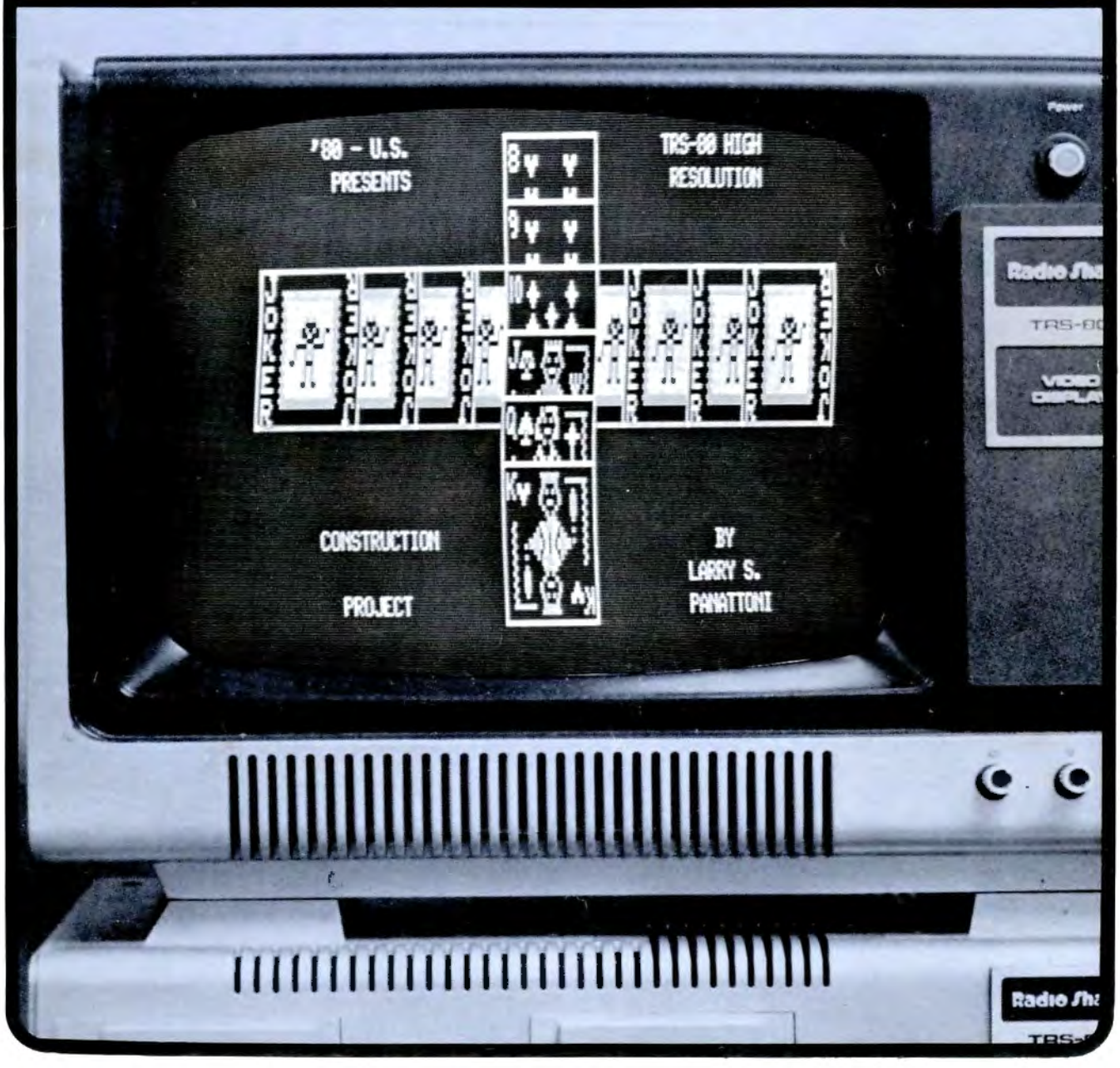
80-U.S.

$3.00 PER COPY
$3.50 in Canada

The TRS-80 Users Journal

Volume III, Number 2 Mar/Apr 1980



Low Cost Add-On Storage for Your TRS-80*. In the Size You Want.

When you're ready for add-on disk storage, we're ready for you. Ready with six mini-disk storage systems — 102K bytes to 591K bytes of additional *on-line* storage for your TRS-80*.

