problem including those having to do with the kitchen sink (at least that's what Forth programmers keep saying).

In 1971, Niklaus Wirth brought out Pascal in response to his dissatisfaction with other languages. Pascal was designed to teach (actually force) good programming technique. It provided all the best in technique. It also was a demonstration of what can be done when a uniform, guiding thought controls the development of such a system. Pascal has become very popular because of its power, but it is lacking in some areas such as input/output. Many versions have been introduced to overcome these problems but usually they lack the tight consistency of the basic Pascal system.

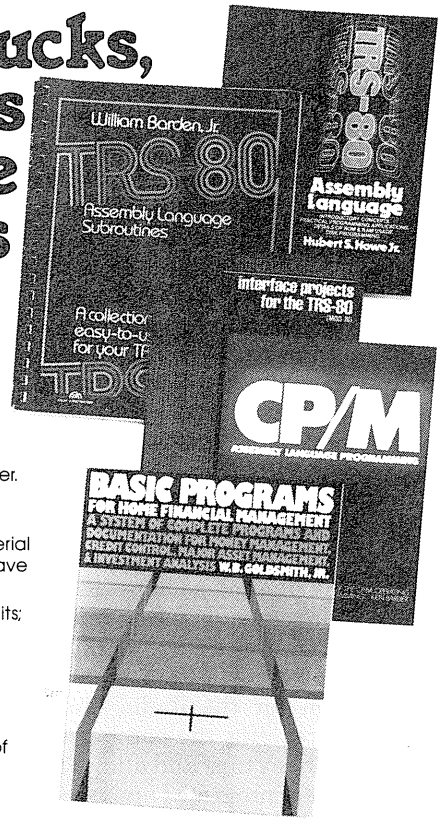
It's a foregone conclusion that I've missed someone's favorite language here. But with hundreds to go through, it would be impossible to even mention them all, even if I knew what they all were.

Even today, new languages are coming into being. They are part of what may be another eternal quest, the perfect programming language. The Department of Defense wanted a single programming language for all its needs, so they rejected all of the existing ones and settled on ADA which was designed specifically for them. It may or may not be beautiful, but you can be sure that it will be important. The DOD will *require* its use!

We may never find the perfect language since there are so many different needs that have to be met. A language that fills them all might be impossible. But people never lose hope, so you can bet that there will be new languages in the future. ■

LOVE AT FIRST BYTE

For as little as 10 bucks, you'll learn fast, easy ways to make your software work so hard it sweats



TRS-80 Assembly Language by Hubert S. Howe, Jr. The first-time user can easily understand this, but even the experienced TRS-80 user can delight in its myriad practical programs and subroutines. Comprehensive tables, charts, and appendices reinforce the detailed info on ROM, RAM, disk operating systems and much more. $9.95, paper.

TRS-80 Assembly Language Subroutines by William Barden, Jr. This handbook courses the speed and compactness of assembly language programming and offers you 65 fully debugged, ready-to-run subroutines that, among others, speed up graphics by a factor of 300, enable you to perform high-speed sorts, or "dump" the video screen to cassette or to read a disk sector. $18.95, paper.

Interface Projects for the TRS-80 (MOD III) by Richard C. Hallgren. You'll find many fully-tested practice hardware projects–including explanations on the necessary interfacing software–including a review of data transfer formats, analog-to-digital and digital-to-analog conversions with the Mod III, serial applications, biofeedback projects, controlling a video playback device, and more. Readers should have a good grasp of TRS-80 basic to wring every last bit of info from these pages. $12.95, paper.

CP / M Assembly Language Programming by Ken Barbier. Microcomputers rely on integrated circuits; and now you can rely on the "integrated learning-by-doing" approach in this book. It details the hardware, its operating system, and assembly language programming. Your hands need leave the keyboard only to turn the page. $12.95, paper.

BASIC Programs for Home Financial Management by W. B. Goldsmith, Jr. If you bought your computer with the hopes of debugging your personal financial picture, you may have found it takes longer to program it than to dig through a shoebox full of records. Here's the simple and fast system of 33 documented programs, including descriptions and sample runs, for money management, credit control, major asset management, and investment analysis. $12.95, paper.

WITH PRENTICE-HALL BOOKWARE™

SEND NOW

Simply complete and return the coupon for fast service. If you wish to order more than two titles, please write to the same address listing all of the books and their authors. ***DEALER INQUIRIES INVITED.***

Spectrum Books

RETURN TO: **Prentice-Hall, Inc. / General Publishing Division / Englewood Cliffs, NJ 07632** ***Attn:*** **Addison Tread**

Please send the following:

AUTHOR TITLE

AUTHOR TITLE

☐ I enclose my check or money order for the price of each book plus $1.00 per title for postage & handling.

Amount enclosed $_____. Bill my ☐ MasterCard or ☐ Visa

#_____ exp. date _____

NAME

ADDRESS APT #

CITY STATE ZIP 80U

TRS-80 languages

For all models

Richard A. Yehle, Sacramento, CA
Cameron C. Brown, Tacoma, WA

There are numerous languages available for your computer. Most of the ones that Terry mentioned are not in this table, but we always thought he had unique tastes. The information has been gathered from reviews, advertisements, and articles in numerous sources. Wherever possible, the configuration and price have been confirmed with the manufacturer. Do not assume that the list is complete. We provide it to help demonstrate the variety that is available to you. Readers are advised to write for the latest specifications and prices. In some cases, shipping and handling charges or state taxes may apply. Usually documentation is available separately.

The Languages

Pascal

The Tiny Pascal sold by Radio Shack is in cassette form. It is a fixed-point, non-array version. Barker Software offers a disk modification for the Radio Shack version. It requires the purchase of the Radio Shack tape and allows for disk storage of programs and p-code as well as disk I/O, line printer commands and alternate tables or compiler p-code. The Alcor Pascal is a complete Jensen and Wirth standard Pascal with a one-pass compiler. It and Pascal-80 from New Classics Software both require 48K systems. See this issue for reviews of Alcor Pascal and Pascal-80. Computerware's tape version includes a supervisor and editor; the disk version requires an ASCII text editor. The Dyanasoft Pascal from Frank Hogg Laboratory requires the Flex operating system for disk Color Computers. For an added $30.00 Frank Hogg Laboratory will include the source code and a run-time version is also available for $89.95.

Compiler BASIC

A compiler will translate your BASIC programs into machine code and thereby execute at greatly enhanced speeds. Radio Shack's compilers are all disk oriented and are not compatible with the interpreter built into your computer. They include single-key ISAM to help organize and retrieve data, cross-reference, interactive DEBUG, easy calls to assembly language or other object code programs. For the Models I/III, two disks, 48K is required and the Model II version is for 64K systems. The Accel 3 from Algorix is an enhanced version of Accel 2 and was reviewed in Oct. 82 *80-U.S. Journal*. Aardvark-80 offers a Color Computer compiler in tape or disk version. It was reviewed in Nov. 82 *Creative Computing* along with many other alternate languages for the Color Computer. Level III BASIC is an enhanced BASIC that offers greater speed and commands not found in Disk BASIC or regular Level II BASIC.

COBOL

(See this issue for a report on COBOL on the Model II.)

C Compiler

Word's Worth compiler requires the Flex operating system for the Color Computer and the version from Dugger's Growing Systems requires an assembler package for its operation. See this issue for a report on a beta-test version of the C programming language on the Model 16.

Forth

This language is available from numerous sources. Frank Hogg Laboratory offers two versions for the disk Color Computer, one of which requires the Flex operating system. Talbot Microsystems offers a regular and enhanced version for Flex users. They are also providing a ROM Pak version for 4K Color Computer owners. Miller Microcomputer Services implementation for the Models I/III was reviewed in July 82 *80-U.S. Journal*. They offer probably the most complete line of Forth utilities, programs, and references for the Models I/III. Their programs have been in use since 1979 and are very popular. Colorforth from Armadillo Int'l. is reviewed in this issue.

FORTRAN

Radio Shack FORTRAN is based upon the ANSI '66 standard. The Model I compiler requires 32K, two disks, the Model III needs 48K, two disks, and the Model II needs 64K, one disk. FORTRAN-80 was reviewed in Mar/Apr 79 *80-U.S. Journal*.

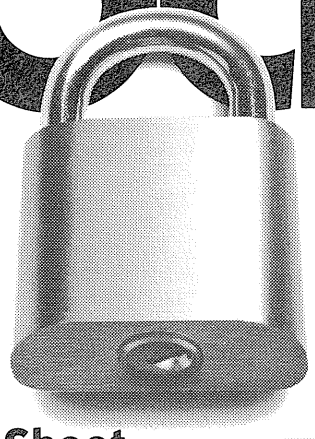
Lisp

The first two packages were reviewed in Dec. 82 *80-Micro*. Supersoft's offering is also available on tape for $75.00. The version from Far West Systems conforms to proposed University of Utah standards and follows the language as described by Winston and Horn. The mu-MATH package from Microsoft is a Lisp-like language that uses a subset of the Lisp instruction set. It was discussed in the Mar/Apr 81 *80-U.S. Journal* issue.

PILOT

The package from Barker

DON'T GET LOCKED IN



Pascal
Basic
Cobol
Forth
Pilot
Fortran
Spread Sheet
Inventory
Logo
"C"

Data Base Management
Word Processor
Communication Utility
Accounts Receivable
Accounts Payable
Engineering Utility
General Ledger
Mailing List
Macro Assembler
Education

TRSDOS*

CP/M**

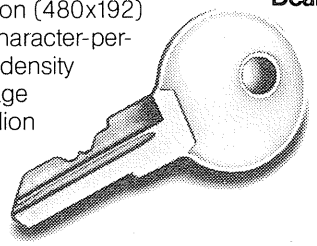
Open your doors to a world of SOFTWARE with LNW computers. You'll get **MORE PERFORMANCE**[1] than with the IBM PC[2], the Apple II[3], TRS80 MODEL II or TRS80 MODEL III[4] along with software support of TRSDOS or CP/M, the TWO MOST WIDELY USED OPERATING SYSTEMS. This means you, the user, can select from the largest base of business or personal software.

Standard Features: A serial RS232 communication port, parallel printer port, Hi-Resolution (480x192) B/W and COLOR graphics, an 80 character-per-line screen display along with Quad-density interface for 5" or 8" floppy disk storage offering immediate access to 3.5 million characters, or optional Hard disk interface to 5 or 10 million characters.

Standard Software: LNWBASIC and DOS PLUS operating system packages, commanding all the above features, are included.

The LNW computer will be the key to your success with the starting price at **$1695.00,** along with a full 6 month warranty.

Dealers: You too can open the door to a successful product. Call for our special dealer programs: (714) 544-5745.



LNW Computers
2620 Walnut Avenue
Tustin, California 92680
(714) 544-5744

*TRSDOS is a trademark of Tandy Corp.
**CP/M is a trademark of Digital Research Corp
1. Performance is based on bench mark test in the JAN 1982 issue of BYTE magazine, pg. 54, with LNW80 II as the comparison.
2. IBM PC is a trademark of IBM CORP.
3. APPLE II is a trademark of APPLE COMPUTERS
4. TRS80 is a trademark of Tandy Corp.

12

International orders please inquire for pricing/shipping cost.

— COMPUTER SHACK —

★ BEST BUY ON DISKS ★

BASF has taken one of the best disks and made it even better. The quality is so good that BASF guarantees each disk for not one year, not five years, but for LIFE. If anything ever goes wrong you can return it to us or to BASF and it will be replaced at no charge. Can you believe MAJOR brand name, LIFETIME guarantee, FAST shipment and at these prices.

35/40 Track Single Density 1 side . . . $19.95
35/40 Track Double Density 1 side . . . $21.95
35/40 Track Double Density Flippy . . . $29.95
The following come in special Computer Shack packaging.
35/40 Track Double Density 2 sides. . . . $29.95
77/80 Track Double Density 1 side . . . $25.95
77/80 Track Double Density 2 sides. . . . $29.95

All of the above come in Boxes of 10 with BASF labels, hub rings and Tyvek sleeves. All disks have one write protect notch and one read hole, except the Flippy's which have two.

Disk Savers (vinyl sleeves) 20 for **$6.00**
Colored Disk box (holds 10 disks) . . . $2.95@
5 Colored Disk boxes (red, blue, green, yellow, and brown asst. colors) **$14.50**

FAMILY TREE

Excellent family genealogy program works on both the Model I or Model III. Along with the normal documentation we send a manual with over 200 pages of instruction on how to research your family tree. You can get three different printouts, a family tree or a family report of each person, or a birthday printout. Sort on Birthdays. The Nebraska State Genealogical Society tested our program and said "Your program will help form a valuable data source that may be accessed an used in different ways. The printouts are excellent."
Disk or Tape **$29.00**

JOVIAN

Newest game by Dunlevy and Frayer. This is Cyborg with a space theme! 9 screens of action simultaneously. Featuring WRAP AROUND ACTION™. Most exciting spaceship type game out for the TRS-80.
Only . . . . . **(19.95/24.95)**

ARACHNID PLUS

Three exciting first class **arcade** games for the TRS-80. A special value pack including a real time arcade game, an exciting car race game, and a strategy game. All excellent games for only . . . . . **(19.95/24.95)**

SUPER DIRECTORY

The Best Catalog program on the market. One program reads Multidos, NewDos 80, Dos Plus, LDos and TRS-DOS. Automatic density recognition! Automatic track count. Reads almost anything. Superfast sort routines! (Sorts on 5 different fields). Super fast find routines (five different find routines including a string search)
ONLY. **$39.95**

NEW! MONTHLY BILL SYSTEM

This is a new program for the businessman who wants to send a monthly or quarterly bill to the same people every billing period. (Landlords, garbage man, etc.) The program will send out a bill to each person (you can set the frequency, monthly, quarterly, etc.). It then has a very easy way of inputing your paid customers. This will help you to keep track of who is past due. Gives you printouts of your financial condition at any time.
Disk only. **$149.95**
We also have a Special program for Auto, and Home Insurance agents, Call for complete details.

Small Business Programs

CHECKING ACCOUNT Mod 1 or III, 48k disk $39
Excellent check writing program for small businessman or for personal use. Menu operated for easy use. Has Screen editor, 99 catagories or expenses. Sorts on payee, Check number, or date. It can print your checks on the printer.

BILLING SYSTEM Mod. I or III, 48k disk $39.00
A billing system written by a businessman for his own use. It is simple, fast and easy to use. Has a screen editor for quick error correction. Prints out invoices, then will send a monthly bill to each customer. It will add interest or carrying charges. This is not a complicated accounts receivable it is a simple but effective way to keep track of who owe's you money for the company that doesn't do a lot of credit business.

E BASIC

There are five major **Dos's** on the market all have there good points. Some folks like one, some like another, but every single reviewer has said that MULTIDOS had the BEST BASIC. Some would stop here, but not Vernon Hestor. He now brings us EBASIC, a new innovative state of the art basic. It makes Graphics and Sound EASY. This will give the basic programer more power than ever! Here are some of the new basic commands: print(a,b), input (a,b), line input (a,b), print$, call, sort, lapeling, array read, array let, cound, shape, circles, cubes . . . . All of this and you still have over 38K for your basic programs. EBASIC is only $29.95 for MultiDos owners.

MULTIDOS

MULTIDOS got bigger and better. New EASY ZAP, New TAPE/CMD, and the New DISK DRIVE TIMER. All of this and its still the cheapest full Dos going. **$99.95**

Z DOS

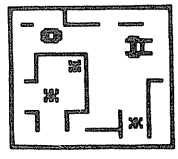
Z Dos the Dos for the person who does not want to spend a hundred dollars or more for an improvement over TRS-DOS. For only $39.95 you will double your fun. For only $39.95 you can get what reviewers in three different magazines said was the BEST BASIC for the TRS-80.
Buy MultiDos at the list price of $99.95 and you can take 20% off of any or all of the following.
E Basic $29.95 . . . . . Super Directory $39.95
Aerocomp Doubler $149.00

SUPER DOS

If you are using TRS-DOS, on the Model III, you can now take out some of its biggest drawbacks. SUPERDOS will make a few automatic zaps to your copy of TRS-DOS and it will boot up instantly! It will give you error messages in English. It has a short directory, and on the long directory it will pause to give you a chance to read the screen.
Only. **$19.95**

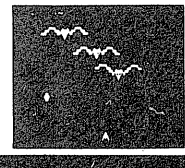
CYBORG

This is the biggest blockbuster to come along since the first BIG FIVE games hit the market. A real TRS-80 classic. The first game for the TRS-80 that features **WRAP AROUND GRAPHICS**™. This exciting game occurs in an interstellar space station. This game features graphics that are super fast and move smoothly in all FOUR directions. Rated #1 three months in a row.
ONLY . . . . . **(19.95/24.95)**



DEMON SEED

The greatest graphics effects yet in a real time arcade game for the TRS-80. Birds of various sizes drop down on you. A gian space ship must be destroyed. A great game based on the arcade game Phoenix™.
Only . . . . . **(19.95/24.95)**



COMPUTER SHACK

1691 Eason ● Pontiac, Michigan 48054
Info: **(313) 673-8700** ● Orders: **CALL TOLL FREE (800) 392-8881**
Master charge and VISA OK. Please add $3.00 for shipping in the U.S.A. - $5.00 for Canada or Mexico - Proper postage outside of U.S. - Canada - Mexico.
Dealers: We are distributors for all items in this ad. Write for our catalog and price list.

Software is advertised as less friendly than Radio Shack's but is cheaper. It comes with a DEBUG package, one command invocation, and file chaining. Radio Shack's PILOT has been renamed to MicroPILOT and is reviewed in this issue. Both offerings require disk systems.

PL/I

The package from Digital Research will run under any CP/M or MP/M operating system.

Sources and Addresses

Aardvark-80, 2352 So. Commerce, Walled Lake, MI 48088.

Alcor Systems, 800 W. Garland Ave., #100, Garland, TX 75040.

Algorix, Allen Gelder Software, P.O. Box 11721, San Francisco, CA 94101 (415) 387-3131.

Armadillo International Software, P.O. Box 7661, Austin, TX 78712 (512) 459-7325.

Barker Software, P.O. Box 5313, Athens, GA 30604.

Computerware, P.O. Box 668, Encinitas, CA 92024 (619) 436-3512.

Digital Research, P.O. Box 579, 160 Central, Pacific Grove, CA 93950 (408) 649-5500.

Far West Systems & Software, P.O. Box 3301, Eugene, OR 97403 (503) 485-5155.

Frank Hogg Laboratory, 130 Midtown Plaza, Syracuse, NY 13210 (314) 474-7856.

Microsoft, 10700 Northrup Way, Bellevue, WA 98004 (206) 828-8080.

Miller Microcomputer Services, 61 Lake Shore Rd., Natick, MA 07160 (617) 653-6136.

New Classics Software, 239 Fox Hill Road, Denville, NJ 07834 (201) 625-8838.

Omegasoft Industrial Products Group, P.O. Box 70265, Sunnyvale, CA 94086.

PCD Systems, P.O. Box 143, Penn Yan, NY 14527 (315) 536-7428.

Radio Shack, any local store or Computer Center.

Supersoft, P.O. Box 1628, Champaign, IL 61820 (217) 359-2122.

STSC, Inc., 2115 East Jefferson St., Rockville, MD 20852 (301) 984-5000.

Talbot Microsystems, 1927 Curtis Ave., Redondo Beach, CA 90278. ■

Language	Source	Price	Model
Tiny Pascal	Radio Shack 26-2020	19.95	III
Tiny Pascal	Radio Shack 26-2009	19.95	I
Tiny Pascal	Barker Software	19.95	I/III D
Alcor Pascal	Alcor Systems	199.00	I/III
Pascal 80	New Classics Software	99.00	I/III
Color Pascal	Computerware	49.95	32K CC T
Color Pascal	Computerware	59.95	32K CC D
Dynasoft Pascal 1.3	Frank Hogg Laboratory	59.95	CC w/Flex
Omegasoft Pascal	Omegasoft Indust. Prod.	425.00	CC w/Flex
TSC Pascal	Frank Hogg Laboratory	200.00	CC w/Flex
Compiler BASIC	Radio Shack 26-2204	149.00	I/III D
Compiler BASIC	Radio Shack 26-4705	199.00	II
Compiler BASIC	Microsoft	195.00	I
Compiler BASIC	Accel 3 - Algorix	99.95	I/III
BASIC Compiler	PCD Systems	225.00	II/16
Tiny Compiler BASIC	Aardvark-80	24.95	16K CC
Level III BASIC	Microsoft	49.95	I D
COBOL	Radio Shack 26-2203	199.00	I/III
COBOL	Radio Shack 26-4703	299.00	II
COBOL Generator	Radio Shack 26-4707	995.00	II
C Compiler	Word's Worth	52.50	CC w/Flex
Small C Compiler	Dugger's Growing Syst.	59.95	16K CC D
X-Forth	Frank Hogg Laboratory	149.95	CC w/Flex
T-Forth	Talbot Microsystems	100.00	CC w/Flex
T-Forth+	Talbot Microsystems	250.00	CC w/Flex
MMSForth	Miller Micro Systems	129.95	I/III D
COLORFORTH	Armadillo Int'l. Software	49.95	16K CC
CCForth	Frank Hogg Laboratory	99.95	CC D
Colorforth	Talbot Microsystems	110.00	4K CC ROM
FORTRAN	Radio Shack 26-2200	99.95	I
FORTRAN	Radio Shack 26-2201	99.95	III
FORTRAN	Radio Shack 26-4701	299.00	II
FORTRAN-80	Microsoft	100.00	I
FORTRAN Compiler	PCD Systems	350.00	II
Lisp	PCD Systems	100.00	II/16
Lisp	Supersoft	100.00	I/III D
UO-Lisp	Far West Systems	199.00	I/III
mu-Math	Microsoft	250.00	I/III D
PILOT	Barker Software	29.95	I/III
MicroPILOT	Radio Shack 26-2205	79.95	I/III
PL/I-80	Digital Research	500.00	II
UCSD p-System	PCD Systems	650.00	III
UCSD p-System	PCD Systems	650.00	II
UCSD p-System	PCD Systems	850.00	16
APL*Plus/80	STSC, Inc.	295.00	III

Oil tank you not to do that

Using defined functions to solve a problem

Models I/III, PMC-80, LNW80

Graham Allan, Maidstone, Kent, England

Use those old DEFFN statements much? Nor did I until someone gave me a tough nut to crack and I found that they made excellent nutcrackers. Read on . . .

Sitting at the keyboard a couple of Thursdays ago, I heard the sound of a car pulling up outside my office and in breezed Nick and Peter, the Braybrooke brothers. Nick and Peter both have strong TRS-80 connections. Nick is a Tandy employee and Peter is an ex-Tandy employee.

"Got a couple of problems," they said. Although they are not twins, they tend to merge into one person when you try to recall a conversation with them.

The first problem was of the everyday, boring variety, about a client who wanted a program written to keep track of four 'Indoor Leisure' clubs, each with up to 2000 members and fifteen categories of membership. It brought to mind all the 'Video Hire' clubs, 'Squash' clubs, etc., etc., all with 2000 members and fifteen categories of membership. I brushed it aside with a yawn and asked about the second problem.

"Well, this one's for Mum and Dad," they said, "They want to measure the volume of used motor oil in cylindrical tanks, and how much spare capacity there is for more used motor oil."

At this my ears pricked up. I couldn't remember the last time I'd had a 'used motor oil' problem. "Used what?" I asked.

"Yes," they said, "you see, these tanks come in all sorts of different sizes. A typical one might be nine feet in diameter by thirty feet long." They paused while I tried to imagine a used oil tank the size of a bus.

"Mum and Dad want to be able to dip in a measuring stick, read off the depth of the oil to the nearest half inch, and know exactly how much they've got, and how much more they can put in."

"This sounds fairly easy," I thought, brushing aside the question of what Mum and Dad used all this motor oil for. "Surely that's just a matter of pi times r squared times the depth."

"It's not quite as easy as you're probably thinking," said the brothers, at one stroke reading my mind and shattering my delusions, "you see, the tank may be standing or it might be lying down."

No . . . it wasn't as easy as I'd thought. $Pi \bullet r^2$ doesn't exactly strain my mathematical resources, although it comes pretty near to it, but this was an entirely different can of worms.

"It would be nice," said the boys, "if Mum and Dad could enter the sizes of any particular tank, and pull off a list of 'volumes' and 'spare capacities' in half inch increments."

"Yes," I thought, "that would be nice."

I knew that Mum and Dad Braybrooke used a Model I with disks, and a PMC-80 in their business, and that they had a Daisywheel II printer. "Leave it with me," I said, "I'll see if I can come up with anything," meaning "when I get home I'll look in the kid's math books, who knows, I might find something."

Well, I didn't find anything. My eldest, Murray, is nine, and his books go up to rudimentary circle theory, but not as far as finding the area of a .... good God, I didn't even know what it was called. It's not a sector, that's a slice of cake. Is it a segment?

I sat down with a paper and pencil, and drew a picture. Obviously we start with a circle. Then we put in a horizontal line to show the level of oil. After gazing at

my picture for a while, I shaded in the oil to add realism. I wasn't getting anywhere. What I needed was a few more lines. I put in a dot for the center of the circle and marked the vertical and horizontal axes. This was looking better. Now a couple of lines to join the center to the points where the surface of the oil met the tank. Hmm . . .

At this point I decided to redraw my diagram on a clean piece of paper and the result was something like Figure 1.

"What we have there," I thought, "right there in the middle, is an isosceles triangle." Not only was it an isosceles triangle, it was also two right triangles, which I knew I could come to grips with. Who could ever forget Pythagoras?

If we could find the area of the isosceles triangle, and the area of the two 'cake slices' above it, take the sum away from the area of the semi-circle that they sit in, we would be left with the cross sectional area of the oil.

Now, if you refer to Figure 1, 'x' is the depth of oil. If 'x' is subtracted from the radius of the circle, we are left with 'a', the height of the right triangles. The hypotenuse has to be the radius, that's easy. All we have to do now is take the square of 'a' from the square of the hypotenuse, 'c', and we're left with the square of 'b', the base. The area of a triangle, I remembered, was half the base times the height, so 'b' times 'a' gives us the area of the whole isosceles triangle. So far so good.

But what about those 'cake slices'? The angle at SXZ must be the same as the angle at YZX because the lines QS and YZ are parallel. Trigonometry can tell us the angle at YZX, and once we know that angle we can find out the area of our 'cake slices', as a known proportion of the whole 'cake'.

Sine = Opposite over Hypotenuse. That's something else that has been travelling the circuits of my brain cells since school days. A quick look at the Level II manual helped to make things clearer. Appendix F gives a run-down of derived functions. What we wanted was one of these 'inverse sines' or 'arcsins' shown on line three. The fly in the ointment is that Level II deals in radians, not degrees, but let's cross that bridge when we come to it.

As far as the theory is concerned we can now consider the problem solved. All that's left is to write the program.

"This," I thought, "is an ideal situation for the use of DEFined FuNctions if ever I saw one." Not that the functions would be called a large number of times; simply for the fact that they could be worked out, defined, and then used as the need arose without cluttering up the code with a lot of formulae.

Functions

I don't know about you, but I tend, in general, to make too little use of defined functions. They only usually put in an appearance in my programs for 'rounding' purposes. (Our Income Tax routines in Britain require *five* different types of 'rounding'.)

The functions are found in lines 140 - 190 of the program listing. Here I'll try to give information about them in detail:

FNGA(VT): Easiest ones first. This just converts cubic inches to gallons. The cubic inches are passed to the function as the argument, and gallons are returned. In case anyone is wondering, in this listing the conversion factor is correct for U.S. gallons, not Imperial gallons.

FNA(DE,DI): This one is given the diameter of the tank and the depth of liquid and returns 'a' in Figure 1.

FNB(A,C): Give this function the radius and 'a', and it does the Pythagorus trick and returns 'b' in Figure 1.

FNAY(B,A): Figure 1's 'a' and 'b' are the arguments, and the area of the isosceles triangle is returned.

FNAT(C): This, given the radius, returns the complete area of the tank's cross section.

FNAS(A,C): This is the tricky one. It gives the sum of the areas of the two 'slices of cake'. In effect it finds the angle at SXZ, using the derived trig. function previously mentioned, doubles it, (two slices of cake), and works out the area as a proportion of the whole circle area. Rather than divide 360 by the number of degrees, as we would if the TRS-80 recognized degrees, twice pi is divided by the angle expressed in radians.

The bonus, as far as I was concerned, was that these

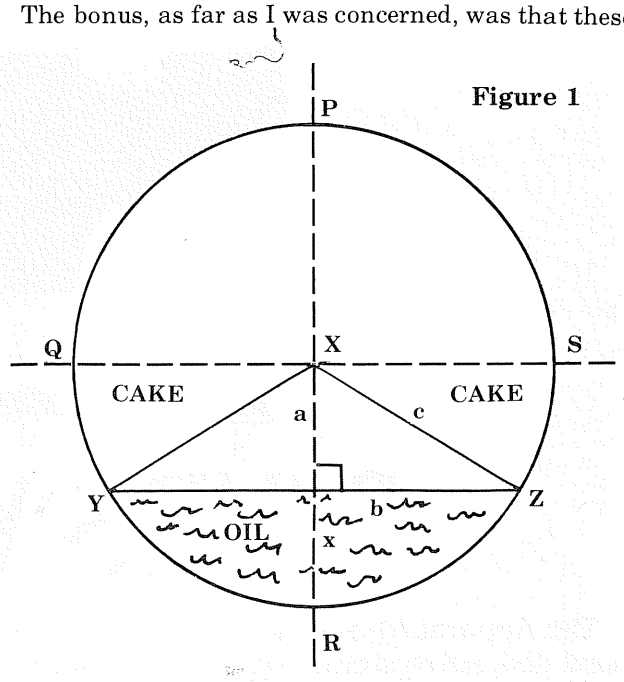


Figure 1

functions work equally well when the level of oil passes the center line and the value of 'a' becomes negative. I must admit that this is an aspect that I hadn't even considered until I started writing this article.

There are some other "why's he done that's" that you should know about. The dimensioning statement with that long list of variables is to allocate 'pecking order' for them. It makes sense that the most frequently used variables should be put at the front, and the less frequently used at the back. Depending on the type of program, this can increase the program speed by up to 30 to 50 percent. An excellent program has been developed by Glen Tesler at Prosoft, to work out this order for you. I recommend it to anyone writing BASIC programs for the TRS-80.

The subroutine at line 5000 scans location 14400 in the keyboard 'scratch-pad' and will halt printing and return to the menu if it finds the space-bar pressed.

The subroutine at line 6000 looks at the printer port,

OURS DOES ONE THING RADIO SHACK'S DOESN'T;

SAVES YOU $500.

The Apparat AT-Series hard disk subsystems for the TRS-80 Model III computer gives you the high performance, complete compatibility and value you demand.

The AT-Series hard disks use the most advanced 5-1/4 Winchester technology for reliable operation. Every AT hard disk comes with NEWDOS/80 Version 2.0 which has been updated for hard disk support. Just plug the AT-hard disk in and your ready for work. And because compatibility has been designed in, all of your programs can quickly and easily be transferred to the hard disk.

The Apparat AT-Series hard disks are available in storage configuration of 5 megabyte (AT-50), 10 Mb (AT-100) and 20 Mb (AT-200). Your system can grow as your needs do. The AT-50 and AT-100 systems are expandable up to 20 Mb. Priced at just $1,975 (AT-50), $2,550 (AT-100) and $3,075 (AT-200), the AT-Series offers you top value in a hard disk sub-system. Prices include all required components and software.

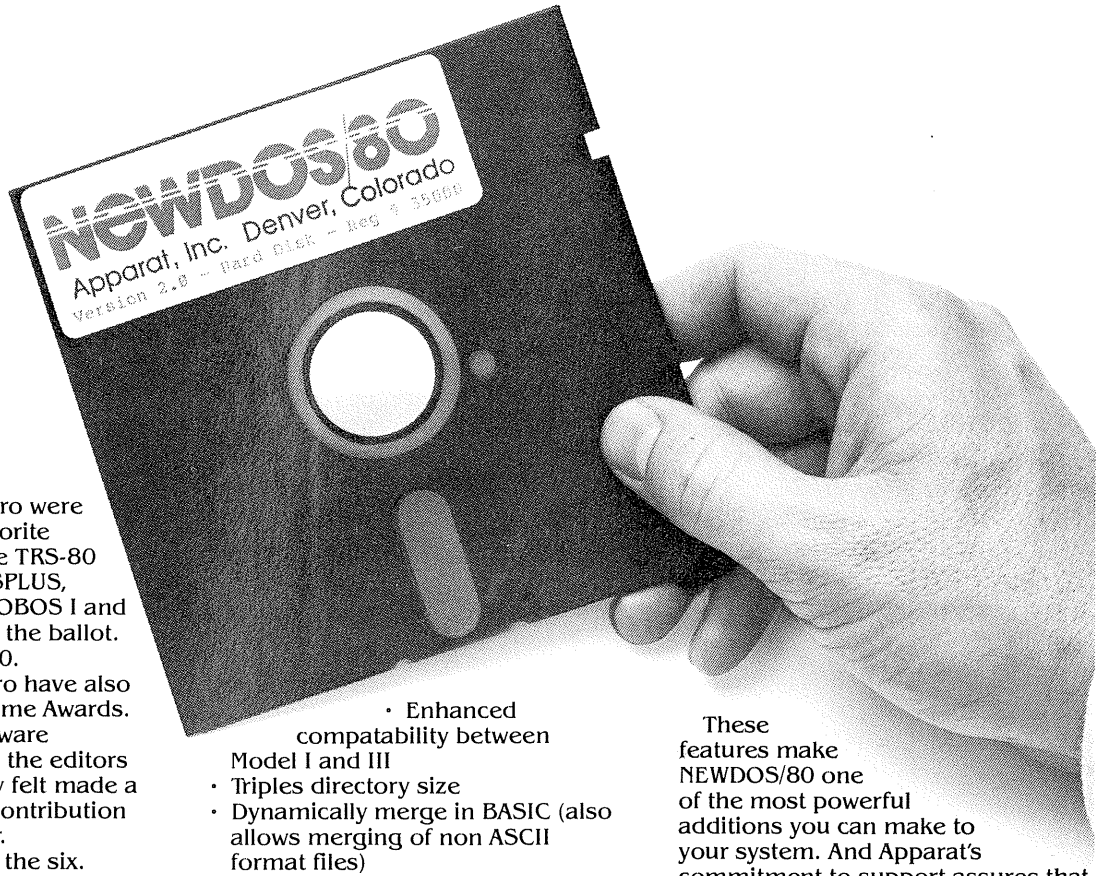
For more information contact your local dealer or write Apparat, Inc., 4401 S. Tamarac Parkway, Denver, CO 80237, 303/741-1778. Dealer inquiries invited.

Radio Shack and TRS-80 are registered trademarks of Tandy Corp.

14

Apparat, Inc.

IF YOU'RE GOING TO BE PICKY ABOUT AN OPERATING SYSTEM SEE WHICH WAS PICKED BEST.



The readers of 80 Micro were asked to select their favorite operating system for the TRS-80 Model I&III. LDOS, DOSPLUS, TRSDOS, MULTIDOS, WOBOS I and NEWDOS/80 were all on the ballot. They picked NEWDOS/80.

The editors of 80 Micro have also awarded their Hall of Fame Awards. From among every software package on the market, the editors picked only six that they felt made a lasting and significant contribution to the TRS-80 computer. NEWDOS/80 was one of the six.

Since we first introduced the NEWDOS operating system we've been stating its features, capabilities and advantages. Thank you 80 Micro readers and NEWDOS/80 users for supporting us.

Version 2.0 . . .
High Performance DOS

NEWDOS/80 Version 2.0 is our highest performance system yet. The versatility and sophistication of Version 2.0 includes features like:

- Double density support on the Model I
- Enhanced compatability between Model I and III
- Triples directory size
- Dynamically merge in BASIC (also allows merging of non ASCII format files)
- Selective variable clearing
- Can display BASIC listings page by page
- Automatic repeat function key
- Routing for peripheral handling
- Enhanced disassembler
- Command chaining
- Superzap to scan files
- Fast sort function in BASIC

Hard Disk Support Now Available

- Support for Apparat's and Radio Shack's Model III hard disk (optional-available upon request for additional $60)

These features make NEWDOS/80 one of the most powerful additions you can make to your system. And Apparat's commitment to support assures that you've purchased a superior product, both today and tomorrow. At just $149.00 it could be the best investment you will make for your TRS-80.

For more information see your local computer store or contact Apparat, Inc., 4401 S. Tamarac Parkway, Denver, CO 80237, 303/741-1778.

TRS-80 and TRSDOS are registered trademarks of Tandy Corp., LDOS – Logical Systems, DOSPLUS – Micro Systems Software, MULTIDOS – Cosmopolitan Electronics, WOBOS I – Western Operations, NEWDOS/80 – Apparat.

15

Apparat, Inc.

and avoids a possible hang-up if the printer is not ready.

The 'file saving' routine at line 65000 not only saves the program, it updates UD$, held in line 110. Type in line 65000 first, then start from the beginning. As soon as you reach line 200, you can run the program, select option 3, and the file will save itself together with the time and date. Be sure to set the time and date at power-up or this feature will have little value. Leave the line in until you are sure that you don't want to make any more changes, (printer codes etc.), then delete it and modify line 200 accordingly.

Mum and Dad Braybrooke are now happily slopping around in their used motor oil, knowing full well that every drop is present and accounted for. Perhaps you keep, or know someone who keeps, some kind of liquid in cylindrical tanks.

Take my word for it, it works just as well with pig swill or perfume. ■

```
10 REM: TANK CAPACITY 1982 by Graham Allan
100 CLEAR 500 :DEFINT F :DIM
A,C,DE,B,VT,FC,GA,AT,AU,GC,DI,LE,U$,PI,I,HV$,FD,FL,FX,
ST,ID,IL,Q$,UD$
110 UD$="07/09/82 14:49:54" :REM: Update Level
120 PI=3.1416 :Q$=CHR$(34) :ST=.5 :GC=231 : 'ST
is the default increment, GC is the cu"/gals
conversion factor.
129
""'12345678901234567890123456789012345678901
23456789012345678
130 U$=" ###.# #####, #####,"
140 DEFFNGA(VT)=VT/GC : ' CONVERT CU. INS. TO
GALS.
150 DEFFNA(DE,DI)=(DI/2)-DE : ' SURFACE TO
CENTRE LINE
160 DEFFNB(A,C)=SQR(C↑2-A↑2) : ' HALF SURFACE
WIDTH
170 DEFFNAY(B,A)=B*A : ' AREA OF ISOSCELES
TRIANGLE
180 DEFFNAT(C)=PI*C↑2 : ' AREA OF TANK CROSS
SECTION
190
DEFFNAS(A,C)=(2*(ATN((A/C)/SQR(-(A/C)*(A/C)
+1))))/(2*PI)*AT : 'AREA OF SUM OF TWO SECTORS
200 CLS :PRINT"Volume of liquid in cylindrical
tanks1. Spot calculations2. Print lists3. Save file
(after modification)Please select"; :INPUTFX
:ONFXGOTO210,300,65000
210 CLS :PRINT"Spot calculations" :GOSUB6000 :'
GET DIAMETER AND LENGTH OF TANK
220 GOSUB 5100 : ' HORIZONTAL OR VERTICAL
230 AT=FNAT(C) :VT=FNGA(AT*LE) :INPUT"What is
the depth of liquid (in inches)";DE
240 IF HV$ = "VERTICAL" THEN GA = FNGA(AT * DE)
: GOTO 260
250 GOSUB5200
260 PRINTUSING"Gallons in tank = #####, spare
capacity = #####, gallons.";GA,VT-GA
:INPUT"Press <ENTER> to continue";FX :GOTO200
300 CLS :PRINT"Print Lists"
310 GOSUB6000 'GET DIAMETER D LENGTH OF TANK
320 GOSUB5100 :'HORIZONTAL OR VERTICAL ?
330 PRINT"What depth increment (default =";ST;Q$;
:INPUT")";ST
340 GOSUB60000 :GOSUB7000 :' CHECK PRINTER
STATUS, PRINT TITLES
350 AT=PI*C↑2 :VT=AT*LE
:IFHV$="HORIZONTAL"THEN500
360 FORI=0TOLESTEPST :GOSUB5000
370 FC=FC+1 :IFFC=56THENGOSUB6990
380 LPRINTUSINGU$;I,FNGA(AT*I),FNGA(VT-(AT*I))
390 NEXT :GOTO200
500 VT=FNGA(VT) :FORDE=0TODISTEPST
510 FC=FC+1 :IFFC=56THENGOSUB6990
520 IFDE=0THENGA=0
:GOTO550ELSEIFDE=DITHENGA=VT :GOTO550
530 GOSUB5200 : ' GET VOLUME OF LIQUID IN
GALLONS
540 GOSUB5000
550 LPRINTUSINGU$;DE,GA,VT-GA
560 NEXT: GOTO200
5000 IF PEEK(14400) AND 128 THEN 200 ELSE
RETURN
5100 INPUT"Horizontal or vertical tank (H/V)";HV$
:IFHV$<>"V"ANDHV$<>"v"THENHV$="HORIZONTAL"
ELSEHV$="VERTICAL"
5110 RETURN
5200 A=C-DE :B=FNB(A,C)
5210 AU=AT/2-FNAY(B,A)-FNAS(A,C) : ' THIS GIVES
CROSS SECTIONAL AREA OF LIQUID IN TANK
5220 GA=FNGA(AU*LE) :RETURN
6000 INPUT"What is the tank diameter (in
inches)";DI :C=DI/2 :FD=INT(DI/12) :ID=DI-FD*12
6010 INPUT"What is the length of the tank (in
inches)";LE :FL=INT(LE/12) :IL=LE-(FL*12) :RETURN
6990 FORFX=1TO10 :LPRINT" " :NEXT
7000 LPRINT"TANK CALIBRATION LISTING"
:LPRINTHV$;" TANK,"FD"FEET"ID"INCHES
DIAMETER,"FL"FEET"IL "INCHES LENGTH." :LPRINT" "
:LPRINT"CALIBRATED AT"ST"INCH INTERVALS" :LPRINT"
"
7010 LPRINT" LIQUID DEPTH LIQUID VOLUME SPARE
CAPACITY" :LPRINT" (Inches) (Gals) (Gals)"
:LPRINT" " :FC=8 :RETURN
60000 PRINT"When printer is ready press <ENTER>"
:INPUTFX
60010 IF(PEEK(&H37E8)AND48)=48THENPRINT"Press
<SPACE-BAR> to stop printing." :RETURN
60020 PRINT :PRINT"===> Printer is not ready
<===" :PRINT :GOTO60000
65000 CLS :PRINT@512,"Saving file under the name
of 'TANK'"
:FX=PEEK(VARPTR(UD$)+1)+256*PEEK(VARPTR(UD$)
+2) :FORI=1TO17
:POKEFX-1+I,ASC(MID$(TIME$,I,1)) :NEXT
65010 SAVE"TANK" :PRINT"File saved - update level
= "UD$
```

Deadstik

A simplified space shuttle program

Color Computer

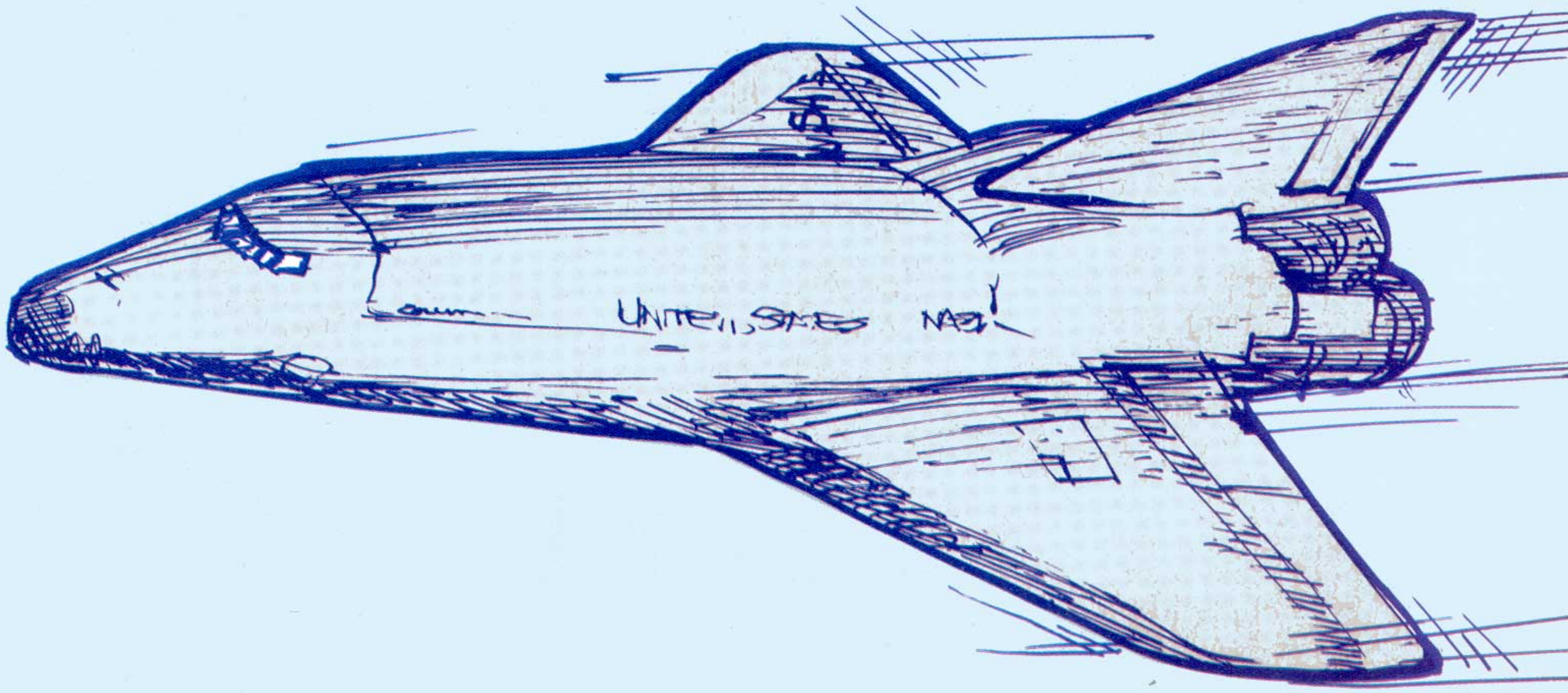
Gary Ludeke, Colorado Springs, CO

Deadstick is a program designed to simulate, in a very simplified manner, the process of flying a powerless aircraft from a given point relative to an airport runway, at a given initial speed, to a safe landing. This is known as making a "dead stick" landing. Mastery of Deadstik will enable the pilot to gain an appreciation of a small part of the challenge that faces space shuttle astronauts returning to earth.

Deadstik begins by drawing the outline of the aircraft instruments (lines 90-165). For memory efficiency, lines are drawn the full length and width of the screen, then erased where required with subroutines beginning at lines 1000 and 2000. Lines 200 to 226 label the instruments as to function. Lines 250 and 255 provide the initial values for speed, angle of descent, distance from the runway centerline, and altitude. Twelve scans of the keyboard are performed each second by lines 300 to 355. In response to keyboard inputs, the descent angle (AN), turn rate (TR), landing gear (G), and spoiler (S) indicators are updated. Lines 400 to 410 determine the compass indication based on the turn rate. Line 415 converts the descent angle (AN) from degrees to radians (RA).

The equation for the acceleration of the aircraft (line 420) is based on the concept of the aircraft sliding down a frictionless plane at the descent angle (AN) (see Figure 1). The force producing the acceleration is the weight (100,000 pounds) multiplied by the sine of the descent angle (RA). Opposing this force is aerodynamic drag represented by a constant (k), multiplied by the velocity (V) squared. Additional drag is created by the spoilers and lowered landing gear (S=1 and/or G=1) which adds additional forces. The value of 2.5 in line 420 was experimentally chosen. Those users familiar with aircraft performance are encouraged to experiment with various values for these constants to attempt to simulate various aircraft in a powerless condition. The display is updated once per second, which allows the velocity equation in line 425 to be correct without a value of time (e.g., V=V+at with t=1 second).

The aircraft stalls (ceases to fly) below 180 mph (line 435). As a warning of an impending stall, line 430 generates a stall warning indication when speed drops below 200 mph. Line 440 computes the descent rate (DR) in feet per second, and the altitude in feet. When the descent rate is displayed, it is converted to feet per minute.



If you just bought another computer,
boy are you gonna be sorry.

Epson.

128

The new Epson QX-10 is unlike any personal computer you've ever seen. It's a computer for people who don't have the time to learn computers; a computer you can be using within minutes.

And fortunately, you don't have to take our word for it. Here's how *Byte*, one of the computer industry's most prestigious magazines, describes the QX-10.

The first anybody-can-use-it computer.

"The Epson QX-10 (is) a computer for less than $3000 that may well be the first of a new breed of anybody-can-use-it 'appliance' computers ... In addition to being a highly integrated word processing/computer system that offers as much usable processing power as almost any existing microcomputer, the QX-10 ... system is designed to be used by people with minimal technical knowledge. We've certainly heard that claim before, but Epson has delivered on this promise in a way and to an extent that *no* microcomputer manufacturer has done."

That's nice to hear from a magazine like *Byte*, of course, but it doesn't surprise us. It's just what we intended the QX-10 to be all along.

More computer. Less money.

But useability isn't the only thing the QX-10 has going for it. As *Byte* says, "the QX-10 gives you a great deal for your money.

"Help is available at any time through the HASCI (Human Application Standard Computer Interface) keyboard Help key ... Text can be entered at any time just as you would in a conventional word processor. The Calc key turns the system into a basic 4-function calculator. Graphics can be created via the Draw key. The Sched (schedule) key gives you access to a computer-kept appointment book, a built-in clock/timer/alarm, and an event scheduler."

Advanced hardware for advanced software.

As for hardware, *Popular Computing*, another industry leader, says: "The QX-10 includes ... a number of advanced hardware features ... The basic components of the system are a detachable keyboard, a high resolution monochrome display, and a system unit containing two 5¼ inch disk drives. The drives use double-sided, double-density disks (340K bytes per disk) and are amazingly compact ... The QX-10 uses an 8-bit Z80A microprocessor. The system contains 256 bytes of RAM. Some of the RAM is ... battery powered ... which lets the computer retain information when the power is off."

You won't have to wait much longer.

The new Epson QX-10 may very well be the computer you've been waiting for. And fortunately, you won't have to wait much longer — it will be appearing soon in computer stores all across the country. In the meantime, write Epson at 3415 Kashiwa Street, Torrance, CA 90505, or call (213) 539-9140. We'll be happy to send you copies of our reviews.

After all, as *Popular Computing* puts it, the QX-10 will "do for computing what the Model T did for transportation."

And we couldn't have said it better ourselves.

EPSON

EPSON AMERICA, INC.

COMPUTER PRODUCTS DIVISION

3415 Kashiwa Street
Torrance, California 90505
(213) 539-9140

Figure 1

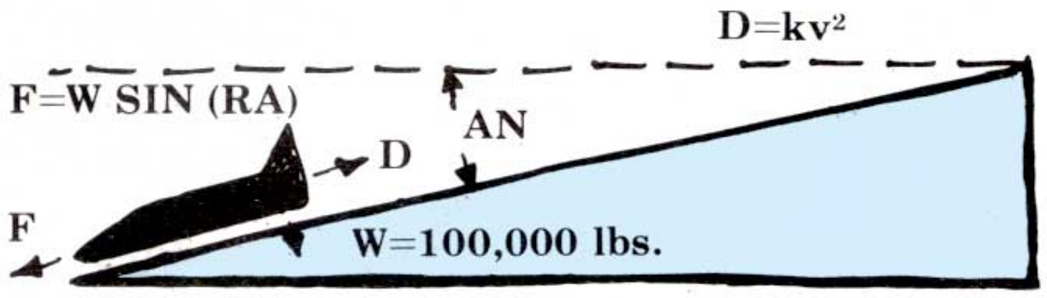
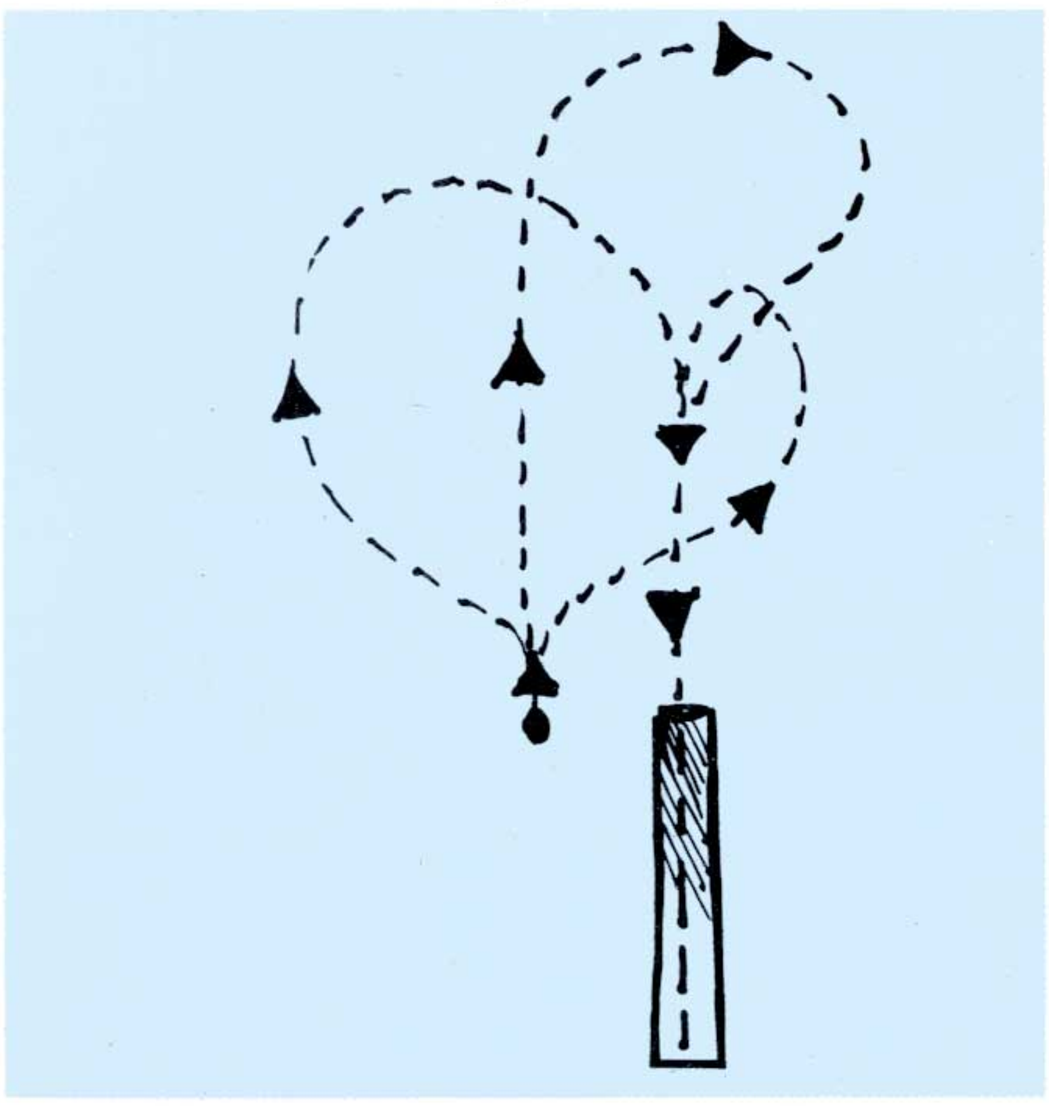


Figure 2



When the altitude reaches zero (line 445), the program jumps to line 950 to begin an evaluation of the aircraft's status. Conditions which will register a crash include a descent rate (DR) greater than seven feet per second, a landing short of the runway (DM>0), a landing beyond the runway end (DM<-3) (the runway is three miles long), a landing more than 100 feet left or right of the runway centerline (line 965), landing gear up (G=0), or a heading not equal to 180 degrees.