- **Choose either 40-track TFD-100™ drives or 77-track TFD-200™ drives.**
- **One-, two- and three-drive systems immediately available.**
- **Systems include Percom PATCH PAK #1™, on disk, at no extra charge. PATCH PAK #1™ de-glitches and upgrades TRSDOS* for 40- and 77-track operation.**
- **TFD-100™ drives accommodate "flippy disks." Store 205K bytes per mini-disk.**
- **Low prices. A single-drive TFD-100™ costs just $399. Price includes PATCH PAK #1™ disk.**
- **Enclosures are finished in system-compatible "Tandy-silver" enamel.**

Whether you need a single, 40-track TFD-100™ add-on or a three-drive add-on with 77-track TFD-200™s, you get more data storage for less money from Percom.

Our TFD-100™ drive, for example, lets you store 102.4K bytes of data on one side of a disk — compared to 80K bytes on a TRS-80* mini-disk drive — and 102.4K bytes on the other side, too. Something you can't do with a TRS-80* drive. That's almost 205K bytes per mini-disk.

And the TFD-200™ drives provide 197K bytes of on-line storage per drive — 197K, 394K and 591K bytes for one-, two and three-drive systems.

PATCH PAK #1™, our upgrade program for your TRSDOS*, not only extends TRSDOS* to accommodate 40- and 77-track drives, it enhances TRSDOS* in other ways as well. PATCH PAK #1™ is supplied with each drive system at no additional charge.

The reason you get more for less from Percom is simple. Peripherals are not a sideline at Percom. Selling disk systems and other peripherals is our main business — the reason you get more engineering, more reliability and more back up support for less money.

In the Product Development Queue . . . a *printer interface* for using your TRS-80* with any serial printer, and . . . the *Electric Crayon*™ to map your computer memory onto your color TV screen — for games, animated shows, business displays, graphs, etc. Coming PDQ!

™ TFD-100, TFD-200, PATCH PAK and Electric Crayon are trademarks of PERCOM DATA COMPANY
*TRS-80 and TRSDOS are trademarks of Tandy Corporation and Radio Shack which have no relationship to PERCOM DATA COMPANY

PERCOM

PERCOM DATA COMPANY, INC.
211 N. KIRBY • GARLAND, TX. • 75042

To order add-on mini-disk storage for your TRS-80*, or request additional literature, call Percom's toll-free number: 1-800-527-1592. For detailed Technical information call (214) 272-3421.

Orders may be paid by check or money order, or charged to Visa or Master Charge credit accounts. Texas residents must add 5% sales tax.

Percom *'peripherals for personal computing'*

THERE IS A DIFFERENCE IN TRS-80 DISK DRIVES CAPACITY

Expansion interface – gives your TRS-80 the disk capacity it needs, and much, much more!

10 to 40 MByte, 8" Winchester drive – expands capacity far beyond Model II storage.

Single sided minifloppy – up to 150 KBytes of storage capacity.

Single or double sided 8" floppies – up to 2.5 MBytes in dual drive cabinet – for the serious TRS-80 user.

LOBO DRIVES' new family of disk memory products provides you with a choice of memory capacities you need to effectively execute the complex business software you've developed for your TRS-80*. LOBO DRIVES' selection of readily available, software compatible drives permits you to expand your inventory, payroll, customer list, and accounts receivable files as your business grows.

And LOBO DRIVES brings you more . . . a new plug-in expansion interface that provides an easy way to add hardware enhancements, communications capability, and programmable features . . . and it comes with the LOBO DRIVES famous 1 year, 100% parts/labor warranty.

Call or write for the complete LOBO DRIVES story. Find out just how competitively priced a family of high capacity drives can be . . .

LOBO drives INTERNATIONAL

935 Camino Del Sur Goleta, California 93017.
(805) 685-4546

"CAN YOU REALLY AFFORD TO PAY LESS?"

Quantity discounts available – Dealer inquiries invited

TR-2

Yes, I want to know more about LOBO Drives and what they can do for my TRS-80. Send me information on:

- ☐ 5 1/4-in. Floppy drive
- ☐ 8-in. Winchester hard disk, 10 Mbyte drive
- ☐ 8-in. Floppy drive Single sided Double sided
- ☐ Double density expansion interface

Name _____

Company _____

Address _____

City _____ State _____ Zip _____

Phone No. _____

If dealer, provide resale no. _____

*TRS-80 is a registered trademark of Radio Shack, a Tandy Company

Editorial Remarks * *

Back in the early seventies, when microcomputing got started, the "fraternity" consisted mainly of dedicated hackers who put their systems together from scratch. They had to - there was not very much "off the shelf" to buy then.

It grew, and companies came and went. Eventually, computer stores came into existance. Many of them were add-on's to existing electronics parts stores or distributors. Byte Shops and Computerlands sprang up and independents opened for business. The trend was away from hackers and games to small business computing. Getting serious with your computer seemed to be the trend (although games, probably because we are still all kids at heart, are still amoung the big sellers).