Lines 500 to 515 calculate the horizontal position of the aircraft relative to the runway threshold. The heading is first converted to radians and then used in the calculation of the east to west (V1) and north to south (V2) components of velocity and distance traveled (D1 and D2, respectively). Lines 600 to 625 update the instrument displays. Should a safe touchdown on the runway occur, line 980 will display the fact, and lines 981to 987 will calculate the braking phase of the landing and cause the display to be updated once per second. Flag "F" in line 980 tells the display update segment of the program (lines 600 to 625) where to go after an update (line 625). The deceleration due to braking is determined as five mph per second (line 983). Line 982 is a delay to cause a one-second display update during braking.

Flying the Simulator

After loading Deadstik (use a CLEAR 20 statement first) and entering RUN, the instrument panel will be drawn and the flight initiated. You will find yourself at an altitude of 30,000 feet, one mile west of the approach end of runway 18 (runway heading 180 degrees), on a heading of zero degrees (north), at an airspeed of 600 mph, and a descent angle of 30 degrees. It is important for those with familiarity with aircraft to realize that the pilot controls the angle of descent of the aircraft, and not the angle of attack of the wing. Due to aerodynamic drag, you will note that speed is decreasing, even though the descent angle is relatively steep.

Using the keyboard controls as indicated in Table 1, your job is to maneuver the aircraft onto the final approach course (heading 180 degrees and within plus, or minus, 100 feet of the extended runway centerline) so as to arrive at, or beyond, the runway threshold at zero altitude. At the moment of touchdown, your descent rate must be at, or below, seven feet per second or you will crash. You are free to maneuver in any manner you choose. Figure 2 gives a few examples of approach patterns that can be used, but the alternatives are virtually limitless.

Points worth mentioning are that the radius of turn is a function of both the turn rate and the speed (slow down for a tighter turn), spoilers allow a greater descent angle without excessive speed buildup, the landing gear creates additional drag when lowered. It cannot be raised again, so be sure you can make the runway with the additional drag before lowering it.

A tip for beginners is to first set up a reasonable rate of descent, then maneuver to intercept the final approach course at 10-15 miles out. Once established within 100 feet of the centerline on a heading of 180 degrees, shift your attention to regulating the descent so as to be stabilized at a five-degree descent angle and 220-230 mph at 1,000 feet and 2 miles from the runway. If your speed is too high and/or your touchdown too far down the runway, you will run off the far end. Happy flying.

Table 1
Simulator Controls

↑	Raise nose (decrease descent angle) one degree per keystroke
↓	Opposite of uparrow
←	Initiate, or increase, a left turn by one degree per second, or reduce a right turn by one degree per second
→	Opposite of leftarrow
G	Lower landing gear
S	Open spoilers
C	Close spoilers

Program Listing for Deadstik

```
5 REM DEADSTIK
90 CLS(0)
100 FORI=0TO63
105 SET(I,0,1):SET(I,12,1)
110 SET(I,16,1):SET(I,28,1)
115 NEXT
120 FORI=17TO20:GOSUB1000:NEXT
125 FORI=43TO46:GOSUB1000:NEXT
130 FORI=62TO63:GOSUB1000:NEXT
135 FORI=0TO31
140 SET(0,I,1):SET(16,I,1)
145 SET(21,I,1):SET(42,I,1)
150 SET(47,I,1):SET(61,I,1)
155 NEXT
160 FORI=13TO15:GOSUB2000:NEXT
165 FORI=29TO31:GOSUB2000:NEXT
200 PRINT@35,"ASI";
202 PRINT@162,"(MPH)";
204 PRINT@44,"DA(DEGS)";
206 PRINT@140,"DIST(NM)";
208 PRINT@58,"ALT";
210 PRINT@185,"(FT)";
212 PRINT@290,"TURN";
214 PRINT@322,"RATE";
216 PRINT@300,"HDG(DEG)";
218 PRINT@395,"DIST.";
220 PRINT@314,"VSI";
222 PRINT@441,"(FPM)";
224 PRINT@486,"GEAR:UP";
226 PRINT@496,"SPOILERS:";
250 V=880:V2=V:AN=30:D1=-5280
255 AL=30000:PI=3.141592654
300 FORI=1TO12
305 A$=INKEY$
310 IFA$="↑"THEN AN=AN-1
315 IFA$=CHR$(10)THEN AN=AN+1
320 IFA$=CHR$(8)THEN TR=TR-1
325 IFA$=CHR$(9)THEN TR=TR+1
330 IFTR<-3THEN TR=-3
335 IFTR>3THEN TR=3
340 IFA$="G"THEN G=1
345 IFA$="S"THEN S=1
350 IFA$="C"THEN S=0
355 NEXT
400 HG=HG+TR
405 IFHG>359THEN GOSUB2100
410 IFHG<0THEN HG=HG+360
415 RA=PI*AN/180
420 AC=(3220000*SIN(RA)-2.5*V*V-
2.5*V*V*S-2.5*V*V*G)/100000
425 V=V+AC:VM=V*15/22
427 VG=V*SIN(PI/2-RA)
430 IFVM<200THEN SOUND128,1
435 IFVM<180THEN900
440 DR=V*SIN(RA):AL=AL-DR
445 IF AL<=0 THEN950
500 RA=PI*HG/180
505 V1=VG*SIN(RA):D1=D1+V1
510 V2=VG*SIN(PI/2-RA)
515 D2=D2+V2:DM=D2/5280
600 PRINT@98,INT(VM);
602 PRINT@78,AN;
604 PRINT@173,INT(10*DM)/10;
606 PRINT@120,INT(AL);
608 PRINT@387,TR;
610 PRINT@334,HG;
612 IFD1<0THEN PRINT@401,"WEST"; ELSE
PRINT@401,"EAST";
614 PRINT@428,INT(ABS(D1));
616 PRINT@376,60*INT(DR);
618 IFG=1THEN PRINT@491,"DN";
620 IFS=1THEN PRINT@505,"OPEN"; ELSE
PRINT@505,"CLOSED";
625 IFF=1THEN982ELSE300
900 PRINT@226,"AIRCRAFT
STALLED-CRASHED!";:GOTO4000
950 IFDR>7THEN3000
955 IFDM>0THEN3100
960 IFDM<-3THEN3200
965 IFABS(D1)>100THEN3300
970 IFG=0THEN3400
975 IFHG<>180THEN3500
980 PRINT@226,"TOUCHDOWN!";:F=1
981 AN=0:AL=0:DR=0:TR=0
982 FORI=1TO450:NEXT
983 VM=VM-5:IFVM<0THEN VM=0
984 IFVM=0THEN5000
985 DM=DM-VM*22/15/5280
986 IFABS(DM)>3THEN3500
987 GOTO600
1000 RESET(I,O):RESET(I,12)
1005 RESET(I,16):RESET(I,28)
1010 RETURN
2000 RESET(0,I):RESET(16,I)
2005 RESET(21,I):RESET(42,I)
2010 RESET(47,I):RESET(61,I)
2015 RETURN
2100 HG=HG-360:PRINT@337," ";
2101 RETURN
3000 PRINT@226,"CRASHED-SINK RATE > 7
FPS";:GOTO4000
3100 PRINT@226,"CRASHED SHORT OF
RUNWAY";:GOTO4000
3200 PRINT@226,"CRASHED BEYOND
RUNWAY";:GOTO4000
3300 PRINT@226,"CRASHED OFF EDGE OF
RUNWAY";:GOTO4000
3400 PRINT@226,"LANDED GEAR UP";:GOTO4000
3500 PRINT@226,"RAN OFF RUNWAY";:GOTO4000
4000 SOUND1,10
4001 GOTO4001
5000 PRINT@226,"STOPPED ";
5001 PRINT@98,0;:GOTO 4001
```

Structured BASIC

Modern programming techniques in BASIC

Models I/II/III, PMC-80, LNW80 with disks

T. R. Dettmann, Associate editor

These days it doesn't take much reading to run into terms like "Modular Programming," "Structured Programming," and "Top Down Programming." At times, it seems like everywhere you look, someone is filling you in on his favorite cure for programming problems.

After all the articles though, many people I've talked to have come to the opinion that all of these sure fire methods for programming better and faster are fine for the author but not anyone else. Some people have given up on finding a cure-all programming method. Some have even taken to laughing at the mention of the above "guru words". But a few have distilled out a useful insight: all the methods are really nothing more than the application of common-sense problem solving to programming.

The essence of top-down programming is looking at the large problems first.

Sure, there are dozens of ways to do it. How many people do you know who approach problems in the same way? Should it then be any surprise that there are numerous ways to program?

What do all the fancy terms mean? It turns out that the concepts are really very simple. Let's look at them one at a time.

Structured Programming

Structured programming arose from the desire to be able to prove in a mathematical way that a program was correct. With the flexibility of a normal programming language such as BASIC, it was found that proving a program to be correct wasn't just difficult, it was nearly impossible. If we limited our programming to using only three simple structures and not using GOTO's, it became possible to prove the correctness of our program.

What are these three marvelous structures? The first is simply a normal program statement with no control. You start at the beginning, do the instruction, and come out the end. No problem.

The IF...THEN...ELSE is also one of the structures. It allows us to make decisions and the flow of control is very simple. One of the most important aspects of this structure is its 'block structure.' The idea is that there can be more than one statement executed in either the THEN or ELSE case. Microsoft BASIC does it to a limited extent, but it should be possible to do it with as many statements as necessary.

The basic form of the block-structured IF statement looks like this:

```
IF (condition) THEN
  one or more statements
ELSE
  one or more statements
ENDIF
```

The last thing we need is a simple loop. A very flexible form is the WHILE statement. The WHILE works like this: at the beginning of the loop, we test our condition; as long as it's true, we keep repeating the loop. We could express it like this:

```
WHILE (condition)
  one or more statements
WEND
```

where the WEND marks the end of the loop like the NEXT does in a FOR-NEXT loop.

By using these simple structures and avoiding GOTO's, we produce programs that are easier to prove correct and in fact easier to make correct in the first place.

Modular Programming

Modular programming is a technique, but it isn't really hard to do. In fact, most of us do it every day without even realizing it. One of the first things most people learn about solving problems is that when you have one that's too big to solve, break it up into little ones and solve them. The big problem will take care of itself.

Sounds too simple to be a modern programming technique but it is. The essence of modular

DOSPLUS 4.0, the perfect score.

The DOSPLUS 4.0 hard drive system is here!

A hard drive without the dynamic new DOSPLUS 4.0 is like an eggshell without the egg. The new DOSPLUS 4.0 is the leading edge—the latest in the line of advanced disk operating systems from MICRO-SYSTEMS SOFTWARE, INC. It's the only current operating system written from the ground up for hard disk operation. Not just a driver but a fully developed system. What that means to you is more bang for your buck! Finally an unlimited TRS-80 for small business. For only **$1899** you get the DOSPLUS "PLUS."

THE DOSPLUS 4.0 FEATURES

- Single volume addressing/Double sided floppies seen as one drive-one file can expand to limit of the hard drive
- Hard Disk—disk editing utilities
- Incredible I/O speed
- Runs any combination of densities or tracks
- Also operates 8" drives with special hardware—comes with expanded users guide and complete DOS technical section on I/O calls and DCB organization
- Ability to use hard drive as the "system" drive.

"PLUS" MANY OF THE SENSATIONAL NEW DOSPLUS 3.4 FEATURES

- BASIC array sort—multi key, multi array
- Tape/Disk—Disk/Tape utility (with relocator)
- Input (controlled screen input)
- Random access and ASCII modification on Diskdump
- BASIC checks for active "DO"
- Backup and Format from a "DO" file
- Much improved Backup (More reliable)

$1899 TAKE YOUR MICRO TO THE MAX. ORDER NOW!

COMPLETE WITH 5 MEG SYSTEM.

- I/O package much faster (disk access time reduced)
- Repeat last DOS command with "/" [ENTER]
- Short directory (filename and extension) available
- Short directory of Model III TRSDOS disks
- Single file convert from Model III TRSDOS
- COMPLETE device routing supported (DOS and BASIC)
- Ability to save BASIC programs directly to another machines' memory (if equipped with DOSPLUS 3.4)

NOTE: The final versions of 3.4 and 4.0 will have almost identical features and documentation.

THE COMPLETE SYSTEM

- Smooth, silent, swift
- Error-free disk I/O
- *Add on up to 4, 10 meg units for a total of 40 megabytes!
- Plugs on the 50 pin data bus.—no loss of floppy drives
- Completely self-contained—just plug it in and go

*10 meg units available Soon.

NOTE: Specify 40 or 80 track when ordering DOS diskette. After initial bootup, user can create any DOS desired.

The first in the industry backed by a lifetime warranty**

**Lifetime warranty on original media

DOSPLUS

DOSPLUS first in quality!
First in the industry!

MICRO-SYSTEMS SOFTWARE, INC.

4301-18 Oak Circle
Boca Raton, FL 33431
Telephone: (305) 983-3390
(800) 327-8724

MS INC.

16

programming is choosing the right breakdown of the major problem. Usually the best course is to follow your instincts.

If you take out a sheet of paper and list the steps, in English, that you have to do, that makes a good starting point for your module selection. But in what order do we do this? That's where Top-Down programming comes in.

. . . all the methods are really nothing more than the application of common-sense problem solving to programming.

Top-down Programming

When we sit down at the computer, especially with BASIC, often the first impulse is to jump in as quickly as possible and start programming. After all, that's where the fun is. The problem we run into is that we can only see results if we start with small parts of the problem. Many people start out this way and wind up building many small blocks of code and then patching them together into programs. Sometimes it works, but usually it creates problems. By creating answers to the small problems without looking at the large ones, we're lucky to get all the pieces to fit together correctly.

The essence of Top-Down programming is looking at the large problem first. We look at the overall problem and write a program that will handle that problem. In the process, anything we don't know how to do is put into a subroutine to be developed later. We just have the subroutine supply some simple response when it's called until we're ready to look at it.

When we're satisfied that the overall program functions correctly, then we drop down a step and look at one of the subroutines we used to bury our problems. Now we apply the same technique there. Look at that as a whole problem, modularize and solve it. As we continue down, level by level, eventually we reach a level where the problem we have left to solve is simple. We just code the answer and at this point, everything else works so the program is done.

Obviously I'm over-simplifying. Yet, even for the most complicated problem, the method still works in the same way.

To try to make it possible to get the flavor of Top-Down, Structured, Modular programming, we've written a program which accepts simple, structured forms and gives us a BASIC program as a result.

It accepts block structured IF...THEN...ELSE, WHILE loops, and it eliminates the ever-present line numbers by allowing us to deal with labels instead of line numbers.

A simple program to run through the processor is the one shown. The idea is to be able to read in an ASCII program file, strip the line number from the front of each line, and then print the result.

The steps to accomplish this are:

```
Get the file name we want to list
Open the file
While we're not at the end of file
  read a line
    strip off the line number
  print the line
End of While
Close the file
```

These steps have been put into a program, called STRUCT/REM, Listing 1, that can be translated by the translator. Listing 2, called TEST/STR is a structured BASIC program that can be translated by STRUCT/REM. The output from the translation is Listing 3, called TEST/BAS. Notice how STRUCT/REM allows you to use structured programming techniques while still working with your existing Microsoft BASIC interpreter.

Our initial program included only a dummy subroutine (what the translator calls a PROCEDURE) which did nothing except return. With only the main program, the program when run would list the lines the way they were and not change them. Once that program worked, the <ELIM-NUM> subroutine was written to strip the line numbers off and the program was complete.

An advantage of this approach is that it is easier to find problems. By actually programming each stage from the main module down we can test them and make sure they work. When an error occurs, we know it has something to do with the module we just wrote.

Why Go To All This Trouble?

All of this effort seems to be a waste of time? Wouldn't it be faster to just sit down and write a program? NO!

Most studies have shown that the "just sit down and write it" approach actually increases the time needed to get a program working AND makes it more prone to error. Furthermore, when it comes time to make changes in it or expand it for new conditions, it's generally less flexible.

Professional programmers and computer scientists have put much effort into finding better ways to program. They haven't turned programming into a science yet, but they've given us some techniques that we can use.

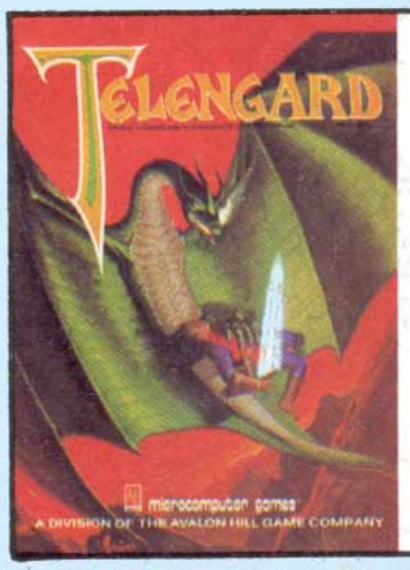
Most studies have shown that the "just sit down and write it" approach actually increases the time needed to get a program working and makes it more prone to error.

Me, I'm just too lazy. I don't want to do any more work than I absolutely have to, so anything that simplifies my programming and makes me more efficient is something I like.

None of these techniques are cure-alls. The best technique is the one that you're comfortable with, but no technique is so good that you should blindly use it. The essense of programming is common-sense problem solving. So get out there and use that common sense.

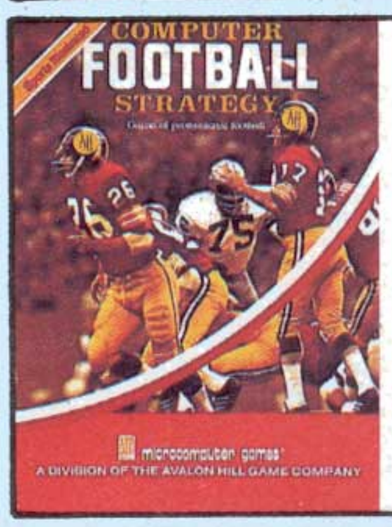
The Thinking Man's Gamemaker

WARNING: Do not buy Avalon Hill Microcomputer Games unless you are above average!



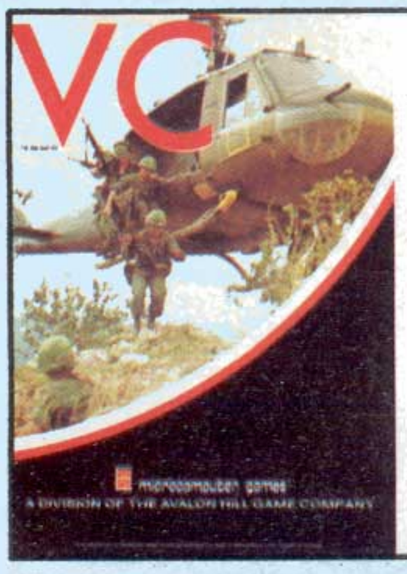
TELENGARD – Microcomputer Dungeon Adventure game. In the mysterious underworld of TELENGARD there are fifty levels of ever-more complex mazes for mighty adventurers to explore. Various means of survival tactics are at the adventurer's disposal and all of the necessary ingredients have been incorporated into this solitaire **real time fantasy and role-playing game.**

G.F.S. SORCERESS – The first **science-fiction adventure game** of the continuing saga of Joe Justin and Selena Sakarov aboard the Galactic Federation Starship Sorceress. In the game, the player takes the part of Joe Justin as he attempts to clear himself of a false charge of mutiny. Beautiful full-color manuals are included giving the player useful clues in his or her attempt to prove Joe Justin's innocence.

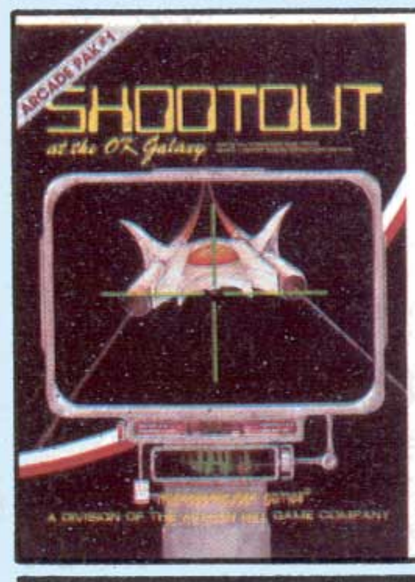


COMPUTER FOOTBALL STRATEGY – Thrilling computer version of Avalon Hill's famous board game. Based on the award-winning Sports Illustrated game of professional football, **Computer Football Strategy** forces the player to constantly make the right decisions about his team's offensive and defensive formations. Match wits against the computer or against a live opponent.

VOYAGER – A solitaire computer game that challenges the human player to explore the four levels of an alien spacecraft's maze-like corridors and rooms in 3-D simulated graphics, all the while avoiding robots programmed to blast any intruders. In order to win, the human must destroy all power generators and escape or hunt out and annihilate the killer robots. **VOYAGER** comes with color-animated graphics and sound capabilities.



V.C. (short for Viet Cong), is Avalon Hill's first game ever on the controversial Viet Nam War. Under your command is the chopper based air-mobile and heavily armed 1/509th Air Cavalry and 9/15th Field Artillery for fire support. It is an **unconventional conflict.** You have the task of bringing the civilian population under your protection where the enemy (played by the computer) can hide amongst the people, and where the politics of terrorism and friendship can turn the people you want to save against you.

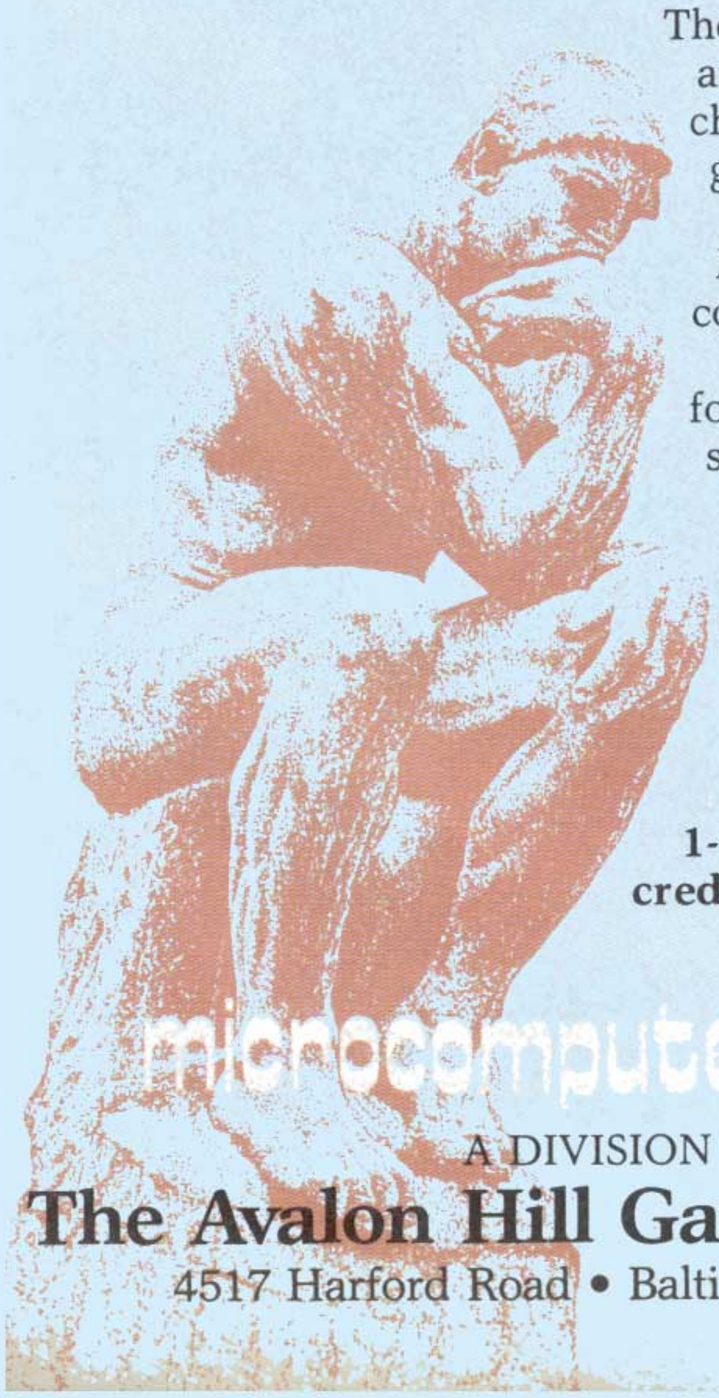


SHOOTOUT – 30 ALIEN WARSHIPS HAVE ENTERED YOUR PATROL ZONE. OK, shields up?, energy level . . . check, azimuth set?, yup. This may sound like the latest summer space movie thriller but in fact it's the preparations YOU will make when playing Avalon Hill's new solitaire arcade strategy game **SHOOTOUT AT THE OK GALAXY.** Over 2 years in the making, **SHOOTOUT** is purely graphical combining arcade excitement with just the right touch of strategy.

SOFTWARE GAMES	CASS. TRS-80 Color	CASS. TRS-80 Models I & III	DISK TRS-80 Models I & III	PRICE
B-1 Nuclear Bomber		16K		16.00
			32K	21.00
Midway Campaign		16K		16.00
			32K	21.00
North Atlantic Convoy Raider		16K		16.00
			32K	21.00
Nukewar		16K		16.00
			32K	21.00
Planet Miners		16K		16.00
			32K	21.00
Lords of Karma		48K		20.00
			48K	25.00
Computer Acquire		16K		20.00
Computer Statis Pro Baseball		16K		25.00
			32K	30.00
Tanktics		16K		24.00

SOFTWARE GAMES	CASS. TRS-80 Color	CASS. TRS-80 Models I & III	DISK TRS-80 Models I & III	PRICE
Computer Baseball Strategy		32K		17.00
Galaxy		16K		20.00
			32K	25.00
Voyager	16K	16K		20.00
Foreign Exchange		16K		20.00
Draw Poker		16K		16.00
Andromeda Conquest		16K		18.00
			32K	23.00
Telengard				
			48K	28.00
VC				
			32K	25.00
G.F.S. Sorceress		48K		30.00
			48K	35.00

SOFTWARE GAMES	CASS. TRS-80 Color	CASS. TRS-80 Models I & III	DISK TRS-80 Models I & III	PRICE
Fredericksburg			32K	35.00
Conflict 2500		16K		16.00
			32K	21.00
Empire of the Overmind		48K		30.00
			48K	35.00
Stocks and Bonds		16K		20.00
			32K	25.00
Computer Football Strategy			32K	21.00
Guns of Fort Defiance		16K		20.00
			32K	25.00
Dnieper River Line		32K		25.00
			32K	30.00
Shootout at the OK Galaxy	16K	16K		20.00
Tank Arkade		16K		15.00



These games are for the sly and clever. Check on the chart at right to see if the game is compatible with your home computer. Most Avalon Hill Microcomputer Games are also compatible with the following home computer systems: Apple II®; Atari 400 & 800®; IBM P.C.®; and Commodore VIC-20, PET CBM & 2001®. **Available at finer computer stores everywhere! Or call TOLL FREE: 1-800-638-9292 for fast credit card purchase. Ask for Operator 80.**

microcomputer games®

A DIVISION OF

The Avalon Hill Game Company

4517 Harford Road • Baltimore, MD 21214

I hereby certify that I am one sharp dude. Therefore, please send me _____ copies
number
of _____ on _____ for _____.
game title *cass. or disk.* *type of computer*

I enclose a total of $_____ (include 10% for postage and handling).

☐ Let me take an IQ test first. In the meantime please send me your FREE full-color game catalog with details on all your software.

Name _____

Address _____

City _____ State _____ Zip _____

☐ CHECK ☐ MONEY ORDER ☐ CHARGE (see below)

☐ AMERICAN EXPRESS ☐ CHOICE ☐ VISA ☐ MASTERCARD

Account No. _____ Expiration Date _____

For fast credit card ordering call Operator 80 TOLL FREE at 1-800-638-9292.

Send Check or Money Order to:

AVALON HILL MICROCOMPUTER GAMES

Dept. 80 • 4517 Harford Road • Baltimore, MD 21214

OUTHOUSE

Is There Nothing Sacred?

DISK VERSION TALKS

JOYSTICK Compatible*

Don't Miss This Outstanding Original Arcade Game!

SSM SOFT SECTOR MARKETING, INCORPORATED

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

Written by Factory Programming - ©1982

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.

Talking and sound effects are playable through the cassette AUX plug. High scores are automatically saved after each game on disk versions. *Compatible with Trisstick.

Call or write for our complete catalog.

MasterCard VISA AMERICAN EXPRESS

The Quality Continues . . .

Program Listing for STRUCT/REM

```
10 REM * *
20 REM
30 REMSTRUCTURED BASIC PRE-PROCESSOR
40 REM(C) 1982 BY TERRY R. DETTMANN
50 REM
60 REMFILENAME:STRUCT/BAS
70 REMVERSION 0.010/82
80 REM
90 REM * *
95 REMMAKE SOME SPACE FOR STRINGS
100 CLEAR10000
105 REMSET UP THE ARRAYS:
106 REMSYM$SYMBOLSLNLINE NUMBERS
107 REMSTK$SYMBOL STACKSTKLINE NUMBER STACK
108 REMKW$KEYWORDS
110 DIM
SYM$(100),LN(100),STK$(25),STK(25),KW$(10)
115 REMLS IS THE SIZE OF THE SYMBOL TABLE
116 REMSP IS THE STACK POINTER
117 REMKW IS THE NUMBER OF KEYWORDS
120 LS=100:SP=0:KW=6
130 DEFFNHDR$(X$) =
STRING$((78-LEN(X$))/2,150) + " " + X$ + " " +
STRING$((77-LEN(X$))/2,150)
131 REMFNHDR$ IS THE HEADER FUNCTION
135 REMSET UP ERROR PROCESSING
140 ON ERROR GOTO570
145 REMNM$ IS USED TO GET RID OF LINE NUMBERS
150 NM$="0123456789"
155 REMRM$ IS A TEMPLATE FOR THE START OF ALL
PROCEDURES
160 RM$="REM - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - -"
165 REMGET THE KEYWORDS
170 FOR I = 1 TO KW: READ KW$(I): NEXT I
180 DATA ENDIF,IF,THEN,ELSE,WHILE,WEND
200 REM ******* *** MAIN PROCESSING LOOP *
205 REMCLEAR THE SCREEN AND GET THE INPUT
AND OUTPUT FILES
210 CLS: PRINT FNHDR$("STRUCTURED BASIC
PROCESSOR"): PRINT: PRINT
220 LINEINPUT" INPUT FILE ==========> ";F1$
230 LINEINPUT" OUTPUT FILE ==========> ";F2$
240 REM - - - - - - - - BUILD SYMBOL TABLE -
245 REMFIRST PASS THROUGH THE FILE BUILDS THE
SYMBOL TABLE
246 REMLN IS THE CURRENT LINE NUMBER COUNTER
247 REMALL LINES WILL BE RENUMBERED IN 10'S
250 OPEN"I",1,F1$:LN=0
255 REMCONTINUE TO PROCESS LINES UNTIL THE
END OF FILE
260 IF EOF(1) THEN CLOSE:GOTO290
265 REMGET A LINE AND PROCESS IT
270 LINEINPUT#1,LN$:GOSUB1000
280 GOTO260
290 REM - - - - - - - - - PROCESS FILE TO
OUTPUT - - - - - - - - - - -
295 REMPAGE EJECT FOR PRINTOUT OF THE FINAL
PROGRAM
300 LPRINTCHR$(12)
305 REMGET THE FILES AND RESTART THE LINE
NUMBER COUNTER
310 OPEN"I",1,F1$: LN=0: OPEN"O",2,F2$
315 REMCONTINUE UNTIL THERE ARE NO MORE
LINES
320 IF EOF(1) THEN CLOSE:END
325 REMINPUT A LINE AND PROCESS IT
330 LINEINPUT#1,LN$:GOSUB2000
340 GOTO320
345 REMALL DONE, WRAP IT UP
350 CLOSE:END
500 REM********** SUBROUTINES *
510 REM - - - - - - - - - PUSH ON STACK -
520 IF SP=25 THEN RETURN
530 SP=SP+1: STK$(SP)=IN$: STK(SP)=IN: RETURN
540 REM - - - - - - - - - POP OFF STACK -
550 IF SP=0 THEN RETURN
560 IN$=STK$(SP): IN=STK(SP): SP=SP-1: RETURN
570 REM - - - - - - - - - ERROR PROCESSING -
- - - - - - - - - - - - - - -
580 IF ERL=250 THEN PRINT"FILE ";F1$;" DOES NOT
EXIST":RESUME220
590 PRINT "ERROR ";ERR;" IN LINE ";ERL
600 CLOSE:END
700 REM - - - - - - - - - DELETE LEADING
BLANKS -
710 IF LEN(IN$)=0 THEN RETURN
720 IF MID$(IN$,1,1)=" " THEN
IN$=MID$(IN$,2):GOTO710
730 RETURN
750 REM - - - - - - - - - DELETE TRAILING
BLANKS -
760 IF LEN(IN$)=0 THEN RETURN
770 IF MID$(IN$,LEN(IN$),1)=" " THEN
IN$=MID$(IN$,1,LEN(IN$)-1):GOTO760
780 RETURN
800 REM - - - - - - - - - DELETE LEADING LINE
NUMBER - - - - - - - - - -
810 IF LEN(IN$)=0 THEN RETURN
820 IF INSTR(NM$,MID$(IN$,1,1))<>0 THEN
IN$=MID$(IN$,2):GOTO810
830 RETURN
1000 REM* ********** PROCESS FOR SYMBOL TABLE
1005 REMINCREMENT THE LINE NUMBER COUNT BY
10
1010 LN=LN+10
1015 REMELIMINATE THE LINE NUMBER AND
LEADING BLANKS
1020 IN$=LN$: GOSUB800: GOSUB700: LN$=IN$
1035 REMIF THIS IS A STRUCTURED COMMENT THEN
FORGET IT
```

```
1040 IF MID$(LN$,1,2)="/*" THEN
LN=LN-10:RETURN
1045 REMLOOK FOR KEYWORDS IN THE LINE, IF
FOUND THEN
1046 REMNOTHING ELSE NEED BE CHECKED
1050 GOSUB1370:IF EF=1 THEN 1070
1055 REMCHECK FOR LINE LABELS OR PROCEDURE
STATEMENTS
1060 GOSUB1090:GOSUB1310
1065 REMIN CASE YOU WANT A PRINTOUT OF THE
INPUT PROGRAM,
1066 REMREMOVE THE LEADING CHARACTER (')
1070 'LPRINT USING"##### ";LN;:LPRINTLN$
1080 RETURN
1090 REM - - - - - - - - SCAN A LINE FOR LINE
NUMBER SYMBOLS - - - - -
1095 REMIF THE LINE IS A REMARK, THEN FORGET IT
1100 FLG=0:IF MID$(LN$,1,3)="REM" OR
MID$(LN$,1,1)="'" THEN RETURN
1105 REMSCAN A CHARACTER AT A TIME FOR LINE
NUMBER
1106 REMLABELS WHICH ALL HAVE THE FORM
"<WORD(S)>"
1110 FOR I=1TOLEN(LN$):C$=MID$(LN$,I,1)
1115 REMIF A LABEL HAS BEGUN, THE ">" ENDS IT
1116 REMFINISH AND ADD IT TO THE SYMBOL TABLE
1120 IF FLG>1 AND C$=">" THEN IN$=IN$+C$:
FLG=FLG+1: IN=LN: GOSUB1180: IN$="": FLG=0:
GOTO1160
1125 REMIF WE FIND "<>" THEN IT ISN'T A LABEL
1130 IF FLG=1 AND C$=">" THEN
IN$="":FLG=0:GOTO1160
1135 REMANYTIME WE FIND "<", IT COULD START A
LABEL
1140 IF C$="<" THEN LC=I: IN$=C$: FLG=1:
GOTO1160
1145 REMIF FLG>0 THEN WE'RE BUILDING A
POSSIBLE LABEL
1146 REMFLG WILL BE ITS LENGTH
1150 IF FLG>0 THEN IN$=IN$+C$:FLG=FLG+1
1160 NEXTI
1170 RETURN
1180 REM - - - - - - - - ADD TO SYMBOL TABLE
-
1185 REMCHECK THE SYMBOL TABLE, IF EF=0
THERE'S NO MORE ROOM
1190 GOSUB1260:IF EF=0 THEN RETURN
1195 REMIF EF>0, THE IT WAS FOUND, BUT IF LC<>1
THEN IT DOESN'T
1196 REMDEFINE THE LINE NUMBER ASSOCIATED
WITH THE LABEL
1200 IF EF>0 AND LC<>1 THEN RETURN
1205 REMIF IT'S FOUND AND IT'S THE FIRST THING
ON A LINE, IT
1206 REMDEFINES THE LINE NUMBER
1210 IF EF>0 AND LC=1 THEN LN(EF)=IN:RETURN
1215 REMNEGATIVE EF MEANS THAT IT IS AN OPEN
SPACE FOR A NEW
1216 REMSYMBOL
1220 EF=-EF
1230 SYM$(EF)=IN$
1235 REMIF THE FIRST TIME WE FIND IT, IT'S ALSO AT
THE
1236 REMBEGINNING OF THE LINE, THEN ADD THE
LINE NUMBER TOO
1240 IF LC=1 THEN LN(EF)=IN ELSE LN(EF)=0
1250 RETURN
1260 REM - - - - - - - - SEARCH FOR IN$ IN
SYMBOL TABLE - - - - - -
1265 REMEF=IS IF THE SYMBOL IN$ IS FOUND AT
POSITION IS
1270 FOR IS=1TOLS:IF IN$=SYM$(IS) THEN
EF=IS:RETURN
1275 REMIF AN EMPTY SPACE IS FOUND AT THIS
POINT, THERE
1276 REMARE NO MORE SYMBOLS, SO MARK THIS
LOCATION FOR
1277 REMPOSSIBLE ADDITION
1280 IF SYM$(IS)="" THEN EF=-IS:RETURN
1290 NEXTIS
1295 REMIF IT GETS TO HERE, THERE'S NO MORE
SPACE AVAILABLE
1296 REMGOOD PLACE FOR AN ERROR MESSAGE
1300 EF=0:RETURN
1310 REM - - - - - - - - - LOOK FOR
PROCEDURE IDENTIFICATION - - - - - - -
1315 REMIS THE KEYWORD PROCEDURE IN THE
LINE?
1320 IF INSTR(LN$,"PROCEDURE")=0 THEN RETURN
1325 REMIF IT IS, LOOK FOR THE BRACKETS FOR A
LABEL
1330 L1=INSTR(LN$,"<"): L2=INSTR(LN$,">")
1335 REMIF NO BRACKETS, IT MUST NOT BE A
PROCEDURE
1340 IF L1=0 OR L2=0 THEN RETURN
1345 REMEXTRACT THE PROCEDURE LABEL AND ADD
IT TO THE
1346 REMSYMBOL TABLE
1350 LC=1: IN$=MID$(LN$,L1,L2-L1+1): IN=LN:
GOSUB1180
1360 RETURN
1370 REM - - - - - - - - - SEARCH FOR
KEYWORDS - - - - - - - - - - - - - - - -
1380 FOR I = 1 TO KW: LC=INSTR(LN$,KW$(I))
1390 IF LC<>0 THEN 1420
1400 NEXTI
1405 REMNO KEYWORD FOUND
1410 EF=0:RETURN
1415 REMKEYWORD I FOUND
1420 EF=1:NK=I
1425 REMGO TO PROCESSING FOR A PARTICULAR
KEYWORD
1430 ON NK GOSUB
1580,1450,1510,1530,1630,1660
```

Innovation...

MULTIDOS...C.E.C.'s deluxe disk operating system for the TRS 80 Model I and Model III. Has forty library commands, twelve utilities, a MIGHTY-MULTI mini-dos, supports three densities (MODEL I version), read/write to/from practically any DOS format and is compatible with most double density boards. Supports clock speeds up to 5.32 MHz. MULTIDOS has, standard, two super BASICs, including BBASIC, the popular "BOSS" single step king and a host of Basic program editing utilities. Versatile, user friendly, fast, efficient. ***Get into the world of DOS FANTASTIC!...$99.95***

Z'DOS...economical disk operating system for TRS 80 Model I and Model III. Inexpensive and feature packed. Forty library commands, supports three densities (Model I version), read/write compatible with most DOS formats and double density boards. Has SUPERBASIC II with 41K of free memory, cross reference, renumber and global editing. ***The fastest DOS in town!...$39.95***

EBASIC...extended disk Basic. Adds high-speed graphics, versatile special functions, twenty-four types of formatted keyboard INPUT, program line labeling, array operations, and multi-key SORT of unlimited arrays. All in low RAM along with Disk Basic and still leaves over 38K of free memory. ***EBASIC just...$29.95***

(EBASIC requires Z'DOS or MULTIDOS Model I version 1.4 or greater, or MULTIDOS Model III version 1.3 or greater.)

BOSS / RENUM90...Best operated single stepper. This machine language BASIC program debugging utility allows you to single step BASIC programs, trace in a controlled area on the display, trace to the line printer, save the screen and display variables, control program execution speed, save Basic programs to high RAM and insert breakpoints into your Basic program. RENUM90 allows you to renumber all or part of your Basic program. **On tape or disk, BOSS/RENUM90 for $24.95 or BOSS only, $15.95**

— Available from the System Innovators —

COSMOPOLITAN ELECTRONICS CORPORATION
5700 PLYMOUTH ROAD
ANN ARBOR, MICHIGAN 48105

Telephone: (313) 668-6660

Add $3.00 shipping/handling. Foreign orders add $10.00. Michigan residents add 4% sales tax. C.O.D. add $1.50. Model I orders specify single, double or P density. Personal checks take 2 weeks to clear. VISA & Master Card accepted.

AUTHORIZED DEALERS

ABC SALES
13349 Michigan Avenue
Dearborn, MI 48126
(313) 581-2896

COMPUTER SHACK
1691 Eason
Pontiac, MI 48054
(313) 673-2224
1-800-392-8881

POWER SOFT
11500 Stemmons Expressway
Dallas, TX 75229
(214) 484-2976
1-800-527-7432

BYTES & NAILS
5110 6th Avenue
Sioux City, IA 51106
(712) 274-2348

SIMUTEK
Computer Prod., Inc.
4877 East Speedway
Tucson, AZ 85712
(602) 323-9391
1-800-528-1149

MICRO-PROG-80
P.O. Box 606
Sunnymead, CA 92388
(714) 653-9429

19

TRS-80 is a registered trademark of Tandy Corp.

```
1440 RETURN
1450 REM - - - - - - - - - - IF STATEMENT - -
- - - - - - - - - - - - - - - - - -
1455 REMIF THERE'S SOMETHING AFTER 'THEN' THEN
THE
1456 REMIF IS AN OLD STYLE IF, NOT A BLOCK IF
1460 LC=INSTR(LN$,"THEN"):IF (LC+4)>=LEN(LN$)
THEN 1490
1465 REMIF THE ONLY THING AFTER 'THEN' IS
BLANKS, THEN
1466 REMIT'S A BLOCK IF
1470 FORI=LC+4 TO LEN(LN$):IF MID$(LN$,I,1)<>"
" THEN EF=0:RETURN
1480 NEXTI
1485 REMSAVE A SYMBOL FOR THE IF ON THE STACK
FOR LATER
1486 REMADDITION TO THE SYMBOL TABLE
1490 IN$=STR$(LN):MID$(IN$,1,1)="I":GOSUB510
1500 EF=0:RETURN
1510 REM - - - - - - - - - - THEN STATEMENT -
- - - - - - - - - - - - - - - - - -
1515 REMNO SPECIAL PROCESSING FOR THEN
1520 EF=0:RETURN
1530 REM - - - - - - - - - - ELSE STATEMENT -
- - - - - - - - - - - - - - - - - -
1535 REMAN ELSE STATEMENT MUST BE MATCHED BY
AN IF ON
1536 REMTHE STACK, IF NOT, THEN THERE'S AN
ERROR
1540 GOSUB540:IF MID$(IN$,1,1)<>"I" THEN
LN$=LN$+"*** ERROR ***":EF=0:RETURN
1545 REMSAVE THE GOTO IN THE SYMBOL TABLE
FOR THE IF
1550 LC=1:IN=LN+10:GOSUB1180
1555 REMSTACK THE ELSE STATEMENT NOW
1560 IN$=STR$(LN):MID$(IN$,1,1)="E":GOSUB510
1570 RETURN
1580 REM - - - - - - - - - - ENDIF STATEMENT -
- - - - - - - - - - - - - - - - - -
1585 REMAN ENDIF STATEMENT IS MATCHED BY
EITHER AN
1586 REMELSE OR AN IF ON THE STACK
1590 GOSUB540:IF MID$(IN$,1,1)<>"I" AND
MID$(IN$,1,1)<>"E" THEN LN$=LN$+"*** ERROR
***":EF=0:RETURN
1595 REMSAVE THE STACKED IF OR ELSE IN THE
SYMBOL TABLE
1600 LC=1:IN=LN+10:GOSUB1180
1605 REMFORGET THE ENDIF
1610 LN=LN-10:EF=1
1620 RETURN
1630 REM - - - - - - - - - - WHILE STATEMENT -
- - - - - - - - - - - - - - - - - -
1635 REMA WHILE IS LIKE AN IF, SO STACK IT
1640
IN$=STR$(LN):MID$(IN$,1,1)="W":IN=0:GOSUB510
1650 RETURN
1660 REM - - - - - - - - - - WEND STATEMENT -
- - - - - - - - - - - - - - - - - -
1665 REMWEND MUST BE MATCHED BY A WHILE ON
THE STACK
1670 GOSUB540
1680 IF MID$(IN$,1,1)<>"W" THEN LN$=LN$+"***
ERROR ***":EF=0:RETURN
1685 REMSAVE THE WHILE IN THE SYMBOL TABLE
1690 LC=1:IN=LN+10:GOSUB1180:RETURN
2000 REM********** PROCESS TO OUTPUT FILE
**********
2005 REMINCREMENT THE CURRENT LINE NUMBER
2010 LN=LN+10:EF=0
2015 REMELIMINATE THE LINE NUMBER AND
LEADING BLANKS
2020 IN$=LN$:GOSUB800:GOSUB700:LN$=IN$
2035 REMELIMINATE COMMENTS
2040 IF MID$(LN$,1,2)="/*" THEN
LN=LN-10:RETURN
2045 REMIF THERE ARE INLINE COMMENTS,
PROCESS THEM OUT
2050 IF INSTR(LN$,"/*")<>0 THEN GOSUB2280
2055 REMIS THIS A PROCEDURE HEADING?
2060 IF INSTR(LN$,"PROCEDURE")<>0 AND
INSTR(LN$,"<")<>0 THEN GOSUB2320:GOTO2100
2065 REMSEARCH FOR KEYWORDS & PROCESS
THEM
2066 REMEF=1 IMPLIES A LINE TO THE FILE, EF=2
IMPLIES A LINE
2067 REMTHAT PROCESSES OUT COMPLETELY
2070 GOSUB2380:IF EF=1 THEN 2100 ELSE IF EF=2
THEN 2110
2075 REMLOOK FOR SYMBOLS TO PROCESS
2080 GOSUB2120
2085 REMTAKE THE FINAL LINE AND ELIMINATE
LEADING & TRAILING
2086 REMBLANKS
2090 IN$=NL$:GOSUB700:GOSUB750:NL$=IN$
2095 REMPUT THE LINE ON THE PRINTER AND THE
OUTPUT FILE
2096 REMYOU CAN EASILY COMMENT OUT THE
LPRINTS
2100 PRINT#2,LN;" ";NL$:LPRINT USING"#####
";LN;:LPRINTNL$
2110 RETURN
2120 REM - - - - - - - SCAN A LINE FOR
SYMBOLS - - - - - - - - - - -
2125 REMSAME SCANNING PROCEDURE AS BEFORE
BUT DIFFERENT
2126 REMACTION IN LINE 2160
2130 FLG=0:IF MID$(LN$,1,3)="REM" OR
MID$(LN$,1,1)="'" THEN NL$=LN$:RETURN
2140 NL$=""
2150 FOR I=1TOLEN(LN$):C$=MID$(LN$,I,1)
2160 IF FLG=2 AND C$=">" THEN
IN$=IN$+C$:GOSUB2230:IN$="":FLG=0:GOTO2210
2170 IF FLG=1 AND C$=">" THEN
```

```
IN$="":FLG=0:GOTO2210
2180 IF C$="<" THEN
LC=I:IN$=C$:FLG=1:GOTO2210
2190 IF FLG>0 THEN IN$=IN$+C$:FLG=2:GOTO2210
2200 NL$=NL$+C$
2210 NEXTI
2220 RETURN
2230 REM - - - - - - - - - REPLACE SYMBOL IN
LINE - - - - - - - - - - - - - -
2235 REMSEARCH FOR THE SYMBOL, IF IT ISN'T
2236 REMFOUND, THEN PRINT AN ERROR MESSAGE
2240 GOSUB1260:IF EF<=0 THEN LPRINT"ERROR -
NOT IN SYMBOL TABLE":RETURN
2245 REMIF IT'S AT THE BEGINNING OF THE LINE,
THEN IGNORE IT
2250 IF LC=1 THEN RETURN
2260 REMOTHERWISE ADD IT TO THE LINE BEING
BUILT
2270 RETURN
2280 REM - - - - - - - - - ELIMINATE IN-LINE
COMMENTS - - - - - - - - - - - -
2290 LC=INSTR(LN$,"/*")
2295 REM/* MARKS ALL COMMENTS TO BE DELETED
2300 LN$=MID$(LN$,1,LC-1)
2310 RETURN
2320 REM - - - - - - - - - PROCESS PROCEDURE
HEADER - - - - - - - - - - - - -
2325 REMSTART WITH THE RM$ MASK AND PUT THE
PROCEDURE LABEL
2326 REMIN THE APPROXIMATE MIDDLE OF THE LINE
2330 NL$=RM$
2340 L1=INSTR(LN$,"<"):L2=INSTR(LN$,">")
2350 IN$=MID$(LN$,L1,L2-L1+1)
2360 MID$(NL$,30)="PROCEDURE "+IN$
2370 RETURN
2380 REM - - - - - - - - - SEARCH FOR
KEYWORDS - - - - - - - - - - -
2390 FOR I=1TOKW:LC=INSTR(LN$,KW$(I))
2400 IF LC<>0 THEN 2430
2410 NEXTI
2420 RETURN
2430 REM - - - - - - - - FOUND A KEYWORD - -
- - - - - - - - - - - - -
2440 NK=I:EF=1
2445 REMPROCESS THE APPROPRIATE KEYWORD
2450 ON NK GOSUB
2620,2470,2540,2570,2650,2750
2460 RETURN
2470 REM - - - - - - - - IF STATEMENT - - - -
- - - - - - - - - - - - -
2475 REMFIND THE IF STATEMENT'S SYMBOL TABLE
ENTRY
2480 IN$=STR$(LN):MID$(IN$,1,1)="I":GOSUB1260
2485 REMIF IT ISN'T THERE, THIS MUST BE A
NON-BLOCK IF
2490 IF EF<=0 THEN EF=0:RETURN
2495 REMFIND THE CONDITION AND BUILD AN IF
STATEMENT
2496 REMAROUND IT
2500 L1=INSTR(LN$,"IF"):L2=INSTR(LN$,"THEN")
2510 NL$=MID$(LN$,L1+2,L2-L1-2)
2520 NL$="IF NOT("+NL$+") THEN"+STR$(LN(EF))
2530 EF=1:RETURN
2540 REM - - - - - - - - THEN STATEMENT - - -
- - - - - - - - - - - - -
2545 REMNO SPECIAL PROCESSING FOR THEN
2550 EF=0
2560 RETURN
2570 REM - - - - - - - - ELSE STATEMENT - - -
- - - - - - - - - - - - -
2575 REMFIND THE ELSE ENTRY IN THE SYMBOL TABLE
2580 IN$=STR$(LN):MID$(IN$,1,1)="E":GOSUB1260
2585 REMIF IT ISN'T THERE, THERE'S AN ERROR
2590 IF EF<=0 THEN NL$="*** ERROR
***":EF=0:RETURN
2595 REMREPLACE IT WITH A GOTO
2600 NL$="GOTO"+STR$(LN(EF))
2610 EF=1:RETURN
2620 REM - - - - - - - - ENDIF STATEMENT - - -
- - - - - - - - - - - -
2625 REMAN ENDIF IS SIMPLY IGNORED
2630 EF=2
2640 RETURN
2650 REM - - - - - - - - WHILE STATEMENT - - -
- - - - - - - - - - - -
2655 REMBUILD A LINE STARTING WITH THE
CONDITION FOR THE WHILE
2660 LC=INSTR(LN$,"WHILE"):NL$=MID$(LN$,LC+5)
2665 REMELIMINATE LEADING BLANKS
2670 IF MID$(NL$,1,1)=" " THEN
NL$=MID$(NL$,2):GOTO2670
2675 REMELIMINATE TRAILING BLANKS
2680 IF MID$(NL$,LEN(NL$),1)=" " THEN
NL$=MID$(NL$,1,LEN(NL$)-1): GOTO 2680
2685 REMBUILD AN IF STATEMENT BY NEGATING THE
CONDITION
2690 NL$="IF (NOT ("+NL$+")) THEN GOTO "
2695 REMSTACK THE WHILE SO WE CAN COME
BACK TO IT
2700 IN$="WHILE":IN=LN:GOSUB510
2705 REMLOOK FOR THE WHILE IN THE SYMBOL
TABLE
2710 IN$=STR$(LN):MID$(IN$,1,1)="W":GOSUB1260
2715 REMIF IT'S IN THE TABLE, FINISH THE IF
STATEMENT
2720 IF EF>0 THEN NL$=NL$+STR$(LN(EF))
2730 EF=1
2740 RETURN
2750 REM - - - - - - - - WEND STATEMENT -
2755 REMA WEND MUST BE MATCHED BY A WHILE
ON TOP OF THE STACK
2760 GOSUB540
2770 IF IN$<>"WHILE" THEN NL$="*** ERROR
***":RETURN
```

```
2775 REMBUILD A WEND INTO A GOTO THAT GOES
BACK TO THE WHILE
2780 NL$="GOTO "+STR$(IN): RETURN
```

Program Listing for TEST/STR

```
10 /**
20 /*
30 /*TEST PROGRAM FOR STRUCURED BASIC TRANSLATOR
40 /*BY TERRY R. DETTMANN
50 /*
60 /**
70 CLS
80 LINEINPUT"FILENAME ====> ";FF$
90 OPEN"I",1,FF$
100 WHILE NOT EOF(1)
110 LINEINPUT#1,LN$/* GET A LINE
120 GOSUB <ELIM-NUM>/* ELIMINATE THE LEADING
NUMBER
130 PRINT LN$/* PRINT IT ON SCREEN
140 WEND
150 CLOSE
160 END
170 /*
180 /*ELIMINATE NUMBER ROUTINE
190 /*
200 PROCEDURE <ELIM-NUM>
210 /*
220 N$="0123456789"/* COMPARISON STRING OF
NUMBER CHARS
230 <REPEAT> IF INSTR(N$,MID$(LN$,1,1))<>0 THEN
240 LN$=MID$(LN$,2)
250 GOTO <REPEAT>
260 ENDIF
270 RETURN
```

Program Listing for TEST/BAS

```
10 CLS
20 LINEINPUT"FILENAME ====> ";FF$
30 OPEN"I",1,FF$
40 IF (NOT (NOT EOF(1))) THEN GOTO 90
50 LINEINPUT#1,LN$
60 GOSUB 110
70 PRINT LN$
80 GOTO 40
90 CLOSE
100 END
110 REM - - - - - - - - - - - - - -PROCEDURE
<ELIM-NUM> - - - - - - - - - - - - - -
120 N$="0123456789"
130 IF NOT( INSTR(N$,MID$(LN$,1,1))<>0 ) THEN 170
140 LN$=MID$(LN$,2)
150 GOTO 130
170 RETURN ■
```

Pascal

The advantages of a new language

For all models

Mark E. Renne, Bozeman, MT

In this article, I'd like to answer three questions: Why learn another language? Why learn Pascal? Which Pascal compiler?

Let's tackle the first question, "Why learn another language?" Most of us like interpreter BASIC because it's very easy to learn and use. You simply turn the computer on, write the program, and run it. If there's an error, a message and line number are displayed. You then make corrections and run again. The language is very simple and straightforward. There is no set structure in BASIC; this is both good and bad. It's good because it's very easy to program in your "own style," it's bad because that lack of structure leads to very bizarre programming.

Interpreter BASIC is very slow compared to a compiled language. There is no way to protect standard interpreter code from potential software pirates. The varied programming styles lead to programs that are hard for other programmers to modify.

Don't misunderstand my critique of BASIC for criticism. I believe that interpreter BASIC is still the best first language for personal computer owners. I also believe that Pascal was hurt more than helped by the tremendous applause it received. One magazine dared to brag that no computer would be sold without Pascal as the "standard" language. Well, it's three years later and BASIC still towers far and above any other language. I know that type of "better than thou" attitude of Pascal users alienated me for quite a while.

I had decided to explore the world outside BASIC. I knew my first step was assembly language. Most users hear the call of speed and try their hands at assembly programming. I couldn't get the hang of it on my own, so I looked for something else. My search then led to a variety of compiler BASICs. I'll explain the difference between compilers and interpreters later. They certainly provided me with speed but they lacked transportability. In other words, if I wrote a program on my TRS-80 I couldn't give it to my friend who owns an Apple. Toward the end of the tunnel a light glimmered — Pascal!

All of your programming problems will not disappear with Pascal. Many people would have you believe that Pascal is completely standardized, I'm sorry to say that's not true. Each manufacturer wants you to buy *his* Pascal. By adding little "goodies" to his Pascal, he can claim his is better than brand X. There is, however, far more standardization in Pascal than any other language. I believe that's true because it's a fairly new language.

What's the difference between an interpreter and compiler? An interpreter simply means the computer must interpret every command it comes across. For example, in the simple program:

```
10 FOR X=1 TO 500

20 PRINT X

30 NEXT X
```

the computer must reduce each line from BASIC into binary code one line at a time. It has to do this five hundred times in this simple three-line program! It's not amazing that interpreters are slow, it's amazing that they aren't slower. A compiler interprets the program all at once. You write the program using an "editor" and then assemble or translate it into machine code. This means that the entire program is directly executable by the computer, thereby increasing speed greatly.

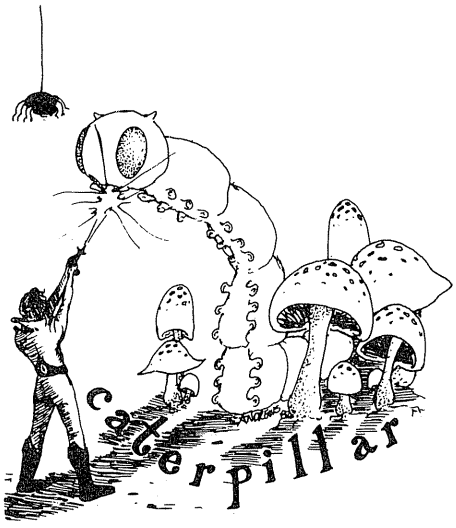
The biggest drawback to compilers is that they include several separate programs which confuse first-time users. You have to load in the editor, write the program, save the program, load the assembler, assemble the program, save the assembled code, and execute the code. If there's an error in the program, you reload the editor, edit the program, save the program, etc. This process is enough to intimidate most users. All Pascal programs are compiler languages and most operate in a similar manner to that described above. I'll mention one later that doesn't. This process is nice for authors, since you keep the source code and the user only gets the object code. Of course, this means you must update the program instead of allowing the user to.

Some confusion about Pascal arises because there is no one Pascal. Standard Pascal refers to the "stripped down" Pascal that was originally created to be a standard language among all computers. It incorporates all the functions needed for most programmers. The University of California at San Diego took this Pascal, added a bunch of procedures and

AARDVARK

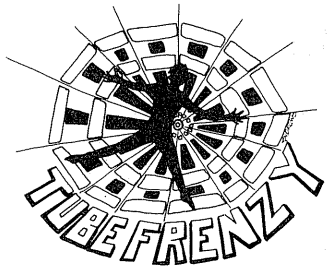
TRS-80 COLOR OSI VIC-64 VIC-20 SINCLAIR TIMEX

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 TRS-80, TRS-80 Color, and Sinclair. 13K VIC-20. $14.95 each.



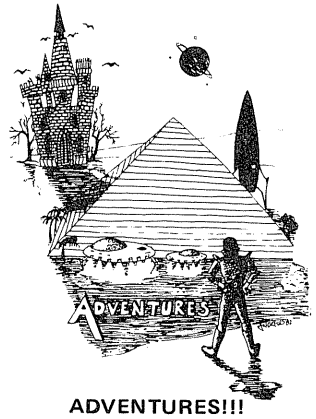
CATERPILLAR

O.K., the Caterpillar does look a lot like a Centipede. We have spiders, falling fleas, monsters traipsing across the screen, poison mushrooms, and a lot of other familiar stuff. COLOR 80 requires 16k and Joysticks. This is Edson's best game to date. $19.95 for TRS 80 COLOR.



TUBE FRENZY
(by Dave Edson)

This is an almost indescribably fast action arcade game. It has fast action, an all new concept in play, simple rules, and 63 levels of difficulty. All machine code, requires Joysticks. Another great game by Dave Edson. TRS 80 COLOR ONLY. 16k and Joysticks required. $19.95.



ADVENTURES!!!

These Adventures are written in BASIC, are full featured, fast action, full plotted adventures that take 30-50 hours to play. (Adventures are interactive 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 TRS80, TRS80 color, and Sinclair. They require 8k on OSI and 13k on Vic-20. Derelict takes 12k on OSI. $14.95 each.

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.

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)

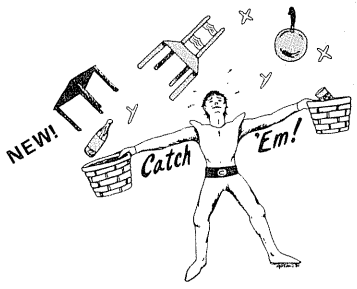
This is our most challenging ADVENTURE. It is a treasure hunt in a pyramid full of problems. Exciting and tough!

HAUNTED HOUSE (by Bob Anderson)

It's a real adventure – with ghosts and ghouls and goblins and treasures and problems – but it is for kids. Designed for the 8 to 12 year old population and those who haven't tried Adventure before and want to start out real easy.

DERELICT
(by Rodger Olsen & Bob Anderson)

New winner in the toughest adventure from Aardvark sweepstakes. This one takes place on an alien ship that has been deserted for a thousand years – and is still dangerous!



CATCH'EM
(by Dave Edson)

One of our simplest, fastest, funnest, all machine code arcade games. Raindrops and an incredibe variety of other things come falling down on your head. Use the Joysticks to Catch'em. It's a BALL! – and a flying saucer! – and a Flying Y!– and so on. TRS 80 COLOR. $19.95.

BASIC THAT ZOOOMMS!!
AT LAST AN AFFORDABLE COMPILER!

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, IF, THEN, RETURN, END, PRINT, STOP, USR (X), PEEK, POKE, *, /, +, -, >, <, =, VARIABLE NAMES A-Z, SUBSCRIPTED VARIABLES, and INTEGER NUMBERS FORM 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, TRS-80 Color, or VIC.

Please specify system on all orders

ALSO FROM AARDVARK – This is only a partial list of what we carry. We have a lot of other games (particularly for the TRS-80 Color and OSI), business programs, blank tapes and disks and hardware. Send $1.00 for our complete catalog.

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

Phone Orders Accepted 8:00 a.m. to 4:00 p.m. EST. Mon.-Fri.

VISA

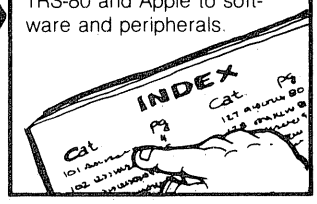
master charge

22

How to Buy or Sell Computer Equipment and Software

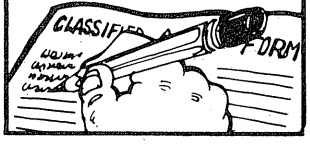
START HERE WITH...

Begin your search in the index of Computer Shopper's bargain filled pages. Locate the category and page number of items that interest you from TRS-80 and Apple to software and peripherals.



Start or add to your computer system by finding money saving bargains in each month's issue from individuals who no longer need their personal equipment.

As you outgrow your system or want to trade up (most dealers won't take trade-ins), use Computer Shopper ads to sell your items to 20,000 readers nationwide for the low cost of 12 cents per word.



If you need help with any computer related problem whether it's an interface problem or advice on the right peripheral for a home-brew system, use the free HELP column especially designed for that purpose.

You've got your computer hardware, but what about the software? Use a Computer Shopper ad to find what you need. Someone advertising in Computer Shopper probably has what you want.

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
23

functions, and created the UCSD version of Pascal. This version is the most popular among large computer users.

The advantages of Pascal are many. It is faster than interpreter BASIC because it is a compiled language. Pascal has a built-in structure that incorporates top-down programming. There are no line numbers in Pascal and "pretty printing" is easy to do. Pretty printing is the ability to indent lines to clearly identify sections of programs or to clearly identify loops. Pascal also allows significant variable names of any length. For example, "Accounts Receivable" and "Accounts Payable" would be two different names in Pascal, BASIC only considers the first two letters.

Pascal has a number of built-in control structures not found in BASIC. Some examples include: While...Do and Repeat...Until. These function in the same way as a FOR...NEXT but give you much more control. Let's say you want the user to input "99" to proceed, you could use the statement "Repeat...Until X=99." The ... could be any normal math function or INKEY function. By the way, Pascal gives no consideration to case of letters. Therefore, "WHILE," "while" and "wHiLe" are all treated as identical. The While . . . Do command is used to repeat a function "while" a certain statement is true or a certain value exists. For example, "While X<>99 Do" would repeat all steps from the next Begin to the next End until X equals 99. Both of these statements give you much more control for loops. The main difference between the two is one comes before a set of steps and the other one after.

The ability to design your own functions and procedures makes Pascal the ultimate versatile language. This means you can create a set of program steps that perform any function you desire. You can create a set of steps that calculates water velocity when fed three variables. By naming this procedure "WaterVelocity," you can call it from any part of your program by simply using the command X:=WaterVelocity(A,B,C), where A, B and C are your variables. The applications of this type of programming are endless.

Yes, I hear that question all the way here in Montana, "Why not just use a subroutine in BASIC?" Two answers: readability and ease of programming. Readability in that there is no question what the function does or what variables are used. Not even the most ardent BASIC fan would claim that GOSUB 5000 makes more sense than its Pascal equivalent. Ease of programming occurs because you write the procedure only once. After it's written it may be merged with your program and called at any time or it can be called from disk. Should the procedures change for calculating the equation, i.e., Sales Tax, it can be easily changed one time for even a series of programs.

By being a procedure-oriented language, Pascal programs are almost self-documenting. This makes it very popular among systems managers. Programs written by one programmer normally make perfect sense to the next programmer since all functions have names reflective of their nature.

Pascal also allows you to create your own data types freeing you from forcing your data into prearranged types. You might define the variable type "Daysoftheweek" to contain "Monday thru Friday" and Weekend" to contain "Saturday and Sunday." You could then compute salaries in your payroll program based on whether your employee worked weekdays or weekends. Certainly, the same type of operation could be done in another language, but not with the same ease.

You now have a basic idea why Pascal is growing in popularity, especially as a teaching language. Its structure creates good programming habits, it allows a modular program that is easily modified, and data structures are handled with ease.

Let's write a program in Pascal and compare it with a similar one in BASIC. Listing 1 is a Pascal program to average three numbers and listing two is the BASIC version. The first line in any Pascal program consists of the keyword "Program" followed by a name for your program. The name can be any name that begins with a letter and contains no spaces or punctuation. The next line is used to define the VARiables Sum, Average, A, B and C. In Pascal all variables must be defined at the beginning of your program. This adds to Pascal's favor among teachers and managers.

The actual program begins on the third line with the word "Begin" followed by no punctuation. See how straightforward Pascal is? The "Cls" command functions the same in both programs; this command may not be found in all versions of Pascal but it can always be accomplished one way or another. Since Pascal has no "Input" command you use the "Write" command combined with a "Read" statement. The Write command prints the information between the single quotes and then holds the cursor at its current position. This is followed by the Read which reads the three values into the three variables A, B and C.

Calculating the Sum and Average is almost the same in both languages. You might notice the slightly different equals sign in Pascal, the := is used in all equations as the = sign is used in BASIC. After the calculations it's time to output the results using the "WriteIn" command. This command prints the information between the single quotes, the value of the variable Sum, and then issues a carriage return. Notice the difference between the Write and WriteIn commands. The "End." tells Pascal this is the end of the program, note the period. This period distinguishes between an end of a procedure, which uses a semicolon, and the end of a program.

You might begin to see one of the disadvantages of Pascal by now. The Pascal program is eleven lines and the BASIC program is only six. (I didn't count the END in the BASIC program because it's really not needed in Microsoft BASIC; old habits die hard!) Yes, Pascal programs do require more lines than BASIC in even the most simple programs. Certainly, you would agree that the Pascal program is more readable and each section is separated by function. The longer the program, the more this is appreciated by the programmer who follows after

you.

Pascal has some other restrictions that may be bothersome to users. There is no string handling in standard Pascal; strings are made up of arrays of characters. Remember, you can create functions and procedures to handle strings. UCSD Pascal does include most string functions. In Pascal all array dimensions must be fixed. Some users may have become adapted to BASIC's variable dimensioning and must remember structure is Pascal's strong suit. It also takes far more time to master Pascal than BASIC, but it's worth it to the serious programmer.

If you've stayed with me this far and are interested in Pascal for your Model I or III, here is some information on available programs. First, there is not a UCSD Pascal available for either system (there is one for the Model II). It's simply a matter of memory, 48K isn't enough for both UCSD Pascal and a program of normal length. This means all versions for Model I or III are standard Pascals of one variety or another. Currently there are three companies offering Pascal programs: Radio Shack, New Classic Software and Alcor Systems.

The Radio Shack version is a tape-based tiny Pascal. Tiny Pascal is a subset of Pascal and in this case is a fixed point, non-array version. It sells for under $20, works in 16K, and is an excellent choice for the tape user or those not sure of their interest in Pascal. It is, of course, very limited in its applications and will require you to work only with tape.

On the opposite end of the software spectrum is the Alcor version. It sells for $324 complete or in modules as low as $199 for the Pascal system. I would not recommend this for the novice programmer. If you're familiar with both compilers and Pascal, this would be the system for you. It is strictly for programmers who need a professional development system. Alcor Systems' address is 800 West Garland Ave., Garland, TX 75040.

My personal choice for disk users who are interested in learning Pascal is the New Classic's version. It sells for $99 and is available from them at 239 Fox Hill Road, Denville, NJ 07834. This is an excellent teaching version of Pascal. It contains most of the functions of standard Pascal except pointer variables, variant records, with operator, and the associated new and dispose procedures. *(Pointer variables have now been implemented in Pascal-80. —Ed.)* Some extras have been added to enhance this version including cls, peek, poke, close, inkey, seek, ex, and fp. It uses both random and sequential files and has 14-digit precision.

The biggest advantage of the New Classic's version is that it is an interactive compiler. This is to say that both the editor and compiler are resident at the same time. By doing this, New Classic has eliminated the biggest confusion associated with compilers in general. A menu is displayed at the beginning allowing you to enter the editor or compile. You write the program using the editor and then compile the program without having to load any other program. This saves many steps if you have

an error since you don't have to keep loading the editor or compiler. The drawback is that this takes up more memory but you have enough room for a 23K program. Programs can be very easily chained also.

Documentation explains only the features and commands of this particular Pascal. If you want to learn Pascal, you'll also need to purchase a good Pascal book; several are listed in the manual. A number of programs are also included — one to create ASCII files of your programs so they may be transferred to other computers, one to create TRSDOS executable files from your Pascal files, one to create Set and Reset commands, one to create Pascal files from ASCII files from other computers, and some others to illustrate various commands.

I highly recommend this Pascal for the disk user who is interested in Pascal but not yet ready for a sizeable financial commitment. It has all of the features you'll need for quite a while and is reasonably priced. Since it's interactive, it makes a compiler just as easy as an interpreter — the best of both worlds!

Learning a new language is an experience filled with challenge and rewards. Being multi-lingual has many advantages in the "real world." Pascal is becoming very popular with many institutions and most people would benefit from an aquaintance with it. Do yourself a favor and explore the world beyond BASIC. After all, didn't you buy a computer to expand your world?

Listing 1

```
Program Average:
Var Sum, Average, A, B, C : Real;

Begin
  Cls;

    Write('Input three numbers (A,B,C)';
    Read(A,B,C);

      Sum := A+B+C;
      Average :=Sum / 3;

    WriteIn('Sum =',Sum);
    WriteIn('Average =',Average);

End.
```

Listing 2

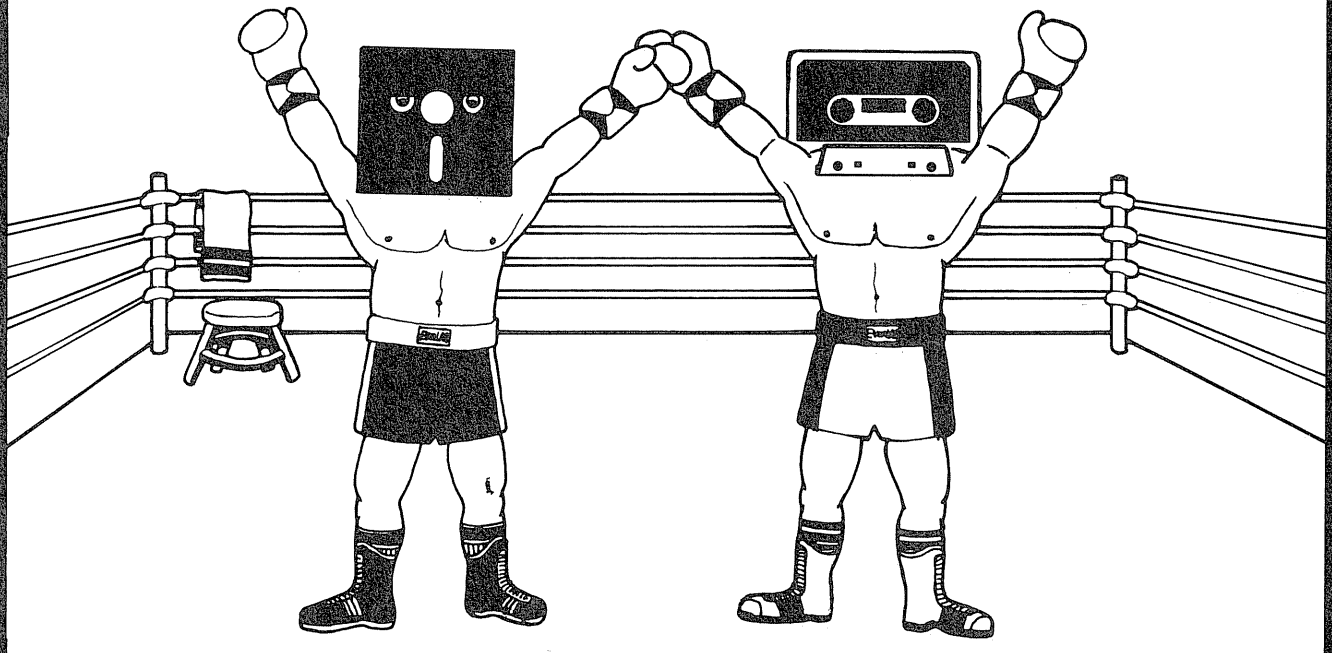
```
10 CLS
20 INPUT "Input three numbers (A,B,C)";A,B,C
30 SUM = A + B + C
40 AVG = SUM / 3
50 PRINT "Sum =";SUM
60 PRINT "Average =";AVG
70 END ■
```

TRS-80
Model I

And In This Corner, Another Champion.....

TRS-80
Model III

CLOAD'S DISK VERSION!



Get 12 tapes OR disks a year containing over 75 quality programs — without knocking out your budget!

We finally gave in . . . After over FOUR years of publishing some of the best (aren't we pompous), useful and fun programs on tape, we are now offering **CLOAD Magazine on disk,** too!

A subscription to **CLOAD Magazine** (tape or disk) consists of 6 to 8 ready-to-load programs delivered by First Class Mail every month. What a winner!

Fight high software prices — Get a subscription to **CLOAD Magazine.** Or just throw a single punch and try a back issue . . .

The Bottom Line:

	Tape	Disk
1 year (12 issues)	$50.00	$95.00
6 months (6 issues)	$30.00	$55.00
Single Copies		
Back Issues	$ 6.00	$11.00
Good Games #1	$12.00	$23.00
Adventures #1	$13.00	$24.00

California residents add 6% to single copies
North America — First Class postage included
Overseas — add $10 to subscriptions and $1 to single copies Sent AO rate

CLOAD MAGAZINE INC.

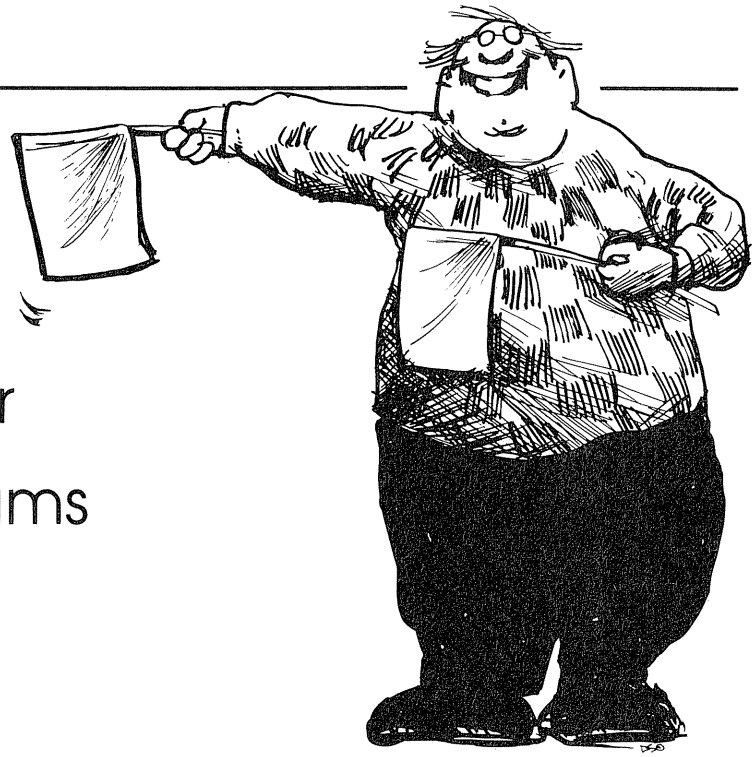
(805) 962-6271

P.O. Box 1448
Santa Barbara, CA
93102

MasterCard/Visa

The Fine Print:
All issues from Oct 78 available on tape Issues from Oct 82 available on disk, also Ask for list (24 Level I issues also available)

27



Crypto

Have your computer help solve cryptograms

Models I/III, PMC-80, LNW80 with disks

Tim Chandler, Springfield, VA

Have you ever tried to work the letter-substitution cryptograms in the Sunday supplement or in a puzzle book? In these puzzles, one letter has been substituted for another wherever it appears in a short phrase or sentence, to produce seemingly meaningless gibberish which you must decode back into plain English. Working by trial and error with pencil and paper, you write and erase, write and erase again, until your worksheet is truly an indecipherable mess! The purely mechanical tasks involved tend to get in the way of producing a solution. I was sure there had to be an easier way, and with that thought held firmly in mind, I set out to write this program which I hope you will find as useful and effective as I have found it to be.

The program, written in Disk BASIC for the Model I or III and running under TRSDOS, NEWDOS, or DOSPLUS, will put your TRS-80 in charge of all the mechanical tasks, freeing you to do what humans do best: think! Once you have used this program, you will never want to go back to pencil and paper to solve a cryptogram again. You can even easily make up your own cryptograms in the twinkling of an eye, and save and load them to and from disk in a matter of seconds. Even a novice like me can solve the most difficult of cryptograms, usually at one sitting. I never even liked cryptograms until I started using this program. I originally wrote it for my father, a former Signal Corps officer and puzzle addict, and over several years, the program has evolved.

The operation of the program is relatively straightforward and largely self-prompting. The start-up screen enumerates the commands available to the prospective code breaker. <ENTER> allows the original cryptogram to be typed in, or if one is already in memory, <ENTER> returns to it. <CLEAR> completely erases the workspace above the cryptogram, useful in starting over from scratch. The uparrow saves the workspace to disk, useful if you wish to save the solution to a hard cryptogram, or if you make up your own. The downarrow saves the original cryptogram to disk. The rightarrow will display the frequency of occurrence of each letter in the cryptogram, along with a table of the normal frequency of occurrence for letters in the English language, according to a table in *The Way Things Work Book of the Computer*, published by Simon and Schuster (see Figure 1). This table is a little different from the ETAOIN SHRDLU learned by a printer's devil, but the differences are minor. This table can be useful when trying to decide which letter to substitute for another, especially in the longer cryptograms. The leftarrow returns to the instructions.

At the bottom of the start-up screen, you are asked if you want to retrieve a cryptogram from disk (D), clear any previous cryptograms from memory (C), or <ENTER> to enter a new cryptogram (or return to one in memory, if one exists). The INKEY$ function is used here. Pressing "D" for disk retrieval sends you to a subroutine that gives a directory of all programs with the appendage, "/CRY", on drive zero (see Figure 2). This subroutine is also called when saving a cryptogram or its workspace. When saving or retrieving a file from disk, you will be asked to supply a filename. It is not necessary to type the extension, "/CRY." The program adds it to the filename in lines 510 and 600, which also check for just <ENTER> being hit when asked for a name. In this case, control is returned from the subroutine to the calling procedure so no error will result.

If, instead, a "C" is pressed to clear memory, the program is RUN again, effectively wiping out any resident cryptogram. If a cryptogram is already in memory when <ENTER> is pressed, you are returned to it, otherwise you are asked to enter your cryptogram. Pressing <ENTER> with no cryptogram entered returns you to the instructions.

Once a cryptogram is in memory, a character frequency count is performed using the subroutine found in Lines 670-790. A running count of the number of characters left to be checked is displayed, just to give you something to watch while you wait. Each character in the cryptogram has 64 subtracted from its ASCII value to give a number from 1 to 26, and then 1 is added

ABSOLUTELY RELENTLESS

The challenge of inner space — the fury of an enemy that seemingly will not die. This is SEA DRAGON — a battle to the death **under** the high seas! Slide into the Captain's chair, take the controls and prepare yourself for the most incredible non-stop action this side of Davy Jones' locker. SEA DRAGON puts you in control of a nuclear sub that's armed from stem to stern with enough firepower to take on King Neptune himself — and you'll need every missile, every torpedo, and every scrap of skill you can muster to survive.

The object of SEA DRAGON is to successfully navigate your sub through an underwater course past mountains and through labyrinthine passageways while avoiding clusters of explosive mines that rise from the seabottom. But the danger doesn't stop there — overhead, surface destroyers lace the water with depth charges; below, deadly attack bases and arcing lasers cut a killing swath that could reduce your sub to bubbling slag. But even these potentially lethal perils are dwarfed by the awesome menace that awaits you at the course's end.

SEA DRAGON — every possible "extra" is here to ensure your playing pleasure: exciting sounds, high score save, machine language graphics and an eye-popping scrolling seascape that extends the equivalent of over two dozen screens placed end-to-end, providing a diverse and unique challenge that will not diminish after repeated playings.

Nothing you've ever seen on your micro could possibly prepare you for this! You are ready now, ready for the ultimate in undersea action with a pace that is absoutely unyielding. SEA DRAGON — the arcade has finally come home.

FOR THE APPLE by John Anderson

FOR THE ATARI by Russ Wetmore

FOR THE TRS-80 by Wayne Westmoreland & Terry Gilman

ALL VERSIONS ARE JOYSTICK COMPATIBLE!

ORDERING INFORMATION

APPLE 2 or APPLE 2 PLUS 48K Disk (DOS 3.3 required)
042-0146 ..... $34.95

ATARI 32K Disk
052-0146 ..... $34.95

ATARI 16K Tape
051-0146 ..... $34.95

TRS-80 32K Disk
012-0146 ..... $24.95

TRS-80 16K Tape
010-0146 ..... $24.95

SEA DRAGON FEATURES

- Fantastic Scrolling Seascape
- Nearly Limitless Game Challenge
- High Score Save (disk version)
- Terrific Sound Effects
- Arcade Action Graphics™
- Apple version "talks" without special hardware!

Adventure INTERNATIONAL
A DIVISION OF SCOTT ADAMS, INC

PRICES SUBJECT TO CHANGE

To order, see your local dealer. If he does not have the program, then call 1-800-327-7172 (orders only please) or write for our free catalog.

Published by ADVENTURE INTERNATIONAL
a subsidiary of Scott Adams, Inc.
BOX 3435 • LONGWOOD, FL 32750 • (305) 830-8194

28

Figure 1

```
        Character Frequency

A  =  3      B  =  4      C  =  0      D  =  0
E  =  6      F  =  0      G  =  6      H  =  2
I  =  3      J  =  0      K  =  0      L  =  5
M  =  0      N  =  0      O  =  2      P  =  6
Q  =  0      R  =  3      S  =  0      T  =  8
U  =  4      V  =  0      W  =  3      X  =  0
Y  =  4      Z  =  2

Normal Frequency of Occurrence
  E   T   O   A   N   I   R   S   H   D   L   C   F
  U   M   P   Y   W   G   B   V   K   X   J   Q   Z

Press <any key> for Cryptogram
```

Figure 2

```
                Cryptograms Available

    JAN31          OCT4          JAN17          MAY2
    AUG23          AUG30         FEB21          JUN6
    NOV22          SEP20         DEC13          JUN13
    AUG16          OCT25         MAY16

Name of File to be Retrieved (C/R to continue)?
```

Figure 3

```
  I S E N     A N T E       A T R I O T     R O T E     I N
L X P A Z I M T Z B A L   U T B O X V B   N O V B A   X Z

" T  O  A S    A I N E "     E N    E    E N T   T O
" B M V R T P  U T X Z A "   N M A Z   M A   N A Z B   B V

T  E    O    S  A T  E  E  T I O N   T I  E .
B M A  U V E E P  T B  A E A I B X V Z  B X R A .

Change letter?
```

Table 1

```
Some cryptograms for your
solving pleasure:

TYRSKZH DAFELYHABIT BZO YLXNA
EMPPRNT IBF UN BOOKDLKSN, UML
KL OYNTZ'L OBIBHN LXN UABKZ.

ARC STIELEBH UOYIOLEE POUCH OY
BYBHBOXXW XTSCXW STHTAEL TY OYW
REPC.

SEWSEWNO CAVEM SEWSDW BIERDZ
PMONRNYNUDZ OPBKAVDWB CNKI AYZ
BKAMD EZUD.

BIG TAX JOBX RANK PANTXKR PBT EA
ZAK RAN; BIG JOBX RAN PBT EA ZAK
RANK PANTXKR.   - C.Z. GYTTYER

BMZZT FCWWT WDCGH UNXBI TPCLM
UHRNX MFKWZ GCMJC DCFMI TKKIC
RYTKM GRNFJ STKBR NKQCK KTFQW
RUCRF HRNGW CIY.
```

The author did not include a list of the solutions for the cryptograms. Sorry, but you are on your own. —Ed.

to the correspondingly-numbered array element, C(). A possible source of error here is if lowercase letters are used in the cryptogram, giving a number greater than 26. Use UPPER CASE ONLY (or write a routine to accept either upper- or lowercase). Then, any numbers or punctuation with ASCII values less than 65 are automatically placed in the workspace string, which is a string just as long as the cryptogram. The cryptogram is displayed in lines 230 to 330, with the workspace string appearing on the line just above the cryptogram string. Line 220 selects either the 32-character mode for shorter cryptograms, or the 64-character display for cryptograms longer than 160 characters. The longest a cryptogram may be is restricted to the maximum legal string length minus 2 (253 characters). Line 230 starts the block of code that actually displays the cryptogram and the workspace. PRINT@ and MID$ functions are used so that everything appears in the proper place, and with the proper spacing. Figure 3 shows a cryptogram whose solution is in progress.

Solving the Cryptogram

At the bottom of the screen, the prompt "Change letter?" will be shown. Since this also utilizes the INKEY$ function, just press the letter you wish to change. This is also the prompt at which the four arrow keys and the <CLEAR> key may be used, as outlined above and in the start-up screen instructions. Lines 340 to 390 test for the arrows, <CLEAR>, or a letter. If a letter is entered, the prompt "To?" will appear. Press the letter you want to change the original letter to. To erase a letter, simply change it to a space, using the spacebar. Lines 430 to 450 search the cryptogram for the original letter, and replace the workspace above that letter with the new letter when a match is found. The replacement may take a little time — in the case of a long cryptogram, perhaps as much as several seconds.

Other Operating Systems

Those of you who are not blessed with the DOSPLUS operating system will have to make some changes to the subroutine that reads the disk directory, starting at line 910. For NEWDOS, change line 920 to read: 920 CMD"DIR" to get a listing of the directory from BASIC, and just delete lines 930 to 1090. Under DOSPLUS, this subroutine opens DIR/SYS and picks out all the non-invisible files with the extension "/CRY", and formats them neatly in four columns, but without displaying the extension. With some experimentation, and armed with *TRS-80 Disk and Other Mysteries*, by H. C. Pennington,

you could probably write a similar short routine for NEWDOS and maybe even for TRSDOS.

Strategies

I don't know if there is any one general strategy to employ that works all of the time. Rather, I usually use a combination of strategies — looking for patterns of letters in words, substituting letters using the knowledge of the frequencies of occurrence, and sometimes just plain brute force, trying something to see if it will fit. The speed and ease with which these strategies can be implemented makes solving almost any cryptogram a breeze. My biggest problem now is finding enough cryptograms to keep me and my father busy!

I have included a few cryptograms that you may wish to try to solve. The hardest ones are those broken up into five-letter code groups. In these, the frequency of occurrence of the letters must be relied upon heavily. The others range in difficulty from easy to hard. A good source for cryptograms is often the Sunday newspaper. Another place to find cryptograms is in the "Saturday Review." These are usually literary in nature, and somewhat longer. Another good source of cryptograms and puzzles is *Games* magazine, published bimonthly by Playboy Enterprises. You can also find many cryptogram and puzzle books at your local drugstores and bookstores.

Program Listing for Cryptograms

```
100 'CRYPTOGRAM SOLVING AID
102 ' BY
104 ' TIM CHANDLER
110 CLEAR2000 :DEFINTA-Z:DIMC(26):F$="## "
120 DIMA$(100):X$="":C$="":A=0
130 CLS:PRINT"C R Y P T
O":PRINTSTRING$(64,95):PRINT"<ENTER> enters
cryptogram, or returns to one in
memory.":PRINT:PRINT"<CLEAR> clears the
workspace above the cryptogram.":PRINT
140 PRINT"<UP Arrow> saves WORKSPACE to Disk
(without CRYPTOGRAM).":PRINT:PRINT"<DOWN
Arrow> saves CRYPTOGRAM to disk (without
WORKSPACE).":PRINT: PRINT"<RIGHT Arrow>
displays Character Frequency.":PRINT:PRINT"<LEFT
Arrow> displays these instructions."
150 PRINT:PRINT"<D>isk retrieval, <C>lear
memory, or <ENTER>?";
160 YN$=INKEY$:IF YN$=""THEN 160
170 IFYN$=CHR$(13)THEN190
180 IF YN$="D"THEN GOSUB470 ELSE
IFYN$="C"THENRUNELSE130
```

```
190 IFA>0THEN220
200 CLS:PRINTCHR$(23)"Enter
Cryptogram:":PRINT:LINEINPUTX$
210 A=LEN(X$):IFA=0THEN RUNELSE
C$=STRING$(A,32):GOSUB670
220 L=32:K=33:IFA>160THENL=64:K=65
230 CLS:IFL=32THEN PRINTCHR$(23);
240 PRINT@0,LEFT$(C$,L); :PRINT@64,LEFT$(X$,L);
250 IFA<=LTHEN330
260 PRINT@192,MID$(C$,K,L);
:PRINT@256,MID$(X$,K,L);
270 IFA<=2*LTHEN330
280 PRINT@384,MID$(C$,2*K-1,L);
:PRINT@448,MID$(X$,2*K-1,L);
290 IFA<=3*LTHEN330
300 PRINT@576,MID$(C$,3*K-2,L);
:PRINT@640,MID$(X$,3*K-2,L);
310 IFA<=4*LTHEN330
320 PRINT@768,RIGHT$(C$,A-4*L);
:PRINT@832,RIGHT$(X$,A-4*L);
330 PRINT@960,CHR$(30);:PRINT@960,"Change
letter? ";
340 A$=INKEY$:IFA$=""THEN340ELSE
IFA$=CHR$(31)THENC$=D$:GOTO240
350 IFA$=CHR$(91)THEN M$="Save
WORKSPACE":GOSUB640:GOTO230
360 IFA$=CHR$(9)THEN GOSUB800:GOTO230
370 IFA$=CHR$(10)THENM$="Save
CRYPTOGRAM":GOSUB560:GOTO230
380 IFA$=CHR$(8)THEN130
390 IFASC(A$)<32THEN330ELSEPRINTA$;
400 PRINT@1000,"To? ";
410 B$=INKEY$:IFB$=""THEN410
420 IFASC(B$)<32THEN400ELSEPRINTB$;
430 FORI=1TOA
440 IFA$=MID$(X$,I,1) THENMID$(C$,I,1)=B$
450 NEXT
460 GOTO240
470 ' Disk Retrieval of Cryptograms
480 CLS:GOSUB920
500 CN$="":PRINT:INPUT"Name of File to be
Retrieved (C/R to continue)";CN$
510 IFCN$=""THEN550ELSECN$=CN$+"/CRY"
520 OPEN"R",1,CN$:FIELD1,2 AS LN$,253 AS
Y$:GET1,1:CLOSE
530 A=CVI(LN$):X$=LEFT$(Y$,A)
540 IFA>0THENC$=STRING$(A,32)
:GOSUB670ELSEKILLCN$
550 RETURN
560 ' Save CRYPTOGRAM to disk
580 CLS:GOSUB 920:PRINTM$
590 CN$="":PRINT:INPUT"Name of File to be Saved
(C/R to continue)";CN$
600 IFCN$=""THEN630ELSECN$=CN$+"/CRY"
610 OPEN"R",1,CN$:FIELD1,2 AS LN$,253 AS Y$
620 LSETLN$=MKI$(A) :LSETY$=X$:PUT1,1:CLOSE
```

```
630 RETURN
640 ' Save WORKSPACE to disk
650 Z$=X$:X$=C$:GOSUB560:X$=Z$
660 RETURN
670 ' Count Character Frequency
680 CLS:PRINTCHR$(23)TAB(2)"Character
Frequency Count":PRINTTAB(8)"In Progress"
690 PRINT:PRINTTAB(2)"Characters Remaining =";
700 FORI=1TO26:C(I)=0:NEXT
710 FORI=1TOA
720 PRINT@240,CHR$(30);A-I
730 J=ASC(MID$(X$,I,1)) :IFJ>64 THENJ=J-64
:C(J)=C(J)+1
740 NEXT
750 PRINT:PRINTTAB(2)"Frequency Count
Completed":PRINTTAB(6)"Please Stand By...."
760 FORI=1TOA
770 IFASC(MID$(X$,I,1))<65 THEN
MID$(C$,I,1)=MID$(X$,I,1)
780 NEXT:D$=C$
790 RETURN
800 ' Display Character Frequency
810 CLS :PRINTCHR$(23);TAB(4)"Character
Frequency" :PRINT
820 FORI=1TO26
830 PRINTCHR$(I+64)" =";:PRINTUSINGF$;C(I);
850 NEXT
860 PRINT@640,"Normal Frequency of Occurrence:
870 PRINT" E T O A N I R S H D L C F U M P Y W G B
V K X J Q Z
880 PRINT@960,"Press <any key> for Cryptogram";
890 A$=INKEY$:IFA$=""THEN890
900 RETURN
910 ' Read disk directory
920 OPEN"R",1,"DIR/SYS"
930 FORJ=1TO8
940 FIELD 1, (J-1)*32 AS DUMMY$,1 AS AT$(J),4 AS
DUMMY$, 8 AS NA$(J),3 AS EXT$(J),16 AS DUMMY$
950 NEXT
960 A$="Cryptograms Available"
:PRINTTAB(32-LEN(A$)/2)A$:PRINT
970 I=1:N=3:R=0
980 GET 1,N
990 FORJ=1TO8
1000 IFASC(AT$(J))<15OR
ASC(AT$(J))>18THEN1060
1010 IF NA$(J)=" " OR NA$(J)="" THEN 1060
1020 IF EXT$(J)="CRY" THEN A$(I)=NA$(J) ELSE
1060
1030 PRINTTAB(16*R+5)A$(I); :IFR=3PRINT
1040 R=R+1:IFR>3 R=0
1050 I=I+1
1060 NEXT
1070 N=N+1:IFN>18THEN1090ELSE980
1090 CLOSE:PRINT
1100 RETURN ■
```

The BASIC/S Compiler

An evaluation of this package from Powersoft

Models I/III, PMC-80, LNW80

Pete Carr, Port Orange, FL

I have a utility program MX-80 users will probably appreciate. It works like this: At DOS Ready, with my printer ready to go, I type the command MX80 DBL ON <enter>. That puts my printer into the doublestrike mode. I decide that I would like my printout to be even darker, so I type MX80 EMP ON <enter>. Now, I'm in the doublestrike and emphasized modes. It's easy to turn a mode off. Just type MX80 EMP OFF. I really get a lot of use out of this program. It lets me set or reset my printer to any of its available modes without having to remember the CHR$ codes! Before I wrote it, I would always have to get out the manual to see what codes I needed for setting emphasized, doublestrike, condensed, print modes. Now, it's easy. If I want to print in the condensed mode, I just type MX80 CON ON <ENTER>.

That nifty machine language program is one that I wrote! Well, not exactly. BASIC/S and I wrote it. I wrote it in BASIC, and BASIC/S transformed it into a machine language CMD file that I can execute directly from DOS READY. BASIC/S is a BASIC compiler written for Model I/III disk systems. It works with all DOSs, but seems to be especially suited for LDOS. For writing utility programs I have found no other BASIC compiler that offers as many of the right features as BASIC/S. Right is the keyword here. Of course, other compilers like BASCOM, RSBASIC, are great for long business programs, but even those big guys can't touch BASIC/S in its areas of expertise!

I like things to operate fast, and I love using disk operating systems. They allow so many possibilities for a computer user. Obviously, being a disk user, I love disk utility programs. There always seems to be some special function that I need, but no program to do it for me! With BASIC/S, I now have the ability to create those special CMD programs. Here is another example of putting BASIC/S to a good use. If you are an LDOS user, you have probably seen Tim Mann's program called BINHEX in the LDOS newsletter. It converts a binary machine language program to an ASCII hex file that can be transmitted over a modem to CompuServe, Micronet, etc. It also allows you to convert the file back to its original binary CMD format. BINHEX is written in BASIC and it does a great job, but it is a little slow. While it is running, the disk drives will turn on and off waiting for BINHEX to catch up. I compiled BINHEX with BASIC/S and it runs much faster. Now the disk drives don't have time to turn off while BINHEX is running! To me, this speed difference transformed a good program into a great one.

Imagine using 18K of disk space for a program whose only job is to set your MX-80 to the print mode of your choice. It would be ridiculous. If I had used one of the more elaborate compilers, like BASCOM or RSBASIC, to compile my MX80/CMD program, that's what would have happened. Why? Those bigger compilers have to be used in conjunction with runtime modules. The BASCOM runtime module takes about 15K of disk space by itself. The RSBASIC runtime modules take about 20K. My whole MX-80/CMD program is less than 3K long. The bigger compilers were just not made for creating those small-sized DOS utility programs. Also, what if I wanted to give someone a copy of my new program? If it was compiled with BASCOM or RSBASIC, that person would have to own one of those himself, or he wouldn't be able to use it! The runtime modules are copyrighted, and you can't just give them away without paying for that right.

BASIC/S does not depend upon runtime modules, or linking loaders of any sort. After you compile your program, it is a true, stand alone, machine language CMD file! The CMD files created by BASIC/S are very reasonable in size. They are usually about one and a half the size of their BASIC counterpart. But, there's always a tradeoff somewhere. BASCOM and RSBASIC can both do functions that BASIC/S can't. BASIC/S could not handle a big business program nearly as well, if at all. It's not its forte. BASCOM or RSBASIC reign supreme for those purposes.

The BASIC/S system consists of two compilers. One is written in BASIC, named BASIC/S, the other is a CMD file named BASIC/S II. The BASIC version runs slower, but offers floating point math. The CMD version runs much faster, and in its stock form, allows use of integer

math only. You can use limited floating point with BASIC/S I, but you have to use a special module that is supplied on the BASIC/S II disk. For writing utility programs, which is mostly what I do with BASIC/S, integer math is all I need.

You won't be able to compile just any off-the-shelf BASIC program with BASIC/S, and will be disappointed if you approach it that way. It should be used as a development tool, and not a magic box that will make all your existing BASIC programs run faster. It uses a little different syntax for certain functions, and doesn't support the full set of Microsoft BASIC. Hence, the name BASIC/S, the "S" meaning subset. On the other hand, BASIC/S offers functions that aren't standard in the full-blown Microsoft BASIC. LDOS has a function called SET EOF. So does BASIC/S. BASIC/S also allows chaining from one program to another, keeping all variables intact. Other useful functions are: HEX$, which takes a number and converts it to its hexadecimal equivalent. HEX$ would be very helpful if you wanted to write a disk zap-type program that displayed data in the usual hexadecimal format! SCAN allows you to read a file or device, a byte at a time. This device-scanning function is perfectly suited for LDOS. You are also allowed to use CMD "DOS Command" and return back to your BASIC/S compiled program for continuation.

A very big plus for BASIC/S is block file manipulation. If I want to open a file, using a record length of thirty-two bytes, BASIC/S will let me do it. I have another very good compiler called ZBASIC. It has some real great features, some I wish BASIC/S had, but it doesn't allow you to open a file using less than 256 bytes. To me, this makes ZBASIC very hard to use. Assume I wrote a program, whose purpose was to read the disk directory and give a printout of where each program resided on the disk. It would make my job so much easier using a record length of thirty-two bytes because each file in the disk directory, and its associated data, make up a block exactly thirty-two bytes in length. It could be done with ZBASIC, but it would be much more work.

BASIC/S comes on three disks, with numerous example programs. These example programs are full of great routines that let you get the most out of the BASIC/S system. One often used routine is called GETPARM. This simulates the DOS call GETPARM (get parameters). Example: MX80/CMD DBL ON. The parameters here are DBL and ON. The documentation explains all of its features and how to best use them. The latest version of BASIC/S has an added feature called INPUT@. INPUT@ is a controlled screen input routine, that allows you to replace those cumbersome INKEY$ data input routines.

Example: INPUT@124,"Enter Your Name ",20,"$*";NA$

This would print the prompt at video location 124. To the right of the prompt would be an underline 20 characters long, showing you where, and how much room, you have for data entry. The "$*" means you are allowed to enter alphanumeric characters; and if you fill the whole 20 byte field, you will automatically be returned to the next line without having to press Enter. If the $ was replaced with a # sign, you would be able to enter only numerals into that field. INPUT@ makes it easy to write "Fill in the form" programs, with a minimum of effort.

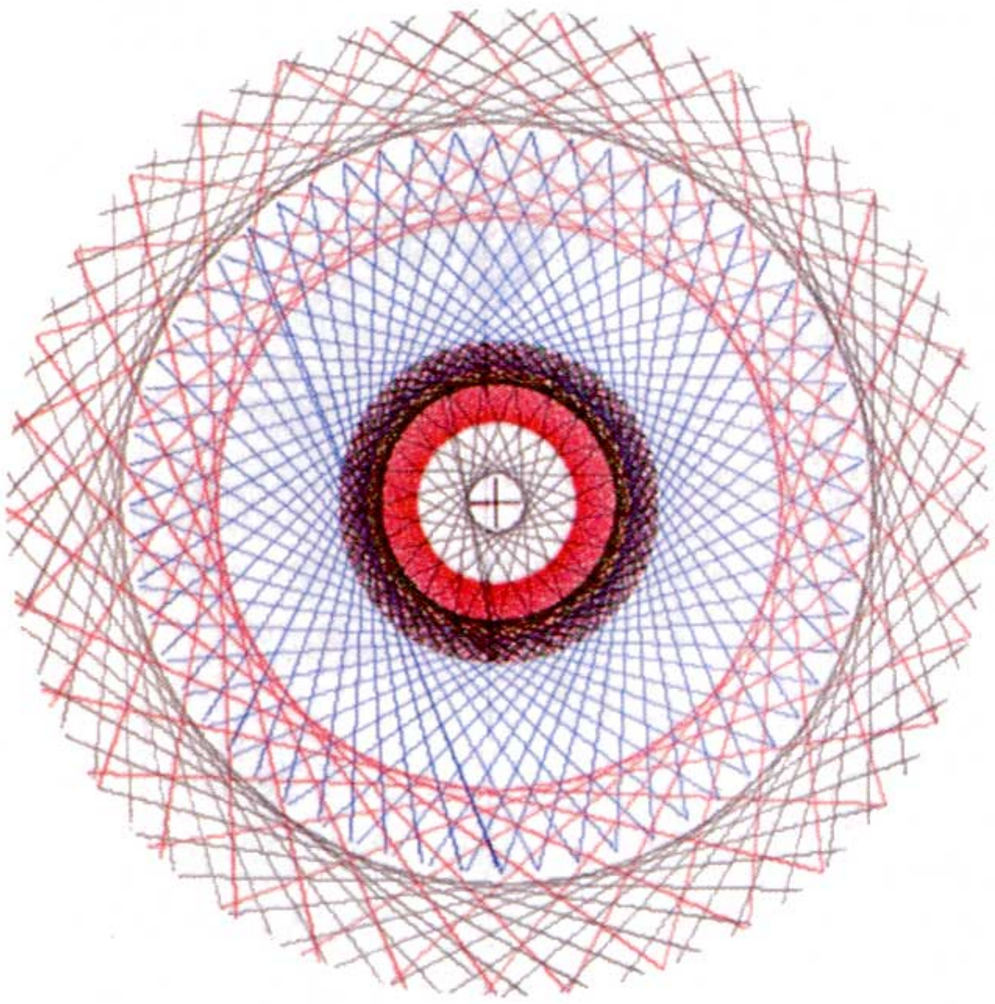
I can't say it's the best, or most powerful compiler available, but I have found it to be best suited for much of the programming I do. There is no tool made for every job. If you have a TRS-80 disk system and are not versed at machine language programming, than BASIC/S will allow you to write many machine language programs that you otherwise couldn't have done. That is exactly why I bought BASIC/S. I know enough about machine language to write and modify certain CMD programs, but I am not anywhere near as versed in machine language as I am with BASIC. Now that I have BASIC/S, I don't have to be. If I could be Kim Watt, or Roy Soltoff for about a week, maybe all that would change. But at this time I can't, so you won't be seeing a Super Utility+, or an LDOS come off my assembler! But in the mean time . . . ■

CGP-115 Printer

A review of this low cost printer-plotter from Radio Shack and a program to help use it

For all models

Jerry Latham, Midwest City, OK



I recently purchased the CGP-115 and it has not ceased to impress me with its abilities, precision, and durability. It has features comparable to units costing more than 5 times its price of $249.95, and it holds its own with Tandy's $1900+ plotter.

The CGP-115 is supplied with one roll of paper, 5 pens (2 black, and one each red, blue, and green), and a 45 page operation manual. No connecting cable is supplied. If you already have a Radio Shack computer or terminal with a printer you may use your present cable to connect the CGP-115 to it. If you don't have a printer you have to get one. The CGP-115 could actually be hooked up to any computer and there is a chart that shows which Radio Shack cable should be used to connect it to each of the different Radio Shack computers including the DT-1 terminal.

The CGP-115 has easily accessible DIP switches that allow you to choose serial or parallel I/O, 80 or 40 characters per line, specify a carriage return or a carriage return with a linefeed, and what type of characters will be used.

The CGP-115 uses 4 1/2 inch by 150 foot roll paper that is sold at $4.95 for a box of three. The ink pens sell for $2.95 for three.

The CGP-115 is capable of drawing lines in one of four selectable colors. The pens are mounted in a revolving holder, similar to the cylinder of a pistol and they are easy to change. Simply sending it a control code 29 will select the next color in the holder, while sending it a "Cn", where n is a value from 0 to 3, will cause the CGP-115 to pick a specific color - depending on how you have them arranged in the holder.

The instructions in the manual are clear and simple to follow. You can have your unit up and running in about 10 minutes. There is a special control button to allow you to eject old pens, turn the cylinder, and select the next pen to be changed. It works simply and smoothly.

Included is a demonstration program.It shows a pie graph and multiple sine and cosine plotting. An excellent piece of software. It wasn't a very well documented program and it was tough to tell what was happening. More comments would have made it a good tutorial.

There are 17 different command codes and sequences that the printer recognizes. It works in one of two modes, as a printer or as a graphics plotter. It powers-up in the printer mode and does a self-test. As a printer it will respond to LPRINT or PRINT#-2 commands.

Tandy anticipated that the CGP-115 would probably used in a lot of graphic applications so they gave us a command to automatically draw the X and Y axis of a graph for us. It even marks the scale on the axis as you tell it to.

Just what do we really get for our $249+? Frankly I am of the opinion that we get quite a lot. If we look at the spec sheet we find that it will print up to 12 characters per second in text mode. Now, that isn't world class, but the print quality is good, and then there is the ability to print in several colors. The drawback is that we are given a piece of paper on which only four inches are used. In the printer mode we can divide that into either 40 or 80 characters per line. In the 80 character per line mode you had better have good eyes or a magnifying glass handy. It would be acceptable for keeping archive copies of listings or data but reading from it becomes tedious. If you just want a printer, put a few more dollars with your $250 and get one that is designed for higher speeds.

If you are looking for an inexpensive plotter and are not concerned with speed, do not mind being restricted to four inch paper, this is the place to put your money.

For the student who wants neat charts and graphs it would almost be perfect. The printing in either mode is very legible and any instructor should be impressed by it.

What about precision? Even after repeated, long range moves and several redefined origins, the pen would end up within 0.2mm of its origin. The error is cumulative, so after four or five movements it can be noticeable. From a practical standpoint, the precision is excellent. I compared the circumference of a circle drawn with the CGP-115 with one drawn by Radio Shack's $1995 plotter and found that there was little difference. The spec sheet says that your effective plotting range is 96mm horizontally and unlimited vertically. That 96mm is further divided into 480 steps, which means that one step on the plotter is only 0.2mm. Now that is just a wee bit smaller than the period on this page.