You still had to put your system together from various manufacturers for the most part. A perfect place for some company to jump in and create a "complete" system, ready to go.

It seemed like Texas Instruments would have been in a good position to do that. So would the Heath Company. But the Tandy Corporation beat them to the wire, and the TRS-80 explosion resulted. Tandy admits in writing that "over 100,000 have been sold". We think it is probably more like double that number. The Heath Company and Texas Instruments have both since come in with their entries. Both look good - but it seems like too little and too late. How can you ask anyone to change systems when they already have shelled out one to four grand for what they already have? Also, what are the software writers writing for? The largest consumer base, naturally.

Since the proliferation of "Shack" computers, many of the old fraternity stores have taken a "down-the-nose" attitude, both toward the computer and the people who own them. They apparently sell "real" computers, not to be confused with the "toy" variety.

Since Radio Shack continues to refuse to sell anything in their stores not made by or for them, other stores are missing an excellent opportunity to do what they are there for - to make money.

Not all stores are like that; some of the more enlightened have already seen the possibilities and are cashing in on it - to everyone's benefit.

But, it is still possible to enter one and ask for what is new in TRS-80 software, only to be rudely pointed to a rack containing a couple of copies of Microchess, while the salesperson dissappears to supposedly "more important" business.

Depressed majorities usually don't stay that way very long. I for one can't wait for this one to pass.

On another note: Did you know how easy it is to get into business? It must be easy, since the government says that about four out of five new businesses fail in their first year. Obviously, the challenge is not getting into business, but staying there once you have started.

It is a challenge, if you have never tried it before, finding out that almost nothing is as you expected it to be. And trying to hang on to all the neat ideals you went in with is something else again. Luck also plays a part.

Last year, I spoke to a gentleman from Omni Magazine. He said that before they published anything, they put over one million into advertising, much of it on television. Their first issue (and all since), have been superb. It takes some kind of planning and finances to get launched like that.

We "lesser lights" have to start out with an idea and a nickel, and try to turn them into a dime. It dosen't always work.

In May 1979, **Ed Thorne** in San Francisco, started a publication for TRS-80 users called **T-PAL** (The Programming Amateurs Newsletter). After two or three issues (which were very good), he found he could not continue.

After some negotiation in November and December 1979, 80-U.S. has agreed to fulfill the obligations of T-Pal with the Journal. Letters have been sent to T-Pal subscribers, giving them the option of continuing with 80-U.S. or recieving a refund. Those already taking both were given the option of an extention to their 80-U.S. subscription.

Our hat is off to Mr Ed Thorne, who could just as easily disappeared, leaving his readers hanging, but instead took the proper way out by offering refund or an alternate publication.

Which brings us back to the beginning. Getting into business is easy, staying there is not so easy, and getting out gracefully is difficult.

Mike

80-U.S.
JOURNAL

EDITOR/PUBLISHER
I Mike Schmidt

EDITOR/ASSEMBLER
Cathy Shappee

ASSOCIATE EDITOR
Terry Dettmann

TECHNICAL EDITOR
Jim Crocker

RESOURCE IN LEARNING CONSULTANT
G B Livingston Ph.D

SOFTWARE DEVELOPMENT
Leo Christopherson
Jim Crocker
Roy Groth
Kristi Schmidt

REVIEWERS
George Blank
Cameron Brown

PRINTED BY
Peninsula Gateway Press
Gig Harbor, Washington

DISTRIBUTED IN CANADA BY
Micromatic Systems Inc
No 101 - 8136 Park Road
Richmond, BC Canada V6Y 1T1

DISTRIBUTED IN EUROPE BY
Hofacker Verlag
Tegernseer Strasse 18
D-8150 Holzkirchen/Obb
West Germany

DISTRIBUTED IN THE UNITED KINGDOM BY
The Software House
146 Oxford Street
London, W.1 England

80-U.S. JOURNAL
VOL III Number 2
MAR-APR 1980

Published bi-monthly in Jan, Mar May, Jul, Sep & Nov. **SUBSCRIPTION PRICE** in the United States is **$16./1yr, $31./2yrs, $45./3yrs. CANADIAN SUBSCRIBERS** contact MICROMATIC SYSTEMS INC No 101 8136 Park Road, Richmond, BC Canada V6Y 1T1. ALL OTHERS: $24./1yr, $47./2yrs, $68./3yrs. remitted in US funds. All except US, Canada & Central America are sent Airmail.

Printed in the United States of America
Application to mail at Second Class Postage Rates pending at Tacoma, WA
POSTMASTER: If undeliverable return to sender. Return postage guaranteed.