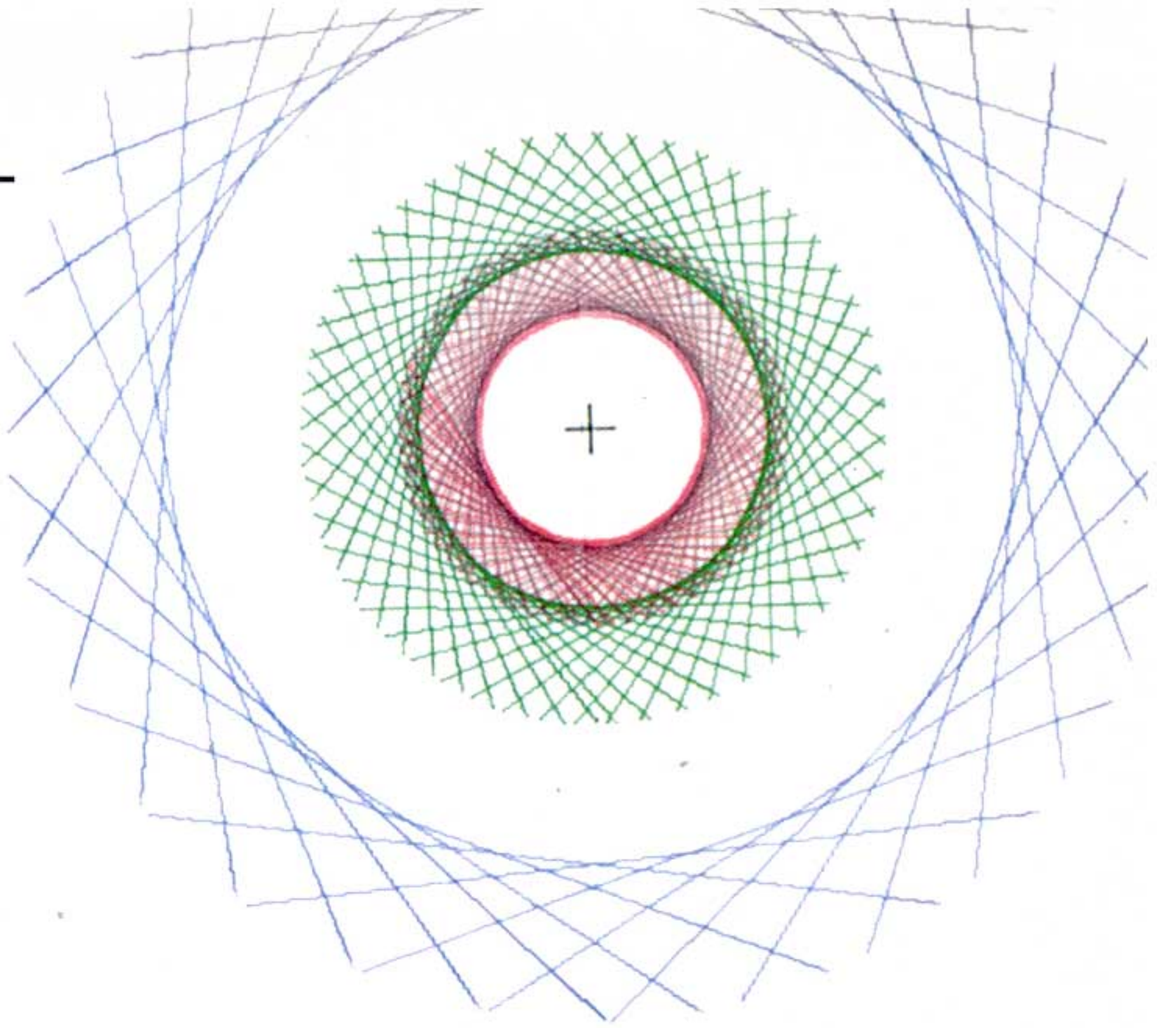
Just how much abuse can it take? I would expect it to last a long time under normal home or office use. The manual does caution about moving the pen cylinder by hand. The weakest components are probably the pens. Like any ball point pen, they are apt to fail. My first red pen ran out of ink, or found a flat spot on the ball during the second week of use. The other pens have held up well.

Each time the unit is turned on it goes through a start-up cycle that involves drawing a small box with each of the pens. I suspect that this is more to get the ink flowing in each of the pens than to actually test the machine. A nice thought on the part of Tandy, I just wish they had provided a way to override it. It is a little strange to start printing something, turn off the machine to go do something else, come back and turn it all on and end up with four little squares in the middle of your work! I did find that if you lift the front lid which covers the pens that they will not come into contact with the paper, but you will still get a linefeed.

The manual is well written, thorough about operating the machine, gives examples for each command, adequately describes their function and the results obtained. There are sections on setting up the unit, changing pens, changing paper, care and maintenance, and specific information about using the CGP-115 with the Model II/16. There are sample programs covering line graphs, underlining, color changes from within a program, how to superscript with the printer, and the general-purpose demonstration program. Unfortunately there are no specific application programs, and the example programs are very poorly documented.

It took me two days to draw a simple circle. I could have had it done in an hour if they had adequately remarked their example programs. For those who have purchased, or plan to purchase, this unit, I have included Listing 1 to show how to draw circles, and even do a little "string art" with the plotter.

The CGP-115 Color Graphic Printer is well worth the money if you need a light-duty plotter with graphic capabilities. At its price, it offers features no other machine has. For the home or office it is a good buy. Its chief limitation is the size of the paper and if that is acceptable, there is nothing that I can see to stop you from having one. ■



```
1 REM ORIGINAL PROGRAM BY: JERRY L. LATHAM
2 REM 1409 EVERGREEN CIRCLE
3 REM MIDWEST CITY, OK 73110
4 REM ALL PROGRAM LINES ARE NUMBERED IN
EVEN INCREMENTS OF 10
7 REM COLOR COMPUTER USERS CHANGE ALL
LPRINT STATEMENTS TO

PRINT#-2, STATEMENTS.

8REM **** COLOR GRAPHIC PRINTER CGP-115
DEMONSTRATION ****
9 REM SET UP INITIAL PARAMETERS AND ASSURE THAT
THE
CGP-115 IS IN THE GRAPHICS MODE AT THE LEFT
SIDE OF THE PAGE.
10 CLS: AN=0: RA=3.14159/180: CI=361: DIM
CX(CI,1): PL=328: TC=64: REM USE PL=264 TC=32
FOR COLOR COMPUTER.
15 REM FOR MODEL II/16 USERS ONLY => ON
ERROR GOTO 400
20 CLS: LPRINTCHR$(18): LPRINT"A": LPRINT
CHR$(18): LPRINT"C0"
25 REM PRINT THE LITTLE MARKER ON THE LEFT SIDE
OF
THE PAGE AND MAKE THAT THE NEW "HOME" OF THE
PEN.
30 LPRINT"I": LPRINT"D20,0": LPRINT"HD0,-10":
LPRINT"HD0,10": LPRINT"H"
40 INPUT"ENTER VALUE FOR THE RADIUS OF THE
CIRCLE.
RANGE CAN BE FROM 10 TO 240 (ENTER 0 TO END)
=>";R: R=ABS(INT(R)): IF R=0 THEN
LPRINT"M0,-400": LPRINT"A": END
50 IF R<10 OR R>240 THEN CLS: PRINT"RESTRICT
YOURSELF TO 10 - 240": GOTO 40
60 PRINT
70 PRINT"CALCULATING AND STORING THE POINTS
AROUND THE CIRCUMFERENCE OF THE NEW
CIRCLE.":PRINT:FOR V=0 TO 500:NEXT V: V=0
75 REM HERE IS THE WAY TO CALCULATE THE POINTS
ON THE EDGE OF A CIRLCE - THEY ARE
```

```
CALCULATED EACH DEGREE ALL THE WAY AROUND,
AND ARE SAVED TO BE USED AS REQUIRED LATER.
80 CLS:AN=0: FOR V=1 TO 361:
CX(V,0)=INT(SIN(AN)*R):
CX(V,1)=INT(COS(AN)*R): AN=AN+RA:
PRINT@PL-TC,"X Y";: PRINT@PL,CX(V,0);CX(V,1);:
NEXT V: PRINT: IF F1=1 THEN F1=0: GOTO 120
85 REM THE CIRCLE'S CIRCUMFERENCE POINTS
HAVE BEEN CALCULATED, NOW MOVE THE PEN OUT
TO THE CENTER OF OUR YET-TO-BE CIRCLE.
90 PRINT"MOVING THE PEN TO THE CENTER OF THE
GRAPHICS SHEET/PAD.": LPRINT"M240,0"
100 PRINT"MOVING PEN DOWN FOR ROOM FOR
THE CIRCLE.": LPRINT"R0,-300"
110 PRINT"RE-DEFINING THE ORIGIN OF THE PEN
TO THE CENTER OF THE CIRCLE.": LPRINT"I"
120 PRINT"MARKING THE CENTER OF THE CIRCLE":
LPRINT"HD-10,0": LPRINT"HD10,0":
LPRINT"HD0,-10": LPRINT"HD0,10": LPRINT"H"
130 PRINT"NOW MOVING THE PEN TO THE TOP OF
THE CIRCLE WITHOUT RE-DEFINING THE ORIGIN.":
LPRINT"M";CX(0,0);",";CX(0,1)
140 PRINT"DO YOU WISH TO:

 1) DRAW THE EDGE OF THE CIRCLE, OR

 2) PROCEED TO THE MANDALA ROUTINE

ENTER YOUR DESIRE (1 OR 2) => ";
150 A$=INKEY$: IF A$<"1" OR A$>"2" THEN 150 ELSE
PRINTA$: ON VAL(A$) GOTO 160,200
160 CLS: PRINT"DRAWING THE CIRCLE.":
LPRINT"M";CX(1,0);",";CX(1,1)
170 FOR V=1 TO CI: LPRINT"D";CX(V,0);",";CX(V,1):
NEXT V
175 REM NOW MAKE THE TOP OF THE CIRCLE THE
ORIGIN, PRINT THE RADIUS, AND THEN HOME THE
PEN TO THE TOP OF THE CIRCLE.
180 LPRINT"I": LPRINT"PRADIUS =";R: LPRINT"H"
185 REM NOW MOVE THE PEN BACK TO THE CENTER
OF THE CIRCLE AND DEFINE THAT POINT AS THE
ORIGIN AGAIN.
190 PRINT"NOW MOVING PEN BACK TO THE CENTER
OF THE CIRCLE.": LPRINT"R";0;",";-R: LPRINT"I"
200 F1=0: CLS: PRINT"DO YOU WANT TO:

 1) CONTINUE WITH THE MANDALA ROUTINE, OR

 2) DRAW ANOTHER CIRCLE WITH THE SAME CENTER
POINT, OR

 3) END THE SESSION?

ENTER YOUR CHOICE (1-3) => ";
210 A$=INKEY$: IF A$<"1" OR A$>"3" THEN 210 ELSE
PRINTA$: ON VAL(A$) GOTO 230,390,220
220 LPRINT"A": LPRINT CHR$(18): LPRINT"M0,-300":
LPRINT"A": END
225 REM
BEGIN THE ROUTINE TO DRAW THE MANDALA HERE.

230 INPUT"ENTER STEP RATE FOR POINTS ON THE
SIDE. ALTHOUGH THE RANGE IS FROM 1 TO 179, I
SUGGEST SOMETHING ABOUT 5 TO 15.

ENTER A ZERO TO END NOW ";ST: ST=ABS(INT(ST)): IF
ST=0 THEN 220
240 IF ST>179 THEN 230
250 INPUT"ENTER THE LENGTH OF THE LINES (1 TO
179).

HERE I SUGGEST SOMETHING IN THE RANGE OF 80
TO 120 => ";LE: LE=ABS(INT(LE)): IF LE<1 OR
LE>179 THEN PRINT: GOTO 250
260 INPUT"ENTER THE COLOR FOR THE LINES:

 0 = BLACK

 1 = RED

 2 = GREEN

 3 = BLUE

ENTER YOUR CHOICE (0-3) ";C: C=INT(ABS(C)): IF
C>3 THEN PRINT: GOTO 260
270 PRINT:PRINT"CHOOSING PEN COLOR NOW.":
LPRINT"C";C
280 PRINT" BEGINNING TO DRAW THE MANDALA
LAYER."
290 V=0: FOR X=1 TO CI STEP ST: V=X+LE
295 REM NEXT LINE ASSURES US THAT V IS IN A VALID
RANGE FOR THE ARRAY CX(N,M).
300 IF V>361 THEN V=V-361: GOTO 300
305 REM FIRST MOVE, WITHOUT DRAWING A LINE,
THE PEN TO THE NEXT START POSITION.
310 LPRINT"M";CX(X,0);",";CX(X,1)
315 REM THEN DRAW A LINE TO THE PROPER PLACE
320 LPRINT"D";CX(V,0);",";CX(V,1)
325 REM LOOP UNTIL DONE
330 NEXT X
335 REM NOW HOME THE PEN BEFORE DECIDING
WHAT TO DO NEXT
340 LPRINT"H"
350 PRINT"DRAW ANOTHER MANDALA LAYER (Y/N)?
";
360 A$=INKEY$:IF A$<>"Y" AND A$<>"N" AND
A$<>CHR$(110) AND A$<>CHR$(121) THEN 360
ELSE PRINTA$
370 IF A$="Y" OR A$=CHR$(121) THEN GOTO 230
380 GOTO 200
390 F1=1: GOTO 40
400 REM FOR MODEL II/16 ONLY => IF ERR=56
THEN RESUME
410 REM FOR MODEL II/16 ONLY => ON ERROR
GOTO 0
```

COBOL

An introduction and comparison of three packages

Model II

T. R. Dettmann, Associate editor

COBOL stands for **CO**mmon **B**usiness **O**riented **L**anguage. It was put together by government and business users to fill a need. They wanted a language that was readable and machine independent (the same program to run on *any* computer without change!). Part of the development was directed toward having the language so simple that anyone could understand and write a program without special training. To the credit of the developers, COBOL met the objectives of readability and machine independence. Despite early hopes, COBOL still required people to be trained in programming.

All this is very interesting, but why bother with COBOL? The TRS-80 comes with BASIC and any problem which can be programmed can be done in BASIC, can't it? Well, yes, but COBOL is more than just your run of the mill programming language.

First, by general agreement, COBOL is the most widely used programming language in the world today. This means that there are more applications available in COBOL than any other language in the world.

Second, COBOL has built-in programming tools that far surpass those available in BASIC. COBOL programs, if written with clarity in mind, also read well. They can be self documenting to a degree that BASIC can never reach.

COBOL also forces the programmer to give more thought to variables, files, and data in general. Everything must be specified before use, and all in one place.

COBOL allows the use of longer variable names than BASIC does AND allows structured variables that will let a programmer refer to, for example, a date as a whole or as any part of the whole.

We'll use a trivial example of a COBOL program to illustrate how it works. This program was designed to balance a check book. It isn't intended to be efficient, but does illustrate different aspects of COBOL.

The Language

Looking at the program listing, it becomes apparent that the program is very much like English. This isn't just accidental, it was designed that way. COBOL words that specify an action to be performed are called 'verbs.' They are always the first word on any program line and they specify what will be done in that 'sentence.' Verbs are 'reserved words,' meaning that they cannot be used for anything but their intended purpose. A reserved word cannot be used as a variable name.

Data names in COBOL can be thought of as 'nouns.' They are the subject of whatever action is specified by a verb. We can add qualifiers to data names which can be likened to 'adjectives.' They are used to specify which record a data name comes from.

COBOL programs are built around four "divisions" which must appear in each program. Let's look at the divisions in more detail to see how they work.

The Program Divisions

The Identification Division: As its name implies, the identification division identifies the program. The sample program includes a minimum of identification for a program.

The Environment Division: For this particular program, the environment division is simply a device for telling what system the program was written on. It serves purely as documentation. However, it could be much more.

The environment division is made up of two sections, the configuration section and the input-output section. As a third paragraph in the configuration section, we could have included a special names section. This section allows the programmer to specify, for example, that the decimal point will be a comma (European convention), or that the currency sign is another character, or that another mnemonic is being used to designate the printer in the program.

The input-output section is the controlling section for all data files. Within this section, we have two paragraphs. First is the file-control section. This section specifies the characteristics of the data files used with the program. This is one of the most important paragraphs in COBOL. Poor file specification makes a job hard, good specification makes it easy.

The second paragraph is the I-O control paragraph. It is used primarily to control sharing of memory space between files within the program. If two files are never going to be open at the same time, they can be designated to share the same memory buffer space in this paragraph.

Unlike BASIC, COBOL requires the programmer to specify ahead of time what variables he will use, how they are to be stored, and what their forms will be. There is a lot of flexibility in COBOL for defining variables and I have only touched on it in the example.

In our sample, we have defined several variables, all to be of level 01. These definitions are in the working storage section of the data division. Each variable definition gives the name the variable is to be known by, and then a "Picture" of the variable which designates its form. For example, the variable CHECK-DATE is designated to be eight alphanumeric characters (a string in BASIC). AMOUNT is a number which has six places; four number places including leading zeros, an implied decimal point between the fourth and fifth digit, and then two decimal places.

Each variable definition fully defines the use of the variable for the particular program. We can also define sub-variables by assigning higher level numbers. For example, we could define DATE to be level 01, and then have three sub variables, Day, Month, and Year each of level 02. Referring to DATE would get all three, but we could refer to only one if we wanted.

We can also have other sections in the data division. There could be a file control section which specifies the variables stored in each file record, a linkage section which specifies variables to be passed to precompiled subprograms, and a screen section that describes screen formats.

The data division is fundamental to the working of the program. Variable choices and descriptions must be well thought out ahead of time or the program will fail.

The Procedure Division details the steps to be taken to solve the problem. It may have a declaratives section at the beginning which specifies programmer designed procedures to handle disk file errors.

In the program listed, there are named paragraphs starting in column five. Each heads a logical grouping of statements which make up one module of the program. Just as in English, each paragraph is a logical thought.

Within each paragraph, COBOL statements are used to specify the action the computer program is to carry out. DISPLAY writes to the screen (like PRINT in BASIC), ACCEPT inputs data from the keyboard into a variable (like INPUT in BASIC), and MOVE sets one variable equal to the value of another ("=" in BASIC). PERFORM is like a GOSUB in BASIC. The nice thing about it is that we can specify what to perform by paragraph name and even specify more than one paragraph at a time to perform! The EXIT verb forces a return from the PERFORM just like a RETURN in BASIC.

COBOL also allows the use of precompiled subroutines which are CALL'ed like subroutines in FORTRAN. It is a valuable means for using and reusing standard routines. With long descriptive names like GET-DEPOSITS-NOT-CREDITED, we can design the main level of the program using descriptive names and later develop the modules to accomplish each task.

COBOL's major failing is that it tends to be too verbose. In order to have the readability, it is necessary to write many lines of code. The English-like readability also leads to some interesting programmer's jokes. For example, an often told story in COBOL programming circles is about the programmer who wrote a whole program, defined all the variables, just so that as the last statement in the program, he could write: ADD GIN TO VERMOUTH GIVING MARTINI. Humorous, but not very enlightening.

Should You Use COBOL?

COBOL is a compiled language. All of the COBOL is translated into machine language or an intermediate language which you can sell without selling your program itself like you must do with an interpreter.

COBOL also has the advantage of being used everywhere for every possible kind of business application. You may find applications already in COBOL that you can use. At least, you will almost certainly find professional programmers who work with COBOL.

Even more important for many applications are the professional level capabilities of a COBOL system. The ability to use special file types such as indexed-sequential, relative, and sequential is important in many data base applications. The data structuring capabilities are far more powerful than BASIC.

On the negative side, COBOL has been around for some time and it is now thought to be a cumbersome and archaic language. More modern languages are available to handle programming problems. Many of these have much more to recommend them for general use. COBOL is also primarily a big machine language. Many programs require big machine capabilities to work efficiently.

Three Packages Examined

I looked at three COBOL packages during this

evaluation. Each of them will run on the TRS-80 Model II system. There has been no attempt to provide benchmark information. It is generally best to do your own benchmarking as it relates to your own application. Each system ran sample programs satisfactorily.

Nevada COBOL was written by Ellis Computing and is distributed by Business Micro Products, 609 S. Livermore, Livermore, Ca. 94550, (415)449-4412. It runs on the CP/M operating system.

Nevada COBOL is a "stripped-down" COBOL. All of the essential COBOL verbs are included and almost any project that could be programmed could be handled in this version. It lacks some of the more sophisticated capabilities of a full COBOL implementation, but what is left out is not essential for many applications.

Ellis Computing has prepared a set of application program packages in Nevada COBOL. Book 1 contains Budget Planning, Personal Financial Reporting, Labels, and PreCOBOL. The last one allows a programmer to invent his own instruction mnemonics and produces well-formatted programs for easy readability.

Microsoft COBOL

Microsoft puts out their COBOL compiler system for both CP/M and TRSDOS. COBOL programs are first compiled into relocatable binary code, then they are linked with routines from the COBOL library or other relocatable modules into an executable package.

The link step is carried out with the LINK80 linker, the same one used in their FORTRAN and BASIC Compiler packages. It is even possible to link to packages prepared with the FORTRAN compiler or the MACRO80 assembler.

The system had almost all of the features I am accustomed to finding in a COBOL system. It does not include multiple-index keys in file handling; a possibly serious limitation. However, I didn't find it to be a problem in any of my tests.

The COBOL Report Writing and Debug capabilities are left out altogether and Microsoft has no plans to implement them. Opinion varies as to how useful these really are. The Report writing capability can be programmed around and the IBM Debug facilities extension is included.

I found Microsoft COBOL to be easy to use. Their documentation is a professional reference manual which assumes you already know what you're doing.

Microsoft COBOL is closer to the standard than is Nevada COBOL, even in the little things like which columns are which. It is a system that could be used to produce good, professional COBOL programs with very sophisticated capabilities.

Radio Shack COBOL

Radio Shack's COBOL package was written by the Ryan-McFarland Corporation. It is available through any Radio Shack outlet and should be available for demonstration at a local Computer Center.

It is a very sophisticated program development

system with a wide range of capabilities including sequential, relative, and indexed file structures, as well as segmentation. This gives the system a great amount of power. It has the further advantage that it is compatible with Radio Shack's BASIC Compiler. Data structures from one system are compatible with the data structures from the other.

The Model II version is a high-level implementation of the ANSI74 COBOL standard. The Model I/III versions are identical except for the limitations in screen size and memory.

Which is Best?

I found that my preferences were hard to pin down. I liked Nevada COBOL for its ease of use. I would highly recommend it to anyone who wants to learn COBOL without burying himself in excessive detail. While it is a stripped-down version, I wouldn't hesitate to develop software using it.

I liked Microsoft COBOL for its power and its compatibility with their FORTRAN and other software. It was easy to use but many people complain about the multiple steps required to get going. It is a professional system and I would *not* recommend it unless you know how to use COBOL and are ready for its power.

The Radio Shack COBOL system performed well and accomplished everything I asked of it. I found it easy to use, certainly easier than the Microsoft package but not so easy as the Nevada COBOL.

Given a choice, for professional development I personally would pick either the Microsoft or the Radio Shack packages. I felt that there were tradeoffs in either case. Microsoft's package has more power, Radio Shack's is easier to use. Both have enough power to solve any problem that needs to be solved. Microsoft's works under CP/M whereas Radio Shack's works under TRSDOS. Nevada COBOL would be my choice for simplicity on CP/M. It is the least powerful of the three, but it is also the easiest to use.

References

To find out more about COBOL, some useful books are:

Tucker, Allen, *Programming Languages*, McGraw-Hill, New York, 1977.

Parkin, Andrew, *COBOL For Students*, Unwin Bros., Surrey, 1975.

Harrison, William, *A Programmer's Guide to COBOL*, Van Nostrand Reinhold, New York, 1980.

Chmura, Louis and Ledgard, Henry, *COBOL with Style*, Hayden, Rochelle Park, 1976.

There is also a complete set of instructional manuals from Anaheim Publishing, 1120 East Ash, Fullerton, CA 92631 (714) 879-7922. The texts, by Gary Shelly and Thomas Cashman, are instructional manuals in big system COBOL. The five books are good and easy to read. The titles are *Introduction to Computer Programming: Structured COBOL, Advanced Structured COBOL Program Design and File Processing, Introduction to Computer Programming: ANSI COBOL, ANSI COBOL Workbook,* and, *Advanced ANSI COBOL Disk/Tape Programming Efficiencies.* ■

Sample COBOL Program

```
A>
0001 IDENTIFICATION DIVISION.
0002 PROGRAM-ID.  CHECKBOOK BALANCING.
0003*****************************************************************
0004*
0005*    ESTABLISH HARDWARE REQUIREMENTS
0006*
0007*****************************************************************
0008 ENVIRONMENT DIVISION.
0009 CONFIGURATION SECTION.
0010 SOURCE-COMPUTER. TRS80 MODEL II.
0011 OBJECT-COMPUTER. TRS80 MODEL II.
0012*****************************************************************
0013*
0014*    ESTABLISH THE DATA NAMES TO BE USED
0015*
0016*****************************************************************
0017 DATA DIVISION.
0018 WORKING-STORAGE SECTION.
0019 01 CHECK-DATE  PICTURE 9.
0020 01 NUMBER PICTURE X(8).
0021 01 AMOUNT PICTURE 9999V99.
0022 01 BALANCE PICTURE S99999.99.
0023 01 FINAL-BALANCE PICTURE S$$,$$$.99.
0024*****************************************************************
0025*
0026*    HERE IS THE ACTUAL COMPUTATIONAL SECTION OF THE PROGRAM
0027*
0028*****************************************************************
0029 PROCEDURE DIVISION.
0030 BEGIN.
0031       DISPLAY "CHECKBOOK BALANCING".
0032       DISPLAY "  ".
0033       DISPLAY "INPUT CURRENT CHECKBOOK BALANCE (7 DIGITS): ".
0034       ACCEPT BALANCE.
0035       PERFORM GET-CHECKS THRU RET1.
0036       PERFORM GET-DEPOSITS-NOT-CREDITED THRU RET2.
0037       PERFORM GET-SERVICE-CHARGES THRU RET3.
0038       MOVE BALANCE TO FINAL-BALANCE.
0039       DISPLAY "ADJUSTED BANK BALANCE: ".
0040       DISPLAY FINAL-BALANCE WITH NO ADVANCING.
0041       STOP RUN.
0042 GET-CHECKS.
0043       DISPLAY "ENTER OUTSTANDING CHECKS".
0044       DISPLAY "  ".
0045 GET-CHECK.
0046       DISPLAY "ENTER CHECK DATE (00/00/00 TO END): ".
0047       ACCEPT CHECK-DATE.
0048       IF CHECK-DATE EQUAL TO "00/00/00" GO TO RET1.
0049       DISPLAY "ENTER CHECK NUMBER:                 ".
0050       ACCEPT NUMBER.
0051       DISPLAY "CHECK AMOUNT (6 DIGITS):            ".
0052       ACCEPT AMOUNT.
0053       ADD AMOUNT TO BALANCE.
0054       DISPLAY "  BALANCE:                          ".
0055       MOVE BALANCE TO FINAL-BALANCE.
0056       DISPLAY FINAL-BALANCE WITH NO ADVANCING.
0057       GO TO GET-CHECK.
0058 RET1.   EXIT.
0059 GET-DEPOSITS-NOT-CREDITED.
0060       DISPLAY "ENTER DEPOSITS NOT CREDITED".
0061       DISPLAY "  ".
0062 GET-DEPOSIT.
0063       DISPLAY "ENTER DEPOSIT DATE (00/00/00 TO END): ".
0064       ACCEPT CHECK-DATE.
0065       IF CHECK-DATE EQUAL TO "00/00/00" GO TO RET2.
0066       DISPLAY "ENTER AMOUNT (6 DIGITS):            ".
0067       ACCEPT AMOUNT.
0068       SUBTRACT AMOUNT FROM BALANCE.
0069       DISPLAY "   BALANCE:                         ".
0070       MOVE BALANCE TO FINAL-BALANCE.
0071       DISPLAY FINAL-BALANCE WITH NO ADVANCING.
0072       GO TO GET-DEPOSIT.
0073 RET2.  EXIT.
0074 GET-SERVICE-CHARGES.
0075       DISPLAY "ENTER SERVICE CHARGES NOT CREDITED".
0076       DISPLAY "  ".
0077 GET-CHARGES.
0078       DISPLAY "ENTER SERVICE CHARGE:  ".
0079       ACCEPT AMOUNT.
0080       IF AMOUNT EQUAL TO ZERO GO TO RET3.
0081       SUBTRACT AMOUNT FROM BALANCE.
0082       DISPLAY "BALANCE:                ".
0083       MOVE BALANCE TO FINAL-BALANCE.
0084       DISPLAY FINAL-BALANCE WITH NO ADVANCING.
0085       GO TO GET-CHARGES.
0086 RET3.  EXIT.
0087 END PROGRAM CHECKBOOK.
```

TRS-80® Printers—No One

DMP-500
1795.00
Cat. No. 26-1252

Daisy Wheel II
1995.00
Cat. No. 26-1158

DMP-200
799.00
Cat. No. 26-1254

DMP-400
1195.00
Cat. No. 26-1251

Triple-Mode Dot Matrix Printers

Our three new dot-matrix printers—the DMP-200, 400 and 500—offer incredible versatility! Each one allows you to select from a *word processing* mode which emulates Daisy Wheel features like proportional spacing, boldface, underline, super and subscripts; a *bit image* mode for impressive, dot-addressable graphics; and a *data processing* mode for fast, efficient throughput. These printers feature bi-directional, logic-seeking 9-wire print heads and support the use of single sheets or fanfold paper.

You get the 96-character ASCII set with 32 special and 30 block graphics characters. Lower case characters have descenders. Print density can be set at 10, 12 or 16.7 cpi in a 9×9 or 15×9 dot matrix. Print proportionally in an n-23 dot matrix. (See samples at right.) Repeat and column-addressing codes make graphics programming easier. Horizontal resolution is 60-100 dots per inch.

Best of all, a handy switch allows you to "custom set" modes, pitches and interface options to make it easier to use a TRS-80 printer with your software. The DMP-500 features a standard parallel interface. The DMP-200 and 400 include selectable parallel and TRS-80 Color Computer-compatible serial interfaces (600/1200 baud). All offer front panel on-line switching and paper advance, including form feed.

DMP-200. An amazingly low price for a full-performance 9 1/2" printer. Prints 120 characters per second at 10 cpi—55 lines per minute at 80 columns. Removable tractor and friction platen. Line feed in 1/6, 1/8 and 1/72 increments.

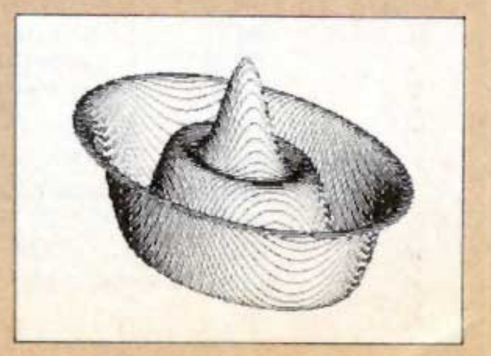
```
10 CPI STANDARD !"#$%&'()*+,-./0
12 CPI ELITE !"#$%&'()*+,-./0123456789
10 CPI CORRESPONDENCE !"#$%&'()*
PS MODE ABCDEFGHI abcdefghi vwxyz{|}~
```

DMP-400. Full 132-column output for under $1200. Prints 140 cps—51 lpm at 132 columns. Compact, low-profile styling with noise-reducing cover. Line feed in 1/6, 1/8, 1/72 and 1/216 increments. Adjustable tractor is also removable. Friction platen.

DMP-500. A business printer that's all business. High performance design gives 220 cps—76 lpm at 132 columns. Adjustable tractor accepts paper from 4" to 15" wide. Line feed in 1/6, 1/8, 1/72 and 1/216 increments.

Low-Cost Dot-Addressable Printer

DMP-100. Graphic density is 60×63 dots per square inch. Prints 80 upper and lower case 5×7 dot-matrix characters at 50 cps, 27 lpm (10 cpi). Features underline, 480-byte full-line dot buffer and selectable parallel or Color Computer-compatible serial interfaces. Adjustable tractor handles 4 1/2" to 9 1/2" fanfold paper.



...e Offers a Wider Variety

DMP-100
399.00
Cat. No. 26-1253

DWP-410
1495.00
Cat. No. 26-1250

ABCDEFG abcdef 01234 :;<=>?

ABCDEFGH abcdefg 01234 :;<=>?

ABCDEFGH abcdefg 01234 :;<=>?

ABCDEFGH abcdefg 01234 :;<=>?

Daisy Wheel Word Processing Printers

Our letter-quality printers give your correspondence and reports that crisp, clean "electric typewriter" look. Interchangeable 124-character print wheels let you select the typeface you want (shown above). "External Program Mode" lets you use print wheels with different pitch or special characters. Switch selectable print density for fixed 10 or 12 cpi or proportional spacing (to give justified right margins). All standard word processing features are included: forward and reverse paper feed, 1/2-line feed, underline and backspace. Use friction-feed platen or an optional tractor-feed. Parallel interface. Each printer includes Courier 10 print wheel and a carbon ribbon cartridge.

Daisy Wheel II. Our finest! Ideal for the business, legal or medical office. Prints at over 500 words per minute. For added convenience, just snap in a tractor-feed (26-1447, $289.95) to use fanfold paper, pin-fed labels, pre-printed and multi-part forms. Fast, optimized carriage movement for surprisingly efficient throughput. The DW II can also be used with our microprocessor-controlled automatic sheet feeder or envelope feeder.

DWP-410. Full-featured letter-quality printouts at a remarkably low price. Prints at over 300 words per minute. 1/120" space and 1/48" line feed. Automatic Paper Set makes paper insertion easy. Or use an easily removable tractor-feed (26-1459, $239.95)—it's ideal for labels, multi-part forms and long DP runs with minimal supervision.

Available Nationwide

You can choose the TRS-80 dot-matrix or daisy wheel printer that meets your needs at any Radio Shack Computer Center, store or participating dealer. Be sure to see our complete selection of color plotters, too. Come in today!

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

Send me your free TRS-80 Computer Catalog today!

Mail To: Radio Shack, Dept. 83-A-185
300 One Tandy Center, Fort Worth, Texas 76102

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

PHONE _____

Retail prices may vary at individual stores and dealers. Special order may be required at some stores.

Basicmon

See what is going on inside your Color Computer

Color Computer

Ronald Constant, Ft. Worth, TX

Basicmon is an easy-to-use and simple-to-understand monitor program. It is designed for the person who wants to do a minimum amount of thinking, yet wants to look at memory and machine language programs. There are some helpful examine, change and debug features in Basicmon.

The program is written entirely in BASIC to help a BASIC programmer fully understand what is being done. In fact, the whole program is written for the programmer who does not yet know assembly language, but wants to begin to explore his memory, ROM and machine language programs. I resisted a strong temptation to include machine language subroutines in Basicmon in two places. The first place is the copy routine (menu choice number six). BASIC is slow in using POKEs and PEEKs as compared to a machine language routine. The second place is the debugger (menu choice number eight). The registers will not be printed when debugging a machine language program and control comes back to the debugger after hitting a breakpoint. To display the registers requires a machine language subroutine. This is a weakness if you want to do serious debugging of assembly language programs.

In all the inputs, you can use either decimal or hexadecimal just like you do with Extended BASIC. For example, you can enter <42> or <&H2A>. You do not have to learn a special way of entering numbers. There is one important exception. That exception is in the examine and/or change memory routine (choice five). I will explain that exception when I get to that topic. The key point is that this monitor is just like any other BASIC program. Many monitor programs have special or unique ways of entering data and they are useful. But, you must learn a new operating system for each program.

Not only do you enter numbers in a way that you already know, but you don't have to remember special commands. There is a menu to select the routine you want to do. When you get to that routine, you will be prompted for every necessary input. In some routines where there are several possible commands to enter, there is a split screen with the commands you need listed at the top of the screen.

For the person who does not think in hex, Basicmon displays memory addresses and values in decimal and hex. You won't have to constantly be making conversions from one base to the other. You will see data in both bases and be able to enter data in both bases.

Let's look at the nine routines of the program.

Program Options

Hex/ASCII Dump— This is a standard dump that displays the contents of memory in hex and CHR$, that is, hex and ASCII. Eight-bit values are corrected to their seven-bit values so that all ASCII codes display their true character. Control codes, values less than 32, are displayed as a period. The display on the screen shows eight memory locations in hex with the ASCII equivalent under each hex value. If you choose the printer option, the printout will be sixteen values wide. The screen will display normally.

View Memory— This option lets you view individual memory locations. The addresses are shown in decimal and hex. The values are shown in decimal, hex and ASCII. The ASCII values are not corrected. They show exactly as the CHR$ they represent. You will see all CHR$ from zero to 255, except for CHR$(13), the carriage return. It shows as a space.

Save Memory on Disk (data form)— This routine allows you to save the contents of a block of memory, including machine language programs, as data, that is, in ASCII format. Sometimes it is helpful to have such ASCII files. If you want to save a machine language program more efficiently, press <BREAK>. Then use the normal SAVEM or CSAVEM.

Load Data from Disk— You can load any file in ASCII format into any block in RAM memory. Normally, it is much faster and easier to use normal LOAD or CLOAD and LOADM or CLOADM commands.

Examine/Change Memory— Here is the routine to use to put values into memory. It can also be used to examine memory. You can examine and enter values any time you want. By using the <UP> or<DOWN> arrows, you can go forward or backward through memory. Addresses and values are displayed in decimal and hex.

We now come to the one exception to input in hex or decimal. To put, or change, a value in a memory location you must use a two-digit hex value only. The two digits cannot be preceded by <&H>; thus you would enter &H3E as <3E> <ENTER>. If you change your mind about entering a number, press the <UP> or <DOWN> arrow instead of <ENTER>. If you make a mistake, use the arrows to go back to that address and enter the correction. There are prompt messages on the screen at all times to remind you of these differences while in this routine.

Block Copy of Memory— One block of memory can be copied to a new location. The source block of memory is not erased or altered; it is simply copied in the new location.

Print ASCII 24 Bytes by Addr— The contents of memory are displayed in blocks of ASCII characters. The address is shown on the left of the screen. Twenty-four ASCII characters are shown on a line representing twenty-four memory addresses. Up to fourteen lines can be displayed at a time. Thus, you can see a block of 336 bytes of memory in a block in ASCII form. This feature is very helpful when you are disassembling a machine language program. It is very easy to find ASCII lists which is important for proper disassembly. Also, it helps when trying to understand how your ROM works. There are other advantages to this routine over the Hex/ASCII Dump for locating ASCII lists. This routine displays much faster. It is easier to actually see an ASCII list. The result is that you can save a lot of time when looking at a long machine language program.

This routine is the best one to use when just looking at any blocks of memory to better understand how your computer works. For example, start looking at the same memory that holds this BASIC program. You can get the start address by viewing memory addresses 25 and 26, using routine number two or number five. The start address will be the hex value in addresses 25 and 26 put together. If you are using a disk system and are in PCLEAR 4, address 25 will have decimal value 38 and hex value 26. Address 26 will have decimal value 1 and hex value 1. Then, you can use this routine to view this BASIC program by inputing <&H2601> when prompted for the <START ADDRESS>. Address locations 27 and 28 will give you the memory addresses immediately after the end of your BASIC program. When you view your program, you will see the shorthand that your computer uses for BASIC commands.

Debugger— This is a very simple routine that allows you to do some debugging by setting breakpoints in a machine language program, not ROM. It doesn't use &H3F, a software interrupt, but rather an &H39, a return from subroutine. As already noted, the registers cannot be displayed. You must manually remove the breakpoints by using the <Y>ANK command. If something should happen, you can use routine number five to manually insert the original value where you set a breakpoint.

Instructions— A short set of instructions is provided in the program. You can delete them if you want.

Cassette Version

Basicmon was written for disk. However, a few minor changes will convert it to cassette.

Line 510: Change DISK to CASSETTE
Line 520: Change FILENAME/EXT to FILENAME
Line 540: Change #1 to #-1
Line 570: Change WRITE#1 to #-1 and change DISK to CASSETTE
Line 620: Change DISK to CASSETTE
Line 630: Change FILENAME/EXT to FILENAME
Line 650: Change #1 to #-1
Line 670: Change (1) to (-1)
Line 680: Change #1 to #-1
Line 710: Change #1 to #-1

The program in Listing 1 with all the remarks uses 5647 bytes of memory. The stripped version with remarks and spaces deleted and lines packed uses 4424 bytes of memory. On disk, the stripped version uses two granules and the unstripped version uses three granules.

Program Listing 1 for Basicmon

```
10 CLS:DIMA(16):DIMD(15):DIMB$(23)
20 'PRINT MENU
30 CLS:PRINT" SELECT A NUMBER"
40 PRINT"1—HEX/ASCII DUMP"
50 PRINT"2—VIEW MEMORY: DEC HEX ASC"
60 PRINT"3—SAVE:MEMORY ON DISK(DATA FORM)";
70 PRINT"4—LOAD: DATA FROM DISK"
80 PRINT"5—EXAMINE/CHANGE MEMORY"
90 PRINT"6—COPY: BLOCK COPY OF MEMORY"
100 PRINT"7—PRINT:ASCII 24 BYTES BY ADDR"
110 PRINT"8—DEBUGGER"
120 PRINT"9—INSTRUCTIONS"
130 C$=INKEY$:IFC$=""GOTO130
140 C=VAL(C$)
150 IF C<1 OR C>9 GOTO30
160 ONC GOTO180,750,510,620,1050,1430,
880,1710,1640
170 'HEX/ASCII DUMP
180 CLS:GOSUB1340:B=—1
190 Z=0
200 PRINT"DO YOU WANT A HARDCOPY (Y/N)?"
210 X$=INKEY$:IF X$<>"Y" AND X$<>"N" THEN GOTO
210
220 FORA=SA TO EA STEP8
230 B=B+1:'FLAG FOR WHEN PRINTER WILL PRINT OUT
FOR LONGER LINE
240 Z=Z+1
250 GOSUB1380
260 PRINT"$";F$;:PRINT" ";
270 IF X$="Y" AND B=0 THEN PRINT#—2,"$";F$;:PRINT#—2,"
";
280 FORD=0TO7
290 S=PEEK(A+D)
300 GOSUB1400
310 PRINT S$;" ";:IF X$="Y" THEN B$(B*8+D)=S$
```

```
320 A$="&H"+S$:A$=CHR$(VAL(A$))
330 IF X$="Y" AND B=1 AND D=7 THEN FOR E=0
TO15:PRINT#-2, B$(E);" ";:NEXT E
340 'FOR ASCII CONVERT 8 BIT TO 7 BIT
350 IF A$>CHR$(127) THEN A$=CHR$(ASC(A$)-128)
360 IF A$<CHR$(32) OR A$=CHR$(127) THEN S(D)=46
370 IF A$>CHR$(31) AND A$<CHR$(127) THEN
S(D)=ASC(A$)
380 NEXTD
390 PRINT:IFX$="Y" AND B=1 THENPRINT#-2
400 FORC=0 TO 7
410 PRINTTAB(7)CHR$(S(C));" ";:IF X$="Y" THEN
D(B*8+C)=S(C)
420 IF X$="Y" AND B=1 AND C=7 THEN FOR E=0 TO
15:PRINT#-2,TAB(7)CHR$(D(E));" ";:NEXT E
430 NEXT C
440 PRINT
450 IF X$="Y" AND B=1 THEN PRINT#-2:B=-1
460 IF INT(Z/7)=Z/7 GOSUB 1300
470 NEXTA
480 IF X$="Y" THEN PRINT#-2
490 GOSUB1320:CLS:GOTO30
500 'SAVE BLOCK OF MEMORY
510 CLS:PRINT"TO SAVE ON DISK WHAT IS
THE":PRINT:GOSUB1340
520 INPUT"FILENAME/EXT";Q$
530 PRINT:PRINTTAB(10)"NOW SAVING"
540 OPEN"O",#1,Q$
550 FORA=SA TO EA
560 B=PEEK(A)
570 WRITE#1,B
580 NEXTA
590 CLOSE#1:PRINT:PRINT"THE DATA IS SAVED ON DISK
UNDER FILENAME ";"'";Q$;"'":PRINT:GOSUB1320
600 'LOAD INTO MEMORY
610 GOTO30
620 CLS:PRINT"TO LOAD FROM DISK WHAT IS THE
BEGINING ADDRESS FOR THE DATA":INPUTSA:PRINT
630 PRINT"WHAT IS THE FILENAME/EXT":INPUTQ$:PRINT
640 PRINT:PRINTTAB(10)"NOW LOADING"
650 OPEN"I",#1,Q$
660 FORA=SA TO &H7FFF
670 IF EOF(1)=-1 THEN 710
680 INPUT#1,B
690 POKE A,B
700 NEXTA
710 CLOSE#1:PRINT:PRINT"THE DATA IS NOW LOADED IN"
720 PRINT" ADDRESSES $";HEX$(SA);" TO
$";HEX$(A-1):PRINT
730 GOSUB 1320:GOTO 30
740 'LOOK AT INDIV. MEM. ALL FORMS
750 CLS:GOSUB1340
```

```
760 Z=0
770 FORA=SA TO EA
780 Z=Z+1
790 S=PEEK(A):PRINT USING"**###";A;
800 GOSUB1380
810 PRINT" ";F$,:PRINTUSING"**#"; S;:PRINT;" ";
820 GOSUB1400
830 PRINTS$;" ";:IF S<>13 THEN PRINTCHR$(S) ELSE PRINT
840 IF INT(Z/14)=Z/14 GOSUB1300
850 NEXTA
860 GOSUB1320:GOTO30
870 'LOOK AT ASCII IN BLOCKS
880 CLS:GOSUB1340
890 Z=0
900 FORA=SA TO EA STEP24
910 GOSUB1380
920 PRINTF$;" ";
930 Z=Z+1
940 FORD=0TO23
950 S=PEEK(A+D)
960 IF S>127 THEN S=S-128
970 IF S<32 THEN S=46
980 PRINTCHR$(S);
990 NEXTD
1000 PRINT
1010 IF INT(Z/14)=Z/14 GOSUB1300
1020 NEXTA
1030 GOSUB1320:GOTO30
1040 'EXAMINE/CHANGE MEMORY
1050 CLS:GOSUB1340
1060 IF EA=0 OR EA>32767 THEN EA=32767
1070 FORA=SA TO EA
1080 S=PEEK(A)
1090 PRINT@480,USING"**###";A;
1100 GOSUB1380
1110 PRINT" ";F$;" ";:PRINTUSING"**#";S;:PRINT" ";
1120 GOSUB1400
1130 PRINTS$;" ";
1140 B=S:IFB>127 THEN B=B-128
1150 IFB<32 THEN B=46
1160 PRINTCHR$(B)
1170 PRINT@0," PRESS <DOWN> OR <UP> ARROWS
<ENTER> TO CHANGE OR
<M>ENU":PRINTSTRING$(32,"=");:PRINT" ADDRESS DEC
HX A CHANGE(HX)";
1180 CC$="":D=0:PRINT@473
1190 C$=INKEY$:IFC$=CHR$(10) GOTO1260 ELSE
IFC$=CHR$(94) GOTO1250 ELSE
IFC$=CHR$(13)GOTO1240 ELSE IF C$="M"GOTO30
1200 IFC$<="F" AND C$>="0"GOTO1210 ELSE
GOTO1190
1210 D=D+1
```

```
1220 CC$=CC$+C$:PRINT@473+D,C$:IF D=2 GOTO 1230
ELSE GOTO1190
1230 CC$="&H"+CC$:C=VAL(CC$):GOTO1190
1240 POKEA,C:GOTO1260
1250 A=A-1:PRINT@474,CHR$(94):GOTO1080
1260 NEXTA
1270 GOTO30
1280 END
1290 'SUBROUTINES USED MULTIPLE TIMES
1300 PRINT" PRESS <C> TO CONTINUE";
1310 IF INKEY$<>"C" THEN 1310 ELSE:PRINT: RETURN
1320 PRINT" PRESS <M> TO RETURN TO MENU";
1330 IF INKEY$<>"M" THEN 1330 ELSE:PRINT: RETURN
1340 INPUT"START ADDRESS";SA
1350 INPUT"END ADDRESS";EA
1360 RETURN
1370 ' SUBS TO PUT HEX VALUES IN STANDARD FORMAT
1380 IF LEN(HEX$(A))=1 THEN F$="000"+HEX$(A) ELSE IF
LEN(HEX$(A))=2 THEN F$="00"+HEX$(A) ELSE IF
LEN(HEX$(A))=3 THEN F$="0"+HEX$(A) ELSE
F$=HEX$(A):RETURN
1390 RETURN
1400 IF LEN(HEX$(S))=1 THEN S$="0"+HEX$(S) ELSE
S$=HEX$(S)
1410 RETURN
1420 'COPY BLOCK OF MEMORY TO ANOTHER LOCATION
1430 CLS:PRINT"BEGINNING ADDRESS FROM WHERE
YOUWANT TO COPY MEMORY":INPUTSA
1440 PRINT"ENDING ADDRESS": INPUTEA:PRINT:IF EA<SA
THEN PRINT"START ADDRESS MUST BE SMALLER THAN OR
EQUAL TO END ADDRESS":GOSUB1300:GOTO1430
1450 PRINT"BEGINNING ADDRESS WHERE TO
COPY":INPUTCA
1460 PRINT"PRESS <C>ONTINUE OR <M>ENU"
1470 C$=INKEY$:IFC$="M"GOTO30 ELSE
IFC$="C"GOTO1480 ELSEGOTO1470
1480 PRINT:PRINT" NOW COPYING":Z=0
1490 FORA=SA TO EA
1500 PE=PEEK(A):PO=CA+Z
1510 IF PO>&H7FFF THEN PRINT:PRINT"CANNOT COPY
INTO ROM AREA":PRINT:GOSUB1320:GOTO30
1520 POKE PO,PE
1530 Z=Z+1
1540 NEXTA
1550 CLS:PRINT"THE MEMORY IN THE FOLLOWING
LOCATION:"
1560 PRINT"DECIMAL:",:PRINTUSING"**###"; SA;:PRINT"
TO ";:PRINTUSING"**###";EA
1570 PRINT"HEXADECIMAL:",:S=SA:GOSUB1400:PRINTS$;"
TO ";:S=EA:GOSUB1400:PRINTS$
```

```
1580 PRINT:PRINT"HAS BEEN COPIED TO THE
FOLLOWINGLOCATION:"
1590 PRINT"DECIMAL:",:PRINTUSING"**###"; CA;:PRINT"
TO ";:PRINTUSING"**###";CA+EA-SA
1600 PRINT"HEXADECIMAL:",:S=CA :GOSUB1400 :PRINTS$;"
TO ";:S=CA+EA-SA:GOSUB1400:PRINTS$
1610 PRINT
1620 GOSUB1320:GOTO30
1630 'INSTRUCTIONS
1640 CLS:PRINT"1 USE THE NORMAL DECIMAL OR HEX
FORMAT THAT 'EXTENDED BASIC'USES, E.G. 142 OR &H8E.
THE EXCEPTION IS WHEN CHANGING A VALUE IN THE
'EXAMINE' MODE; USE ONLY HEX THEN.";
1650 PRINT" WHEN CHANGINGMEMORY ENTER TWO HEX
DIGITS ONLYAND PRESS <ENTER>. IF YOU ENTER WRONG
VALUE, PRESS <DOWN> ARROW INSTEAD OF <ENTER>."
1660 PRINT"2 THERE ARE NO COMMANDS TO REMEMBER.
EACH OPERATION WILL PROMPT YOU WHAT YOU NEED TO
DO AT THAT TIME."
1670 GOSUB1300
1680 PRINT"3 THE BREAKPOINT IN THE DEBUG-GER IS
REALLY A RETURN FROM SUB-ROUTINE. YOU MUST
REMOVE IT FROMTHE ML PROGRAM WITH <Y>ANK."
1690 GOSUB 1320:GOTO30
1700 'DEBUGGER
1710 CLS
1720 PRINT@0,"<B>REAKPOINT <Y>ANK REAKPOINT"
1730 PRINT@35,"<G>O ML PROGRAM <M>ENU"
1740 PRINT@64,STRING$(32,"=");
1750 PRINT@480,"COMMAND";:INPUTQ$
1760 IF Q$="B" GOTO 1780 ELSE IF Q$="Y" GOTO 1840
ELSE IF Q$="G" GOTO 1880 ELSE IF Q$="M" GOTO 30
ELSE GOTO1750
1770 'SET BREAKPOINT
1780 INPUT"ADDRESS";BA
1790 IF A$="Y" THEN PRINT"BREAKPOINT ALREADY AT
$";HEX$(YA):GOTO 1720
1800 A=PEEK(BA):YA=BA:A$="Y"
1810 POKE BA,&H39
1820 GOTO 1720
1830 'YANK BREAKPOINT
1840 IF A$="N" THEN PRINT"NO BREAKPOINTS":GOTO1720
1850 POKE YA,A:A$="N"
1860 GOTO1720
1870 'GO TO ML PROGRAM
1880 INPUT"ADDRESS";GA
1890 DEFUSR0=GA
1900 B=USR(0)
1910 PRINT"RETURNED AT $";HEX$(BA):PRINT
1920 GOTO1720 ■
```

Captain 80

The adventures of a software secret agent

Bob Liddil

Max has been kidnapped by entities unknown. No ransom demand has been forthcoming so far. Meanwhile, I have set forth on a quest to save him. Downloaded into the world of various programs, I have succeeded in locating the much sought after and quite elusive Professor Megabyte who had apparently taken up residence in the Computer Shack blockbuster arcade game Cyborg. The Professor rescued me from certain destruction by freezing time within the game, allowing escape. Meanwhile, Max languishes in an unknown location. His only hope of rescue is the sideslip codes that boot, at times, to move me through different programs toward his prison.

"You know," Professor Megabyte was telling me, as we made our way along the outskirts of Cyborg to his home in a graphics block, "things were a lot simpler when Adventure was king of the programs."

I nodded in agreement as he continued.

"Life was uncomplicated in this world when I first got here. I spent quite a while on Ed Juge's Star Trek, you know. That was a good program world. Nothing sinister, just your average everyday Klingons and the good old USS Enterprise."

He was getting misty eyed.

"Those were the good old days before high technology wrecked everything.

Suddenly the floor began to shake and the air grew heavy with electronic fog. The walls of the room fuzzed out just a little and it was difficult to breathe. Then everything returned to normal.

"What the deuce was that?" I exclaimed, thoroughly shaken.

"Oh, that was Twitch, testing his doomsday machine."

"Twitch? Who's Twitch?"

The Professor explained to me that Twitch was created initially a few years ago, actually, about the same time as Max, but aborted by its creator as being unsuitable for the gaming audience because of logic flaws. It had achieved sentience much the way Max had, through a power surge, and had taken up residence in High Ram where the realm of the programs seldom extends. Since that time it had been just sitting there malevolently drawing power from operational ram.

Sort of a despotic electronic taxation, to hear the Professor tell it. In all likelihood, he surmised, it was Twitch, and the minions of evil,

which have the power to mimic denizens of any game, that no doubt kidnapped Max.

As he told me all of this, he was packing his possessions into a small knapsack. He'd decided to come with me to help save Max. I was beginning to develop precognition about sideslipping and I knew our time in Cyborg was just about over. The Professor now was furiously pounding on a Pocket Computer with a telescoping antenna on it. He had just hit ENTER when the room de-rezzed. When everything came back into focus he was gone. I was alone again.

It took me a moment to get my bearings. Obviously, this was no arcade game, that much was clear. For I was standing in an open field, facing thirty Tygers, in company with a legion of giant bats, bears, beavers, several centaurs and a dragon. On the ground lay a cloak of invisibility, an orb of searching and a very deadly-looking sword. Suddenly I had a craving for a laser pistol and a good Wookie by my side. In the distance I could see the castle of Ra, the evil and all powerful.

Mommie . . .

The Sword of Zedek is from Krell Software and was gleaned from the shelves of The Program Store, where almost anything that has ever been published in the history of the TRS-80 can very likely be found. The zip-locked and monocolored documentation did little to attract me but I have always been a sucker for a dragon and the picture on the front sold me.

Sword of Zedek is not an adventure in the traditional word association sense. There are no particular puzzles to solve and much of the action is spent exploring the countryside looking for magic devices and collecting allies.

The operation of the program is alpha-keyed; that is, in response to "What do you want to do?" the player encoded <G> for GO or <T> for TAKE, <L> for LIGHT and so forth. The player will respond to inquiries for employment by non-player characters. "YOU MEET RIKKI, BANE OF SNAKES, HE WILL JOIN YOU FOR 1500 GOLD PIECES, WILL YOU PAY?" <Y>es and you have an ally, <N>o and he huffs off to join Ra to fight against you.