80-U.S.

The JOURNAL for TRS-80* Users

TRS-80 is a trademark of the Tandy Corporation

VOL III Number 2 March-April 1980

CONTENTS

Letters to the Editor

Two observations - the first is that somewhere I read that G-2, the maker of Level III Basic (which I own and is great) has gone out of business. Too bad! The other thing is that you should do a review on the Exatron Stringy Floppy, which I own. It is fantastic for those who as of yet cannot afford an expansion interface and disk. It records a 16K program at a rate of 7200 baud in about 20 seconds and then verifies it. I have saved over 400K worth of programs without an error! With Level III Basic I now have all functions (nearly) of a disk and 10K of memory. Its great!

John M Delaney
Wood River, IL

G2 Level III Basic was created by Bill Gates, of Microsoft and was reviewed in our Nov-Dec 79 issue, page 58. We certainly agree with your satisfaction with the Exatron Stringy Floppy, see 80-U.S. Jan-Feb 80, page 60.

I wish you would print a word of praise for the folks at Exatron, makers of the Stringy Floppy. Mine arrived two weeks ago, and I became an immediate convert! I have called the factory twice on their WATS line, once to check on an order, and once for some technical information. Both times my questions were answered immediately. An excellent product from a company with an excellent attitude!

David Anderson
Northampton, MA

See previous letter. Again, we agree, and isn't it nice to know that some people really care?

Your Jan-Feb 79 issue had a program on page 22 titled "A Basic Text Editor". On page 24 line 1550 seems to be incomplete, 1550 IFE$="L"THEN
I had just started using DOS and got your program out when I noticed this. Could you please let me know if there is an addition to this statement. Surely do enjoy your publication and programs.

John J Foster
Laguna Beach, CA

Ignore line 1550, it was left in from a previous re-write, and then as now, proofreading is still a problem.

One of the first things I noticed about my TRS-80 was the lack of X-Y positioning in the graphics mode except for SET or RESET. To print anything else, other than just turn a spot on or off, one needs to convert to a "PRINT AT" location from 0 to 1023. This is a useful little addition to any graphic program which converts X and Y coordinates into the appropriate "PRINT AT" location:

```
D=Y*64+X:PRINT@D, ....
```

It would be best to provide for X being between 0 and 63 and Y between 0 and 13 to keep the item printed from going to the next line or the screen from scrolling.

Chuck Doherty
Providence, RI

I have an idea that might be of use to others and thought I would pass it along. - Why not a program for a super index type? For example, as your issues pile up it will become harder to find items you remember but not the details. Your program on telephone numbers could be used *(see Vol I, No 1, page 8)* Also, I see a use to have an index type of list to combine books. Let's say several authors discuss the trait "Anger" in various ways. I think it would be useful if the computer would tell me which books discuss anger and then I could go to the books I want. - Perhaps this is old stuff, but the concept is new to me and I think that all of us are looking for ways to make the computer more useful than just playing games.

James C McCord
Fairbanks, AK

Yes - see the article "A Search Program", in this issue.

In case you or your readers are interested in using an IBM custom print element, it is a modified 12 pitch Letter Gothic element. It includes *greater than,less than,up arrow* and a cancelled zero. Greater than and less than are mapped to upper and lower case ¼ and ½ respectively, up arrow replaces the cent symbol (upper case 6), and zero is of course zero. The IBM ordering information is as follows:

Blanket part number is 1175704
Standard part number is 1167138
CHANGES
Position 41 upper case =X99013 (Greater than)
Position 41 lower case =X99016 (Less than)
Position 19 upper case =X99004 (Up arrow)
Position 35 lower case =X14107 (Cancelled zero)

At the time of this writing (Dec 79), this element costs less than $70. with the above changes (and is metalized like the standard element for durability). However, it takes from six to eight weeks to manufacture. Other custom configurations are available, but not all symbols are available at all positions and the more changes, the higher the cost.

Clayton E Schneider Jr
Bethlehem, PA

I subscribe to about 8 TRS-80 monthly magazines. The only reason I have not considered yours is because it is bi-monthly. When you go monthly (even at double the price) I will subscribe. I am also waiting to see 80-Microcomputing. This may eliminate my subscription to all others. (And at $1.00 per issue).