Now the structure of this game is such that the sophisticated player would write it off immediately. However, with a little scrutiny an amazing discovery emerges. The Sword of Zedek, for all its simplicity, is *fun!* It doesn't brain-bust like true Adventure and it doesn't require the dexterity of a NASA space chimp, like the arcades. It leaves nothing and yet everything to the imagination, so that by the time Ra attacks, ready or not, you want to see that sucker die!

I loved it! It gave me the same involvement as Taipan, the same wide judgmental ability as Santa Paravia and the classic personalization that has hallmarked many fantasy games that are now sadly fading into obscurity. For a kid, or someone who hasn't forgotten how to love the imaginary, this game is ideal.

I have the Torch of Zog (continuous light). I have the Sword of Zedek (powerful handweapon). I have myriad creature allies, including bats, beavers, centaurs, a dragon, trolls, orcs and a snake.

Ra attacks.

He strikes without warning, leading an army much like my own except for one difference. He leads 300 demons and the banes of many of my allies. The battle begins.

When the dust settles each of our armies lies in ruin. It is Ra and me alone in single combat. He rains blows down on my shield with endless energy but it does not yield. Soon I get the best of him and he is at my mercy. Then his shape begins to change from human form to something hideous and doubly evil. This is not part of the game. It is one of the minions of Twitch, disguised as Ra. Its face twists cruelly as it fills the air with laughter. I touch it with the sword and it explodes in white light and video noise. I can still hear the laughter.

I see the Professor coming up the stairs from a nearby dungeon, muttering as he does, and punching equations into his Pocket Computer. He grins tersely in satisfaction as he punches the <ENTER> key. The scene around us fuzzes out and a new scene swirls in. I still have the magical items that were in my possession at sideslip. Interesting.

The new scene is vaguely familiar, a street scene, gas lights and a sign: Madame Rosa's Massage Parlor. The Professor wanders onto a side street. I go forward to investigate.

Ah, Max, the things a fellow will do for his friends. Rescue is on the way. *To be continued.* ■

Exploring VisiCalc

Creating titles and split screens

Models I/II/III, PMC-80, LNW80

Timothy K. Bowman, Spokane, WA

This month as we continue our exploration of VisiCalc, let's consider how to use VisiCalc's screen-formatting capabilities so that you or another user of your VisiCalc-created spreadsheets can be more productive. VisiCalc has two key screen-formatting commands: Titles and Windows. Before we discuss each one of them, type in the example shown in Figure 1 (it's really a test of your knowledge of the label function). You'll note that Figure 1 is in reverse order from the way that you would normally type it in as it is a screen-to-printer dump using the command sequence "/SS:P". Alternately, you could save some typing and load a similar spreadsheet you may have previously created and stored on disk. Your spreadsheet must be larger, both horizontally and vertically, than the screen display and it must have labels in horizontal and vertical directions. Beyond that it doesn't matter for this demonstration whether it contains data or not.

Titles

The Titles function allows you to prohibit the cursor from entering certain portions of the screen. This is especially important when other people are using your VisiCalc-created spreadsheets so they don't replace label information with value information. It also allows key labels to remain on the screen despite any scrolling action that occurs. There's nothing quite like typing input material on line 63 of column AA with no visual description because that position's description is found in position A63 or AA1.

To demonstrate the use of the Titles function, position the cursor at position D5 and type /T. You will be presented with four choices in the upper left of the screen: Titles: H V B N. Since our example contains both horizontal and vertical labels press the B key. If you now use the arrow keys to move the cursor to the right and down and then try to move it back to the D5 position using the arrow keys you'll find that you can't. In order to move to any position which is to the left and/or above D5 you must use the GOTO command (right carat or shifted period) and answer the prompt with the location you wish to move the cursor to.

The H and the V represent horizontal and vertical. Again position the cursor at D5 and type /TV. You should find that once you move the cursor to the right using the arrow keys and then attempt to move it back to the left, it won't move past column E. Reposition the cursor at location D5 using the GOTO command and type /TH. Move the cursor down and to the right on the spreadsheet and then try to move up above line 6. You'll find that you can't.

In order to remove the Title function barrier, type /TN. Your cursor will now have unrestricted access to the screen.

There is a case in which the Titles function is undone. If you have fixed vertical titles encompassing columns A, B and C and increase the column width so that only column A shows on the screen, the Titles function will be undone. If you return to a column size that permits displaying columns A, B and C on the screen and you want the Titles function to work, you must reset it.

Split Screen

I have found that split screens are one of the more valuable VisiCalc aids when setting up worksheets. The VisiCalc user can have two entirely separate portions of the screen side-by-side or above one another so that important information in one portion of the screen can be used or compared with information in the other portion of the screen. It's an electronic version of the principle of divide and conquer.

To demonstrate the use of split screens let's use the example from Figure 1 and split the screen vertically using the Window command. Position your cursor in column F. Type /W. The prompt line should now show "Window: H V 1 S U." Press the H key and there should now be two screens. The right one

was formed in the column starting to the right of the cursor location. Your cursor is in the left screen and it can be moved anywhere on the screen without affecting the display of the right screen. To jump to the right screen press the semicolon (;) key. You now have unrestricted access in the right hand screen with no effect on the left screen display.

If you find that it would be desirable for the information in the two screens to scroll in a synchronized manner, position both screens on the same line and type /WS. Moving the cursor up and down in either screen will cause the other screen to move with it. If at any time you want to stop the scrolling feature, type /WU.

When you complete your viewing of the split screens and want to return to one screen, type /W1. If you want to have an upper and a lower screen, position the cursor in the row you desire and type /WH. Again, if you want the upper and lower screens to scroll together, type /WS.

When using split screens, it is important to note several features. You can use the Titles command which we discussed above. This adds even greater flexibility in designing and using viewing screens. Note that split screens are simply two different pictures of the same worksheet. If you create worksheet material in the right screen which does not show in the left screen, jump to the left screen and move to the same location as shown in the right screen. You'll observe that the information shown is the same. Another helpful use of split screens is to use one screen for an input area and the other for displaying the results of the calculations which can be from an entirely different section of your worksheet.

The visual display of our VisiCalc worksheets is important to the ease of their use. We spend more time viewing the screen than the VisiCalc program spends in recalculating our worksheet's results!

Please note the following corrections for the November, 1982, Exploring VisiCalc column on page 76:

C17 is +A21 + (.001*C14/(C14-C16))
A14 is +A2+@NPV(A3, B2.F2) downarrow
C21 is + C17 + (.001 * (C18/(C18-C20))) ENTER >E4 ENTER
E5 is /FR @ ABS(+C21-A21)<= .00001

If you have any questions, feel free to write me in care of *80-U.S. Journal*, 3838 South Warner Street, Tacoma, Washington 98409. Be sure to include a SASE if you desire a written response. Questions of a general interest may be included in a future Exploring VisiCalc. ■

---

(1) The VisiCalc program is copyrighted by VISICORP and VisiCalc is a registered trademark of VISICORP.

Figure 1

```
>A17:"Repairs
>A16:"Interest
>A15:"Postage
>B14:"essing
>A14:"Data Proc
>B13:"lities
>A13:"Other Uti
>B12:"ty
>A12:"Electrici
>A11:"Telephone
>B10:"e
>A10:"Automobil
>C9:"rtization
>B9:"ion & Amo
>A9:"Depreciat
>A8:"Insurance
>B7:"axes
>A7:"Payroll T
>A6:"Salaries
>A5:"EXPENSES:
>P4:"December
>O4:"November
>N4:" October
>M4:"September
>L4:" August
>K4:"  July
>J4:" June
>I4:"   May
>H4:"  April
>G4:"  March
>F4:"February
>E4:"January
/W1
/GOC
/GRA
/GFL
/GC9
/X>A1:>A1:
```

Com 80

Getting on-line

For all models

Donald L. Stoner, Mercer Island, WA

The November 1982, telecommunications issue revealed a suspected truth. A large number of our readers have telecommunications capability. The majority of the remainder are interested in the subject and plan to add this capability to their system in the near future.

Based on this revelation, *80-U.S. Journal* plans to present a regular column called COM 80, devoted to the subject of computer communications. We will work diligently to ensure that the information is of interest to the majority of readers.

The Ground Rules

First and foremost, the material presented will be written with the newcomer in mind. Most articles on telecommunications seem to assume the reader is an "ole pro" and understands completely what the author is discussing. "Ole pros" tend to forget that telecommunications is an entirely new area of interest for the beginning computerist.

Typically, Joe Keypresser has recently purchased a Color Computer, mastered the fundamentals of BASIC, bought a few games and heard vague mention of something called a modem. When Joe, or his counterpart . . . Jane Keypresser, read about Universal Asynchronous Receiver Transmitters (UARTs), parity bits, baud rates and the like, they are literally "blown away."

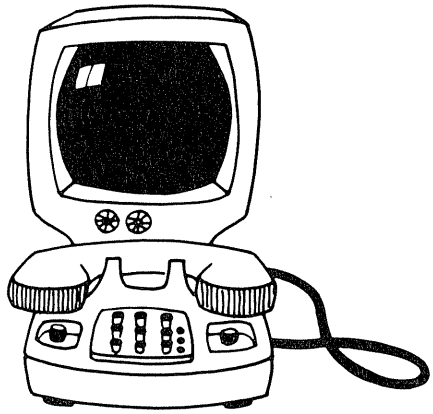
So . . . for all the Keypressers out there, this is for you.

One other "ground rule" should be mentioned. I will be avoiding discussion of the relative merits and/or features of commercial telecommunication products. There is a rather fundamental reason. I am deeply involved in the telecommunication business and (A) there is a strong desire and tendency to "crow" about my products and (B) it would pain me deeply to extol the virtues of a competitor's "gizmo." Thus, I'll leave this area to the New Products Editor.

I do need reader input, however. If there are specific areas that need explaining, please write to me in care of *80-U.S. Journal*. If your questions are of general interest, they will be used as the basis for a column. The things that confuse you, or are not easily understood, are probably of interest to the majority of readers.

What is Telecommunication?

The answer is many things to many people. To the radio amateur, it conjures up visions of chatting on a ham radio station. The telephone company thinks of it in terms of telephones . . . lots of telephones! In the context that we will be using it here, it means connecting computers together over the telephone network.



Depending on reader interest, and the number of readers who are Radio Amateurs (my call letters are W6TNS/7), we may extend the definition to include communications over two-way radios. There is a growing interest on the part of "hams" in connecting computers to their "rigs." Let us know if you would like this subject included in the column.

Sounds Great— How Do I Get Started?

You've probably read a dozen articles extolling the virtues of hitching your computer to the "twisted pair" (that's telephone jargon for the telephone line). Literally hundreds of pages could be written telling of all the marvelous things you can do when your computer is connected to another computer.

Unfortunately, no one tells you how to get started. Most articles imply that one should visit their favorite computer dealer to learn all about modems, terminal programs and so on. What these articles overlook is the fact that most people working in computer stores these days know less about telecommunications than you do! At least you read *80-U.S. Journal*. With the exception of Radio Shack Computer Centers, I have yet to see a

permanent telecommunication demonstration set-up in a computer store.

The answer to the question of how to get started is so simple and straightforward that everyone seems to have overlooked it. The best approach is this . . . visit a friend who is already on-line (has his or her computer connected to the telephone network). Most people are so pleased with what they are able to accomplish and how far they have extended their computing horizons, they are delighted to show off their system to someone else.

You can make contacts through your local computer club or associates at your place of employment. So many people have acquired modems (the device that couples your computer to the telephone network), that it should not be too difficult to find someone and arrange a demonstration.

I may be a relatively sophisticated "telecomputerist." Even so, I am still in awe when I see data streaming across the screen of my TRS-80, transmitted to me from hundreds, or even thousands, of miles away. Keep in mind that the "telecommunicator" is actually running a Digital Equipment Corp. PDP-11 or causing the disks to whir on a PRIME or IBM mainframe clear across the country.

I love telecommunicating and I think you will too. Unless you are a very jaded person, you cannot fail to be impressed the first time you plug in the modem and dial up a remote computer.

Whom Can I Keyboard With?

As mentioned earlier, that's a pretty large subject. There are three principal computer-to-computer categories which will be the subject of future columns. They are (1) communicating with another computer owner and exchanging programs, (2) contacting free bulletin board systems (BBSs) for low-cost electronic mail and (3) using the computer networks to communicate with national time-sharing services.

The last category applies to thousands of systems at the other end of your telephone line. However, there are two principal systems of interest to Mr. and Ms. Keypresser. They are CompuServe and The Source.

CompuServe — I must admit to favoritism regarding CompuServe Information Service. I have been a member of this group almost since its inception (my number is 70371,111 and electronic mail is answered at least once a week) and "cut my teeth" on this time-sharing system back in Columbus, Ohio. CompuServe uses Digital Equipment Corp. PDP computers and the system functions beautifully. I consider a membership ($19.95 at your local Radio Shack store) to be a real bargain.

Newcomers, particularly, appreciate the fact that CompuServe is "menu driven." This means that as soon as you check in, you are presented with a selection of menu choices, asking what you want to do. You enter a number, selected from the menu. The system then presents you with either another menu or the information you have requested. In

popular jargon, it is very "user friendly."

I started the nation's second bulletin board, called Hamnet (Richard Taylor, with MNET80, was the first). Today there are bulletin boards galore (now they are called SIGs, or Special Interest Groups) for Heath, CP/M, the TRS-80 Color Computer and dozens of other special areas such as photography, writing and so on.

The biggest group, in terms of membership and messages, is MNET80. This system is devoted II, III and, of course, the new Model II, II and, of course, the new Model 16. This is where the "pros" hang out, discussing the latest bug they have found in someone's software, praising or damning the latest "gizzwhiz" designed to make their TRS-80 more (or less) efficient. If you are considering the acquisition of some new hardware and/or software, all you have to do is ask the members what they know about it and/or the company that sells it. You'll soon learn if it is a bargain or a "beast." You can easily recover your membership costs if you avoid even one "unfortunate" purchase. To access this system, type R MNET80 from command level (when you see the 'OK' prompt).

I am the SYSOP (system operator) of a bulletin board or SIG devoted to telecommunications. To access it, simply type R MCONN from command level (the letters OK) and CompuServe's magic carpet will whisk you away. There is no charge for membership, just leave me a message requesting that you be allowed access to the programs and database.

Hamnet is an extremely active SIG and many prominent radio amateurs check in at least once a week. To access, type R HAMNET.

The Source — I have also been a member of The Source since its inception and have experienced its "growing pains" (each and every one!). The original management never anticipated the flood of members that would join (even at $100 per!). Thus, the equipment (PRIME computers) quite often went into shock and response delays were in the order of minutes, rather than seconds. After the Reader's Digest organization bought The Source, there was significant improvement.

My identification number on this information utility (another name for a database or collection of data) is TCD142. I do not check in for messages as often as on CompuServe.

The Source has an incredible amount of information available, particularly data of interest to businessmen and professionals. Finding your way around this database is something like playing an Adventure game, with dozens (no, make that hundreds) of data "caves" to explore. The Source does provide you with exceptionally good documentation to assist you in your quest for information. I recommend that you read it thoroughly a couple of times before you dial them up.

In the next issue, we will be discussing some "sugar coated" fundamentals of telecommunications so you can have an insight into how this miracle of modern technology is accomplished. ■

In the chips

The first in a new series on machine language programming

Models I/III, PMC-80, LNW80

Spencer Hall, Associate editor

If you have a superbrain . . . if you can handle complex concepts and follow the most complicated logic with ease, you are not qualified to read this series. It's about some of the theory behind microcomputers. These ideas are so simple that only a person with the precious gift of normal intelligence can really appreciate them.

Too much intelligence can be a real handicap for a computer programmer. It's an even bigger handicap for anyone who wants to write about computers. It takes a simple mind to understand computers well enough to tell others about them. That's why the editor chose me to write this series.

Let's begin with a line from Henry Wadsworth Longfellow's poem, "The Midnight Ride of Paul Revere." If and when British troops come to put down the threatened rebellion, Paul plans to warn the Massachusetts farmers. Are the British going to disembark at Boston or are they going to march in from another landing site? He has to tell the farmers which way to point their guns. He's going to ride thru Middlesex County on the north shore of the Charles River. His friends are across the river in Boston. To save time, he tells them to signal with lights from the tower of the old North Church when they learn which way the British are coming. He says, according to Longfellow:

"One if by land and two if by sea and I on the opposite shore will be."

Simple? Yes! A modern computer specialist would call this scheme "telecommunication via electromagnetic radiation in the visible portion of the spectrum using a binary byte consisting of two bits." These words, of course, stand for some very simple ideas. First of all, the telecommunication medium was ultra-sophisticated. No modern technology could get the message across the Charles River any faster than the speed of light. Beyond that fact, the scheme contains a number of things in common with computer theory. Each lamp was a bit. Like a bit, each one could be either on or off. There were two lamps on the tower, so the byte had a length of two bits. It meant one thing if one bit was on or "high" and another if both bits were "on." The information in the tower was in a higher level language . . . spoken English. Revere's informants mentally "compiled" it into a primitive binary code and "input" it to the lamps. After it was transmitted via electromagnetic radiation, Paul had "programmed" himself to convert the binary data into an English message which he then proceeded to "output" to the farmers at the top of his voice.

I told you this was going to be simple!

How many messages could have been sent with those two bits if Paul and his friends had thought it necessary? Two? Wrong. What if the British didn't come? Longfellow doesn't say, but I suspect the tower would have remained dark and Revere would have gone back to bed. Two bits low. The British aren't coming.

Suppose that it had been important to know if the British were coming overland from the north shore or the south shore. Could two lamps have told Revere which? They could have if one was on each side of the tower. One side high, North shore . . . other side high, South shore. What would two lamps have meant then? They're coming both ways . . . what else? Four different messages could have been sent with that two-bit byte. Now it would have required a third lamp in the middle of the tower to mean the British were coming by sea. Actually, with a three-lamp byte they would have sent eight messages. That would have ruined Longfellow's poem.

A bit "high" in a microcomputer can be a wire with a current passing through it (usually at five volts, more or less). A bit "low" can be another wire with a current of lower voltage (usually in the neighborhood of one volt). The majority of bytes in today's microcomputers contain eight bits. Often two bytes are combined to produce a sixteen-bit byte. That's one reason why computer gear has to be connected with ribbon cables. When I think of a byte, I always see eight light bulbs in a row. If I were Paul Revere I would see eight whale oil lamps. That, as we just said, means eight conductors, collectively called a data bus. Often these conductors are so small they are called traces. Inside a computer chip they are so tiny you could almost measure their width in molecules. Computers were originally designed to compute, so the "message" of each bit in a data byte is a number. The right hand bit means 1. The bit next to it stands for 2, the next one, 4, etc. Each is double the last until the leftmost bit stands for 128. A byte with all bits off is, logically enough, zero. A byte with all eight bits on will, if you add all bit values, equal 255. Take my word for it or add them up for yourself and see. Now you know why computer types insist on numbering things by starting at 0 instead of at 1 like more sophisticated people.

To show which bits are on in an eight-bit byte, we use ones and zeroes. This makes for some terrifying numbers that no really smart people ever try to read. If

you are like me and panic whenever you see a binary byte in print, take 00001111 minutes out right now and enter the BASIC listing below. If you're a good typist you can probably do it in 00001010 minutes. If you try to type it in 00000101 minutes, you'll probably get syntax errors when you try to use it.

Now RUN this program and have fun entering decimal numbers. The computer will tell you which bits are on and what their values are. It will also show you the horrible result in 0's and 1's. Just for kicks, start with a small number such as six. Enter it and see the bit pattern. Then watch what happens to the pattern when you double the number (twelve) and keep doubling it. Play with this program and you will find yourself on easy terms with the basic idea behind all digital computers. It's good for your nerves, but don't forget to come back. There's more.

After you read this next paragraph you can forget about bits forever, unless you plan to write machine language programs. Enter 17 in the program you have just been playing with. You get a binary byte of 00010001, don't you? If you were too lazy to enter the program, that's okay, but then you'll just have to believe me. The program told you that the 1 on the right stood for one. It also told you that the other 1 stood for sixteen. That's why the byte has a value of 17. Now listen closely, because this is real neat. Let's put a hyphen in the middle of this byte and get 0001-0001. Each half, taken alone, stands for one. Multiply the left hand 1 by the value it represents and you get sixteen. Now just add the right hand one and you get 17. For clarity why not represent the values of these two parts by writing 1-1, which means 1 times 16 + 1 times 1. Now try the same thing on byte 00100010. Break it up, so: 0010-0010. Each half has a value of 2 and in our new number system this would be 2-2. The first two would mean two times sixteen and the other would mean + two. that totals 34. Check this with our program.

If we drop the hyphen, we've got a way to replace those 0's and 1's with just two familiar digits . . . or have we? What about byte 11111111 (or 1111-1111)? Each half has a value of 15. Sure enough, fifteen × sixteen + fifteen is 255, just the way it should be. But now we need some new digits to get us above 9. Let's use A for 10, and so on thru F for 15 and call it the hexadecimal system! "Hex" is Greek for "six" and "decem" is Latin for "ten." Put them together and you get "sixteen" in a sort of ancient pidgin language. Now 11111111 can be written FF as well as 255.

That's enough for one session. I'll leave you with a sobering thought. There is nothing stored in the memory of your TRS-80 except eight-bit bytes . . . numbers with a value anywhere from 0 to 255. It has to be that way. I told you microcomputers were simple gadgets! ■

We recommend that readers who wish to follow along with this series obtain a copy of DEBUG (R.S. Cat. #26-2000). For those of you with disks, the utility is already on your TRSDOS disk (Models I, II and III). —Ed.

Program Listing for In the Chips

```
10 CLS: CLEAR 1000
20 V$ = "128 64 32 16 8 4 2 1"
30 BE$ = STRING$(3,"140") + " "
40 FOR J = 1 TO 8: BR$ = BR$ + BE$: NEXT
50 PRINT @333,V$: PRINT @397,BR$
60 PRINT @0,: INPUT"DECIMAL BYTE";B
70 IF B<= 255 THEN 120
80 PRINT"BYTE TOO LARGE!"
90 FOR J = 1 TO 500: NEXT
100 PRINT @64,CHR$(208)
110 GOTO 60
120 T = 128
130 A = B/T
140 IF A<1 THEN BI$= BI$+"-"+ STRING$(4,32):
BT$=BT$+"0": GOTO 160
150 BI$ = BI$ + "*"+ STRING$(4,32): BT$= BT$+"1":
B=B-T
160 T = T/2: IF T<1 THEN 170 ELSE 130
170 PRINT @462,BI$
180 PRINT @724, "BINARY VALUE = "BT$
190 PRINT @853, "USE ANY KEY TO REPEAT"
200 PRINT @917, "OR <BREAK> TO EXIT"
210 Z$ = INKEY$
220 IF Z$="" THEN 210
230 BI$ = "": BT$ = "": GOTO 60
```

Radio Shack double density kit

Model I

Harry Avant, LaCrescenta, CA

Radio Shack has added double density capability to the venerable Model I. To me, the most surprising aspect of the Model I double density conversion is why Tandy waited so long. Percom, Aerocomp and LNW have been selling double density conversion kits for the Model I for quite a while.

Double density modification does not double the disk storage space, but does increase it by 80 percent. This will allow up to 175K bytes of storage on a 40-track data disk. Table 1 shows the storage capacity for various track sizes and disk configurations. A granule with Radio Shack's Double Density Disk Operating System is composed of three sectors, or 768 bytes. Each track contains six granules, or 4608 bytes per track.

Hardware modification consists of a small printed circuit board (about 3x4 inches) that plugs into the existing floppy disk controller chip socket. Radio Shack has apparently borrowed from their very good Model III double-density controller in designing the Model I conversion hardware. Three Western Digital chips, a 1791B, 2143 and 1691, are used here just as they are in the Model III.

A new Disk Operating System is supplied with the kit and here Tandy has done some interesting things. First, it should be noted that the manual states that a Model III directory can be read with the Model I double density operating system. This is not so. The new operating system will not read Model III disks, nor will the Model III read double density Model I disks. Some day, perhaps, the group that develops the Model III and the group that works on the Model I in Fort Worth may discover each other's existence. The continuing lack of compatibility between the two models is just not sensible. I have had compatibility between both types of computers for months now, using LDOS, NEWDOS/80 and DOSPLUS. For those of you who are considering the Radio Shack doubler with a non-Radio Shack operating system, the fact that NEWDOS/80 and LDOS both boot should be of interest. I tried to boot DOSPLUS 3.4 without luck. I suspect that DOSPLUS will have a patch for this new hardware very soon.

The new Radio Shack Disk Operating System does offer a large number of features that are more or less standard for the Model III. Nearly all of the Model III library commands have been added along with some new commands. A total of forty-six library commands are available in the new Disk Operating System.

An interesting command is Erase, which will clear all of the bytes of a file to zeroes. This will really "kill" a file, whereas, the regular kill only removes the filename from the directory but leaves the data on the disk. A single user-defined library command, LIB, can be created, which is a neat idea. As an example, if an audio beeper has been attached to the I/O port, a library command, beep, could be created to turn it on.

A CONFIG command has been added to allow the setting of number of tracks and stepping rates for the drives. BLINK will toggle the cursor from blinking to non-blinking. A BUILD and DO have been added, which allow for ease of turnkey type operations. Perhaps the best of the new library commands is WP, which will write protect a drive. This can be very useful in preventing the accidental over-writing of a file.

A COPY command that copies from single to double density, or

double to single density, is also included. BACKUP will backup and format to double density only. In order to format a single density disk, the original single density disk operating system must be used. PATCH is a library command that changes the contents of a file. This is bound to be useful for adding fixes to the operating system, as Tandy releases them.

FILFIX is a library utility that loads a file and allows the user to modify it. This utility is similar to NEWDOS Superzap, or DOSPLUS Diskzap.

FREE will display a free space map of the selected disk drive in the same manner as it does on the Model III. At first, this type of display takes a little getting used to. Its real advantage is to show not only how much space is left on a drive, but how segmented it is.

CREATE allows space to be reserved on a disk prior to writing a file to the disk. This can speed up disk I/O time for certain types of programs.

Several enhancements to Disk BASIC have also been added, including sixteen CMD statements called from BASIC. Table 2 is a listing of the new Disk BASIC statements.

The DOS manual lists several of the approved ROM subroutine entry points for use by assembly language programmers. As would be expected, there are no new calls listed here that have not already been well-defined in other Tandy publications.

The price listed for the double-density kit is $149.95 plus installation. This price, assuming the installation charges are reasonable, is very competitive with non-Radio Shack double-density modifications. Because of the similarity between the new Model I doubler and the Model III circuit design, this hardware should prove to be quite reliable.

Time will tell as to how good the Disk Operating System is. As with any new DOS, a few bugs are bound to have been missed. Radio Shack Computer Stores that have a TSR (Technical Service Representative) generally have current fixes for bugs. If you purchase a double-density kit and find problems with the Disk Operating System, it pays to check with the local computer store's TSR.

I would like to thank Steve Fobes, manager, and Dan Darby, TSR, of the West Los Angeles Radio Shack Computer Center, for making the double-density kit and operating system available for evaluation. ■

Table 1

Double Density Storage Capacity

	Tracks	Grans	Bytes
Full System Disk	35	114	87,552
	40	144	110,592
TRSDOS plus BASIC	35	120	92,160
	40	150	115,200
TRSDOS w/o BASIC	35	128	98,304
	40	158	121,344
Data disk	35	198	152,064
	40	228	175,104

Table 2
Disk BASIC Statements

CMD A Returns to TRSDOS with error message.
CMD B Enables or disables break key.
CMD C Deletes program remarks (REM) or spaces.
CMD D Displays disk directory for specified drive.
CMD E Displays previous TRS DOS error.
CMD I Executes a TRSDOS command.
CMD J Conversion of calendar date format.
CMD K Turns clock display on or off.
CMD L Loads an assembly language program.
CMD O Sorts a string array.
CMD P Checks printer status.
CMD R Enables real time clock interrupts.
CMD S Return to TRSDOS.
CMD T Disables real time clock interrupts.
CMD X Cross reference for BASIC programs.
CMD Z Dual routing screen to printer.
NAME Renumbers program lines.

PMC SOFTWARE

Proven Software Solutions for the TRS-80, PMC-80 & 81 User At VERY Special Prices

GAMES

Title	List	Our Price
Advanced Operating Systems		
+ VOYAGE OF THE VALKYRIE		
D 32K I & III	39.95	32.00
T 16K I & III	29.95	24.00
Adventure International		
ARMOR PATROL		
D 32K I & III	24.95	19.00
T 16K I & III	19.95	15.00
COMBAT		
D 32K I & III	20.95	16.00
T 16K I & III	19.95	15.00
CONQUEST OF CHESTERWOOD		
D 32K I & III	20.95	16.00
T 16K I & III	19.95	15.00
+ DEMON VENTURE #1— REIGN OF THE RED DRAGON		
D 32K I & III	24.95	19.00
ELIMINATOR		
D 32K I & III	24.95	19.00
T 16K I & III	19.95	15.00
GALACTIC EMPIRE		
T 16K I & III	14.95	12.00
GALACTIC REVOLUTION		
T 16K I & III	14.95	12.00
GALACTIC TRADER		
T 16K I & III	14.95	12.00
+ GALACTIC TRILOGY		
D 32K I & III	39.95	30.00
LUNAR LANDER		
D 32K I & III	20.95	16.00
T 16K I & III	14.95	11.00
MACES & MAGIC #1—BARLOG		
D 32K I	29.95	23.00
MACES & MAGIC #2— STONE OF SISYPHUS		
D 32K I	29.95	23.00
MACES & MAGIC #3— MORTON's FORK		
D 32K I	29.95	23.00
MISSILE ATTACK		
D 32K I & III	20.95	16.00
T 16K I & III	14.95	11.00
OTHER VENTURE #2— CROWLEY MANOR		
D 32K I & III	20.95	16.00
T 16K I & III	19.95	15.00
OTHER VENTURE #3— ESCAPE FROM TRAAM		
D 32K I & III	20.95	16.00
T 16K I & III	19.95	15.00
OTHER VENTURE #4— EARTHQUAKE SF 1906		
D 32K I & III	20.95	16.00
T 16K I & III	19.95	15.00
OTHER VENTURE #5—		
+ DEATH PLANET		
T 16K I & III	19.95	15.00
PLANETOIDS		
D 32K I & III	20.95	16.00
T 16K I & III	19.95	15.00
SCOTT ADAMS' ADV. #1, #2 & #3		
D 32K I & III	39.95	31.00
SCOTT ADAMS' ADV. #4, #5 & #6		
D 32K I & III	39.95	31.00
SCOTT ADAMS' ADV #7, #8 & #9		
D 32K I & III	39.95	31.00
SCOTT ADAMS' ADV #10, #11 & #12		
D 32K I & III	39.95	31.00
SEA DRAGON		
D 32K I & III	24.95	19.00
T 16K I & III	19.95	15.00
SHOWDOWN		
D 32K I & III	20.95	16.00
T 16K I & III	14.95	12.00
+ SILVER FLASH-PINBALL 1		
D 32K I & III	19.95	15.00
T 16K I & III	14.95	12.00
SKY WARRIOR		
D 32K I & III	20.95	16.00
T 16K I & III	14.95	11.00
SPACE INTRUDERS		
D 32K I & III	20.95	16.00
T 16K I & III	19.95	15.00
STAR FIGHTER		
D 32K I & III	29.95	23.00
T 16K I & III	24.95	19.00
STAR TREK 3.5		
D 32K I & III	19.95	15.00
T 16K I & III	14.95	11.00
TREASURE QUEST		
D 32K I & III	19.95	15.00
T 16K I & III	14.95	11.00
Automated Simulations (EPYX)		
CRUSH, CRUMBLE & CHOMP		
D 32K I & III	29.95	23.00
T 16K I & III	29.95	23.00
DUNJONQUEST— HELLFIRE WARRIOR		
D 32K I & III	39.95	30.00
T 16K I & III	39.95	30.00
DUNJONQUEST— KEYS OF ACHERON		
D 32K I & III	19.95	15.00
T 16K I & III	19.95	15.00
DUNJONQUEST— SORCERER OF SIVA		
D 32K I & III	29.95	23.00
T 16K I & III	29.95	23.00
DUNJONQUEST— TEMPLE OF APSHAI		
D 32K I & III	39.95	30.00
T 16K I & III	39.95	30.00
DUNJONQUEST— UPPER REACHES OF APSHAI		
D 32K I & III	19.95	15.00
T 16K I & III	19.95	15.00
+ RICHOCHET		
D 32K I & III	19.95	15.00
T 16K I & III	19.95	15.00
Big 5		
ATTACK FORCE		
D 32K I & III	19.95	15.00
T 16K I & III	15.95	12.00
COSMIC FIGHTER		
D 32K I & III	19.95	15.00
T 16K I & III	15.95	12.00
DEFENSE COMMAND		
D 32K I & III	19.95	15.00
T 16K I & III	15.95	12.00
GALAXY INVASION		
D 32K I & III	19.95	15.00
T 16K I & III	15.95	12.00
METEOR MISSION		
D 32K I & III	19.95	15.00
T 16K I & III	15.95	12.00
ROBOT ATTACK		
D 32K I & III	19.95	15.00
T 16K I & III	15.95	12.00
+ STELLAR ESCORT		
D 32K I & III	19.95	15.00
T 16K I & III	15.95	12.00
SUPER NOVA		
D 32K I & III	19.95	15.00
T 16K I & III	15.95	12.00
Cornsoft		
+ BOUNCEOIDS		
D 32K I & III	19.95	15.00
T 16K I & III	15.95	12.00
SCARFMAN		
D 32K I & III	19.95	15.00
T 16K I & III	15.95	12.00
SPACE CASTLE		
D 32K I & III	19.95	15.00
T 16K I & III	15.95	12.00
Med Systems		
ASYLUM		
D 32K I & III	22.95	18.00
T 16K I & III	19.95	15.00
+ ASYLUM II		
D 32K I & III	22.95	18.00
T 16K I & III	19.95	15.00
DEATHMAZE 5000		
D 32K I & III	17.95	14.00
T 16K I & III	14.95	11.00
LABYRINTH		
D 32K I & III	17.95	14.00
T 16K I & III	14.95	11.00
WARRIOR OF RAS— DUNZHIN		
D 48K I & III	29.95	23.00
T 48K I & III	29.95	23.00
WARRIORS OF RAS— KAIV		
D 48K I & III	29.95	23.00
T 48K I & III	29.95	23.00
WARRIORS OF RAS— THE WYLDE		
D 48K I & III	29.95	23.00
T 48K I & III	29.95	23.00
Melbourne House Software		
PENETRATOR		
D 32K I & III	24.95	20.00
T 16K I & III	24.95	20.00

Now Twice As Many Programs!

EDUCATION

Advanced Operating Systems		
MOSTLY BASIC EDUCATIONAL PKG. 7 PRG.		
T 16K I & III	24.95	20.00
MOSTLY BASIC SCIENTIFIC PKG. 7 PRG.		
T 16K I & III	24.95	20.00
TIME DUNGEON— AMERICAN HISTORY		
T 16K I & III	24.95	20.00
TIME DUNGEON— + WORLD HISTORY		
T 16K I & III	24.95	20.00
Automated Simulations (EPYX)		
JABBERTALKY		
D 32K I & III	29.95	23.00
T 16K I & III	29.95	23.00

BUSINESS

Small Business Systems Group		
ACCOUNTS PAYABLE		
D 48K I & III	195.00	155.00
ACCOUNTS RECEIVABLE		
D 48K I & III	195.00	155.00
GENERAL LEDGER		
D 48K I & III	195.00	155.00
INVENTORY CONTROL		
D 48K I & III	195.00	155.00
PAYROLL		
D 48K I & III	195.00	155.00

WORD PROCESSING

Aspen Software		
GRAMMATIK		
D 32K I	59.00	46.00
PROOF EDIT		
D 32K I	30.00	23.00
PROOFREADER		
D 32K I	54.00	42.00
SOFT-SCREEN		
D 48K I	69.00	54.00
SOFT-TEXT		
D 48K I	69.00	54.00
Michael Shrayer Software		
ELECTRIC PENCIL		
T 16K I	100.00	24.00

INFO. PROCESSING

Adventure International		
+ MAXI CRAS		
D 48K I & III	99.95	79.00
MAXI MANAGER		
D 48K I & III	99.95	79.00
+ MAXI MANAGER UTILITY PACK #1		
D 48K I & III	49.95	39.00
+ MAXI STAT		
D 48K I & III	199.95	147.00
Auk's		
+ COMPUTER FILING SYSTEM (CFS)		
D 32K I & III	69.00	55.00
Dan Haney Associates		
ELECTRIC SPREADSHEET		
D 32K I & III	69.95	65.00
T 16K I & III	34.95	30.00

OPERATING SYS.

Micro Systems Software		
DOS PLUS V3.3S		
D 32K I	100.00	80.00

UTILITIES

Adventure International		
+ DIRECTORY INFO. MGT II (DIM II)		
D 32K I & III	24.95	19.00
Howe Software		
MON3		
T 16K I	39.95	31.00
MON4		
D 32K I	49.95	39.00
STERM		
T 16K I	69.95	55.00
SYSTEM DIAGNOSTIC		
D 32K I & III	99.95	79.00

MISCELLANEOUS

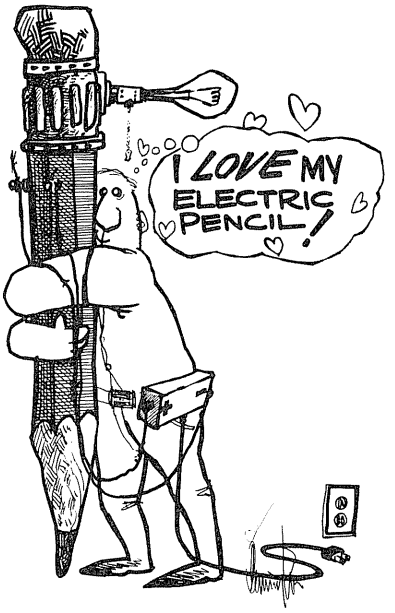
Adventure International		
HINT SHEET for single Adventure #1 thru #12	1.00	1.00
Software Affair		
ORCHESTRA-85		
T 16K I	99.95	95.00
D 16K I	99.95	95.00

+ NEW # 44

PMC SOFTWARE
475 Ellis St., Mt. View, CA 94043
Orders: (415) 968-1604

Terms: FOR FAST DELIVERY, send certified checks, money orders, VISA or MasterCard number and expiration date. Personal checks require 3 weeks U S A sales only. PRICES INCLUDE UPS continental delivery (do not use P.O. Box). CALIFORNIA customers add 6½% tax. Prices subject to change

Word Processing Software for TRS-80* Model I



World's most popular word processing package. Fully proven Electric Pencil* now enhanced and offered in 16K cassette version only.

- Easiest to learn
- Simple to operate
- Full screen editing
- Global search & replace
- Powerful insert/delete
- Dynamic print formatting
- No control key mod.

Send check or MO for $24.00 plus sales tax for Calif. cust. UPS shipping in USA prepaid.

$24.00

$10.00 for manual only # 45

*Electric Pencil licensed to PMC Software by Michael Shrayer. TRS-80 trademark of Tandy Corp.

PMC SOFTWARE
475 Ellis St. Mtn. View, CA 94043
Orders (415) 962-0318
Info (415) 962-0220

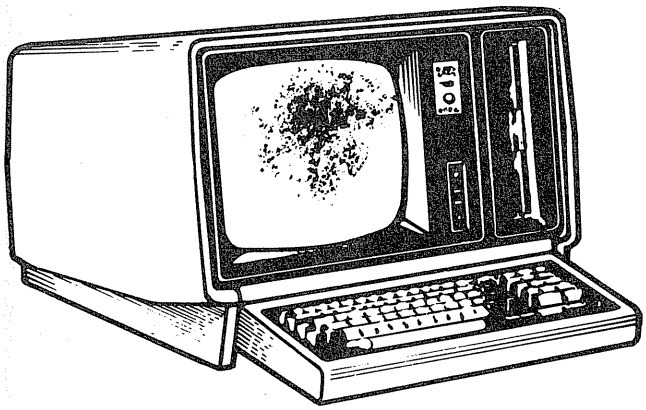
Supervisor calls

Part II: Flipping screens on the Model II and the beginning of a disk directory program

Model II

T. R. Dettmann, Associate editor

Another very useful SVC is the BIDRAM routine (SVC 94). Its purpose is to take a portion of regular memory and instantaneously swap it to the video screen or swap the video screen into RAM. With this available, it's almost possible to treat the screen as a part of memory without knowing anything about the technical



details of how the screen is really accessed.

This SVC is a step more complicated, though, than the scroll protect. In order to make use of it, we have to have a buffer in memory. We're using an integer array to reserve the buffer space and we have to let the routine know where it is.

The explanation of VIDRAM (owner's manual TRSDOS page 4/44A) notes that to use it, we have to do the following: 1. Place 94 in the A register. 2. Place a code in the B register, 0 to go from RAM to video, <>0 to go from video to RAM. 3. Place the address of the buffer in RAM in the HL register, this buffer must be 1920 bytes long in 80-character mode and 960 in 40-character mode. To do this, we'll make use of the fact that when we execute a USR routine instruction, the argument in parentheses is made available to the USR routine if it wants it. We'll use this to pass the address of the buffer in memory.

We'll also create two copies of the routine, one for video to RAM and the other for RAM to video. The basic routine is composed of the numbers in the data statement in line 130. It goes like this:

Get the buffer address into the HL register	94	35	86	235
Load 94 into the A register	62	94		
Load 0 into the B register	6	0		
Execute the RST 8	207			
Return to the BASIC program	201			

Line 140 creates the routine for RAM to video (B register 0). In line 150, we set CD(8)=1 (the number which goes into the B register) so it will form the video to RAM routine.

Now that we have created the two machine language routines, we can execute them in the subroutines at 1000 and 1020. In the sample program, line 200 clears both buffers and puts an identifying line in each. Then the subroutine at 2000 is executed.

Subroutine 2000 allows us to move around the screen with the arrow keys and type characters on the screen anywhere. Hitting the F1 key will cause the current video display to be dumped into memory and the other display to be swapped to the screen by doing a GOSUB 2300 which executes the VIDRAM routines in the proper order.

If you look carefully, you'll notice that the memory buffer (array BUF) has more space allocated than

needed. Since each array location in an integer array is 2 bytes, I only need 960 locations per screen. That's a total of 1920 array locations for two screens. After running the program, list BUF completely to see how the screens are stored. The unused portions of the array will form a background of zeroes in your listing.

So You'd Like a Disk Directory?

The final example is the beginning of a Model II directory program that you can use to catalog your disks. The critical SVC here is called RAMDIR (SVC 53). It is explained in the owner's manual on TRSDOS, pages 4/60D and 4/60E.

RAMDIR is our most complicated routine yet. In order to use it, we have to provide a buffer in RAM of up to 3265 bytes in length, and we have to set the A, B, and C registers. The routine that does this is in line 130 of the sample program. It goes like this:

Get the buffer location into the HL register	94	35	86	235
Load 0 into the B register	6	0		
Load 0 into the C register	14	0		
Load 53 into the A register	62	53		
Execute the RST 8	207			
Return to the BASIC program	201			

As explained in the manual, the B register is to hold the drive number that we want to look at (0, 1, 2, or 3) and the C register is loaded as follows:

0	Get the entire directory
1-96	Get only one record
255	Get free space only

The sample program makes use only of the 0 option in the C register.

Subroutine 1000 sets up the RAMDIR routine by first placing the disk number (DN) into CD (6) which will be loaded into the B register and then, using the same technique as before, build the USR routine. Once built, subroutine 1100 executes the USR call to get the whole directory into the memory buffer (integer array BUF).

The second page (4/60E) in the manual describes the layout of the directory in RAM in detail. All we're going to do is pull out the filenames, sort them, then print them.

Each individual directory entry is 34 bytes long and starts with a colon (:). If a # occurs instead of ":", it marks the end of the directory.

Bytes 2-16 of each entry are the filename, extension, and drive number in standard format. The subroutine at line 1300 knows about this layout and it extracts the filenames from the integer buffer and converts them into strings which it stores in the array NM$.

To break out the names, each integer position where a filename should be is examined. The first eight locations are broken up into individual bytes, converted to their equivalent string characters, and added into the string X$. If the first character is not ":", it's assumed that there are no further entries (we really should check for #,

but this is good enough for the demonstration).

Once we have all the entries, subroutine 1400 sorts them and subroutine 1500 prints them after first scroll protecting the screen.

We won't go into more detail here on the directory program, but in the next article, we'll expand the directory program to include a database management system that can keep a complete directory of your Model II disks.

Final Note

In looking at using several of the SVC's, we've worked out a number of routines for specific purposes that are able to load the A, B, C, and HL registers and execute an SVC. Try experimenting with these routines and other SVC's that interest you. Knowing how to load these four registers will give you access to a wide range of SVC's.

For those of you who would like a more convenient reminder of the SVC requirements, NANOS Systems, P. O. Box 24344, Speedway, IN 46224, (317) 244-4078, introduced a Model II SVC reference card which I used heavily in writing this article. The information I found didn't always satisfy me, but it was a real help that kept me from having to go back to the massive owner's manual over and over again. I highly recommend it if you're going to be using SVC's quite a bit.

Nanos reference cards are available for the Models I, II, III and Color Computer for BASIC assembler languages and system utilities. They range in price from $2.95 to $5.95. Cards are also available for other computers and peripherals. ■

Listing 1 — Supervisor Calls

```
20 REM
30 REMMODEL II SCREEN FLIP TEST
40 REM(C) 1982 BY TERRY R. DETTMANN
50 REM
60 REMVERSION 0.009/82
70 REMFILENAME: FLIP/BAS
80 REM
90 REM
100 CLEAR1000:DEFINTA-Z
110 DIM BUF(4000),CD(10),SV(10)
120 FORI=1TO10:READCD(I):NEXTI
130 DATA 94,35,86,235,62,94,6,0,207,201
140 J=0: FORI=1TO10STEP2:
SV(J)=CVI(CHR$(CD(I)) + CHR$(CD(I+1))):
J=J+1: NEXTI
150 J=5: CD(8)=1: FORI=1TO10STEP2:
SV(J)=CVI(CHR$(CD(I)) + CHR$(CD(I+1))):
J=J+1: NEXTI
200 CLS: PRINT"SCREEN 2": X=0: GOSUB1000: CLS:
PRINT"SCREEN 1": X=960: GOSUB1000
210 PRINT"ENTER A SCREEN FULL OF DATA"
220 GOSUB2000
999 END
1000 REM SCREEN TO BUFFER
1010
```

```
Y=0:DEFUSR0=VARPTR(SV(5)):Y=USR0(VARPTR(BUF
(X))):RETURN
1020 REM BUFFER TO SCREEN
1030
Y=0:DEFUSR0=VARPTR(SV(0)):Y=USR0(VARPTR
(BUF(X))):RETURN
2000 REM PUT STUFF ON THE SCREEN
2010 Y=11*80+40:Y1=Y:Y2=Y
2020 PRINT@Y,;:GOSUB2200
2030 IF ASC(C$)=1 THEN GOSUB2300:GOTO2020
2040 IF ASC(C$)=2 THEN CLS:RETURN
2050 IF ASC(C$)=8 THEN Y=Y-1:PRINT@Y," ";
2060 IF ASC(C$)=28 THEN Y=Y-1
2070 IF ASC(C$)=29 THEN Y=Y+1
2080 IF ASC(C$)=30 THEN Y=Y-80
2090 IF ASC(C$)=31 THEN Y=Y+80
2100 IF ASC(C$)=13 THEN Y=(ROW(X)+1)*80
2110 IF Y<0 THEN Y=Y+80 ELSE IF Y>1918 THEN
Y=Y-80
2120 IF ASC(C$)<32 OR ASC(C$)>127 THEN 2020
2130 PRINT@Y,C$;:Y=Y+1:IF Y>1918 THEN Y=Y-1
2140 GOTO2020
2200 REM GET A CHARACTER
2210 C$=INKEY$:IFC$=""THEN2210ELSERETURN
2300 REM SWAP SCREENS
2310 IF X=960 THEN Y1=Y: GOSUB1000: X=0:
GOSUB1020: Y=Y2 ELSE Y2=Y:GOSUB1000:
X=960:GOSUB1020:Y=Y1
2320 RETURN
```

Listing 2 — Supervisor Calls

```
10 REM
20 REM
30 REMMODEL II DISK DIRECTORY
40 REM(C) 1982 BY TERRY R. DETTMANN
50 REM
60 REMVERSION 0.109/82
70 REMFILENAME: DIRECT/BAS
80 REM
90 REM
100 CLEAR1000:DEFINTA-Z
110 DIM BUF(1636), SV(5), CD(32), CV(10),
NM$(96)
120 FORI=1TO18:READCD(I):NEXTI
130 DATA 94, 35, 86, 235, 6, 0, 14, 0, 62, 53, 207, 201
131 DATA 6, 0, 62, 27, 207, 201
160 DEFFNHDR$(X$) =
STRING$((78-LEN(X$))/2,150) + " " + X$ + "
"+STRING$((77-LEN(X$))/2,150)
200 REM MAIN LOOP
210 CLS:PRINTFNHDR$("DISK
DIRECTORY"):PRINT:PRINT
220 LINEINPUT"DISK NAME & NUMBER (DEFAULT 0)
===> ";DN$
```

Supervisor calls

```
230 L=INSTR(DN$,":")
240 IF L=0 THEN DN=0 ELSE
DN=VAL(MID$(DN$,L+1)):DN$=MID$(DN$,1,L-1)
250 IF DN<0 OR DN>3 THEN PRINT"DISK NUMBER
ERROR":GOTO220
260 GOSUB1000
270 GOSUB1100
280 GOSUB1300
290 GOSUB1700
300 GOTO200
999 END
1000 REM SETUP DIRECTORY CALL
1010 CD(6)=DN:J=0
1020 FORI=1TO12STEP2: SV(J)=CVI(CHR$(CD(I)) +
CHR$(CD(I+1))): J=J+1: NEXTI
1030 RETURN
1100 REM GET DISK DIRECTORY
1110 Y=0: DEFUSR0=VARPTR(SV(0)):
Y=USR0(VARPTR(BUF(0)))
1120 RETURN
1300 REM PULL OUT NAMES, SORT & DISPLAY
1305 NM=0
1310 FORI=1TO96
1320 X$="": FORJ=1TO8: K=(I-1)*17+J-1
1330 X!=BUF(K):IF X!<0 THEN X!=65535-X!
1340 X1=INT(X!/256): X2=X!-X1*256
1350 X$ = X$ + CHR$(X2) + CHR$(X1)
1360 NEXTJ
1362 IF MID$(X$,1,1)<>":" THEN 1385
1365 Z=INSTR(2,X$,":")
1370 NM$(I)= MID$(X$,2,Z-2): NM=NM+1
1380 NEXTI
1385 GOSUB1400:GOSUB1500
1390 RETURN
1400 REM SORT DIRECTORY ENTRIES
1410 GP=NM
1420 IF GP<=1 THEN RETURN
1430 GP=INT(GP/2)
1440 FG=0
1450 FORI=1TONM-GP
1460 IF NM$(I)<=NM$(I+GP) THEN 1480
1470 SWAP NM$(I),NM$(I+GP):FG=1
1480 NEXTI
1490 IF FG=1 THEN 1440 ELSE 1420
1500 REM PRINT DIRECTORY
1505 CLS: PRINTFNHDR$("DIRECTORY"): CD(14)=1:
GOSUB1600
1510 FORI=1TONM:PRINTNM$(I):NEXTI
1520 LINEINPUT"PRESS ENTER TO CONTINUE";A$
1530 RETURN
1600 REM SCROLL PROTECT
1610 J=0: FORI=13TO18STEP2:
CV(J)=CVI(CHR$(CD(I)) + CHR$(CD(I+1))):
J=J+1: NEXTI
1620 Y=0: DEFUSR3= VARPTR(CV(0)): Y=USR3(0)
1630 RETURN
1700 REM SAVE DIRECTORY ENTRIES
1710 RETURN
```

Basically BASIC

User-defined functions

For all models

James A. Conrad, Seattle, WA

"There's no dubiosity about it, Dandy, Manu Korzeniowski has got to be, without uncertaintude, the most undah-rated playah in the NFL!"

"That's for sure, Howard! Why, on any given Sunday . . ."

I'd like to acquaint you with the most undah-rated playah in the BASIC league: the user-defined function known as 'DEFFN.'

Like most hackers I know, upgrading to disk so excited me that I breezed past the Disk BASIC manual's discussion of the DEFFN statement. Get back to that, I figured, after I get my tapes onto disk and play with those neat random access files. And, whenever I glanced at the section on DEFFN, I was disenchanted by its inane examples. Not until I started studying Louis Rosenfelder's excellent book, *BASIC Faster and Better*, did I discover the power in these little one-liners. Now it's difficult to write even a simple program without using at least one of them.

What's so great about user-defined functions?

1. They are powerful. Functions generally use less memory and are faster than other methods, such as subroutines and IF...THEN...ELSE statements.

2. They are efficient. Often, you can write functions that can be used in several programs — no need to reinvent the wheel every time you write a program. Many of us function addicts have substantial libraries of functions to merge into the programs we write, saving programming time.

3. They are fun to write. I get a lot of satisfaction from stuffing several lines of program into a one-line function which will, in turn, condense data to be stored on disk to two-thirds, or less, of its original size. Warning! The Surgeon General has determined that functions are addictive and although beneficial to your program's health, they may be hazardous to your mental health! Learn to write them and you may never return to the low-priced spread.

Only lack of imagination, creativity, and knowledge limits uses of functions. A few uses are: rounding, compressing and uncompressing data, string manipulation, bit manipulation, input checking, menu writing, and hash coding.

If user-defined functions are so great, why aren't they

Table 1
Definitions

Statement: An instruction to the computer. Contains (or is) a verb (e.g., PRINT, GOTO). Tells the computer to do something.

Assignment statement: Assigns a value to a variable <LET (OPTIONAL) A=B+3>.

Expression: Mathematical formula or equation.

Operators: Used to perform the operation(s) in an expression. There are three kinds of operators: *arithmetic* (+ - * / 1) perform arithmetic operations, *relational* (< > = <= >= <>) test if relationships are true or false, *logical* (not and or) perform logical operations.

Function: A variable so related to another that for each value assumed by one, there is a value determined for the other. Most functions compute <SQR(X)> or convert <STR$(X)>. There are two kinds of functions: *intrinsic* (those that come with the language) or *user-defined* (those you write or define yourself, using the DEFFN statement).

Argument: The value (variable, constant or expression) given to a function so it can compute or convert a result and return it as the value of the executed function.

Dummy argument: An argument which is not used by the definition expression. (This differs from many texts which use the terms dummy argument and argument interchangeably.)

Call: Execution of the function. The calling argument is passed to the definition expression as defined in the DEFFN statement. The result is calculated and returned as the value of the function.

Pass: Substitution in the definition expression of the value of the calling argument variable(s) for the definition argument variable(s).

Return: Assignment of the calculated value of the definition expression (result) to the function.

used more than they are? Why isn't more information available about them? There are several reasons.

1. They require quite a bit of study to understand and can get complicated very quickly.
2. A programmer must have a good understanding of IF...THEN logic to use functions efficiently.
3. Literature on DEFFN is practically non-existent and most books that do mention it don't go into enough detail to explain it.
4. On Models I and III you don't get DEFFN until you get Disk BASIC. The Color Computer requires Extended BASIC and then you get only single variable numeric functions.

Just what is a user-defined function? Probably the easiest explanation is that it's a one-line subroutine. To gain a thorough understanding of these little gems, it's a good idea to define some terms first. Table 1 contains definitions of the terms we will be using and Table 2 lists rules for user-defined functions. Look over these tables — then we will write a few functions.

Numeric Functions

There are three steps in defining and using a function:

1. Define the function.
2. Give it a value to use in its computation.
3. Call it to return the results of the computation.

To illustrate these steps, we'll add the constant 2 to variable X and return the result as variable Y. We could do this by using an assignment statement: Y = X + 2. Or, we can define a function to do it.

The form for writing a user-defined function is: DEF FN NAME (ARGUMENT) = DEFINITION EXPRESSION. Let's write a function for our little example: DEF FN A(X) = X + 2. Read this "define function A of X equals X plus 2." We have named the function A, assigned the variable X to its argument, and used the expression X + 2 as the definition expression.

The function will do several things when called:

1. Take the value given to the argument in the call.
2. Substitute this value in the definition expression wherever the variable of the definition argument appears.
3. Calculate the result.
4. Return the result to the function.

The form for the call is: FN NAME (ARGUMENT). The call passes its argument value to the function definition which, as noted, calculates and returns the result.

For our example, we'll set X=3, use FN A(X) to call the function, and assign the result to variable Y. Here's a program to do this:

```
10  DEF FN A(X) = X + 2
20  X=3
30  Y=FN A(X)
40  PRINT Y
```

Line 40 prints Y as 5. What happens in the other lines? We define our function in line 10 and name it A. In line 20, we assign X the value of 3 to use in the calling argument in line 30. Read line 30 "Y equals function A of X." It calls the function definition, passes the value of the argument (3) and takes on the value of the result (5).

We used the same variable, X, in the calling argument

as we used in the definition in line 10. We could have used any variable, C for example:

```
10  DEF FN A(X) = X+2
20  C=3
30  Y=FN A(C)
40  PRINT Y
```

Y is 5. We see that the calling argument can contain any valid variable.

The calling argument can also be an expression. We'll try this by changing the argument in line 30 to the expression X+5.

```
10  DEF FN A(X) = X+2
20  X=3
30  Y=FN A(X+5)
40  PRINT Y
```

Y is 10. The call passed the argument value 8 to the definition and returned 10 as the value of the function.

Now we'll eliminate the variable X from our program and use the constant 3 as the calling argument.

```
10  DEF FN A(X) = X+2
20  Y = FN A(3)
30  PRINT Y
```

Y is 5. So now we've used a variable, an expression, and a constant as the argument of the call. We'll return to this shortly, but first we'll make another slight change in the program we've been using. We'll eliminate the assignment to Y and print the function directly.

```
10  DEF FN A(X) = X+2
20  PRINT FN A(3)
```

Again, 5 is printed. We see that we can use the function, FN A(3) ("function A of 3"), as we would use any variable. One more quick example: change line 20 to:

```
20  PRINT FN A(3) * 6
```

We print 30. We know that the value of function A of 3 is 5. We multiply this by 6 and get 30. Here, again, we're using the function directly as a variable.

We've seen that we can use a constant, an expression, or a variable as the argument in the call. This variable doesn't have to be the same one we used in the definition. We can use the function as we would use any variable. We'll demonstrate all of this as a FOR...NEXT loop.

```
10  DEF FN A (X) = X+2
20  FOR J = 1 TO 3
30  PRINT FN A (J)
40  NEXT J
```

We print 3, 4 and 5. Let's analyze this and see what happens. In line 10, we define function A of X as X+2. In line 20 we set up a loop assigning the values 1, 2 and 3 to variable J. Line 30 takes these values, calls the definition <FN A (X) = X+2>, and passes the values to the definition expression which calculates the value of the function and returns it as the function value.

In the first step of the loop J=1. Line 30 takes this argument value (1), calls the definition in line 10 and passes the value as X to the definition expression, X+2. The result, 3, is calculated and returned as the value of function A of J, and is printed. The program again steps through the loop, setting J=2. Line 30 calls function A of 2, takes on the value of 4, and is printed. In the final step, J=3. Line 30 becomes function A of 3 and prints the number 5.

"So now, Dandy, indubitably you can comprehend

why DEFFN is the most undah-rated playah in the BASIC league."

"I . . . aah . . . guess so, Howard."

Use

1. Use a function as you would use any variable, e.g., PRINT FN A(X), ON FN J(Z) GOTO . . .

2. You can assign to a variable, e.g., Y=FN B(J), then use the variable.

If the function is returning logic (i.e., -1 or 0 for true or false), you can use it in IF . . . THEN statements with true implied: IF FN A(X) THEN . . . (read "if function A of X is true then . . .).

Errors An error in the definition expression will show in the calling line: DEF FN X$(A)=23 (string name with numeric expression) will give a type mismatch error in the calling line number. ■

Table 2
Rules for
User-defined Functions

1. Functions must be defined before use. It's a good idea to define them in the opening program lines.

2. Functions can be redefined anywhere in the program.

3. Functions can be nested (one function calling another).

Defining the Function

Form: DEF FN Name (Argument) = definition expression

Name

The name can be any valid variable.

The value of the name variable isn't changed by use elsewhere in the program.

The type of variable used (e.g., string, integer, double precision) is the type of value the function will return.

The Color Computer allows only numeric functions.

Argument

Passes values to the definition expression when the function is called.

Can contain any valid variable, constant or expression.

Can contain more than one variable (except Color Computer).

The values of variables used in the argument aren't changed by their use elsewhere.

Model I requires an argument, even if it's not used (dummy), e.g., DEF FN PI(DUM)=3.14159.

Definition Expression

Must be an expression.

Must be one line (no ":" as line separator).

Can't contain any verbs (e.g., GOTO, PRINT).

Can contain logical operators (and, or, not), but no logical IF . . . THEN statements.

Can call another function (nesting).

Calling the Function Form: FN Name (Argument)

Name

Must be the same as used in the DEFFN statement.

Argument

Passes values to the function definition.

Must be in the same form as in the definition argument (e.g., if defined as $, it must be called as $).

Can contain constants, expressions, or variables.

The variables can be different than the variables in the function definition.

Values must be assigned to argument variables before calling the function.

Tandy topics

Ed Juge, Director, Computer Merchandising
1500 One Tandy Center, Fort Worth, TX 76102

Well, I had my dadgummit column done ahead of time for once, but Mike's mailing schedule for the February issue messed up my plans. I had planned to give you some news, but the magazine mailed about a week before I could let it out. So, be watching for next month's Topics... I think you'll like what I have to say!

"Outside" Software Revisited

Today alone, I've gotten calls from a couple of outside software vendors who figured it was "impossible to approach Tandy" about software. Not so! Back in November, I told you about Phil Kitchen's appointment as manager of our new outside software support group. The mention was sketchy because, although the plan was firm, all the details were not. Well, we know a lot more now than we did then, and applications for membership in our software support program are being accepted. Since there still seems to be some misunderstanding on our "approachability" (and since this column is supposed to give you a feel for how we work at Tandy), a further discussion about the program seems in order.

Phil's group will be administering two primary activities. One is simply support for software development firms, publishers, or educational institutions who want to develop TRS-80 software for sale through their own (or our) channels. Our software support program will offer them several benefits. First, there will be technical information guides covering the hardware on which they're developing software. These will include information not generally available elsewhere. Information not covered in these publications may also be requested. Problems reported by members (and their solutions) will be added to these guides as they come up.

Second, there will be software development guides, illustrating procedures used in creating thorough documentation and functional, smooth programs. It includes tips to help you get your product ready for market. We'll also share with you some of our internally-developed library routines, such as our INKEY routine for BASIC program inputs, an assembler routine that allows full programmer control of keyboard and graphics in COBOL for Model I and graphics in COBOL for Models II and III, an advanced joystick input routine for the Color Computer, and more.

We'll make available licenses for TRSDOS, so you can duplicate and distribute our operating system and BASIC interpreter with your applications software. You'll be able to purchase BASIC compiler and COBOL runtime disks at reduced "commercial" discounts, purchase copies of TRSDOS source code, the parts to build your own ROM packs for Color Computers, and more. Then, of course, you'll be on our information distribution list for notices about changes in or patches to our systems software. Under certain circumstances, you may be able to receive advance information about new hardware so you can start development on software before the hardware is available in our stores. And, of course, if you're interested in our considering your software for sale through our stores, we'll do it... although membership in this program isn't a prerequisite for consideration.

Now, I do want to make it clear that membership in this program is limited to software development firms, publishers, and educational institutions. Radio Shack reserves the right, at its sole discretion, to determine those persons and entities it will accept in the program and to remove any person or entity from the program at any time. If you're interested, write (don't call) Phil Kitchen at 1300 One Tandy Center, Fort Worth, TX 76102 for an application.

A second function for Phil's group is to evaluate software being offered for TRS-80s by outside vendors. This will result in the best programs for specialized fields being referred to customers so they may contact the vendor directly and buy from him. We know we can't do it all. Even if we did try to stock more vertical market programs, our people can't learn enough about all that software to adequately demonstrate and discuss it with prospective customers. Good software vendors writing reliable and well-documented software are much better able to serve those specific markets than we are. So, we'll try to help get them together with our customers. Our recommendations will depend heavily on evaluations we receive *from actual users* of these packages.

What Software Are We Looking For?

For Phil's program, we're looking for vertical market programs with a reasonably wide appeal. Obviously, we can't go for the super-specialized program that calculates stresses in buildings over 200 stories tall, or solar wind drag on space ships. We're interested primarily in good, solid applications for business, and widely-used scientific or engineering programs, or really useful home or personal software for any of our

computers.

Are we still looking for software to sell in our stores? You bet! We are, however, getting much more selective. As I've hinted above, we'll leave most of the vertical markets to the specialists. We will continue to supply popular languages, WP, DBM, spreadsheet, and accounting packages with very wide appeal, software for a few selected vertical markets, games, educational, and home programs. Both our in-house efforts and our interest in outside-developed software will be devoted to improving our offerings in those areas. Games will have to be truly unique, original, and outstanding. Super graphics are an absolute must. If you think you've invented the program of the decade, but you're not a member of our Software support program, send a one-page writeup to Software Submissions, 1500 One Tandy Center, Fort Worth, TX 76102. Include a description, and sales pitch. Tell us what makes it so great, and why people will break down our doors to get to it. And, please, don't forget to tell us *which* TRS-80 it runs on! Our folks who handle this type of evaluation get bogged down, so give us sixty days to reply . . . we might need it.

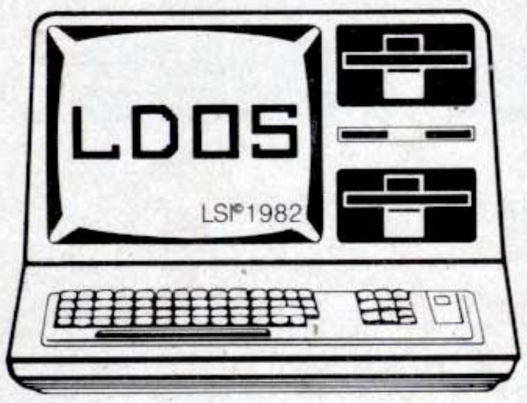
Computers Everywhere

On a personal note, my wife Jo and I just got back from a great week of R&R in Hawaii. Sightseeing, sunburning, picture taking, and indulging in Jo's hobby — shopping. Our returning 747 was packed with tired, sunburned bodies sporting a variety of Aloha shirts and dresses, and funny hats. I'm a very relaxed flyer, but I've never learned to sleep sitting up, even on overnight flights like this one. The fellow across the aisle, apparently also a non-sleeper, was reading (you guessed it) a computer magazine. Once the bug bites, even an island vacation can't take your mind off the subject! See you next month. ■

LDOS MICRO REVIEW

Volume 1 No. 1 ★ SPECIAL EDITION ★★★ January 1, 1983

The BASIC Answer



You'll think you've made the DOS strike of the decade when you turn your micro on to LDOS. You'll find a bonanza of features like full keyboard type-ahead; a true background spooler; file backup by date, class, and between different drive types; hard disk support; data transportability between Model I and III; and a complete communications utility including disk file send and receive. Support for Radio Shack's Doubler and selected others is also provided.

With our Job Control Language, you get true "hands off" running of your application programs — give a single command and then walk away. The 400 page manual includes examples of all commands and utilities. The Operator's Guide gives you step by step instructions on how to use LDOS with your applications. Stop running with only "half" a computer! Let LDOS provide the missing features to speed up and simplify your TRS-80 computer system! Visit a dealer or contact LSI for more information on the most popular sophisticated operating system for your TRS-80.

LDOS is available worldwide through thousands of dealers for just $129.

The BASIC Answer is a BASIC text processing utility. It is designed to allow the BASIC programmer to build code in a structured manner. "Source" code is written with a word processor or text editor which allows the user to exploit the powerful editing and movement features characteristic to those types of editors. Source code can even be created by your own BASIC interpreter. **The BASIC Answer** is then used to process these files into normal interpretive BASIC code.

Free Yourself from Line Numbers

The BASIC Answer allows substitution of labels for **line numbers!** This means that your BASIC code now can read like a novel. Instead of the typically undescriptive "GOSUB 1000", a label such as "GOSUB @Search.Name" is used. Imagine yourself reading code filled with such descriptive branches and understanding it at a glance, even years later. This feature even allows totally relocatable BASIC routines without the renumbering problems.

TRS-80 is a trademark of Tandy Corporation. LDOS is available for the TRS-80 Model-I and Model-III. Prices and specifications subject to change without notice. LDOS and The BASIC ANSWER are products of Logical Systems, Inc.

```
ORLP!=2TOHA!PRINT@32,"primes found
FHA!/LP!=INT(HA!/LP!)THENGOTO48"CH
EXTLP!:IFVAL(FA$)=LO!THENFA$="* Pr
R!(CO%)=LO! on this scan"USING"##,
R$(CO%)=FA$LEN(FA$)-1)FORLO!=ST!T
O%=CO%:PS%=PS%+1ELSEFA$=LEFT$(FA$,
ORLP!=ØTO1ØPRINT@Ø,"factoring "US
RINT@64*LP!+192,PR!(LO%),PR$(LO%):
O%=LO%-INPUT"ORIGIN OF SCAN";IN$Ø
FLO%=-1IFVAL(IN$)<2THEN11NG"###,##
EXTLP! ST!=INT(VAL(IN$))##";PS%;R
O%=CO%+INPUT"   END OF SCAN";IN$(
FCO%=11EN!=INT(VAL(IN$))IMPR!(1Ø),
FHA!/LP!=INT(HA!/LP!)THENGOTO48"CH
EXTLP!:IFVAL(FA$)=LO!THENFA$="* Pr
R!(CO%)=LO! on this scan"USING"##,
R$(CO%)=FA$LEN(FA$)-1)FORLO!=ST!T
O%=CO%:PS%=PS%+1ELSEFA$=LEFT$(FA$,
ORLP!=ØTO1ØPRINT@Ø,"factoring "US
RINT@64*LP!+192,PR!(LO%),PR$(LO%):
O%=LO%-INPUT"ORIGIN OF SCAN";IN$Ø
FLO%=-1IFVAL(IN$)<2THEN11NG"###,##
EXTLP! ST!=INT(VAL(IN$))##";PS%;R
O%=CO%+INPUT"   END OF SCAN";IN$(
ORLP!=2TOHA!PRINT@32,"primes found
FHA!/LP!=INT(HA!/LP!)THENGOTO48"CH
```

A New Concept in Variable Usage

The BASIC Answer allows variable names to be as long as 14 characters and ALL 14 are significant. Imagine reading:

"IF ACCNT.OVERDUE #> 0 THEN GOSUB @PRINT.DUN"

rather than

"IFAO#>0THEN GOSUB52130"

Which would you rather read? It also introduces to BASIC the concept of Global and Local variables. This feature circumvents the tedious problem of variable tracking because a Local variable is only viable in its own subroutine!

NOW AVAILABLE
LDOS 5.1 Quick Reference Card
$5.95.

End the Multiple Machine Hassle

The BASIC Answer introduces the concept of **"Conditional Translation."** This feature allows the programmer to place different "machine dependent" code simultaneously into the same Source Code. **The BASIC Answer** can be "switched" when processing to ignore the unwanted or include extra code! No more multiple master programs to confuse maintenance. All the masters could now be rolled into the same program. Modify the one master and you've modified them all. Process the same code with different switches set, and get two or more versions from the same source.

The BASIC Answer combines the self-documenting power of COBOL with the relative ease of BASIC together with the power of a word processor.

The BASIC Answer is available for just $69.00.

LOGICAL SYSTEMS INC.

11520 N. Port Washington Rd.
Mequon, WI 53092
(414) 241-3066

58

PLEASE ANSWER THESE 5 QUESTIONS

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

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

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

Maxi Manager *by Dale Kubler*

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

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

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

Maxi Mail *by Dale Kubler*

Maxi Mail is a powerful mail list management system that:

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

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

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

Maxi Cras *by Dale Kubler*

(Check Register Accounting System)

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

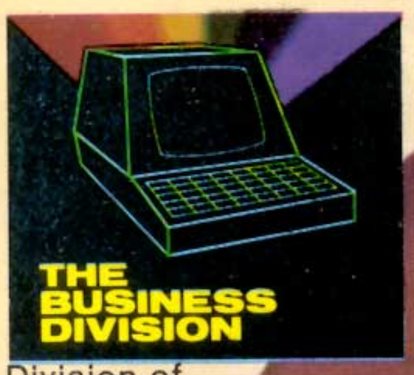
Model I & Model III. Minimum 2 drives required **012-0145 $99.95**

Maxi Stat *by David Walonick*

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

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

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



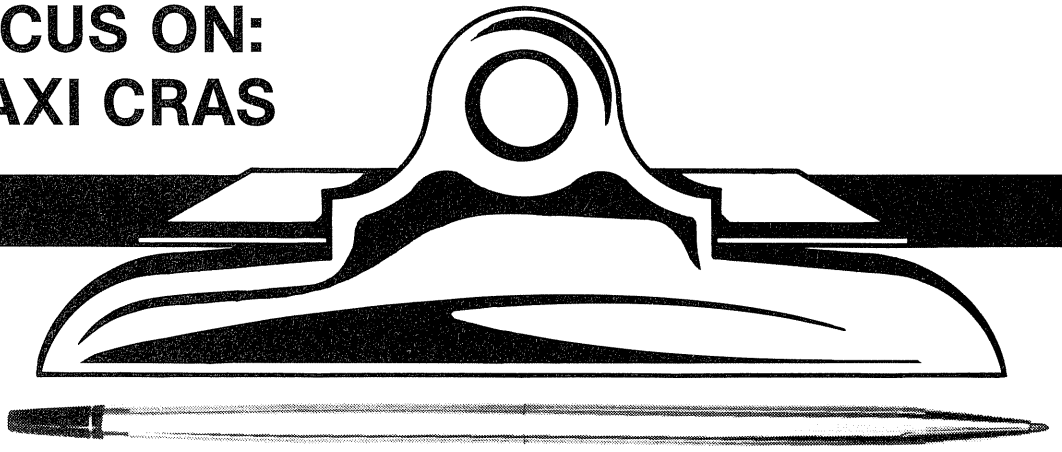
A Division of
Scott Adams, Inc.

119

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

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

FOCUS ON: MAXI CRAS



What is Maxi CRAS?

Maxi CRAS (for Check Register Accounting System) takes the work out of printing checks, balancing your account, and reconciling it with bank statements. And that's only the beginning!

Who can use it?

Maxi CRAS is powerful enough for small businesses, and easy enough to handle for family or personal use. Best of all, it won't make you change the way you do business. Write checks by hand and enter them into the computer later, or let Maxi CRAS do all the work, and print checks automatically.

But what makes it different from all the rest?

We're glad you asked! Disk-based Maxi CRAS handles a virtually unlimited number of checks and deposits each month. And to keep track of all those financial transactions, Maxi CRAS supports up to 223 separate income and expense accounts, and your transactions can be assigned to one or even all 223 accounts.

Why is this so important? For some transactions it's not. Suppose you write a check for $250.00 to Jolly John's Jalopy Jumpers for repairs to your automobile. You would simply assign the check to the AUTO REPAIRS account. But suppose you had Jolly John fill the tank, and you picked up a gallon of milk while you were there. With Maxi CRAS you can simply add the $50.00 worth of gas and the $2.00 worth of milk to the $250.00 you owed him for the new windshield wiper, and write a check for $302.00. Many systems would force you to assign this check to a single account. But Maxi CRAS lets you charge $250.00 to AUTO REPAIRS, $50.00 to your GAS & OIL account, and $2.00 to your FOOD account. Six months, or even six years from now, you'll still know exactly what that $302.00 went for. That's power!

I'm interested. Tell me more!

Organizing the data is just half the picture. Maxi CRAS provides six essential reports and statements in a fraction of the time required to do them by hand. At tax time, you'll have all the data you need at your fingertips. No more sifting through shoeboxes full of cancelled checks!

THE BUSINESS DIVISION
A Subsidiary of Scott Adams Inc

MAXI CRAS
By Dale Kubler
For your TRS-80 Model I or III with 48K, two disk drives, and an 80-column printer.
$99.95
Soon to be released for IBM/PC
THE BUSINESS DIVISION
BOX 3435 • LONGWOOD, FLORIDA 32750
(305) 862-6917

For detailed forecasting and budgeting, Maxi CRAS now interfaces EASILY with VISICALC(TM), the electronic spreadsheet program sold by Radio Shack. A Maxi CRAS exclusive!

The 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 (NEBS 9020).
- Data Entry routine second to none — saves time AND eliminates errors. You don't need to be a computer expert to use Maxi CRAS.
- The best selection of printed reports available — complete check register, income and expense subtotals, bank statement reconciliation, list of check register notes, and an account distribution statement. Compare!
- For even more detailed analysis, Maxi CRAS data is readable by VISICALAC(TM).
- Checkbook balance is constantly updated and instantly accessible. No more embarrassing overdrafts!
- A notes option can be used to flag tax-deductible transactions.
- Handles up to 223 income AND expense accounts. Many other systems only allow ONE income account.
- Assign transactions to a single account, or distribute over multiple accounts. Assign specific amounts, or pro-rate by fraction or percentage. Cash transactions can be recorded in any account without affecting check-book balance.
- Check Address data base stores up to 40 addresses and automatically prints them on your checks.
- Fast and easy bank statement reconciliation. Compare!
- Maxi CRAS is supplied with a complete User's Manual, sample printouts, and TDOS, a special version of the DOSPLUS operating system.

WE INVITE YOU TO COMPARE.

Compare Maxi CRAS with any other system available. We did, and we're sure you'll find Maxi CRAS to be the most versatile, easy to use Check Register system available — at any price!

60

BASIC and Forth

A comparison of the two languages

For all models

Anthony Scarpelli, Scarborough, ME

Forth is the most powerful language in the world. A rather bold statement, of course, but it is an option concerning the one language that I love to use. How many other languages can you say that about?

If you use BASIC and don't know any other language, BASIC is probably your language of preference. However, if you've heard of Forth, are slightly curious about it, and are interested in learning something about this unique language, this article will let you in on some of what it's all about. It might even convince you to try it.

BASIC is a high-level language. Forth can also be considered a high-level language. However, some people place its level between a high-level language and a low-level language such as assembly language. It all depends, of course, on your own point of view, but since a listing of a Forth program can look very cryptic, you might consider it jibberish instead of a language at all.

The reason Forth looks strange to some people is because the essential elements of it, called words, can consist of any single (or combination of any of the) ASCII characters except for a space, backspace, or the carriage return. Thus, a word can be as simple as a {:} colon. It means the combination of characters following it, up to the next space, is to be put into the dictionary, and is to be defined by the word(s) following it. (As opposed to BASIC, one or more spaces must always separate Forth words.)

The word {;} semicolon is used to end a definition. I now want to create a word that will produce the square of a number. I will call it "square". It will simply multiply a number by itself.

Square will consist of two previously-defined words: the word {*} star, which is the same as the BASIC multiply operation, and the word {DUP}, which duplicates the number that precedes it.

The new word can now be defined as: :SQUARE DUP * ; .

Before I go on, I will explain something that some people consider to be degenerate in a modern language such as Forth, and that is the use of "postfix notation."

Forth is a stack-oriented language. That is, it uses a stack for all its operations. You put a number (or numbers) onto the stack, follow it with a word that acts on that number (or numbers), and usually some result is left on the stack. It is a very simple concept, but it means that operations must follow numbers. Instead of saying 2*3, as in BASIC, you would have to say 2 3 *. This is called postfix notation, or reverse Polish notation (RPN).

At first, it seems strange to do things this way, but after using it for a while, it becomes quite natural and is no longer disturbing. You can quickly learn to visualize stack operations.

Now, back to our example. To use the word {SQUARE}, you would merely write the number you want to square on the screen, then a space, then the word SQUARE. When you press the ENTER key, the number is placed onto the stack, then the word SQUARE is interpreted. Since SQUARE consists of two words, they are interpreted next. The first word, {DUP}, duplicates the number on the stack. Now there are two copies of the same number — one on top of the stack, and the other is second on the stack.

The next word (*) multiplies them together and leaves the result on the stack. To see the result, we would use the word {.} dot, which removes the top number from the stack and sends it out to the output device, which, in this case, would be the screen. It is very similar to BASIC's PRINT.

If we chose the number 5 to square, the BASIC operation would be "PRINT 5†2." In Forth, it would be "5 SQUARE .".

We can just as easily combine this last operation in another word. The definition ": PSQ SQUARE .;" could stand for print square. To use it, we would say "5 PSQ". There is no easy means of producing new words or operations in BASIC. This is why Forth is such a powerful language. You define a word, test it out, define other words with it and, finally, you can define the whole program with just one word.

When you purchase the Forth language, it has many words already defined in it which are similar to BASIC's reserved words. The number of these words vary with the implementation you purchase. The version for the TRS-80 Model I that I use is from Miller Microcomputing Service. There are about 300 words already defined and ready to be used by themselves, in definitions, and in programs.

Some of these words do the same job as BASIC statements. For instance, the FOR . . . NEXT loop in BASIC has its counterpart in Forth. This is the DO . . . LOOP. It is used slightly differently, though, because of our stack machine. In BASIC, you would say, FOR I=1 TO 100 : some process : NEXT I. In Forth, you would say, 100 I DO some process LOOP.

There are other loops in Forth

which have no exact counterpart in Level II BASIC. The BEGIN . . . UNTIL, and the BEGIN . . . WHILE . . . REPEAT loops are an example. In the BEGIN . . . UNTIL loop, a process after BEGIN is repeated UNTIL a true flag is left on the stack. In the other loop, a process is done after BEGIN, and WHILE a flag is true another process is done. Then the loop will REPEAT.

Of course, all of the BASIC arithmetic and logic operations are supported in Forth, including operations on double-precision numbers. In all of these, postfix notation is always used.

In Forth, you can change number bases in mid-stream, which you can't do in BASIC. If you wanted to use binary, or hex, or octal, or any other base in the middle of a definition, you can do it since there are words defined for this.

In BASIC, we have PEEK and POKE to directly manipulate memory. In Forth, there are many more memory operations. The Forth words {@} fetch, and {C@ } C-fetch, remove the address which was placed on top of the stack and replaces it with the address's two-byte or one-byte contents respectively. The words {!} store, and {C!} place two bytes, or one byte, into an address.

In addition, Forth has {MOVE}, {–MOVE}, {FILL}, {ERASE}, and {BLANK} which are operations that can move or change whole blocks of memory.

There is also a slew of words for our terminal. Included in these operations are {PAGE}, which is the same as Level II's "CLS", {PRINT}, which directs all output to the printer, {CRT}, which directs it back to the screen, and {PCRT}, which sends the output to both at the same time.

My version of Forth I is disk-based. However, the disk is used like an extension of memory. This is called virtual memory. RAM is used to store the dictionary, arrays, constants and variables. Two buffers are used to transfer data from RAM to disk and from disk to RAM. The size of a buffer is 1024 bytes. This is called a block, or a screen. It exactly fits into one video screen of 64 characters by 16 lines.

Writing programs involves editing blocks of data. The process "16 EDIT" reads block number 16 from disk and puts it on the screen. It will either be empty, or will have previously-stored data in it. Now it can be modified by writing words and creating definitions. The editor provided with this version is an excellent screen editor which makes modifications quick and easy. When finished with the block, it can be sent back out to disk for safe storage. The next step is to say "16 LOAD". This will compile the definitions into the dictionary. The program can now be tested by typing out the word that was defined for it. If the word doesn't work, we can {FORGET} it or any others that don't work to remove them from the dictionary and try again.

Forth also supports byte, single-precision, double-precision, and string arrays. String operations are provided and many of them are very similar to Level II, such as {MID$} or {RIGHT$}. Others are different, such as {$COMPARE} or {$XCHG}.

If you need random numbers or graphics, the version of Forth that I have supports them also. They are very similar to Level II operations.

You can very easily create words that contain assembly language routines by using 8080, or Z80 mnemonics. Forth executes much faster than BASIC, but when you use assembly language code, it becomes just as fast as machine code.

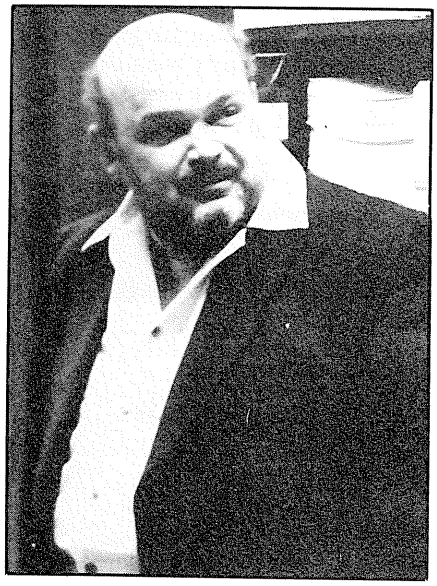
The Forth word {:} is called a defining word because it initiates and defines a certain class of words. We can very easily create other defining words as well. We can also create different vocabularies, or even other languages. You simply cannot do things like this with BASIC.

Forth is one of the best control languages around. It is the language used by many radio observatories all over this country to control their telescopes. It is also a language that is not easy to learn because of its power. When you do learn it, you can say that you truly are master of the computer. ■

80-U.S. interviews John Harding

80-U.S. staff

A.J.(John) Harding is the owner of Molimerx Ltd., the largest software house in Europe. They carry numerous items for the Models I and III as well as the Video Genie, a TRS-80 work-alike. 80-U.S. had the opportunity to talk with John at a recent conference and as any Englishman will tell you, English is certainly another language for most Americans. We hope you enjoy the discussion as much as we did. -Ed.



80-U.S.:How about a little personal history? How did John Harding get attracted to the micros?

John: I was a lawyer in England and trained in electronics in the Air Force. My wife is Canadian and while living there, I decided to set up an electronics company called J&J Electronics. We sold semi-conductors and the market was right. It took off and is still succeeding. I gave up law and was making money selling electronic parts. Around 1973, we restructured the company so we could return to England and run it from there. We structured it so well, there was little to do but make decisions and sign checks. I got myself a little Tandy to play around with and from there it's the same story as anybody else. While living in Canada, I met a number of people in the States, so I asked a few to send over some software, perhaps I could sell it. And from there it's now a big international company.

80-U.S.: Where did the company name come from?

John: You're not a Latin scholar. Moli is soft and merx is ware.

80-U.S.: How would you describe Molimerx's main thrust to our readers?

John: We're general purpose. We currently have 250 programs, utilities, business, games - the lot. For every TRS-80 machine except the pocket one. But also, the Video Genie, as we call it. You call it PMC, it is strong everywhere except in the States, for obvious reasons. And it is a big market for us too.

80-U.S.: Have you ever run into incompatibility problems with TRS-80 software on the Video Genie?

John: We did at the beginning. The original Genie was lacking two arrow keys and a clear key. That has been fixed and now it is truly compatible. There are two small differences, one in the printer address and the RS-232 is completely different. But it is nothing that gives us any great problem.

80-U.S.: We recently saw advertisements for the Genie II and Genie III in the German magazine ELCOMP. What do they equate to?

John: There are really four machines, the Video Genie, Genie I, II, III, and now a Color Genie as well. The Video Genie is the Model I and the Genie I is that, souped up. They use the 2K just above the ROM to put in a monitor, lower case and gadgets like that. The Genie II is the same but with the cassette on board.

The Genie III is an entirely different machine. It might be closer to the IBM if anything. CP/M and NEWDOS/80 both come with it. It has two on-board disk drives that can be configured any way you want. They will probably be double sided soon. You get a normal 64x16 screen, or 80x24. You get two versions of your DOS to match the screen format. It's nice, but it is not particularly attractive, I haven't met anyone who likes its appearance.

80-U.S.: How do you see the future of the TRS-80 peripheral market in England?

John: It's completely different than here. Your Model I is almost gone; in England it's still going strong and the Genie backs it up, because it is a Model I also. I understand that the Color Computer

is doing well. In England it has fallen almost completely flat. It's not really Tandy's fault. We have small colour machines coming out almost twice a week. There is tremendous competition.

80-U.S.: What is the overseas reaction to American software?

John: The spelling is important. After all, what if the average American picked up a software package and the word color is spelled k-u-l-i-e-r? He wouldn't like it too much. It wouldn't be a big sales problem, but if he had another one exactly the same that spelled it correctly, he'd choose that one.

80-U.S.: What advice do you have for American writers who would like to possibly market overseas? Is that a difficult task for them?

John: Well, it depends on the author. We have two and they submit their programs to us. We have a standard royalty agreement and we treat them the same as anybody else. But they are further away, so there has to be some trusting of people. We account quarterly and have the agreements drawn up by lawyers.

80-U.S.: There is no problem with customs and duty?

John: For an author, no. He just sends us one master and that's it. When a supplier sends a product then we have to go through customs and duty and that gets quite complicated, but an author wouldn't be involved in that at all. He could always submit material to us.

80-U.S.: Do you recommend to American writers that they patch their work and make them country specific?

John: Oh yes. Whether they'll do it or not I don't know. People in America don't realize the software market that there is in England. Now, we really haven't gone around yelling about it because we don't want someone cutting away at our roots, but there is a fantastic amount of very good TRS-80 software written in England. We have ninety-eight authors that write for us, four of them full-time. We have our own adventure series, you have Scott Adams and we have the Mysterious Adventure series. We already market it and one or two others over here.

80-U.S.: Can we order directly from England?

John: No, we will be shipping through Logical Systems Incorporated. And this is just good stuff. There is no point in sending over garbage.

80-U.S.: Thank you.

John: Well, maybe because you've got enough garbage already. But for whatever reason, there is no point in doing it. So, you're going to get the creme de la creme of our software. Prozap is the best "zapping" program there is. There is a program called Impakt which is essentially a tool kit for the BASIC programmer. It's a 747 simulation that is so sophisticated that I'm not at all sure one should call it a game.

You Americans, you like to think that you are international, but you're not really. You're provincial. You don't really go outside of the country for things. In some cases you act to totally discourage it. You miss quite a lot.

80-U.S.: Do you see the microcomputer as possibly unifying world language? Or at least getting us to have a common base in English?

John: No. No, the program will always have to be written in English, won't it? Essentially in English if it's in BASIC. What we now do is translate the documentation when it is feasible. When we get something like LDOS it's tricky. If it's a few pages, it is only a few pounds to have it translated.

80-U.S.: Is software protection a problem in England?

John: I feel very, very strongly about this. Reporters go around saying it's such a terrible thing, everybody is getting ripped off. I have not seen a single magazine, a single reporter, come up with any statistics whatever to support that statement.

All they say is "Oh, there must be piracy. I feel dishonest. I would do it, therefore everybody else does it." What we've done is carry out tests. We've put out the same program protected and unprotected at the same time of year. We sell more programs unprotected than protected.

You will never get rid of user clubs. You sell a guy a ten pound program and there isn't a person in the world who won't give a copy to his brother that has a TRS-80. You've got to put up with that. But real pirating in England? There isn't any. We had one case where we sued under the copyright laws and succeeded. That was that, they are out of business. It was a stupid thing to do.

I seriously wonder if you have the same problem here. Anyone who says it, let them bring out the statistics to prove it. Sell a program protected and unprotected over the same time and compare sales. Particularly, let them come up and say "I found 2000 illegal copies of this." I don't believe it. There is some, obviously. But it makes good copy doesn't it?

At this point the interview was halted for it was time for a poker game to begin. The address of Molimerx Ltd. is 1 Buckhurst Rd., Town Hall Square, Bexhill-On-Sea, E. Sussex. Telephone (0424) 220391/223636 or TELEX 86736 SOTEXG. Next month, an interview with another TRS-80 industry leader.-Ed. ■

PILOTPlus (MicroPILOT)

An evaluation of this CAI language from Radio Shack

Models I/III, PMC-80, LNW80

Ranes Chakravorty, Salem, VA

There are three main groups of people who use home computers. The first are the real experts — they know assembly language, are constantly experimenting with the latest peripheral, and if they are not already on-line to a communications network they will soon be. The second is the pure end-user who uses the computer and commercial software without any interest in how or why the system works. The champions in this group are glassy-eyed individuals who are trying to score yet higher in the latest computer game. The third group are basically interested in what the computer can do — hence they also depend largely on commercially available software, however, they do have some interest in what makes things happen and thus experiment a lot with the programs they have. Most programs they have fall short of their expectations, but they do not have the expertise to improve the program. I am a confirmed member of the third group and the review of the Radio Shack program PILOTPlus (now called Micro-PILOT) is written from that viewpoint.

A large segment of my professional activities involves teaching medical students. By and large, medical students are highly intelligent and motivated. They are hard-working and have enormous workloads. As a result, they have little time to spare and long free periods for continuous study are rare. I felt that the microcomputer should be an ideal instrument to provide reasonably short lessons which a student could go through quite quickly to assess his knowledge of any given subject. At the same time, he would have a rapid evaluation of what he knew, what he needed to know, and where he could get the information he required.

Computer assisted instruction (CAI) has a long history in medical education. However, the excellent programs that are available are generally long and quite intricate. They are the product of the efforts of expert programmers. These programs are generally long and time consuming and require expensive hardware. I decided to try and write my own programs.

Instructional programs can be written in BASIC. It is a formidable task needing more expertise and time than I have. PILOT is a language that was developed specifically for the purpose of CAI. Since Radio Shack had a PILOT program available I decided to test it to see if it would suit my purpose.

PILOTPlus requires at least 32K and one disk drive. It is supplied on a 5¼" diskette with a set of instructions, two graphics reference grids printed on both sides of a sheet and a reference card. They are all nicely assembled in a handsome three-ring binder. For the most part, the instructions are clearly written and assume no prior knowledge of the Radio Shack Model I (or Model III) computers. Step-by-step advice is given in starting the computer up, inserting the PILOTPlus diskette and getting a backup made, and also in preparing a formatted diskette for saving programs written.

Following the introductory segment, the instructions are divided neatly into five sections labeled "DIRECT COMMANDS," "PROGRAM COMMANDS," "EDITING COMMANDS," "EDIT MODE COMMANDS" and "GRAPHICS COMMANDS." Printed in large type, the command is first explained and summarized in a neat box at the end of the section. This is a useful aid to quick recapitulation. In addition to a good index, each page has a top-right heading which indicates the section on the left and the command on the right. This helps to locate sections quickly. At the end of each section there is a summary of all the commands discussed.

For the person with some idea of Radio Shack BASIC, PILOTPlus is easy to learn since many of the commands are identical. Once the disk is operative, the command PILOT loads the program. Load completion is signified by the message PILOT READY with a # sign and a blinking cursor on the next line. Any of the three types of activity mentioned above, i.e., programming (text or graphics), editing a previously prepared program or running a loaded PILOT program, can now be undertaken.

The Direct commands include — NEW, SAVE, BOOT, LIST, PLIST, RUN, LOAD, CLS and BREAK. The only new command is PRO, which initiates the PROGRAM MODE for creating the program. The initiation of the program mode is indicated by an asterisk at the left margin with a following indented cursor.

The commands in the program mode are the essential feature of the PILOT language. They are all single letters and each is explained in detail in the instruction book with examples that are completely worked out. Many of the commands are specific to the PILOT language, some are old friends from BASIC in a new guise and a few are condensations of BASIC subrou-

tines. The commands specific to PILOT are the following:

T: allows entry of text into program (can be placed at any specific location by &/xxx& which is the equivalent of PRINT@).

A: accepts student's response.

M: matches student's response to expected response. Somewhat like the IF...THEN sequence in BASIC, however, the M: command is much less rigid and the degree of match expected can be predesigned from a complete congruence to much less exact identity. This is one of the most useful features of PILOT.

V: this allows verbatim reproduction of the program text so that letters having a command significance (such as the &) can be printed out.

O: useful when the length of a program exceeds available memory; this command loads subprograms that are overlaid on existing programs in memory. The operative length of a PILOT program can thus exceed available memory.

There are some old friends from BASIC neatly incorporated into PILOT under new command names. These are:

J: same as GOTO.

U: and R: same as GOSUB and RETURN.

W: same as INPUT.

I: same as INKEY$.

Y: same as the IF...THEN or =.

Z: resets all variables to zero and is used in calculating students' scores.

Some short routines have been condensed and incorporated into PILOTPlus. Thus:

B: institutes double-width characters.

N: erases screen and returns to normal characters (like CLS).

D: delays execution of the program by the number of seconds specified after the command.

The command C is used to start calculating designated variables so that the scores of individual students can be tallied separately.

The editing and the edit mode commands are close to the editing commands in BASIC. Thus:

I: inserts a line of text.

D: deletes a line of text.

E: displays a line of text that can be edited as in BASIC with various commands such as L: for list, I: for insert, H: for Hack, X: for extend, etc. The S(earch), Q(uit), A(gain), C(hange) commands of the edit mode are identical with those of BASIC.

There are two methods of adding graphics to the programs. The first is the graphics mode which uses the SET/RESET sequence of BASIC. The functions are facilitated by allowing the capability of drawing lines or boxes by simple combined commands such as G:S and G:R which draw (or clear) a line between two defined coordinates or G:SB and G:RB which draw (or clear) a box between defined coordinates. A screen-grid reference card supplied with the instructions helps plan the display. The other graphics aid is the program GRAPHICS/PLT which has to be loaded separately. When this program is loaded, the cursor can be used to draw forms on the screen and the final product saved as a subprogram which can be accessed in the body of another program as needed. The saved program can be reedited, changed and brought back as a new display. TRS-80 graphics characters can also be displayed using the T: command in the &CHR$(xxx)& form.

The last part of the book of instructions has some condensed discussion on arrays, mathematical functions, strings, etc., which would be helpful to the programmer who has a good knowledge of BASIC, but would probably be unintelligible to a person interested in using PILOTPlus purely as a programming tool.

The PILOTPlus program is well planned. The instructions are well written and the text nicely arranged. Even for the beginner, the construction of effective programs should be easy after a little practice. A sample program is included on the diskette and the anatomy of the program's function analyzed in detail. The analysis is good but difficult to read. The graphics programs are concise, intelligible and easy to use. Line numbering is automatic and therefore a major chore in writing a program is obviated. If new lines are inserted during the editing mode, all subsequent lines are automatically renumbered (this also happens

when program lines are deleted).

Tone generation is possible using the <: sign and specifying the frequency and duration. Correct and incorrect responses by the student can therefore be signalled by characteristic tones. Also, audible cues can be given for concomitant use of other materials (for example slides, illustrations, etc.) during the instruction or the questioning phase of the program.

Scores of each student going through a program can be tabulated during execution. The scoring can be adjusted according to the number of attempts needed to get the correct answer.

PILOTPlus has more sophistication than readily meets the eye. There are quite a few different ways of achieving the same end. This versatility is useful but confusing to the beginner. Arranging the scoring of answers given by students has been the most difficult part of programming for me. The sample exercise is helpful but leaves much to be desired — this is the major weakness in the instructions supplied. The algorithms used in the sample program could be incorporated verbatim but that is restrictive. The graphics work well only with rectangles and straight lines. Diagonal lines are stepped and are particularly awkward when using GRAPHICS/PLT. Circles and curved lines are next to impossible. That is not a fault of the program but arises from the lack of high resolution in the display. This is still a significant drawback.

An instructional program written in PILOTPlus cannot be run without having PILOT running at the same time. An astute student could thus break into the edit mode during a program run and examine or change the instructions. That possibility does not detract significantly from the excellence of PILOTPlus as the time needed to do this would be considerable and a dishonest student is the exception rather than the rule.

The program in its present form occupies a considerable amount of space. With a 48K system initiation of the PILOT program leaves 29191 bytes free. During the writing of a program, the direct command FREE will return the number of bytes still available. For extra-long programs new segments can be called in overlaying older sections in memory (see O: command above).

The PILOTPlus program marketed by Radio Shack is a well-planned, well-documented aid to computer-assisted instruction. It can be used by the novice programmer but some considerable knowledge of BASIC and prolonged practice is needed to fully utilize the considerable potential of the program. ■

PILOTPlus, now called Micro-PILOT is available for the Models I and III. The Model I disk system (#26-2205) is $99.95 and the Model III version (#26-2718) is $119.95.

BASIC bits

Jumping from DOS

Models I/III, PMC-80, LNW80

Thomas L. Quindry, Burke, VA

John Wicker, of Burke, Virginia, whose father just added disk drives to his Model III, very efficiently solved a problem that was bugging him. John was annoyed that he couldn't jump directly to a machine language program in memory from TRSDOS without first going to BASIC and using the SYSTEM command, or else going to Debug and using the J command. Upon reading the Model III TRSDOS manual (in the technical section for the description of $CMDTXT), John found a fairly innocuous statement: "On entry to a program, (HL) = First non-blank character following the program name."

He decided to make use of this fact with the following program, which he calls JUMP/CMD. The assembly code listing is:

```
CALL   1E5A    ;converts ASCII to integer in DE

EX     DE,HL   ;put value in HL

JP     (HL)    ;jump to that address
```

In TRSDOS, when he calls the program and follows it by an ASCII decimal address, he will jump to that address. For instance, JUMP 40000 or JUMP/CMD 40000 will allow you to jump to the decimal address 40000.

You can POKE the machine code from a BASIC program to any non-conflicting location in memory and then save it using the TRSDOS DUMP command. There is an area in low RAM that doesn't appear to be used for anything. In both the Model I and Model III, the addresses from 40B8H (16568 decimal) to 40D2H (16594 decimal) appear to be unused. This looks like a good place to put the five-byte routine. Use the BASIC program:

```
10 DATA 205,90,30,235,233
20 FOR N = 16568 TO 16572
30 READ A: POKE N,A
40 NEXT
```

Then get into your DOS and save (DUMP) the program from 40B8H to 40BCH with a transfer address of 40B8H. See your DOS manual for details on how to DUMP the program.

Question: I am trying to write a checkbook balancing program. I have data for each check in a five-dimensional array, but don't know how to print out all the information for each check. Can you give me a suggestion?

Answer: Let's say you want to give information in the following categories: 1. check number, 2. date, 3. pay to, 4. amount, 5. payment category. Assuming your string array is A$(N,Y), you should dimension your array A$(N,5) where N is the maximum number of checks you plan to store as data and the 5 stands for the five categories listed above. To display each check and the other data for it, use:

```
100 Z=1
110 FOR X = 0 TO N
120 FOR Y = 1 TO 5
130 PRINT A$(N,Y);" ";
140 NEXT Y
150 PRINT
160 Z=Z+1: IF Z=16 THEN INPUT "PRESS ENTER
    TO CONTINUE";Y$:Z=0
170 NEXT X
```

The count on Z lists fifteen lines and then pauses so

BASIC bits

you can review the data before continuing. You could send the information to the lineprinter by substituting LPRINT for PRINT and eliminating lines 100 and 160.

Question: I'm running a survey and am entering the numerical data using INKEY$. How can I convert it to decimal from the string data?

Answer: You can convert any string data (including those generated by INKEY$) to decimal notation by defining a variable to the value of the numerical string. For example, assuming your string is A$, let A=VAL(A$). The string, A$, must begin with numerical data to be useful. For instance, if A$="100 PEOPLE", then VAL (A$) would give you a value of 100. For a string starting with alphabetic characters, VAL (A$) would return a value of zero.

Question: I have set a string to A$="332,445,544" and have then saved it to tape using PRINT#-1. Why can't I read back the data properly? I only get A$="332" when I read the data using INPUT#-1.

Answer: When you saved your data to tape, you saved all of the information. However, since the PRINT#-1 function saves it by ASCII code, the commas, which are in ASCII also, are read out by the INPUT#-1 function as data separators which are also ASCII code commas. If you used the INPUT#-1 function to get three strings, A1$, A2$, and A3$, and then concatenated them by A$=A1$+", "+A2$+", "+A3$ you would have your desired result. ■

Reviews

ColorForth
Armadillo International Software
P.O. Box 7661
Austin, TX 78712
(512) 459-7325
$49.95

Do you tire of the slowness and restrictions of BASIC and wish you had an alternative? One that gave you complete freedom to program as you wanted and use the full power of the computer? With the introduction of ColorForth, that alternative is here!

ColorForth is shipped on a cassette with a 36-page user's guide. Side one of the cassette contains the cassette version of ColorForth, while side two contains the disk version. (Yes, you actually receive two versions for the price of one!) Thus, when you're ready to move up to disk, you don't have to buy another version of ColorForth — just use the other side of your cassette.

The authors begin by urging you to make a backup of the cassette. I've purchased several tape products and this is the first to urge making a backup tape.

Be careful that you don't make the mistake I did. I loaded their program and tried to use it, but the program kept destroying itself with errors. Then I noticed that I'd loaded the disk version on my cassette system! After reloading the correct version of the program, it worked perfectly.

ColorForth is an implementation of the F.I.G. (Forth Interest Group) model with most of the high-level commands taken from the public domain fig-Forth model. In addition, several extensions relating to the Color Computer have been added. The program contains CLOADM, CSAVEM, disk commands, and printer commands to enable printing the results generated. The command reference section of the user's manual lists each command, a description of the condition of the parameter stack, both before and after execution, and characteristics of each command that are necessary for compiling other commands.

In addition, Armadillo International Software includes an application command called TURNKEY. After creating an application program (i.e., spaceship), you would execute the TURNKEY command to delete essential information regarding ColorForth. Thus, the user cannot load your application, execute ColorForth, and modify your program. Pretty neat! Unfortunately, the command doesn't strip out unnecessary Forth words to reduce the size of your application. Also, ColorForth applications aren't relocatable.

Remember that ColorForth consists of both cassette and disk versions? If you're running a cassette system, ColorForth simulates disk storage in upper memory. Thus, as far as the program is concerned, it always

thinks you have a disk system. The manual shows you how to set up the program for varying numbers of simulated disk buffers. You must remember to CSAVEM the simulated disk prior to turning off the computer or you'll lose any programs you've created!

Overall, Armadillo International Software has produced an excellent product. The manual is professionally bound and designed for heavy use, but they should have included a table of contents and an index. I spend a lot of time trying to find things in the manual that I'd rather spend programming.

The software has two shortcomings that should be addressed. The authors spent some time developing commands relating to the Color Computer, but they totally ignored the graphics commands. If you're adapting a product for a machine, you should use that machine's capabilities. Secondly, ColorForth doesn't include a Forth assembler. The assembler is under development, so we should see it in the near future.

My overall rating for ColorForth is excellent. Even with the two items mentioned above, I feel my money was well spent. If you want a powerful language for the Color Computer, check out ColorForth.

Darrell Wright

Alcor Pascal
Alcor Systems
800 W. Garland Ave., Dallas, TX 75040, (214) 226-4476
Alcor Pascal $199
Adv. Devel. Pkg. $125

Every Radio Shack computer system user is familiar with BASIC as an example of a "high-level" language, but familiarity with other languages is not widespread. BASIC is easy to learn and use, but in some ways, it is an inefficient language, particularly in execution speed and memory management. It also lacks portability between various computer systems.

Pascal was developed in Switzerland in the early 1970s by Niklaus Wirth. It was designed not only to teach, but also to emphasize programming skills. Pascal is a very structured language. Structured programming is the ideal for which professional programmers strive, since it enhances flow and readability, an important feature when changes and maintenance are necessary. Pascal makes structuring a simple task with the program logic flowing smoothly from top to bottom.

Unlike BASIC, Pascal is truly portable between all computer systems which support the same implementation of the language. Programs written in Pascal are first compiled to an intermediate form called "pseudo-code," or p-code for short. Then, as with BASIC, the p-code is processed at run time by an interpreter. Any system which has the same Pascal system available can process the p-code into its own computer machine code. As an added benefit, since the original program has been partially

processed into p-code, execution by the interpreter is generally faster.

One of the most powerful features of Pascal is its data handling capability. Most languages are limited to real, integer, text, and Boolean data. Pascal includes these, and adds character, set, file, array, and record. In addition, you can define any other data types that you need. BASIC forces the use of numerical data to represent data types. For example, keeping track of package sizes would require using the numbers 1, 2, and 3 to represent small, medium, and large in BASIC. With Pascal, you could define a variable type called size, to consist of the data elements small, medium, and large. Variables would then take on these values directly. When trying to follow program flow, reference to a variable "small" is much clearer than referring to the numeral 1.

Alcor Systems' recent release of Pascal for the Models I and III is the first step in their implementation of the language. Additional releases planned for the near future include standard CP/M, the Osborne system, the APPLE II via a Z-80 softcard, and the TRS-80 Models II and 16. Additional packages are in store for the Heath/Zenith Z89, and the IBM personal computer. With all of these systems supported, true program portability will be achievable for the computer user blessed with more than one system. Alcor will sell their interpreter separately for other computers. This means that you need to purchase the complete compiler package for only one system, use it to write and compile programs, then run the programs on other systems which have the interpreter available.

The System

The programs are designed for execution under TRSDOS, NEWDOS, LDOS, and DOSPLUS. You need 48K and at least one disk drive to develop programs with the Alcor package, although compiled programs can run in less memory. Other Pascal systems have memory requirements greater than 48K, eliminating them from use on the Models I and III. Alcor overcame this memory constraint by developing two compiler variations. A fully memory-resident compiler provides the fastest compilation time, but its size limits source code to around 1,000 lines. To allow larger programs, an overlayed compiler is included in the package. Since various overlays must be called from disk to complete the compilation process, this version runs slower, but source code can grow to 4,000 lines in a 48K system. Programs can also be broken down to separate components for compilation, and called into memory as needed during execution.

The documentation supplied is easy to read and follow. A Beginner's Guide provides very simplified instruction on editing and compiling programs. A separate section gives more detailed instructions for using the text editor. For the beginner, a 70-page tutorial section on standard Pascal includes numerous examples. All of the example programs are included on disk in source code, ready for study and compilation. This is followed by a 25-page system implementation manual outlining the specifics of the system for the Models I and III. A 103-page reference section gives complete details on the language, and includes descriptions of compiler options, error messages, and language extensions. A handy five-panel quick reference guide completes the package.

The system implements the full standard Pascal language. However, over twenty language extensions have been added in the Alcor version, including external procedures and functions, common variables which remain defined after a procedure ends, the ability to find a variable's address, integer representation in hex, mixed mode arithmetic, a message procedure for terminal string output, and the addition of an "otherwise" clause to Pascal's CASE statement.

The Model I/III package also includes additional runtime library routines to fully utilize each computer's strengths. This includes procedures that duplicate SET, RESET, POINT, PEEK, INP, OUT, CLS, INKEY$, and USR for interfacing with machine language routines. String functions, normally limited in Pascal, are implemented, including LEN, MID$, LEFT$, RIGHT$, STR$, INSTR, concatenation, and others. None of the string manipulation power of TRS-80 BASIC, missing on other systems, is lost. Over thirty-five such library routines are included. However, since these take advantage of the TRS-80's hardware, care must be taken if programs are to be transported to other computer systems that do not include them.

An extremely powerful full-screen text editor is used to enter source code. This same editor can be used for BASIC programs, eliminating all the constraints of the 80's line-oriented editor. Program size is limited only by available disk space. As a protection feature, all editing is performed on a work file, leaving the original file intact. The original is deleted by the system only after the file has been successfully updated. Thus, complete loss of a file through hardware or software glitches is practically eliminated. "HELP" files can be called from the editor if needed. Space limitations prevent a full description of the Blaise editor. Suffice it to say that it is, in itself, a mini-wordprocessor, more powerful for program composition than Scripsit.

Efficiency and Speed

Alcor compared their system to

other available Pascals by using a benchmark program published in *Byte* magazine in September, 1981. On a Z-80, running at 4MHz, Alcor was clearly the fastest. Speed is not as fast on the Models I and III since they run at 1.7 and 2.0 MHz, respectively.

The compiler is designed to generate compact p-code, resulting in a very efficient memory use. In large programs, size can still be critical. An additional advanced development package is available which includes a p-code optimizer. Running the compiler-generated p-code through this option results in a further size reduction of 25 to 30 percent for even more memory savings.

Since p-code is an intermediate language requiring interpretation, execution speed still cannot equal that of machine language. In some applications, speed is a critical factor. To solve this problem, the advanced package also includes a native code generator. This program takes the p-code and translates it into Z-80 native machine language for direct CPU execution, which can further enhance speed by three to five times. Note the word "native." What this means is that the machine code is produced for the Z-80 as used in a particular system. The generated code is no longer transportable to other systems.

Machine language programs tend to use more space than their BASIC or p-code counterparts. This tradeoff between space and speed exists in all cases. Alcor provides two nice solutions to this dilemma. First, native code and p-code can be mixed in any application. Second, the compiler allows external functions and procedures. With these enhancements, program sections that need increased speed can be translated into machine language while non-speed dependent sections can be left in optimized p-code. Using a linking loader, all sections can be joined together to give a program which makes optimal use of memory while still running as fast as possible.

Impressions

Other versions of Pascal have been released for the TRS-80, but all suffer from limitations. UCSD Pascal from FMG Corporation uses so much memory that only around 250 lines of code can be compiled. Tiny Pascal from Radio Shack is an extremely limited subset of the language, without built-in disk access. Pascal-80 is not implemented on other computer systems, and does not use the full Pascal language (it is missing the WITH command and varient records). Alcor's system appears to overcome all of these limitations. It runs under all popular DOS systems, incorporates a runtime library to avoid sacrificing those features which make the TRS-80 an outstanding computer, uses overlays to allow compilation of 1000 to 4000 lines, adds useful language extensions, optimization and machine language conversion capability, and is (or will be) fully implemented on a variety of computer systems to allow programs to be swapped between systems that you own.

My past programming experience has included BASIC, FORTRAN, ALGOL, and only a passing familiarity with Pascal. After using Alcor Pascal for several weeks, I am sold! Pascal provides some of the best features of all high-level languages while making it much easier to follow program logic. Alcor's documentation is excellent, and the tutorial section provides a good introduction to the language. I would purchase one of the many fine teaching guides on Pascal to more fully learn the intricacies of the language.

Pascal has rapidly become a favorite language of instructors and professional programmers, and I hope microcomputer users will begin to feel the same. Now, if only someone would write a program that would translate all my "spaghetti" BASIC programs into Pascal . . .

Paul R. Prescott, M.D.

"Starting Forth"
Leo Brodie
Published by Forth Inc.
Available at Major Bookstores
$15.95

When I first saw this book in the store, I must admit it didn't look very interesting. I felt I was a pretty good Forth programmer and didn't need an introductory book about the language. I'm thankful that I don't always listen to my first

impressions of things!

Starting Forth is one of the most amazing books I've ever read! This book contains information for everybody from beginning to the advanced Forth programmer. Mr. Brodie has written the book for the beginning to average programmer, then included footnotes that specifically address the beginning and experienced users.

Each chapter covers a specific area of Forth (i.e., Chapter four is devoted to IF...ELSE...THEN conditional statements, while Chapter six is devoted entirely to programming loops). Within each chapter, you'll find many examples and tidbits of information that prove extremely helpful. Chapter ten, which examines Input/Output commands, includes the complete definition of a random number generator as one of its examples. You could incorporate this definition in one of your programs and have it work flawlessly!

Mr. Brodie has spent considerable effort on "detail". When he describes a Forth word, he gives a complete description of the word and includes examples showing how it is used. You can enter the examples at your machine and they work! I've read many articles and books that include program examples, but when I enter the examples in my machine I can't get it to work properly. It's a refreshing change to find workable examples.

At the end of each chapter, you'll find a summary of all the words covered within the chapter. In addition, you'll find review problems to help you practice using each of the words. Within each chapter, Mr. Brodie uses numerous illustrations to show what is happening during the execution of a word.

While the entire book is extremely good, Chapters nine, ten, and eleven are of special interest to the experienced Forth programmer. Chapter nine discusses how Forth actually works internally. The chapter covers the operation of the interpreting and compiling phases of Forth. Chapter ten contains information on communicating with the terminal and the disk. Chapter eleven describes the words that are necessary for extending the compiler. Using these words, you can create a set of compiling words unique to your own application!

Chapter twelve ties the entire book together by presenting three complete examples of Forth applications. By analyzing or using these examples, you can gain an appreciation for the power of Forth.

If you're looking for an excellent way to learn how to program in Forth, I strongly recommend this book. It's one of the best investments you can make.

Darrel Wright

Bytewriter
Model I, II, III
125 Northview Blvd.
Ithaca, NY 14850
(607) 272-1132
$795, Cable $39

When I decided to purchase a printer for my TRS-80 Model I system, I made a list of those features that were most important for me to have in a printer. I am a student and the primary uses I have for a printer are school papers and letters. Therefore, I decided one requirement was to avoid the "dotsy" look of most computer printers.

After looking at many printers, I finally decided upon the Bytewriter, made by Bytewriter, in Ithaca, New York. The Bytewriter is an Olivetti Praxis 30 typewriter with a computer interface added. The Praxis is an electronic typewriter and prints using a daisywheel. The Praxis keyboard is a standard typewriter keyboard, and features a "Keyboard II" mode. In this mode, several standard keyboard characters are replaced by symbols needed for foreign languages. Missing from the keyboard are the "less than" symbol, "greater than" symbol,and the up arrow. The interface accepts standard, parallel-printer output from a computer and drives the Bytewriter.

The Bytewriter sells for $795 plus cable $39, and it is available from dealers or directly from the company. Cables are available for TRS-80 Models I, II, and III, Apple II, Osborne I, and the IBM personal computer. The machine still functions as a typewriter and is serviceable by Olivetti dealers, and the 90-day typewriter warranty is still honored by Olivetti. Service for the interface is provided by your local dealer or by Bytewriter. Daisy wheels cost about $30, and ribbons cost about $4, and are available from Olivetti dealers.

I ordered my Bytewriter directly from the company, and it arrived slightly more than a week later. It connects directly to my computer and operates as advertised, at eight to twelve characters per second. The print quality is excellent, especially with carbon ribbon that is available. The printer prints 10 characters per inch as delivered, and the print scale on it is for 10 cpi; however, twelve and fifteen characters per inch are switch selectable. The daisywheel must also be changed to obtain the proper appearance in the other print densities.

The Bytewriter prints a space for any character it cannot print, such as the arrows and the "greater than" and "less than" symbols. The printer automatically supplies a line feed with a standard carriage return. This can be overridden, however, by a switch which is inside the typewriter. Regardless of the position of the switch, ASCII code DC2 will generate a carriage return without a line feed. Underlining can be obtained by two methods, as no "underline off" codes are recognized. Method one is to send a carriage return without a linefeed, and then print underline characters in the appropriate places on the line. Method two is to backspace at the end of a phrase to be underlined and then print the underline characters. I have modified tape Scripsit to underline by method two.

The Bytewriter operates well with Scripsit. Since the print head of the Bytewriter always returns to the left margin, I found that it speeds up printing to set the left margin manually on the Bytewriter and to set the left margin in Scripsit at zero.

The Bytewriter has performed flawlessly with one exception. When I first got it, it occasionally dropped a character near the beginning of a line. I called Bytewriter, and they told me they knew of the problem and had a fix for it. I sent my interface board back to them, and they replaced it free of charge. Their turnaround time was excellent; I mailed my board one Friday and received the replacement the following Friday. The problem has not occurred since I installed the new board.

I am pleased with the Bytewriter, and would recommend it for someone with requirements similar to my own, and who understands that 8 to 12 characters per second is SLOW. The Bytewriter compares favorably with other low cost daisy wheel printers, and has the added advantage of being a typewriter.

Steven B. Greene

Reviews

Penetrator
Model I/III Disk or Tape
Melbourne House Software, Inc.
347 Reedwood Drive
Nashville, TN 37217
(615) 361-3738
$24.95

Penetrator is a fast-action arcade game for the TRS-80 Models I/III that is written completely in machine language. Supplied on a disk or tape, it is a fairly recent release from Melbourne House Software and costs $24.95.

This is the game version of the popular arcade game "Scramble." Penetrator features sound, graphics, and one of the most entertaining and exciting game scenerios I have ever seen on the TRS-80.

After the game is loaded, a slowly forming cursive written "Penetrator" title page appears. This is interesting to watch the first few times the game is played, but in later games, the CLEAR and BREAK keys may be hit simultaneously to skip this title page creation.

The player is given five ships to penetrate a series of caverns which store an illegal, neutron-bomb. This storage must be destroyed, and the player must return through the series of four caverns to succeed. The caves become sucessively more difficult to penetrate.

For example, the first two caves are defended with both missiles and radar dishes. The third cave is actually a long stretch of concrete canyons, which are very difficult to maneuver through. The fourth cave is extremely difficult because, in addition to radar bases, there are paratroopers who can easily destroy the player's ship.

The controls used to move the ship are fairly straightforward and easy to master. The up and down arrow keys are used to move vertically. Holding the left arrow key slows the ship down. Holding down the right key increases the ship's speed and pressing it rapidly causes laser fire to be emitted from the ship.

One of the best offense/defense weapons on the player's ship is the bomb. Release of the bomb is accomplished by means of the space bar, and this can be a real life-saver.

Of course, the enemy has many ways of destroying the player. There are missiles which fly up and may annihilate the ship and there are radar bases which gather information to allow more missiles to be fired. The paratroopers are most difficult to evade and will destroy one of the five ships should they come in contact with it.

If the ship makes it through every one of the caves, your object is to destroy the large neutron storage area by dropping a bomb down a small shaft. If the bomb misses, then the ship is destroyed. However, if you hit the storage, a fantastic explosion occurs and a little song is played to congratulate you.

At this point, you must return to control center by going back through all of the caves, with the aliens even more eager to destroy. This is extremely hazardous, and one would be very lucky to make it.

One of the highlights is the well-done graphics. Missiles, ships, radar bases and caverns are very impressively made. I was particularly astounded at the explosion which occurred when the neutron bomb storage was hit. It is all fascinating to watch.

The sounds are also a big highlight of this game. Not only are there high-quality sounds for explosions, lasers, and bombs, but the song which I mentioned earlier is also amusing. The game isn't the same without the sounds, which goes to show how well done and important they are.

Five high scores can be retained with the disk version of this game, along with the names of the players who made them. If you want a completely different cavern, or perhaps fewer missiles and radar bases in one stage of the game, you may accomplish this by using the landscape editor. This editor is available in both the cassette and disk versions of this fine game, and it introduces even more versatility.

Finally, there is a training mode for this one-or two-player game, in case a person is not experienced with Penetrator. An especially helpful aspect of this is that any cave may be selected for training.

Never before have I seen or played a game made for the TRS-80 as exciting as Penetrator. On the day that I got it, I played it until two in the morning. It is addictive and offers features not found in most games. I highly recommend it to any TRS-80 owner who enjoys games. Penetrator is a wonderful creation, and it will prove to be fun for all.

Tim Knight

Architectural Engineering Library

50 programs in a total of 10 categories including:

Heat Loss/SLR
Timber Beams,
Steel Beams,
Wood Trusses (4 types),
Concrete Beams, Walls,
Slabs and Footings.

TOTAL COST: **$225** (U.S.)
Individual Categories: **$35**
Disk, ESF or Cassette

ERIC CLOUGH
Box 52, Winlaw, B.C.
Canada VOG 2J0
or
DHU-GLAS — Box 1664
Sausalito, CA 94966

91

Dental Computer Newsletter

E. J. Neiburger, D.D.S., Editor
1000 North Avenue
Waukegan, IL 60085

The D.C.N. is an international group of dentists, physicians and office management people who have interests in office computers. Though the emphasis is on microcomputers, many members use minis. We cater to all makes and brand names.

Annual membership dues $15.00. Membership runs from January to January. If you join mid-year, we will supply you with the year's back issues.

92

TRS-80 MODEL I[T.M.]* GOLDPLUG - 80

Eliminate disk re-boots and data loss due to poor contact problems at card edge connectors. The GOLD PLUG - 80 solders to the board card edge. Use your existing cables.

CPU/keyboard to expansion interface $18.95
Expansion interface to disk, printer, RS232, screen printer (specify) . . . . . .$9.95 ea
Full set, six connectors. . . $54.95

VISA master charge

EAP COMPANY
P.O. Box 14, Keller, TX 76248
(817) 498-4242
*TRS-80 is a trademark of Tandy Corp.

93

COMTRONIC SYSTEMS

PMS-1
PROPERTY MANAGEMENT SYSTEM

- Saves Hours of Bookkeeping
- Increases Accuracy
- 100% User Support
- User Friendly System
- 350 Units Per Diskette
- Complete Tenant File
- Current Tenant Report
- Late Rent Report
- Late Rent Notices
- Rent Payment History
- Vacancy Report
- Income Statement
- Expense Ledger
- Change of Rent Notices

For TRS-80 Model I and III 32K One Disk Drive.
Excellent for Management Companies or Individuals.
$169.95 on Diskette with Manual, Documentation and Sample Files.

LOAN AMORTIZATION PROGRAM

Prints or displays an amortization schedule for a given mortgage contract. Perfect for Real Estate Investors.

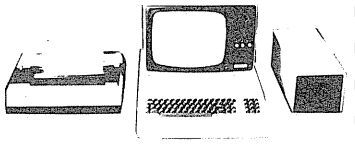
For TRS-80 Model I and III 16K Computers
$12.95 on tape and $15.95 on Diskette

"SATISFACTION GUARANTEED"

COMTRONIC SYSTEMS
4028 Somerset Lane, Kent, WA 98032

94

LNW80 COMPUTERS



IN STOCK NOW!! CALL FOR SPECIAL PRICES ON LNW80s, NEC PRINTERS, RGB COLOR MONITORS, AND TANDON DISK SYSTEMS.

***** NEW LNW SOFTWARE! *****

EXCELLONIX

7180 Woodrow Wilson Dr
Los Angeles, CA 90068

(213) 650-5754 - (714) 973-1939

95

MODEL II & 16 OWNERS

Enhancements to enable your Radio Shack Single Drive Basic Programs to do more for you and custom written programs.

Radio Shack
General Ledger
Payroll
Accounts Payable
Custom Programs
Hotel Reservation Systems
Financial Eval. Calcaluations
1982 Income Tax System

Write To:
Joseph W. Brooks, Accountant
24 Woodland Terrace
Lake Placid, NY 12946
(518) 523-2320

We use all in our business and support all systems.

96

MODEL I

SOFTWARE BY: *Rip*

"Masque"

MASK AND PROTECT YOUR BASIC PROGRAMS. EDITING OF MASKED LINES IS VIRTUALLY *IMPOSSIBLE*. MAKES MONITOR AND HARD-COPY LISTINGS DIFFICULT TO READ.....

*** MASQUE1 ***
INSTANT MASKING AND DE-MASKING OF BASIC PROGRAM LINES..... $24.95

*** MASQUE2 ***
PERMANENTLY MASKS BASIC PROGRAM LINES. CANNOT BE DE-MASKED.... $29.95

*** MASQUE3 ***
MASQUE1 AND MASQUE2 COMBINED IN ONE PACKAGE..... $39.95

- REQUIRES 48K, DISK DRIVE -
- SEND CHECK OR MONEY ORDER TO -

GLENN/CLIFF ASSOCIATES
8301 EAST MONTEBELLO
SCOTTSDALE, ARIZONA 85253

97

Easy File: Database, Mail List, ect... **$15.00***
Easy File: Disk, 32K Min., Model I/III **$15.00**
Stock Quest: 39-week Data File..... **$11.00**
Household Inventory: Database Program..... **$7.00***
Zaars Citadel: RND Situation Game.... **$5.00**
Miners: Hammurabi Type Game..... **$4.00**
Stock Game: Easy Type Stock Market Game..... **$5.00**

DATA MANIA, INC.

51625 Chestnut Road
Granger, IN 46530

Add $2.00 P/H
* 16K or 32K, Specify
For 16K TRS-80[m] Color and Model I/III Tape Only. Specify

98

IEEE-488 TO TRS-80* INTERFACE

Everything needed to add *powerful* BASIC GPIB-488 controller capability to TRS-80 Model 1 or 3, Level 2 or DOS with a minimum of 16K.

488-80B
For Model 1 Operation

488-80C
For Model 3 Operation

Model 488-80B or 488-80C Price: $375.
+ shipping, insurance & tax

WHEN ORDERING SPECIFY DISK OR TAPE

SCIENTIFIC ENGINEERING LABORATORIES
11 Neil Drive • Old Bethpage, NY 11804
Telephone: (516) 694-3370

*Trademark of Tandy Corp.
There is no affiliation between Scientific Engineering Laboratories and Tandy Corp. or Radio Shack.

99

NO LIMIT **Epson Giveaway** NO LIMIT

Epson MX70/80 Cartridges

$5.00 EACH

Min. 3 of same color

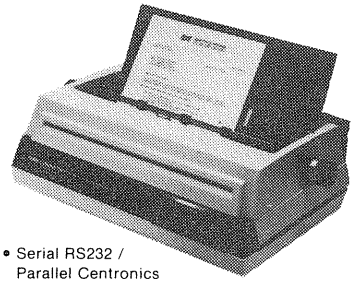
Reloads $2.50 each Min. 12
$30.00 a Doz. of Same Color
Cartridges and Reloads Available
In Black, Red, Green, Blue, Brown

AR systems
Dept. 8A, 35 Cherry Court
East Northport, N.Y. 11731

N.Y.S. Residents Add Tax. Add $2.00 Shipping & Handling
Prices Subject to Change
Allow Clearing Time for Personal Checks
Money Orders & Certified Checks shipped same day

100

SMITH-CORONA TP-1
DAISY WHEEL PRINTER



- Serial RS232 / Parallel Centronics
- Ask about our discount catalog

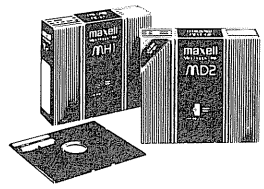
$575

MICRO TECHNOLOGY
7817 Ivanhoe Ave., San Diego, CA 92037
(714) 457-2149

101

maxell

the top of the line



at the lowest price!

Call our Modem Hotline (anytime) - 619.268.4380 for exclusive monthly specials. Our free catalog contains more than 600 fantastic values.

ABC Data Products
(formerly ABM)
8868 CLAIREMONT MESA BLVD.
SAN DIEGO, CALIFORNIA 92123

ORDERS ONLY 800-850-1555 | ITT TELEX 4992217 | INFORMATION 619-268-3537

102

Modem
$99.95*

At Last -for the TRS-80 Mod I

* Needs no expensive RS-232
* Direct Connect
* Connects to Cassette port
* Up to 300 baud
* Half duplex send and receive
* Includes instructions and Software
* Communicate with other Host systems- Compuserve, Source, Bulletin boards, ect.
* Available Soon for Mod III & TRS-80 Color computer
* Price includes shipping*

Send check or M.O. to:

Comstar Research
PO Box 771
Madison Heights, MI 48071

* Alaska, Hawaii, or C.O.D. add $3.00 shipping

103

LARGECAPACITYSYSTEMSLAR GECAPACITYSYSTEMSLARGECAPACITYSYS

MI **M3**

GENERAL LEDGER

400+ACCOUNTS

5000+TRANSACTIONS/MONTH

NO OTHER SYSTEM OFFERS....

+REPORT FLEXIBILITY/CAPACITY
+DEPARTMENT P & L (UP TO 5)
+UNLIMITED ACCOUNT CATEGORIES
+STATEMENT OF CHANGES (ASSETS)
+PERCENT P & L comparison
100% sales MTD vs YTD
or net sales QTD VS YTD
or total Exp. MTD VS QTD
+WITH TDOS (A MINI DOSPLUS)

149.95

reg. 32K 2 drives

manual $30.00 test set $50.00

HOLMAN D-P SERVICE
2059 W. LINCOLN
OROVILLE, CA 95965 VISA/MC 3.00 S&H
(916) 533-5992 COD

STEMSLTYSYSTEMSLARGECAPACITYSYSTEMSLARGECAPACITYSYSTEMS

104

AIDS(TM) USERS...NOW YOU CAN

KONVERT!

+ REFORMAT AIDS(TM) FILES!
+ ONE-PASS FILE CONVERSION
+ ADD, CHANGE, DELETE MULTIPLE FIELDS
+ HANDLES ALL FIELD TYPES
+ FILE SIZE NOT LIMITED BY AVAILABLE MEMORY
+ EASY-TO-USE...MENU-DRIVEN AND INTERACTIVE
+ STILL MORE FEATURES!
+ INCLUDES MANUAL

ONLY $29.95 POSTPAID FROM:

SEASOFT

PO BOX 7362, SEATTLE WA 98133
PHONE: (206) 775-9801

REQUIRES MODEL I, 2-DISK, 32K
WA RESIDENTS ADD TAX
AIDS(TM) BY META TECHNOLOGIES

105

Toucan Presents:

A new generation of adventures — with graphics!!!

Subterranean Encounter — A fantastic medievai adventure which will take you from above ground with it's castles, wizards, moats, and alligators to the subterranean depths whose evil and danger knows no bounds. A new plateau in adventuring — from Toucan Software, who else?? (TRS-80* Model I or III (Please specify), 32K, 1 Disk Drive). Just $29.95 at fine software stores everywhere or directly from us.

TOUCAN SOFTWARE
4024 Canonero Court
Fair Oaks, CA 95628
(916) 966-4241

*TRS-80 is a registered trademark of Radio Shack, a division of Tandy Corporation.

106

TAXES

Software to assist you in organizing and preparing an accurate Tax Return.

TAX I
- 1040, Schedule A and B
- For 16K Timex-Sinclair, Commodore, Texas Instruments, Apple, Radio Shack Color, Mod I/III
- Program on cassette, tax deductible **$29.75**

TAX II
- Tax Estimator
- 1040 and frequently used schedules
- For 48K Apple II, Radio Shack Mod I/III.
- Program and Tax Tables on Disk, tax deductible **$34.75**

Specify machine with your order
(Pa. Residents add 6% Sales Tax)

Eugene H. Bock & Associates
Consultants 215-362-5082
135 Heartwood Dr., Lansdale, PA 19446

BONUS 1983 H & R Block 266 page Tax Workbook ($5.95 value) included if you order before Mar. 15th

107

BARCLAY WHYTE ASSOCIATES

WORDPROC I·III
VERSION 3·0

Basic plus machine language Word Processor

TRS-80* Model I/III 48K Disc

Special Version Now you can embed control codes in your text for

Superscript **Subscript** **Underline**
Italics **More** Supports all MX-80** typefonts

Still only **$14·95 each, with disc**
$10·95 each, documentation only

BARCLAY WHYTE ASSOCIATES
P.O. Box 948,
New Westminster
B.C. Canada. V3L 5C3.
Add $1·00 for postage
B.C. Residents add tax at 6%

*Trade mark of Tandy Corporation
**Trade mark of Epson America

108

MAILING LIST & COIN INVENTORY — MODEL I PROGRAMS —

Mailing List — high capacity sort by any field, 30-char. last & company name, 20-char. remarks, prints labels specific printing orders.

Coin Inventory — high capacity, quick M.L. sort print any or all coins.

Arctic Computer Systems
5031 Nemiskam Rd. NW
Calgary, Alta, Canada T2K-2P8

$29.95 each or **$45** for both.
Add $2.50 for shipping.
NEWDOS/80 2.0 & Disk required.

109

COMTRONIC SYSTEMS

TRS-80 Model I, III and Color Computers. 16K RAM Required.

ARCADE GAME	TAPE	DISK
B-52 Strategic Bomber	$10.95	$12.95
Centurion	$10.95	$12.95
F-15 Death Pilot	$10.95	$12.95
Raid on Entebbe	$10.95	$12.95
Street Fight	$10.95	$12.95
Submarine Attack	$10.95	$12.95
Air War	$ 9.95	$11.95
Killer Satellite	$ 9.95	$11.95
ADVENTURE GAMES	**TAPE**	**DISK**
Space Colony	$10.95	$12.95
Air Cav	$ 9.95	$11.95
WAR GAMES	**TAPE**	**DISK**
Battle of Midway	$10.95	$12.95
Next War	$10.95	$12.95
Nuclear Holocaust	$ 8.95	$10.95

DISCOUNTS:
Buy 2 Games, Save 10%
Buy 3-4 Games, Save 15%
Buy 5 Games, Save 20%

Add .55 shipping or $2.55 for C.O.D.
"SATISFACTION GUARANTEED"
COMTRONIC SYSTEMS
4028 Somerset Lane, Kent, WA 98032

110

MULTI DOS* is the New DOS LEADER with more features than any other DOS and at the best price–99.95

ORDER TODAY FOR SURPRISE FREE BONUS.

TRS80 MOD I/III VISA/MC

C D C
13715 Vanowen Street
Van Nuys, CA 91405
(213) 873-6621

* TM

111

Quatrk Computer Systems, Inc.
P.O. Box 638
Bixby, Oklahoma 74008-0638
(918) 369-1359

TAX PROGRAMS FOR TRS-80 MOD III
Complete programs for tax preparers only
Programs (screen print only) for anyone who wants to do their taxes on the TRS-80 MOD III

BASIC PACKAGE
SMAL LDOS OPERATING SYSTEMS
FORM 1040 — U.S. INDIVIDUAL TAX RETURN
SCHEDULE A & SCHEDULE B
TAX PREPARER'S WAIVER & INVOICING

FEATURES
MENU DRIVEN PROGRAMS
DIRECT PRINTING OF THE FORMS & SCHEDULES WHEN USED WITH MOST DOT MATRIX PRINTERS
THE 1040 FORM REQUIRES PROPER SETUP TO BE ACCEPTED BY THE IRS
ALL TAX RECORDS ARE STORED ON DISK

COST

TAX PREPARERS BASIC PACKAGE	$320.00
SCREEN PRINT ONLY	170.00
MANUAL & PRINT OUTS ON THE IDS PRINTER	50.00

ADDITIONAL SCHEDULES (*requires extended system)

SCHEDULE C	$ 80.00
SCHEDULE D	90.00
SCHEDULE E	110.00
SCHEDULE F	n/a
SCHEDULE G	80.00

BASIC SYSTEM REQUIREMENTS
TRS-80 MOD III WITH 48K & TWO DISK DRIVES

***EXTENDED SYSTEM REQUIREMENTS**
TRS-80 MOD III WITH 48K & THREE DISK DRIVES

TRS-80 — TRADEMARK OF TANDY CORP
SMAL LDOS — TRADEMARK OF LOGICAL SYSTEMS, INC
PRICE OF MANUAL & PRINTOUTS APPLICABLE TOWARD COST OF PACKAGE PURCHASED

112

LARGECAPACITYSYSTEMSLARGECAPACITYSYSTEMSLARGECAPACITYSYSTEMS

SMALL & LARGE CAPACITY
ACCOUNTS RECEIVABLE
FOR MODEL I/III
5000+CUSTOMERS
15000+ TRANSACTIONS
BALANCE FORWARD 99 TRANS CODES
30-60-90-120 AGED STATEMENTS SHOW
DATE/INV#/DESCRIP/AMT (WITH AGEING)
SELECTIVE FINANCE CHARGES & RATES
RATES FAST ENTRY POSTING W/AUDIT
REPORT SUB-ACCTS % OF CREDIT LIMIT
DATE OF LAST PAYMENT SALES ANALYSIS
SPECIAL 90 DAY ACCOUNTS LABELS

(REQ. 32K 2 DISKS)

$149.95
TEST SET $50.00 MAUAL ONLY $30.00

HOLMAN D-P SERVICE
2059 WEST LINCOLN 3.00 S&H
OROVILLE, CA 95965 VISA OR MC
916-533-5992 COD*
*COD (CASH, CERTIFIED, OK, MONEY ORDER)

LARGECAPACITYSYSTEMSLARGECAPACITYSYSTEMSLARGECAPACITYSYSTEMS

113

Get
FREE
Programs
for the
Color Computer
Join the
East Texas Color Computer Club

2101 East Main St.
Henderson, TX 75652

114

Mayday

HELP/CMD...a utility program for your NEWDOS-80* system. A HELP request for any of the commands returns a mini-page of information to the screen on the subject. Can be modified to get what YOU want to see. Machine language program that can be HELPful to you .....**$16.50.**
BACKUP/CMD...Does a full disk backup for your NEWDOS disks from a simple and quick program .....**$16.50.**
COMPACT/BAS...Squeeze those disks some more for extra space and faster operation of your programs .....**$16.50.**
LISTR/BAS...Now you can UNSCRAMBLE that compacted program and print out a neatly organized and easy to read listing. Works on most any ASCII saved program .....**$16.50.**

Order **all four** of the above on one disk for the special price of....**$46.50.** Remember...ALL of these prices INCLUDE shipping. Please state I/III, S or DD needs.
MAYDAY software wants to be your SOURCE for more UTILITIES, GAMES and HARDWARE items. Write for more listings.

Wisc. residents add 5% tax
Personal checks should allow 2 weeks extra

MAYDAY software
P.O. Box 66 Rock Creek Road
Phillips, Wisconsin 54555
(715) 339-3966
* APPARAT, Inc. **VISA/MC** are welcome

115

Arrow Ambush

an exciting new game by SOFTLEX Games

- TRS-80 mod. 1 level II
- gets harder as you get better
- written in unprotected BASIC for easy user modifications
- $9.95 (cassette, postage paid)

SOFTLEX Games
40 Highland Avenue
Lexington, MA 02173

Name_____
Address_____
City_____State_____Zip_____
Send $9.95 NOW!

116

ribbons

For Radio Shack Printers

New Ribbons For:
- Daisywheel II
- Line Printer III & V
- Line Printer VI & VIII
- Line Printer VII

Recycle Your Used Ribbons:
- Service available for most printer ribbon types.

Buy directly from a major manufacturer and save!
FREE CATALOGUE!

*Aspen Ribbons, Inc. is not affiliated with the makers of Radio Shack products.

Aspen Ribbons, Inc. 1700 N. 55th St.
Boulder, CO 80301 (303)444-4054
End User Orders: 800-525-0646
Wholesale Orders: 800-525-9966

117

FREE
business software directory

Micro Architect, Inc.
96 Dothan St., Arlington,

MA 02174

125

CONVERT-A-DISK

Single-sided disk drive users increase your disk storage by 100%!!!

Convert-A-Disk will work with any 5¼-in. single-sided disk drive no matter what type of computer you are using.

Why pay hundreds of dollars to increase your disk storage?? Convert-A-Disk will allow you to convert all your existing disks to give you a 100% increase disk storage capacity.

Send $15 check or money order to:

CONVERT-A-DISK
P.O. Box 15277
Portland, OR 97214

126

GIANT Introductory Offer!

NEW FLIGHT SIMULATION GAME WITH ADF NAVIGATION!

Mission–bomb and strafe. Take off, navigate by ADF/DME to target and back and land on runway. You might have a dogfight, stall, run out of fuel, overshoot landing...lots can happen! Instrument panel, air strip, fighters, etc. ...all in graphics.

SUPER BONUS!
7 personal programs including Income Tax, Mail List, etc..
OVER $100 VALUE!

All for $29.95 Disk only
Requires 32K–TRS 80 M I/III

TM Tandy Corp.

BAP$ Software
6011 San Felipe Houston, TX 77057

127

Do you have a few holes in your collection of **80-U.S. Journals?**

We might be able to fill them!

Contact:

80-U.S. Back Issues

3838 South Warner
Tacoma, WA 98409
(206) 475-2219

129

– Professional –

REAL ESTATE SOFTWARE

for APPLE, TRS-80 & CPM SYSTEMS

- **PROPERTY MANAGEMENT SYSTEM: $450**
  - 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
  - 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
  - Loan Sales/Purchase
  - Tax Deferred Exchange
  - APR Loan Analysis
  - Loan Amortization
  - Depreciation/ACRS Analysis
  - Loan Wrap Analysis
- **WORD PROCESSOR – WORD STAR: $295**

master charge VISA

Software Realty Company™

At Computer Stores Everywhere or Order COD Direct
Cal Residents add 6½% Sales Tax
(213) 372-9419

Suite E, 1116-8th Street, Manhattan Beach, CA 90266

130

CP/M DATABASE

*The **toughest DBMS** on the block and the **only one** with all these features*

- ★ **40 fields, 21 field types, data compression**
- ★ **Search** with selectors on any, all, or selected fields by string, substring, range or comparison or **instant** retrieval by record number or binary search
- ★ **Multi level sort** any, all, or selected fields in any order
- ★ Print **reports** to the screen or to the printer on any, all, or selected fields in any order
- ★ **Transfer, post** or **move** data in **any, all,** or **selected** fields to one or several databases
- ★ **Arithmetic** in stored file, in calculated field or apply varied calculations to any applicable field
- ★ **3 years** in development—**proven** in 2 years of test marketing
- ★ **Production input** on any number of fields

Now available for: TRS 80 model 1/111 $250*
*TRS 80 model 11/16 *Osborne 1—*CPM $300*

C D C
13715 Vanowen Street, Van Nuys, CA 91405
Write or phone (213) 873-6621

*Registered TM ISA, Tandy, Osborne, Digital Research

131

MODEL I/III SOFTWARE BY: Rip

*** RIPCHECK ***.....$59.95
SUPER FAST MENU DRIVEN PAYABLES SYSTEM (4) USER DEFINED ACCOUNTS - REVIEWS - (156) VENDOR NAMES/ADDRESSES - LPRINT CHECKS/REG. - (4) SEARCHES WITH TOTALS

*** RIPBONDS ***.....$89.95
A COMPUTERIZED PORTFOLIO FOR THE BOND INVESTOR - YTM - CURRENT ACCRUED INT. OUTSTANDING CURRENT/SOLD BOND REPORTS

*** PAYROLL5 ***.....$79.95
CONVERT RADIO SHACK'S ® DISK PAYROLL TO OPERATE 5 SEPARATE DIVISIONS/DEPTS. WITH INDIVIDUAL & CONSOLIDATED RECAPS - SEND ORIGINAL PROGRAM DISKETTES -

*** SHUTTLE COMMAND ***.....$19.95
TRANSPORT YOUR GRP FUEL CELLS THROUGH BI-DIRECTIONAL METEOR SHOWER - MACHINE LANGUAGE ARCADE TYPE GAME - AUTOMATIC HIGH SCORE SAVE - ONLY ORDER IF YOU'RE READY FOR "THE ULTIMATE CHALLENGE"

- - SEND CHECK OR MONEY ORDER - -

GLENN/CLIFF ASSOCIATES
8301 EAST MONTEBELLO
SCOTTSDALE, ARIZONA 85253

132

TRS-80 DISK and TAPE USERS!!!
Protect your investment in disks and tapes! END loading problems One **TOOL** does it ALL!

Tape Reproduction System
Tape or **Disk Utility for Model I** or **III**

- Read TAPE and/or DISK
- Write TAPE and/or DISK
- Verify TAPE written will load
- **ANY** TRS-80 protocol TAPE
- Identifies BASIC, SYSTEM, EDTASM
- Change program name as desired
- Select tape speed H/L (Mod III)
- Select tape port 0/1 (Mod I)
- **OFFSET** load addresses
- **UNOFFSET** load addresses if offset
- Displays operational STATUS plus
  - Diskette directory (0-3)
  - Load Map (start-end addresses)
  - NAME, LENGTH, FORMAT of data
  - START, END, EXECUTE addresses
- Automatic protocol conversion
- **SINGLE** disk drive copy (Mod I)
- Archives on TAPE for DISK or TAPE
- Run TAPE programs from DISK
- 14 Menu-driven features in all!
- DISK features require TRSDUM

TRSTUM-16K tape systems ..... ONLY $16.95
TRSDUM-1 drive TRSDOS systems ..... ONLY $17.95
Specify Model I or III - add $1.00 postage and handling
Send check, money order or SASE to:

CRB Microtools
"Software MTools for the 80's"
14835 N. First Avenue
Phoenix, AZ 85023
TRS-80 & TRSDOS - trademarks of Tandy Corp

133

CARTRIDGES FOR EPSON MX-80*

Factory Fresh New Stock. Your choice of Black, Blue, Red Brown or Green.

1 for $6.95
3 for $16.00
12 for $59.95

All Prices Postpaid. Mix or match any way you want. Michigan orders add 4% tax. Mail Sales Only. USA Orders Only.

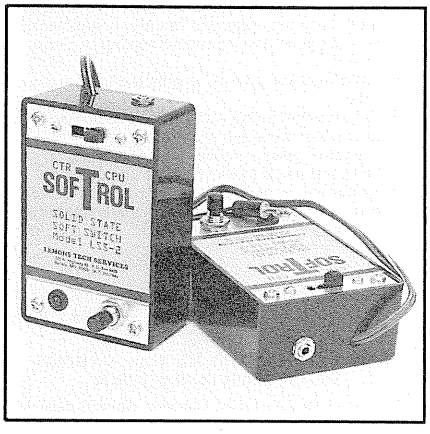
Pete Skeberdis
PO Box 27
Fremont, Michigan 49412

Prices good only while current supplies last.

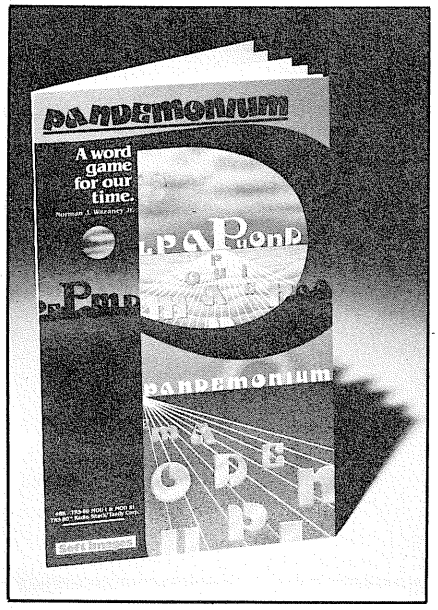
* EPSON MX-80 Trademark of Epson America Inc.

New products

Cassette Controller



Pandemonium



C-language Compiler

MISOSYS announces LC, a C-language compiler for the TRS-80 models I and III. It is an integer-only compiler that supports all statements except "struct", "union", and "typedef". LC supports all operators except "->", ".", "sizeof", and "typename".

LC supports I/O redirection, command line arguments, dynamic memory management, and sequential files for read, write, and append. Floating point routines in ROM are accessible and an extensive library supports graphics and string routines as well as DOS calls. LC generates assembler source code that is compatible with EDAS IV.

LC requires a two-drive, 48K model I or III and LDOS 5.1.x. It includes 200 pages of documentation, and is priced at $175 plus $4 shipping. Contact MISOSYS, P.O. Box 4848, Alexandria, VA 22303-0848, phone (703) 960-2998.

#200

Cassette Controller

SOFTROL is a solid-state cassette recorder controller from Lemons Tech Services. It allows for pushbutton positioning of tape without removing plugs, reduces "switching noise" on tapes, protects CPU relay contacts from early failure, has a motor-off delay to prevent capstan/pinchroller pressure at the end of programs and automatically gives a gap between saves of a program.

The new unit can be used with any standard plug recorder and is easily connected to your system. It can also be used to computer switch any DC load of five to fifteen volts and up to 1.5 amperes. Price is $18.99 from Lemons Tech Services, P.O. Box 0429, Buffalo, MO 65622, phone (417) 345-7643.

#201

Versatile Wiring Adapter

A unique wiring adapter which will connect any two RS-232 devices in any pattern is offered by B&B Electronics. The unit may be installed temporarily or permanently. Ten plug-in wire jumpers are included for the postpaid price of $24.95. Contact B&B Electronics, Box 475, Mendota, IL, 61342 or call (815) 539-5827.

#202

Graphics Analysis Pak

Radio Shack announces the Business Graphics Analysis Pak for the Model III two disk system. It makes it possible to create a variety of graphs including pie, bar, line and scatter charts. You can select type as well as size of chart. Entering the data can be from the keyboard or from disk files, including VisiCalc. It includes features for easy data manipulation, editing, scale range adjusting, shading, framing and more. Charts can be produced on selected Radio Shack printers and on the multi-pen plotter.

Contact Radio Shack Computer Centers or participating stores and dealers. Price is $174.95

#203

Pandemonium

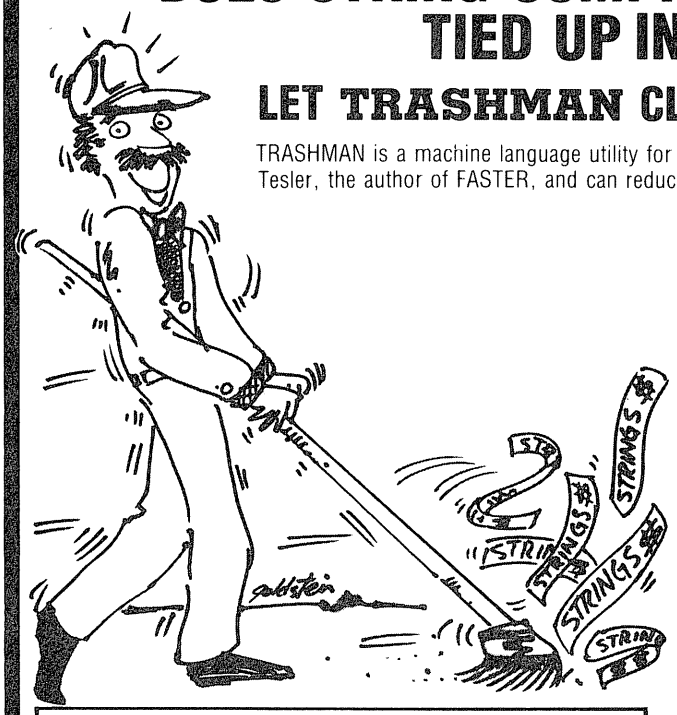
A challenging new word game for the models I and III has been introduced by SOFT IMAGES. This educational game contains a built-in 6000 word dictionary and requires strategy as well as skill. It can be played by any number of players and time limits can be set to equalize different age players.

The object of the game is to place twenty-five randomly generated letters into a five by five matrix and

DOES STRING COMPRESSION HAVE YOU TIED UP IN KNOTS?

LET TRASHMAN CLEAN UP THE MESS!

TRASHMAN is a machine language utility for the TRS-80 Models I and III. It was written by Glenn Tesler, the author of FASTER, and can reduce BASIC's string compression time by 95% (see table below).



WHAT'S STRING COMPRESSION?

When a BASIC program changes a string (words, names, descriptions), it moves it to a new place in memory, and leaves a hole in the old place. Eventually, all available memory gets used up and BASIC has to push the strings together to free up some space. This takes time. Lots of time. The computer stops running for seconds or minutes, and you may even think it's "crashed". The keyboard won't work, and until all the strings have been collected, you just have to sit and wait. Then things run for a while, until string compression is needed again. And again.

If you're using your computer for business, that wastes your money. If you're using it personally, it wastes your time.

WHAT'S THE SOLUTION?

As soon as you start using TRASHMAN, those delays almost disappear. It uses less than 600 bytes of memory, plus 2 bytes for each active string. It works with other machine language programs and with all major operating systems. It's easy to use, comes with complete instructions, and can be copied to your own disks.

WHAT'S THE CATCH?

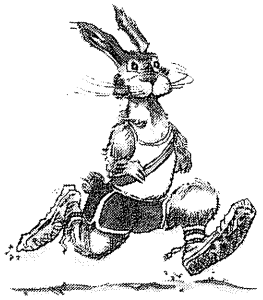
If a BASIC program uses only a few strings, very little time is wasted in string compression, and TRASHMAN won't be helpful. But, if hundreds of strings, including large string arrays, are used, TRASHMAN is just what you need.

# STRINGS	SECONDS DELAY NORMAL	SECONDS DELAY TRASHMAN	PERCENT IMPROVEMENT
250	11.8	0.7	94
500	45.8	1.6	96.5
1000	179.6	3.5	98
2000	713.2	7.8	98.9

TRASHMAN is available on disk for just $39.95.

(All timings done on TRS-80 Model I. Model III 15% faster, but pct. improvements identical. Listing of timing program available on request.)

SAVE TIME WITH FASTER



"FASTER" speeds up most TRS-80 BASIC programs by 20-50%. It's helped hundreds of satisfied people and it can help you. Detailed instructions make it easy to use. FASTER analyses your BASIC programs *while they run*, then displays a simple change, usually one line, that sequences program variables so the ROM will find them faster.

You can use FASTER to speed up programs you've bought, as well as programs of your own. Since it isn't a compiler, your BASIC programs can be read and changed afterwards. FASTER works on business programs, models, and games. The more complex your program, the better the results.

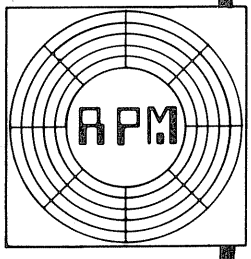
Does FASTER really work? Yes! Just check the reviews in *Personal Computing*, May, 1981, p. 116: **"FASTER is effective and easy to use"**; *80 U.S. Journal*, April, 1982, p. 106: **"I recommend FASTER to everyone"**; and *80 MICRO* (April, 1982, p. 40): **"If you...would like a significant increase in the run-time speed, then buy FASTER."**

FASTER runs on the TRS-80 Models I and III, 16-48K tape or disk, and all major operating systems. **$29.95**

"QUICK COMPRESS" takes only 276 bytes of memory, and removes the blanks and remarks from even the largest BASIC program in less than 3 seconds. It produces smaller, faster programs without altering their logic. **$19.95**

SPECIAL: FASTER and QUICK COMPRESS: **$39.95**

ERRATIC DISK DRIVES?



You can avoid unnecessary disk errors and repair bills by using **RPM.** This easy-to-use program measures the rotational speed and fluctuations of your disk drives, and warns you if they are running **too fast, too slow, or unevenly.**

Incorrect or erratic speed is a common cause of unexplained disk errors and loss of data. RPM's documentation explains how to **detect** *and* **correct** these problems quickly and easily. As *80 MICRO* (April, 1982, page 41) said: **"If your drives have problems I recommend RPM before paying to get it repaired."**

RPM is supplied on diskette for the TRS-80 Models I and III. We suggest you order a copy *before* you need it.

$24.95

ORDER FROM YOUR LOCAL SOFTWARE DEALER, OR 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.

PROSOFT®

Dept. B, Box 560, No. Hollywood, CA 91603

TERMS: We accept VISA, MasterCard, checks, or even cash. Please add $2.00 shipping/handling within U.S.A. or Canada, and $5.00 overseas. C.O.D. charge is $2.00 in U.S. only. We ship within one day of receiving orders.

construct three, four, and five letter words. The computer checks validity of the words and scores each player. It requires 48K, one disk drive and sells for $39.95. Contact SOFT IMAGES, a division of Decision Systems, Inc., 200 Route 17, Mahwah, NJ 07430 or call (201) 529-1440.

#204

Space Saver Printer Stand

The B.T. Space Saver Printer Stand allows continuous form paper to be stored under the printer and gives you easy stacking of completed forms. The clear, plexiglass stand is available in sizes for 80-column and the larger 132-column printers. Optional shelves for storage and slots for feeding through the stand are available. Prices start at $29.95. It is sold at most computer stores or contact B.T. Enterprises, 10B Carlough Rd., Bohemia, NY 11716, phone (516) 567-8155.

#205

Cost Estimation

The EXBIDITE software package enables salesmen to efficiently create itemized estimates based upon inventory items and services requested by the customer. The estimate lists items, services, quantities, prices, and totals as well as complete customer information, terms and type of project. The estimate can be stored on disk, and easily updated.

Prices can be updated and profit margins adjusted according to individual items or for the entire estimate. EXBIDITE includes a program for creating inventory tables and will support an unlimited number of inventory items. It is written in BASIC and is available for the models I or III. One disk drive, printer and 48K are required. Its price is $39.95. For further information, contact Grout & Associates, 26324 Edgewater Blvd. S.W., Poulsbo, WA 98370 or call (206) 779-5149.

#206

Fabric Ribbon Renewal

Dark as original print impressions can be obtained by using the Ribbonizer on used fabric ribbons. Ribbons may be renewed repeatedly until the fabric wears out. Models are designed for popular letter-quality and dot-matrix printers. The black ink is blended to meet supplier specifications and can be ordered separately. The Ribbonizer retails for under $40. For more information contact The Ribbonizer, P.O. Box 1727, Redlands, CA 92373 or phone (714) 792-0831.

#207

Color Computer and Model III Software

Universal Data Research, Inc. announces a new product line for Color Computer owners with the Flex operating system as well as for the TRS-80 Model III. Included are a data base manager (two versions), balanced billing system, a single entry general ledger, and a church contributions package. The advanced data manager and the billing programs are only available for the Color Computer and they retail for $150. The contributions package is also $150 and the general ledger system is $95. All programs allow for updates, easy operator use and provide for a variety of reports. For further information contact Universal Data Research, Inc., 2457 Wehrle Drive, Dept. A, Buffalo, NY 14221 or phone (716) 631-3011.

#208

Type Element Cleaning Kits

Eliminate problems with legibility and document appearance by keeping print characters sharp and smudge free. PerfectData Type Element Cleaning Kits are available in two forms, one for cleaning daisywheel elements and a second form for cleaning ball elements such as in the IBM Selectric. The kit consists of a cleaning unit, pad, and solution. Each kit provides for about twenty-five cleanings and new pads and solution can be purchased as needed. Suggested retail price is $19.95. Contact Innovative Computer Products, 18360 Oxnard St., Tarzana, CA 91356 or phone (213) 996-4911.

#209

Intelligent Buffer

The PrinterMaid is a 64K byte buffer which interfaces from and to the computer in either serial or parallel mode. It accepts parallel data at 3600 baud and serial data at 50 to 19200 baud. For messages less than 64K, PrinterMaid releases your computer and sends data to your printer at the rate specified by the printer. It is hardware and software programmable and can resolve numerous communication and printer problems. It can be programmed for carriage return delays and four serial protocols (CTS/RTS, DSR/DTR, XON/XOFF, ETX/ACK) can be used. All software commands are preceded by a command mode character and the character can be any ASCII character. Suggested retail price is $399, volume discounts available. For more information contact Personal Micro Computers, Inc., 475 Ellis St., Mt. View, CA 94043 or phone (415) 962-0220.

#210

Integrated Accounting Package

IAP version H is a hard-disk accounting package. It includes general ledger, accounts payable, receivables and payroll. The number of transactions per period is virtually unlimited. It features an automatic backup with the date stamped at the end of a period. Priced at $717, IAP comes with four diskettes and over 500 pages of documentation. Each program can be run alone and each is priced at $199. All programs require 64K Model II, a hard disk, and a 132 column printer. User references are available on request. For more information, contact Micro Architect Inc., 96 Dothan St., Arlington, MA 02174 or call (617) 643-4713.

#211

Epson Print Fonts

Two new disks containing ten spectacular type styles each are now ready for your Epson MX80/100 with Graftrax 80 Plus. No hardware modifications are necessary and the new fonts can be used with Dot Writer. Fonts such as Chancery Medium, Computer, Medium Bold 3, Stencil, Fancy, Broadway, Shadow, Balloon and others are available in normal or italic versions. The font disks and Dot Writer are available

from J.F. Consulting, 74355 Buttonwood, Palm Desert, CA 92260 or call (619) 340-5471. They are also available from ACM Computers, 221 Hirschfield Dr., Williamsville, NY 14221 or call (716) 634-3026.

#212

Color Geography Pac

Geography Pac is an easy way to learn world or U.S. geography. It is a collection of five 16K extended BASIC programs, each using sound and color with machine language subroutines. They teach the topological location of countries or states, their capital, largest non-capital city, major industry and currency (or date of statehood). A four-color high resolution map is used and answer study sheets are included.

Keep the student interested by using the flash test which is speed selectable. All five games may be purchased for $29.95 on cassette or $33.45 on disk. Individual programs for the U.S., Asia, Africa, South or Central America, and Europe are $9.95 cassette or $13.95 disk. Schools can obtain multiple copies of the study and answer sheets. Contact Spectral Associates, 141 Harvard Ave., Tacoma, WA 98466 or call (206) 565-8483.

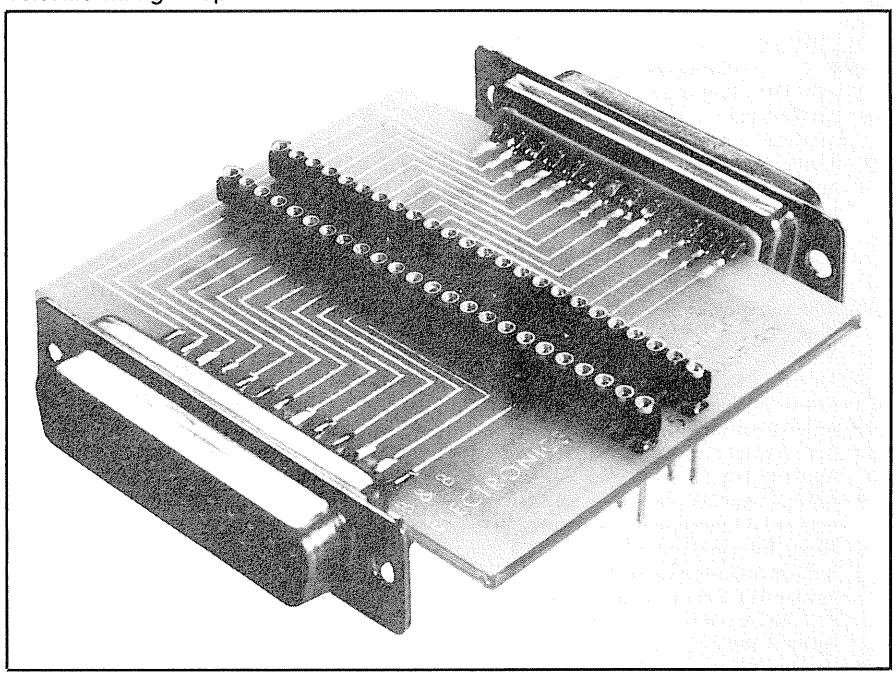
#213

Electric Notebook

Electric Notebook allows you to record facts and update and retrieve them at will. ENB supports data inter-dependencies of any complexity and permits insertion and retrieval on any field. It is a true relational database manager and is applicable to an enormous variety of applications. Use it for customer files, sales, purchasing, medical records, household accounts, a host of possibilities.The program easily handles multiple indexes, variable length fields, record blocking, multi-level and multi-disk files. Available for the models I and III, 48K, one or more disk drives. Retails for $140 plus $3 shipping and handling. For more information contact ALGORIX, P.O. Box 11721, San Francisco, CA 94101 or call (415) 387-3131.

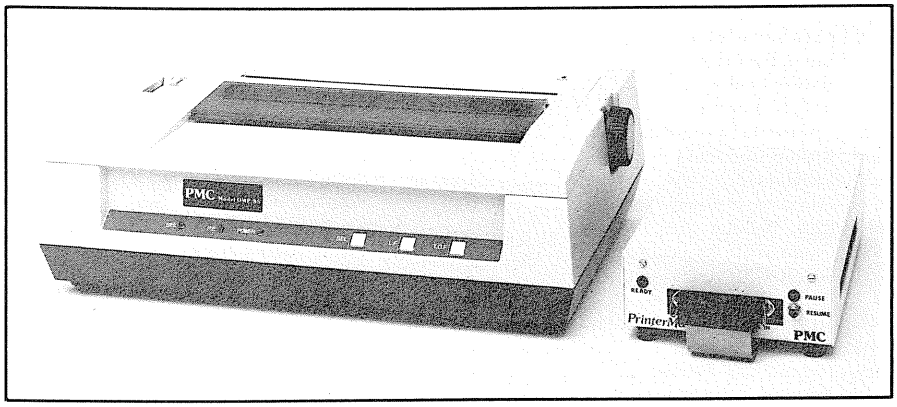
#214

Versatile Wiring Adapter



Type Element Cleaner

Intelligent Buffer



Advertiser index

RSN#	Advertiser	Page
102	ABC Data Products	128
100	AR Systems	128
22	Aardvark-80	49
120	Abacus Associates	130
137	Access Unlimited	140, 141
42	Ace Computer Rentals	88
28	Adventure International	57
135	Adventure International	137
10	Algorix	19
136	Alpha Products	139
62	Alps	111
76	Alps	119
122	American Computer	130
46	American Small Business Computers	93
69	Anitek Software	116
14	Apparat, Inc.	28
15	Apparat, Inc.	29
109	Arctic Computer Systems	129
66	Armadillo International Software	115
117	Aspen Ribbons, Inc.	129
★	Astro Star Enterprises	104
17	Avalon Hill Micro. Games	39
48	B. T. Enterprises	95
127	BAP$ Software	131
108	Barclay Whyte Associates	128
37	Binary Devices	77
107	Bock, Eugene H. & Associates	128
96	Brooks, Joseph W.	127
59	Business Division, The (A Div. of Scott Adams, Inc.)	106
60	Business Division, The (A Div. of Scott Adams, Inc.)	107
111	CDC	129
130	CDC	131
27	CLOAD Magazine, Inc.	55
132	CRB Microtools	131
70	Cer Comp	117
26	Chromasette Magazine	54
91	Clough, Eric	127
83	Comnet Software Assoc., Inc.	124
20	CompuKids Magazine	45
49	Computer House	96
119	Computer Label Co.	130
67	Computer Plus	116
13	Computer Shack	24
23	Computer Shopper	50
81	Computer Trader	122
7	Computing Teacher, The	13
103	Comstar Research	128
94	Comtronic Systems	127
110	Comtronic Systems	129
126	Convert-A-Disk	131
19	Cosmopolitan Electronics Corp.	43
★	D&M Software	83
74	Data Associates	119
98	Data Mania, Inc.	127
92	Dental Computer Newsletter	127
124	Digital Images	130
123	Disk 'n Data	130
93	EAP Company	127
114	East Texas Color Computer Club	129
78	Eigen Systems	120
★	80-NW Books	63
★	80-U.S. Journal	102, 131, 138
72	Electronic Specialists	118
128	Epson America, Inc.	32, 33
95	Excellonix	127
2	Fantastic Software	3
55	Fink, William	102
97	Glenn/Cliff Associates	127
131	Glenn/Cliff Associates	131
★	Gooth Software	104
33	HPB Vector	70
141	Hexagon Systems	9
104	Holman D-P Service	128
113	Holman D-P Service	129
80	Holmes Engineering, Inc.	122
51	Howe Software	99
8	I.J.G.	15
34	I.J.G.	71
50	I.J.G.	98
★	Interpro Corp.	51
31	J&M Systems, Ltd.	61
6	J. F. Consulting	12
88	Jan Tech	125
73	Jessup Company, The	118
61	Jimscot, Inc.	109
68	KCH Consulting	116
64	Kwik Software	114
12	LNW Research	23
77	Langley-St. Clair Systems, Inc.	120
75	Lawyer's Microcomputer, The	119
140	Leading Edge, The	144
65	Lemons Tech Services	114
4	Lindbergh Systems	8
1	Logical Systems, Inc.	2
58	Logical Systems, Inc.	105
32	MISOSYS	69
40	MISOSYS	85
29	Marymac Industries, Inc.	59
115	Mayday Software	129
39	Mercer Systems	83
125	Micro Architect, Inc.	131
53	Micro Management Sys., Inc.	101
118	Micro Moonlighter Newsletter	130
16	Micro Systems Software, Inc.	37
138	Micro Systems Software, Inc.	142
101	Micro Technology	128
121	Micro-Grip, Ltd.	130
5	Micro-Labs, Inc.	11
43	Microsette	90
56	Midwest Comp-U-Tron	102
★	Nepenthe Programs	130
63	New Classic Software	113
41	Nocona Electronics	86
54	Oasis West	102
38	Orion Instruments	79
44	PMC Software	91
45	PMC Software	91
24	Perry Oil & Gas, Inc.	52
47	Pioneer Software	94
71	Powersoft	117
79	Powersoft	121
82	Powersoft	123
89	Powersoft	125
87	Practical Computer	125
11	Prentice Hall	21
90	Programmer's Guild, The	126
3	Prosoft	7
134	Prosoft	133
★	Publishers Exchange	130
112	Quatrk Computer Systems, Inc.	129
36	REMarkable Software	76
35	Radio Shack	72, 73
139	Radio Shack	143
52	Rainbow, The	100
129	Realty Software Company	131
105	SEASOFT	128
99	Scientific Engineering Labs.	127
133	Skeberdis, Pete	131
18	Soft Sector Marketing	40
116	Softlex Games	129
21	Software Options, The	47
★	Specialized Software, Inc.	78
84	Sugar Software	124
86	Tops Programming Enterprise	124
106	Toucan Software	128
30	Trend Software	60
57	Triple D	104
25	Wadsworth Electronic Publishing	53
85	Western Operations	124

★ **Contact these advertisers directly.**

Bulletin board

This bulletin board space is available free to individuals with single or unusual items for sale or trade, and for other announcements of interest to the general readership of this magazine. 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 announcements must be typed, contain 75 words or less and include complete name and address.

Computer Swap America. The first of three shows in 1983 will be held in San Jose, California at the Santa Clara Fairgrounds on Saturday, February 5th. Admission is $5 with hours from 10 am to 6 pm. Sellers, both companies and individuals, should call (415) 494-6862 for a Seller's Information Package. May 21st and September 10th are dates for subsequent shows.

Wanted: Assistance in learning machine language programming. What did you use? I've tried different books to no avail. Please squander a twenty cent stamp and some time to help a struggling neophyte. Bud Meyers, 2 Church Street, Box 498, Washburn, ME 04786 (207) 455-8373.

Used cassette programs for sale. Cosmic Fighter, Robot Attack, Penetrator, and many, many more. Ten dollars each, for TRS-80 models I/III. Guaranteed to load. Call or write Kenny Chan, 3535 55th Ave N.E., Tacoma, WA 98422 (206) 927-7504.

Pocket Computer for sale. Radio Shack PC-1 with printer, cassette interface and two sets of business software. Only $230. Contact Richard Reis, 711 Copley Lane, Silver Spring, MD 20904 (301) 384-0540.

Learn Radio Shack COBOL's built-in ISAM by studying source code for a five-key mailing list and a five-key article index. Both include printing programs. Source code can be typed into a model I, II, III, or 16 COBOL and then compiled. Receive printouts of all source code for $10. If typing bugs you, $25 gets it on a model III data disk. Check or money order to R.J. Bueche, 5704 Spring Valley #1056, Dallas, TX 75240.

Conference proceedings: The ERIC Clearinghouse on Educational Management has published a 236 page volume of proceedings from the July 1982 conference on "The Computer: Extension of the Human Mind" which was held at The University of Oregon. Twenty-three papers are included, by such nationally known experts as Alfred Bork, Ramon Zamora, David Moursund, Karen Billings and others. Copies are available for $10 each from Editor's Office, ERIC Clearinghouse on Educational Management, University of Oregon, Eugene, OR 97403. Full payment or purchase orders must accompany all orders. Make checks payable to ERIC/CEM Publications.

Model III— I will be glad to pay for technical information, data, and schematics for the drives and controller for the Percom units that fit into the model III. Cecil White, P.O. Box 2827, Garland, TX 75041 (214) 475-4556.

For Sale: Eaton LRC 7000+ dot-matrix printer. Prints 20,32,40, or 64 char. per line on 3 7/8 inch paper. Upper/lower case. Excellent hobby printer. Includes eight ribbons, $225 or best offer. Jim Downie, 1522 5th Ave N. E., Aberdeen, SD 57401.

DOUBLE FEATURE
FOR YOUR TRS-80

HERE'S YOUR TICKET TO MANY THRILLING HOURS OF ENJOYMENT

TWO BRAND NEW GRAPHIC ADVENTURES IN ONE PACKAGE — FOR YOUR TRS-80

By Roger Schrag

SPOOK HOUSE

Horrors! A deranged madman has locked your unconscious form inside of a deserted creep house at a crazy carnival. You awake to some sobering facts: a bomb is set to explode, and only 30 real-time minutes stand between you and eternity. Somewhere, somehow you must locate and defuse the explosive — but we'd be less than honest if we thought you had more than a ghost of a chance! Brrrr — a real cliffhanger!

TOXIC DUMPSITE

Something's gone very, very wrong at the Toxic Dumpsite where the treatment and burial of deadly contaminants take place. The entire plant will explode like the Fourth of July in less than 30 minutes — unless you can avoid the many traps and protection systems and shut the plant down in time. Of course, time is the one luxury you haven't got . . . !

This is it — the blockbusting debut of the long-awaited Graphic Adventures! Now, your TRS-80 can hobnob with the best of 'em. Over 50 full screens of sharp, quality graphics breathe a realism and vitality into Adventuring, the likes of which you've yet to see on the TRS-80. Both Graphic Adventures feature game save, extensive vocabulary, real-time action (yes — we said real-time!) and accept FULL sentences! Exclusive area showings brought to you by Adventure International. Now playing at a TRS-80 near you!

TRS-80 48K Models 1 & 3 disk 012-0164 — $39.95

To order, see your local dealer. If he does not have the program, then call 1-800-327-7172 (orders only please) or write for our free catalog.

Adventure INTERNATIONAL

PRICES SUBJECT TO CHANGE

Published by ADVENTURE INTERNATIONAL
a subsidiary of Scott Adams, Inc.
BOX 3435 • LONGWOOD, FL 32750 • (305) 830-8194
135

80-U.S. BACK ISSUES

May/Jun 1979
String packing techniques
Determine functions of the brain
How to win Nim-type games

Jul/Aug 1979
Create fast graphics
Renew lost programs
No-hardware lowercase mod.

Nov/Dec 1979
Function grapher/root finder
Home heat loss program
Restoring killed disk files

May/Jun 1980
Telecommunications with the TRS-80
BASIC game program technique
Produce sound with BASIC programs

Nov/Dec 1980
Simple payroll program
Digital plotter interface
Produce keyboard typeahead

May/Jun 1981
Line packing techniques
How to use "PRINT USING"
Animation and the TRS-80

Jul/Aug 1981
Descending lower case for Model I
Student timetable program
Easy tape loading for Model I

Sep/Oct 1981
Keyword search database program
Compute a retail installment contract
The vertical mill: a 3-D plotter?

Nov/Dec 1981
Comparison shopper program
Real time clock construction
Pocket Computer biorhythms program

Jan 1982
Microcomputers in business

Feb 1982
Micros and word processing

Mar 1982
Microcomputers and medicine

Apr 1982
Microcomputers and investments

May 1982
Space: An infinite frontier

Jun 1982
Games issue

Jul 1982
The TRS-80 in law offices

Aug 1982
Microcomputers in education

Sep 1982
Graphing and graphics

Oct 1982
Microcomputing tips and tricks

Nov 1982
Telecommunicate with your computer

Dec 1982
Disks and DOSs

Please send the following back issues postpaid:

- ☐ **May/Jun 1979**
- ☐ **Jul/Aug 1979**
- ☐ **Nov/Dec 1979**
- ☐ **May/Jun 1980**
- ☐ **Nov/Dec 1980**
- ☐ **May/Jun 1981**
- ☐ **Jul/Aug 1981**
- ☐ **Sep/Oct 1981**
- ☐ **Nov/Dec 1981**
- ☐ **Jan 1982**
- ☐ **Feb 1982**
- ☐ **Mar 1982**
- ☐ **Apr 1982**
- ☐ **May 1982**
- ☐ **Jun 1982**
- ☐ **Jul 1982**
- ☐ **Aug 1982**
- ☐ **Sep 1982**
- ☐ **Oct 1982**
- ☐ **Nov 1982**
- ☐ **Dec 1982**

Name _____

Address _____

City _____ *State* _____ *Zip* _____

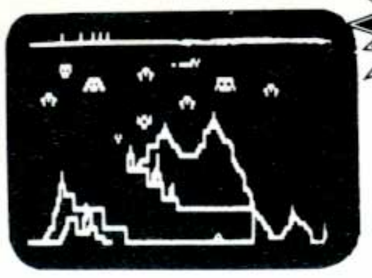
Visa/MasterCard # _____ *Exp. Date* _____

Please remit $4.00 for each back issue ordered.

80-U.S. Back Issues
3838 South Warner St., Tacoma, WA 98409, (206) 475-2219

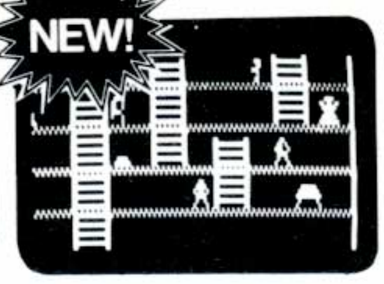
REAR GUARD

Deadly waves of enemy Cyborg craft attack your fleet from the rear. You are the Mothership's sole defender. You have unlimited firepower but the Cyborgs are swift, nimble attackers. Your abilities are tested hard in this game of lightening fast action and lively sound from Adventure International. Price: B



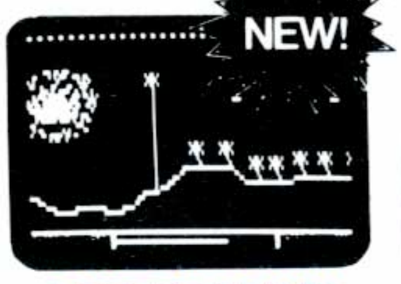
STRIKE FORCE

As the primary defender of a world of cities under deadly alien attack; your weaponry is the latest rapid fire missiles, long range radar and incendiary star shells. Your force field can absorb only a limited number of impacts. A complex game of strategy, skill and reflexes from Melbourne House. Price: A



PANIK

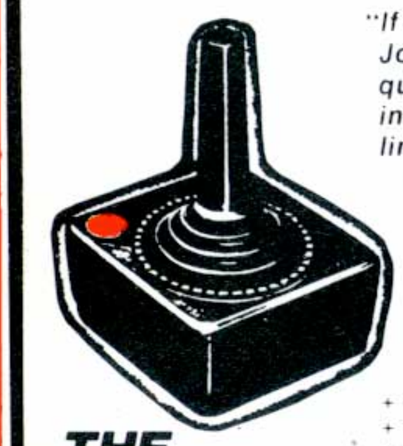
Trapped at an enemy building site, your fate seems certain. Your laser is empty and evil Mzors are closing in. You'll have to climb ladders and think one step ahead of the various monsters. A challenging game for agile minds. From Fantastic Software with voice (Disk has larger vocabulary). Price: B



SEA DRAGON

Your submarine, the U.S.S. Sea Dragon, penetrates a mined enemy channel. Armed with missiles and torpedos, you engage the enemy while navigating unknown waters. Succeed or come to a salty end in this game. 29 screens of horizontally scrolling sea-scrape and sound from Adventure International. Price: B

SAVE 10, 15, 20%



"If you purchase Alpha's Joystick you get the exquisite pleasure of enjoying (action games) to the limit of arcade-style realism."

–80 Microcomputing 80 Reviews, Jan '82

THE ALPHA JOYSTICK

ONLY $39.95

- Features the famous Atari Joystick
- Works with all Model I or III systems
- Compatible with any other accessories
- Saves your keyboard from abuse
- Experiment in BASIC. Use A = INP(0)
- Complete, ready to plug in and use
- Model I: plugs into KB or E/I
- Model III: plugs into 50 pin I/O bus

Price includes Joystick + Alpha Interface + Instructions + Demo Program listing. Please specify Model I or III.

14 DAY MONEY BACK GUARANTEE

© 1982 ALPHA Products

THE BEST FOR LESS

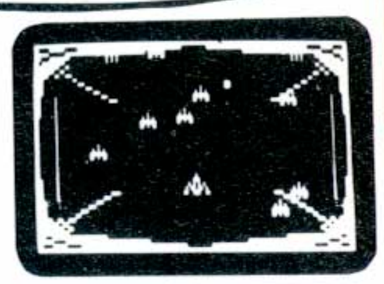
As you can see, all the best games from the top producers are joystick compatible. These games are fun without the joystick but we hope that you are one of the many thousands who enjoy the advantage of real joystick action.

Now you can deduct up to 20% on the price of games: buy any 2 games deduct 10%, buy any 3 games deduct 15%, buy any 4 games deduct 20% from game prices.

TOP TEN

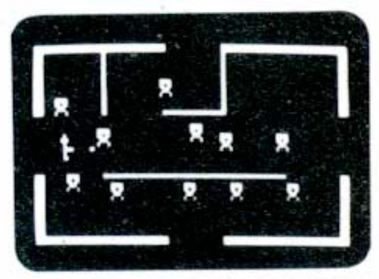
1. SCARFMAN - *All time favorite*
2. PANIK - *Remarkable Voices*
3. PENETRATOR - *Rave reviews*
4. ARMORED PATROL - *Super 3D graphics*
5. CATERPILLAR - *Good rendition*
6. CRAZY PAINTER - *Unique game concept*
7. DEFENSE COMMAND - *Tough struggle*
8. STELLAR ESCORT - *Fast and Challenging*
9. ROBOT ATTACK - *With voice*
10. SEA DRAGON - *Amazing "Seascape"*

STELLAR ESCORT



The latest super action game from Big Five. As the Federation's top space fighter you've been chosen to escort what is possibly the most important shipment in Federation history. The enemy will send many squadrons of their best fighters to intercept. With sound. Disk version has voices. Price: A

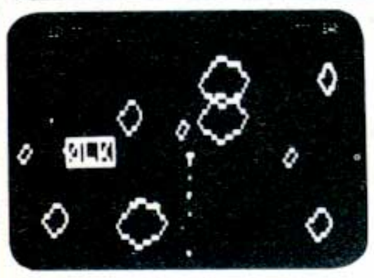
ROBOT ATTACK



Talks without a voice synthesizer through the cassette port. With just a hand laser in a remote space station you encounter armed robots. Some march towards you, more wait around corners. Careful, the walls are electrified. Zap as many robots as you dare before escaping to a new section. More robots await you. Price: A

LUNAR LANDER

As a vast panoramic moonscape scrolls by, select one of many landing sights. The more perilous the spot, the more points scored -- if you land safely. You control LEM main engines and side thrusters. One of the best uses of TRS-80 graphics we have ever seen. From Adventure International. With sound. Price: A

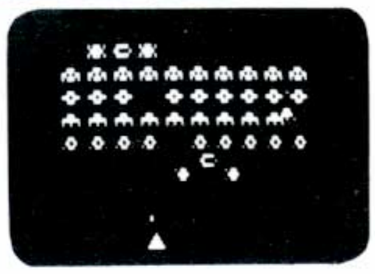


SUPER NOVA

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. Price: A

OUTHOUSE

You are the mighty protector of this small (but important) wooden structure. For reasons unknown, a bizarre gang of miscreants wish to vandalize, loot and otherwise destroy the little half moon house. Your patrol craft has lasers and smart bombs to deal with this terror. From SSM with sound. Price: A

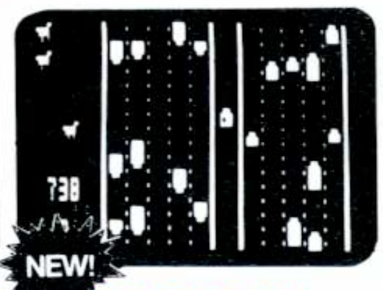


GALAXY INVASION

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! Price: A

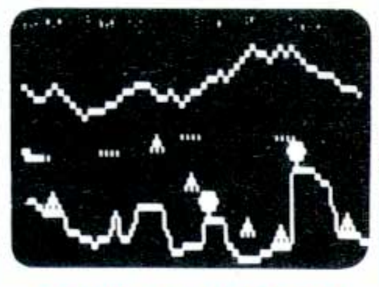
LASER DEFENSE

In this game of ICBM's, high-energy lasers and particle beams, you control the U.S. strategic defense satellite system. From your viewpoint high above the globe, you intercept Soviet nuclear missiles in flight and attempt to destroy their scattered missile silos. With sound from MED Systems. Price: A



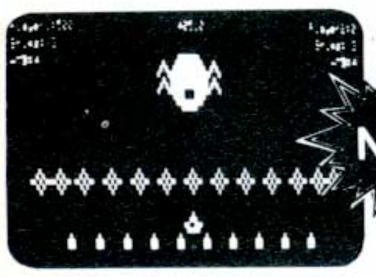
CHICKEN

Will the chicken cross the road? That's up to you. Can you guide these helpless little chicks across the perilous 10 lane super highway to safety? Or will you bumble, littering the blacktop with a storm of chicken feathers? A humourous yet challenging game of nerves from SSM with sound. Price: A



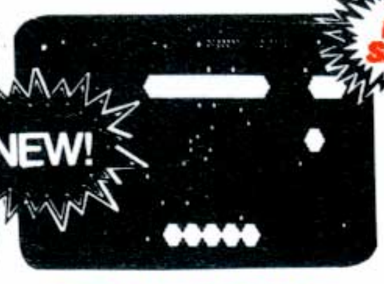
PENETRATOR

Soar swiftly over jagged landscape, swooping high and low to avoid obstacles and enemy missiles 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. Price: C



DEFENSE COMMAND

The invaders are back! Alone, you defend the all important nuclear fuel canisters from the repeated attacks of thieving aliens, repeatedly. An alien passes your guard, snatches a canister and flys straight off. Quick! You have one last chance to blast him from the sky! With sound and voice. Price: A

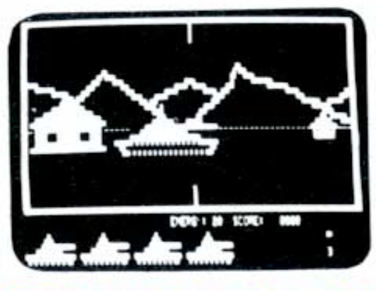


BOUNCEOIDS

Huge boulders careen off the walls. You're in the middle, in danger of being flattened. Keep your wits about you as you blast these bounceoids from the screen. Large ones break into many small ones. Clear a screen and enter a fast-paced challenge stage with a chance for big bonus points. From the Cornsoft Group. Price: A

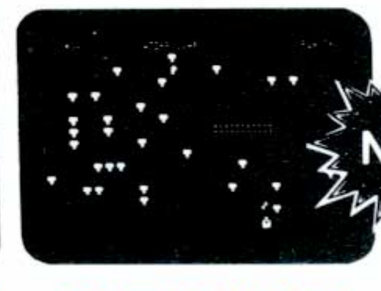
SCARFMAN

This incredibly popular game craze now runs on your TRS-80! It's eat or be eaten. You run 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. Price: A



ARMORED PATROL

A realistic tank battle simulation. Your view is a 3-D perspective of an alien landscape. Maneuver 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. Price: B



CATERPILLAR

An arcade favorite! Stop these multi-sectioned crawlers before they creep down through the mushrooms. Zap one and it splits into two smaller bugs, each with its own sense of direction. There are moths and tumble bugs too. It all adds up to lots of fun for kids and adults alike. From Soft Sector Marketing. With sound. Price code: A

CRAZY PAINTER

You have to paint the floor white. We give you the paint and brush. Sounds easy? Hah! You'll be confounded by stray dogs, snakes, sloshing buckets of turpentine, even a ravenous paint eater! A crazy, imaginative new game with ten selectable levels of skill for new or seasoned game players. Lot's of laughs. Price: A

LATEST RELEASES:

FROGGER
LICENSED BY SEGA
TAPE: $19.95 DISK: $22.95

WEERD
FROM BIG FIVE
TAPE OR DISK: $19.95

WITH ANY ORDER FROM THIS PAGE
STICK-ON LCD CLOCK-CALENDAR

FREE!

BLACK COLOR. BATTERY INCLUDED
LIMITED: ONE PER CUSTOMER

GAME PRICES

A: TAPE: $15.95 DISK: $19.95
B: TAPE: $19.95 DISK: $24.95
C: TAPE: $24.95 DISK: $24.95

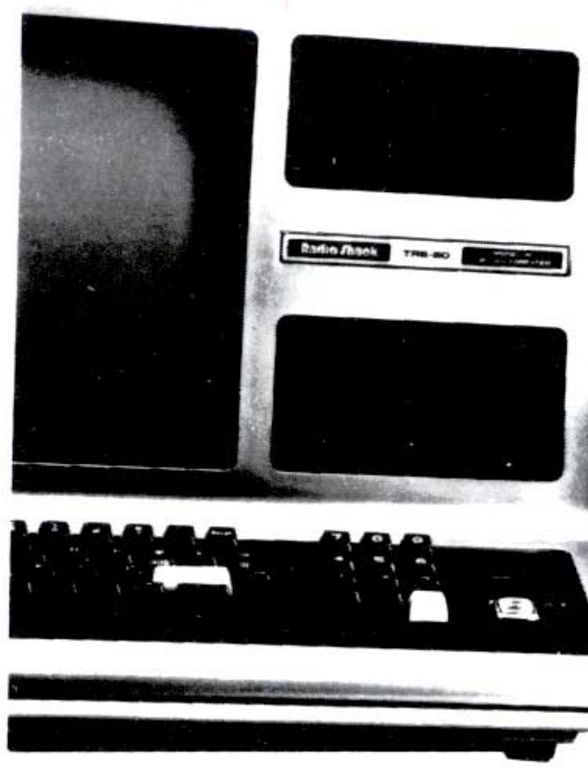
TAPE: Model I&III, 16K Level 2
DISK: Model I&III, 32K, 1 Disk
All games are joystick compatible or may be played using arrow keys.

ALPHA Products
79-04C Jamaica Ave., Woodhaven, NY 11421

Toll Free Order Line
136 **800-221-0916**
Orders Only. NY & Info call (212) 296-5916. Hours: 9-5 E.S.T

Add $2.00 per order for shipping/handling.
We accept Visa, Mastercard, Checks, M.O.
C.O.D. - Add $3.00 extra.
N.Y. Residents add sales tax.
Overseas, FPO, APO: Add 10%
Dealer discounts available.

ACCESS UNLIMITED



FREE

FREE! DOSPLUS 3.4 disk operating system with any Percom or Access Unlimited 1st Drive System for the Model III***

We have added approximately $400,000.00 worth of brand new business and game software with more arrivals daily.
Call or write for our new FREE catalog.

SAVE $50.00 on a fully-tested PERCOM Disk Drive for Model III* Reg. $499.00. Now $449.00 ea or $750.00 Dual

Percom's internally-mounted drives, with their widely-acclaimed disk controllers, are completely pre-tested with a 48 hour Burn-in. So, you know it works right when you get it. Choose single or double-density storage capacity.

A First-Drive System includes the four-drive disk controller, one drive, power supply, mounting hardware, cables and a fully-illustrated, easy-to-understand owner's manual.

NEW PRINTERS "CENTRONICS" — "C'ITOH" — "OKIDATA" — "STAR" — "TALLY" — "EPSON" — "GEMINI". VALUES TO $3500.00. Call for latest price.

Make sure it's done right. Let us install your First-Drive System.

Purchase your Model III* Drives from us and for $39.95 plus shipping, you can have our experienced professionals perform a 48-hour burn-in of your Model III* computer, install the drive system, and check out your expanded system to make sure everything works correctly. Call for pricing of complete Model III* system with 2 to 4 Access Unlimited or Percom drives.

Get the best! Our own fully-tested drives* for as low as $275.00 Model I*

Every Access Unlimited Floppy Disk Drive is electrically and mechanically tested, then burned-in for 48 hours under operating conditions. The signed test list in your shipping carton is proof of that.

Our floppy disk drives offer you either single or double density. With double-density, it stores up to 364 Kbytes, depending on the format.

SATISFACTION GUARANTEED! If you're not completely satisfied with your Access Unlimited Disk Drive, return it within 15 days for a full refund. All our drives have a comprehensive 90-day limited warranty.

Save $$ on our most popular items!

Percom Data Separator (reg. $29.95)	now **$23.95**
Screens for Models I*, II*,III Green, Lt. Blue, Dark blue, and Amber. Bronze for color video (reg. $24.95)	now **$12.95**
Head Cleaning Kit (reg. $29.95)	now **$19.95**
Drive Numbering Tabs, pkg. 0-3 (reg. $4.50)	now **$3.95**
Flip-N-File 5¼" (reg. $39.95)	now **$19.95**
Flip-N-File 8" (reg. $54.95)	now **$28.95**

MEDIA FOR LESS

BASF 5¼" Single Sided Double Density
Lifetime Limited Warranty. reg. $44.95 now **$24.90** bx of 10

ATHANA complete with hub rings & one year limited warranty.

Single sided Single density 5¼"	**19.90** bx of 10
Single sided Double density 5¼"	**22.80** bx of 10
Double sided Single density 5¼"	**29.90** bx of 10
Double sided Double density 5¼"	**33.50** bx of 10

VERBATIM Single Sided Double Density **30.90** bx of 10

8" Diskettes from $22.75

LIBRARY CASES — Holds 10 Diskettes Sale **$2.95 ea.**

Bare Drives for "IBM PC" — Internal or External — $269.95 ea.

Beautiful Oak Finish Computer Furniture
50" Split Level Desk $249.00
Matching Printer Stand $89.00

DOS + 3.4 Reg. $159.00. Sale $99.95

Anti-Static Mats — Colors: Russett, Blue & Gold, Natural Brown, & Golden Brown.

3' x 5' — Reg. $75.60	Sale: **$57.00**
4' x 6' — Reg. $120.90	Sale: **$91.00**
4' x 8' — Reg. $161.30	Sale: **$122.00**

"PAGEMATE" Typing Easels—$14.95

SAVE on an Arrick Quick Switch‡

Changes a TRS-80* printer port or a peripheral between computers instantly and easily. Available for RS-232 and Centronics. Plugs included.
Now from $99.95. Cables from $31.95.

• LIMITED TIME OFFER/LIMITED QUANTITIES •
Prices subject to change without notice.

" " Reg. Trademarks Prices do not include state taxes.

137

MICRO SHOPPING CENTER

Percom Quality for your Model III

Nothing but the best for your Model III*. Percom internally mounted drive systems: including 4 drive controller with gold edge connectors, double density disk drive or drives, all hardware and cabling. A free copy of DOS Plus 3.4 is also included with every first drive purchase.

TFD 340N1	one drive single sided double-density	**$449.00**
TFD 340N2	two drive single sided double-density	**$750.00**

Brand Spankin' New! SALE Dual Headed Drives for the Price of Flippies!

Now you can have a *dual headed PERCOM Drive System for your Model III!!

TFD344N1	One drive dual headed double-density	**$560.00**
TFD344N2	Two disk dual headed double-density	**$860.00**

***completely compatible with programs existing on single sided or double sided diskettes.**

WE HAVEN'T FORGOTTEN THE MODEL I*!!

Upgrade your Model I to Double Density **All for 159.95!!**

You have a good system and you just don't want to sell out to a Model III-Upgrade with the Percom Doubler II, the overwhelming favorite double density adapter for over **2 years**! Simply plug the adaptor into your expansion interface and run either single or double-density programs. Comes complete with FREE DOS Plus 3.4 but will also run with LDOS NEW DOS 80 and TRS DOS.

Drives for your storage needs

AFD40-1	40 track single sided drive	**$275.00**
AFD42-1	40 track flippy drive	**$329.00**
SALE TFD100-1	**40 track flippy**	**$278.90**

Percom Hard Disk—Quality for your System

*works with existing floppy drives — Reg. from $2495
*can be daisy-chained up to four hard disks
***DOS Plus 4.0 or LDOS included FREE** — **Now from $1995**
*Now available for immediate delivery in 5 & 10 megabyte configuration
Can be used for Model II*, III*, "Apple II" or "IBM PC" computers.

Check out the benefits of Access Unlimited Hard Disk Drives For TRS-80*, "IBM-PC", "APPLE II"

- ✓ 5, 10 Megabyte and larger drive units.
- ✓ lets controller handle up to 4 drives.
- ✓ works along with existing floppy disk drives.
- ✓ includes host system support software.

From $1,995 (5 MB)

Your present system too small?

Take advantage of the iBEX company business system and stand alone word processing-systems—overbuild mistake-**model No. 7202**

Regular retail $9070.00
Now for a limited time only—while supply lasts—$3995.00

Look what you get for $3995

- Z80 • CP/M compatible
- 64 kilobyte RAM.
- Dual 8" floppies (total 2.4 Mbytes)
- Switchable to IBM format
- 12" green phosphor monitor (80 x 24 characters)
- Centronics compatible printer interface
- Serial interface
- Full function keyboard
- Clock timer and calendar (with battery)
- Free software: CP/M Operating System, M/Basic Interpreter, Perfect Writer, Perfect Speller, Perfect Calc, Perfect Filer
- ADS Bus. Software

Enjoy 1-stop shopping and BIG savings

Order your disks, software, supplies, and accessories from Access Unlimited. We offer low prices on automatic Datamarc*** Sheet and Envelope Feeders for Radio Shack*, QUME, Ricoh, Diablo Daisywheel, and Dot Matrix Printers. And the more you buy at one time, the more you can save.

For more information, call (214) 340-5366 Monday through Friday; (214) 690-0207 after 5 p.m. and Saturdays.

To order or for FREE literature, call TOLL-FREE

VISA **1-800-527-3475** master charge

Ordering Information

Order by phone or by mail. We accept Visa, MasterCard, cashier's checks, certified checks, and money orders. With personal checks, allow additional time for bank clearance. Your bankcard will not be charged until your order is shipped. On orders over $1,000, we pay freight (surface only) and insurance; please add $3.00 shipping and handling under 50 lbs. Over 50 lbs., add $5.00 for orders under $1,000.00. Texas residents add 5% sales tax. Allow 2 to 4 weeks for delivery.

EXPLANATION OF SYMBOLS
* Trademark of Tandy Radio Shack Corp
†† Trademark of Apparat Corporation
TM Trademark of Access Unlimited
† Trademark of Percom Data Company, Inc.
** Trademark of Micro-Systems Software, Inc.
‡ Trademark of Arnick Products Company
*** Trademark of Datamarc, Inc
x Trademark of Digital Research

☐ Please send me a FREE catalog. I'm not ready to order at this time.
☐ YES, I'm taking advantage of your Sales prices.

Name _____
Company Name _____
Address _____
City _____ State _____ Zip _____
Phone Number ( ) _____

Quantity	Item	Unit Price	Subtotal
		Subtotal	
		State Sales Tax (Texas residents only)	
		handling charge	
		Total	

Check one:
☐ payment enclosed ☐ Visa ☐ MasterCard*

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

*If MasterCard, numbers above name: ☐☐☐☐

Expiration Date: ☐☐ — ☐☐

Authorized signature, if charged

ACCESS UNLIMITED

DEPT. C2/401 N. Central Expwy. #600/Richardson, Texas 75080
Tel. 1-800/527-3475 214/340-5366
214/690-0207—Sat. and Evenings Only

M I C R O T E R M

More and more hardware and communications services are allowing speeds up to 1200 baud. Soon, some may be going faster than that. Today's terminal software simply can't keep up. But now there is an alternative. Micro-Systems Software introduces MicroTerm, the high speed terminal.

Model III MicroTerm will communicate, without insertion of null characters, at 4800 baud. Guaranteed. No cop-outs, no question. MicroTerm is so fast that you can exit from the terminal to the main menu, adjust video width, open the buffer, turn on the printer, or any one of dozens of other functions, and return to the terminal mode **without missing a thing!**

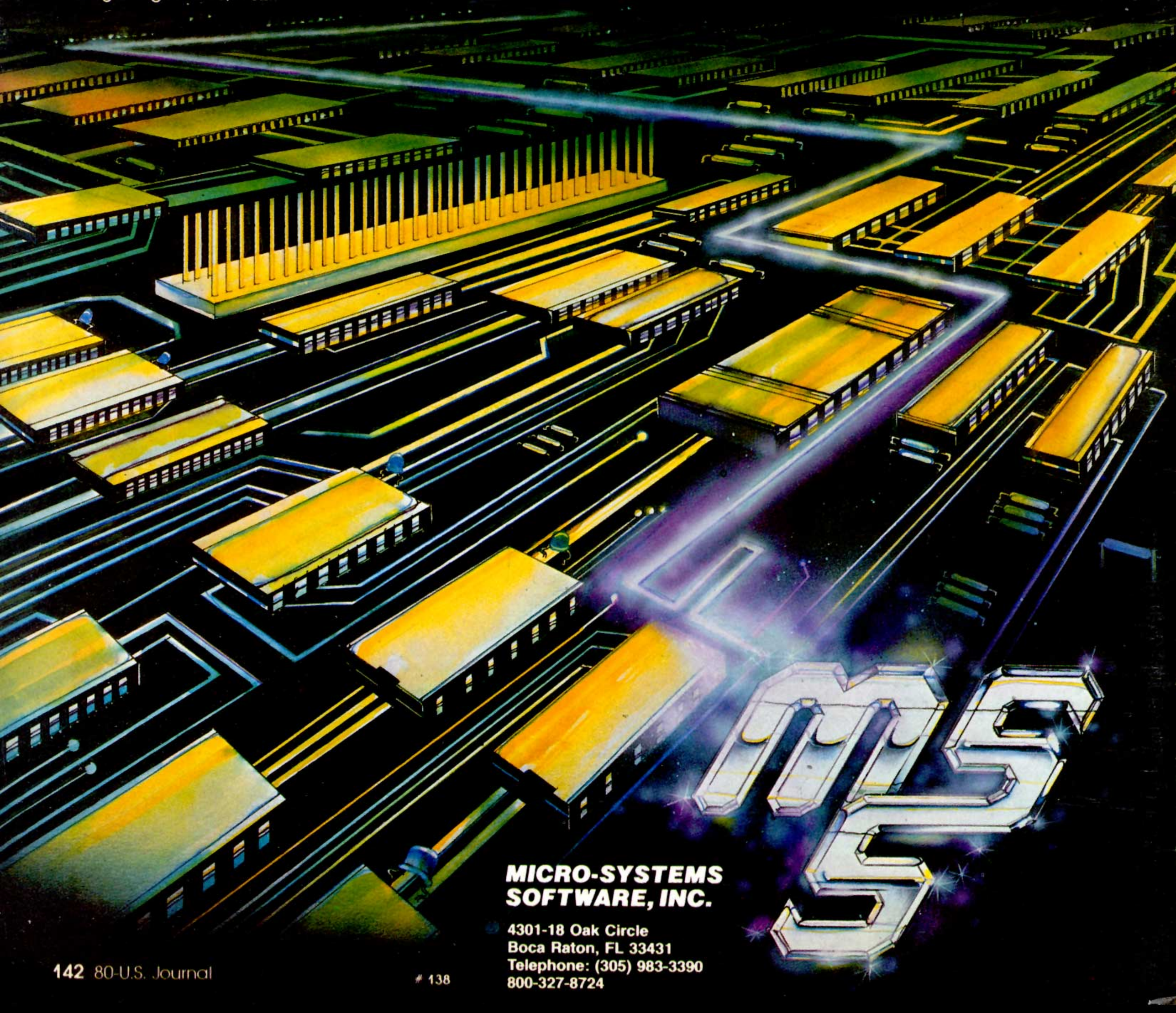
MicroTerm continues to input from the RS232, even while at the main menu. This is the only terminal capable of such an astounding feat. MicroTerm offers you most of the features that "Brand X" smart terminals have, plus it gives you: • Ultra high baud rate operation (up to 9600 in certain cases). • Input while at menu. • Easy to use translation tables. • Easy to use phone number listings. • Maximum auto dial support — most major brands. • Direct file transfer companion program included at no extra cost (compatible with DFT). • DOS commands from menu without exiting program. • Over 34K of capture buffer (in a 48K TRS-80). • Can be set to automatically dial telephone and transmit buffer at preset time without **any** operator intervention.

And many, many more great features, MicroTerm is so fast you must see it to believe it. The various menus are displayed so fast, they seem to jump out at you. Status of various functions can be displayed and altered in split seconds.

For the computerist who wants the ultimate, state-of-the-art terminal software, there is no other choice.

MicroTerm retails for $79.95, but registered DOSPLUS owners can purchase it for only $59.95. $20.00 off the retail price! MicroTerm comes complete with the terminal program, the direct file transfer program, some standard translation tables, and documentation.

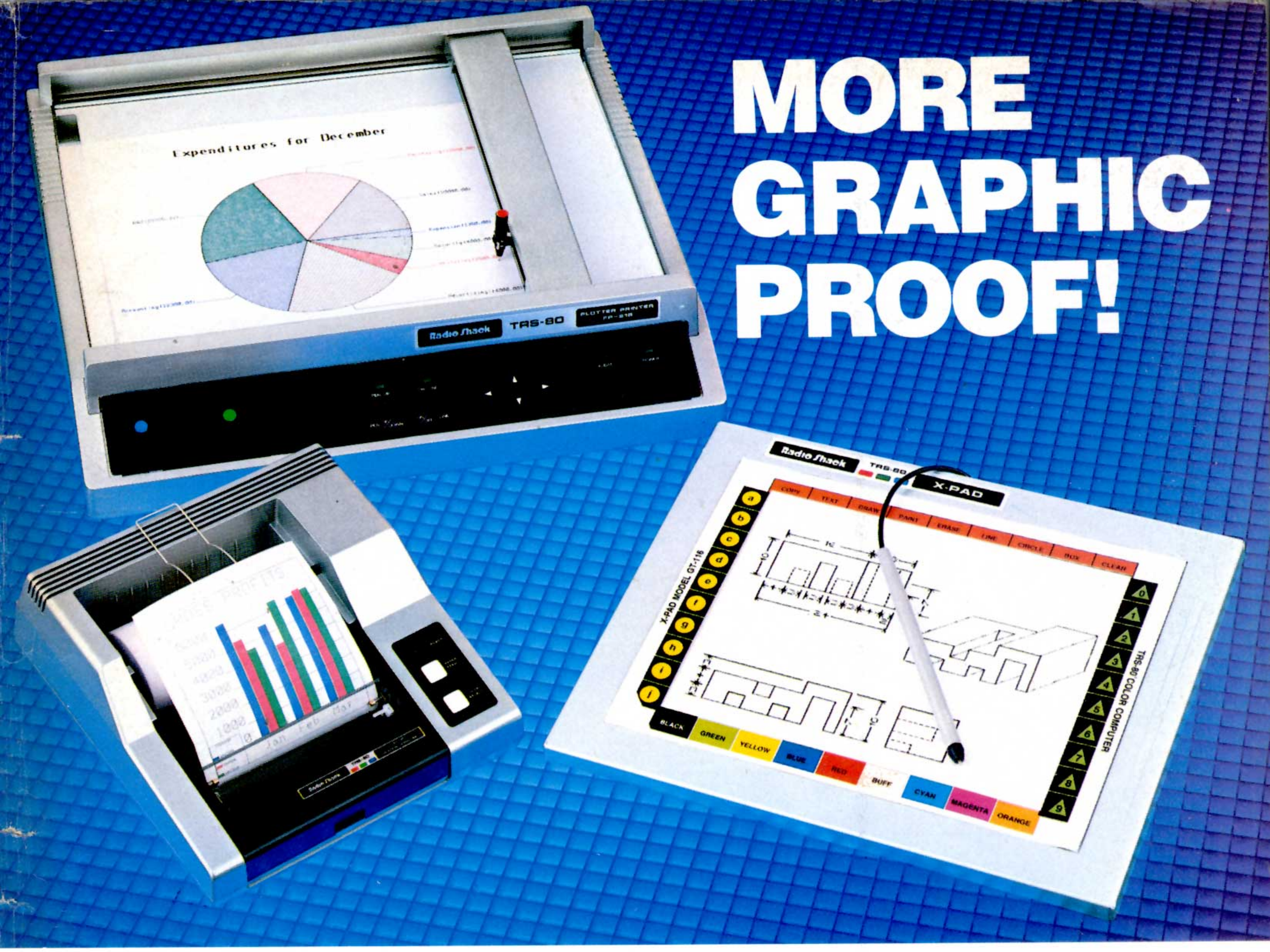
Don't delay, order yours today! Specify when ordering: Model I or III and whether you want it on 40 or 80 track media. Requires a 16K TRS-80 with one disk drive. We recommend 48K for serious communications work. MicroTerm will be available beginning June 30, 1982.



MICRO-SYSTEMS SOFTWARE, INC.

4301-18 Oak Circle
Boca Raton, FL 33431
Telephone: (305) 983-3390
800-327-8724

138



MORE GRAPHIC PROOF!

Radio Shack's TRS-80® is #1

These new peripherals let you explore an exciting dimension of your TRS-80—*Graphics!* They're three more reasons Radio Shack's TRS-80 is the best supported microcomputer in the world!

New! FP-215 Flatbed Plotter/Printer. This intelligent, 4-color, single-pen graphics device draws charts and graphs using BASIC ASCII command codes and parameters, and prints text horizontally or vertically using the BASIC LPRINT command. Parallel and Color Computer-compatible serial interfaces. Cat. No. 26-1193, $995.00.

New! CGP-115 Color Graphics Printer. Ultra-compact—yet produces beautiful graphics in red, blue, green and black on standard 4½" roll paper and prints alphanumerics at 40 or 80 characters per line. Built-in commands simplify drawing and plotting. Parallel and Color Computer-compatible serial interfaces. 26-1192, $249.95.

New! GT-116 Graphics Tablet. Instant graphics at the touch of a pen! What you draw or trace is read into your TRS-80 Color Computer. Compatible with Extended BASIC graphics commands. Menus and other options simplify data input. 26-1196, $349.95.

See Them All at Radio Shack—Your Graphics Headquarters. Discover how to turn your TRS-80 computer into a complete graphics system. Stop by your nearby Radio Shack Computer Center, store or participating dealer to get a "hands on" demonstration of these peripherals. Be sure to ask about our six-pen color plotter and our 11 × 17" digitizer, too.

Radio Shack®

The biggest name in little computers™

A DIVISION OF TANDY CORPORATION

Send me your free TRS-80 Computer Catalog today!

Mail To: Radio Shack, Dept. 83-A-176
300 One Tandy Center, Fort Worth, Texas 76102

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

PHONE _____

Retail prices may vary at individual stores and dealers. Special order may be required at some stores.

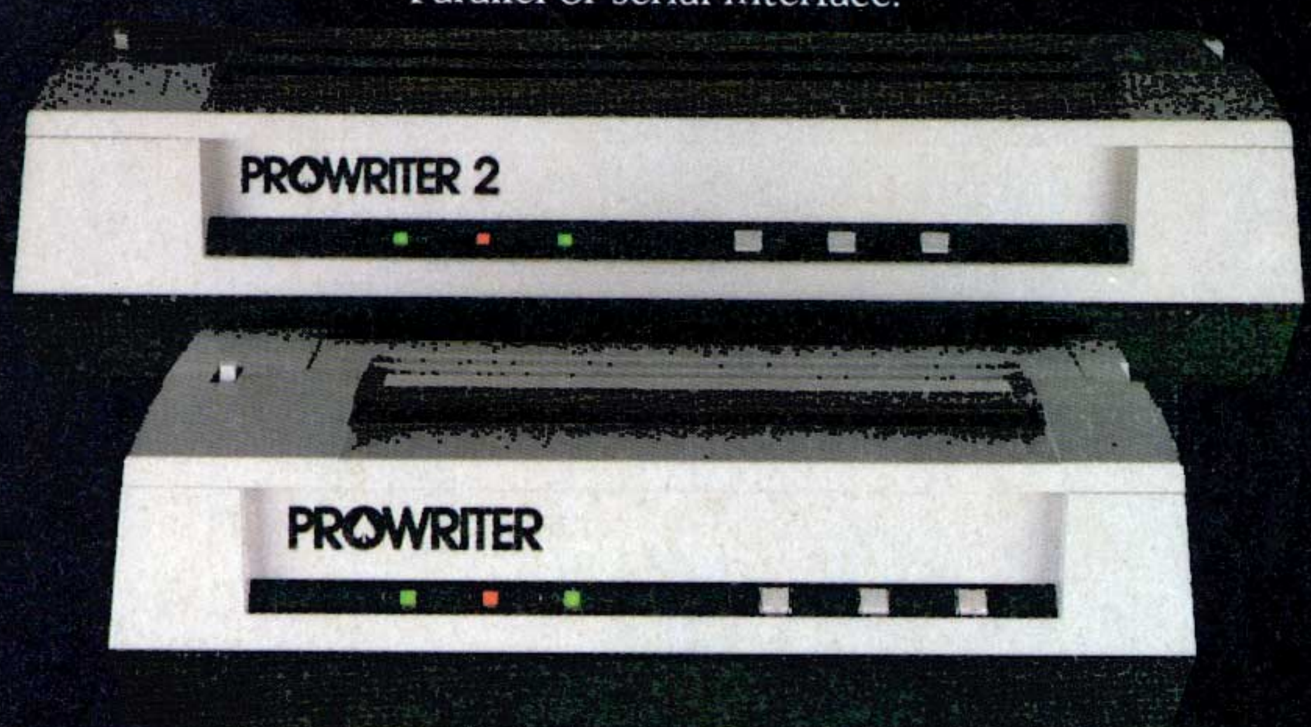
THE LEADING EDGE IN PRINTERS

ONE GREAT LINE. ONE GREAT WARRANTY.

Finally, there's one full family of printers that covers every business or word processing application—all from C. Itoh, a company known for packing more product into less price; and all distributed exclusively by Leading Edge, a company known for searching out and providing that very thing. Which means that one call to one source can get you any printer, any time you need it, for any purpose. All backed by a full years' warranty from Leading Edge. (Try *that* on any other line of printers.)

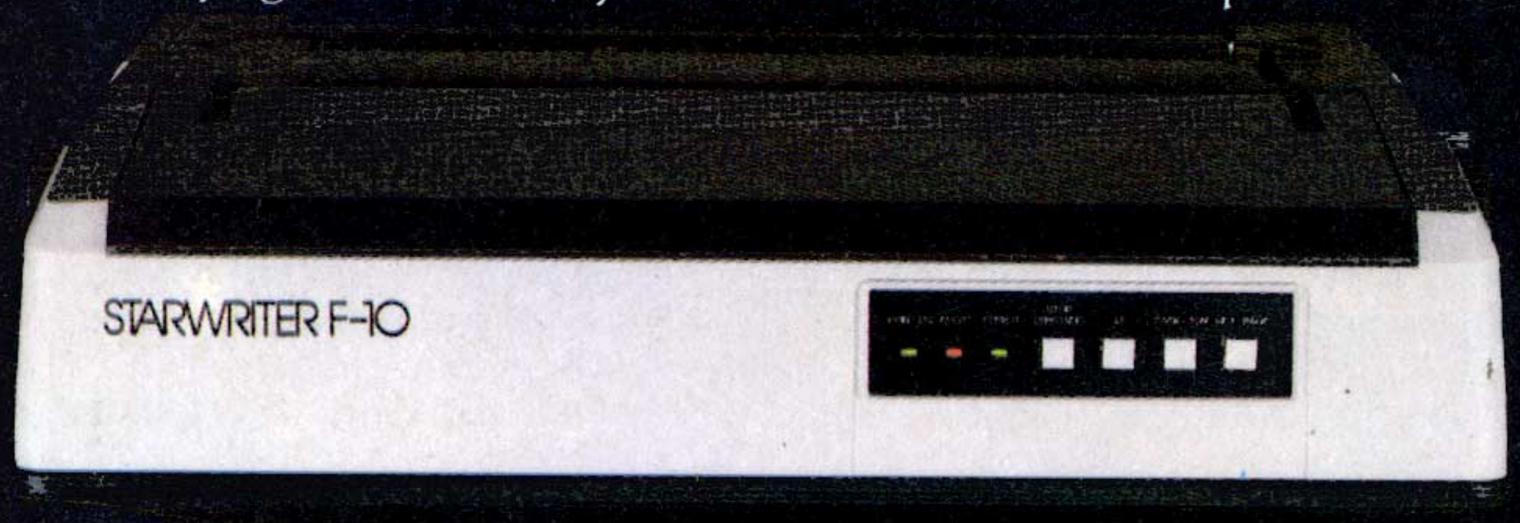
THE PRO'S.

The Prowriters: business printers—and more. The "more" is a dot-matrix process with more dots. It gives you denser, correspondence quality copy (as opposed to business quality copy, which looks like a bad job of spray-painting).
Prowriter: 120 cps. 80 columns dot matrix compressable to 136. 10" carriage. Parallel or serial interface.
Prowriter 2: Same as Prowriter, except 15" carriage allows full 136 columns in normal print mode. Parallel or serial interface.



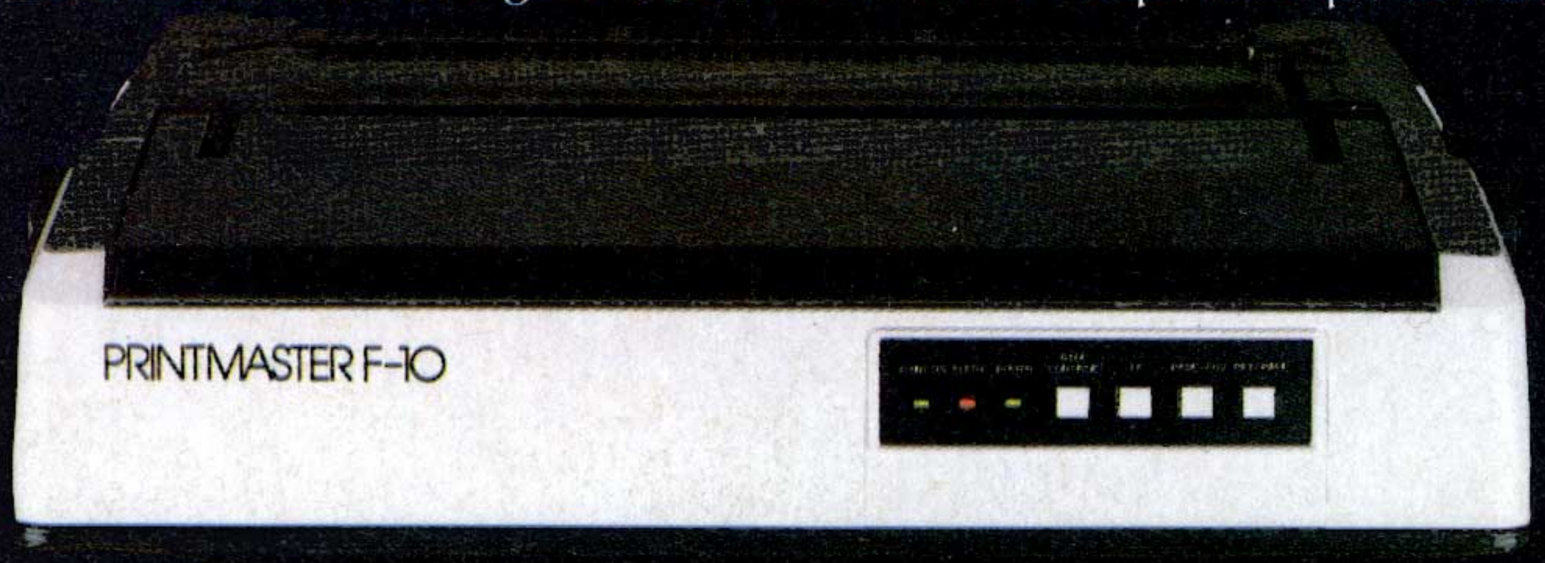
THE STAR.

The Starwriter F-10. In short (or more precisely, in a sleek 6" high, 30-pound unit), it gives you more of just about everything—except bulk and noise—than any other printer in its price range. It's a 40 cps letter-quality daisy-wheel with a bunch of built-in functions to simplify and speed up word processing. It plugs into almost any micro on the market, serial or parallel.



THE MASTER.

The Printmaster F-10. Does all the same good stuff as the Starwriter except, at 55 cps, the Master does it faster.



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.