Kenneth Foshey
Menomonee Falls, WI

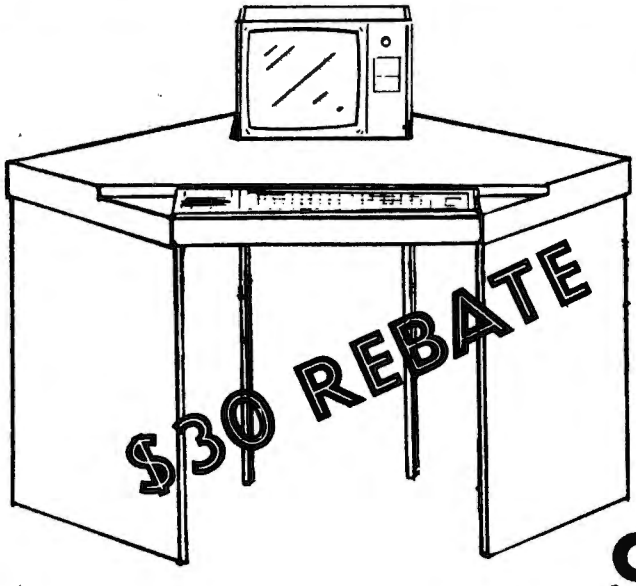
We have found that you generally get what you pay for.

I have come across a neat little thing that I find makes the -80 a little bit more enjoyable. Nothing gets to me more than when running a program, having cute little error statements like "?SN ERROR" pop up. I have devised a short subroutine which will add a lot to the error handling ways of the computer. What it does is simply this: The normal error display is changed into some sarcastic or funny remark by simply incorporating the "ON ERROR GOTO" command. When an error is encountered, the computer randomly selects a statement and displays it instead of the normal error statement. Sometimes it is necessary to know the type of error. Line 8 below takes care of that, by printing the numerical error code in parentheses after the commmand. It is fun to change the comments to correspond to particular programs. Of course, additional changes would have to be made if the resident program already includes an ON ERROR

(Continued on Page 6)

Product Release...

AVS, manufactures of the *Home for the TRS-80*, that high quality all wood, hand rubbed, walnut finish furniture line, announces the release of their *Economy Version* of their CF-80 Custom Furniture Series. Made from office furniture standard of simulated walnut formica, the CF-90 Corner Console completly builds in the TRS-80 basic system of keyboard. cassette, monitor, and expansion interface. *Absolutely* no visible cables. Plus, provides cushion arm rest and copy shelf.



FEATURES:

- One Homogeneous Unit
- No Visible Cables
- Space Saver
- Copy Shelf
- Arm Rest Cushion
- Economical
- Enhances Any Decor

ONLY $129.95

REBATE:::REBATE:::REBATE:::REBATE:::REBATE:::REBATE

ORDER NOW

24 HOUR

PH 408-946-1265

VISA MasterCard

AVS AUDIO-VIDEO SYSTEMS

$30 REBATE on all orders received prior to April 15, 1980, as a speial introductive offer. That's right, for only $99.95 you will receive a custom corner desk console that provides total build-in capability for the TRS-80 keyboard, interface, monitor, and cassette.

NOTE: All orders prepaid, F.O.B. San Jose, no C.O.D.'s

Dealer inquiries invited.

2485 AUTUMNVALE AVE. SAN JOSE, CA. 95132

GOTO statement, and care should be taken not to duplicate variables. Here is the subroutine:

```
1 RANDOM
2 ON ERROR GOTO 4
4 ZZ = RND(3): ON ZZ GOTO 5,6,7
5 A$ = "YOU IDIOT, YOU GOOFED AGAIN!":
  GOTO8
6 A$ = "WHOOPS, YOU BLEW IT SOMEWHERE!":
  GOTO8
7 A$ = "OH, OH, YOU MADE A BOO BOO!"
8 PRINTA$;"(";ERR/2+1;")":RESUME NEXT
9 CLS
```

Naturally, the above should be put at or near the beginning of the resident program. It can even be used by itself when using the computer manually. The program will always be remembered after it is run, even after the break key is pressed.

Chris Knoll
Gig Harbor, WA

First the kudoo...each copy of your magazine brings very welcome, high quality information. Please keep going in this fashion and enjoy a long and hopefully prosperous life.

Second, a couple of comments on Phil Pilgrim's excellent utility program "Append" in the Sep-Oct 79 issue. The program appears to need a "NEW" after input and execution with the "/". After that, all is well. Without NEW, the first attempt to call the routine causes an error message. Another comment: APPEND works just great with G2 Level III. After bringing up Level III in the normal way and executing it, go to SYSTEM and enter APPEND. After executing APPEND with the "/", use Level III's ability to set Shift Keys to special purpose to allow for one key entry and exit from APPEND. I chose "o" for on and "f" for off and they work just fine. Since I have 48K, I adjusted the ORG address accordingly. Thanks for a great magazine.

Rik Karlsson
Oakton, VA

It is stated in the Sep-Oct 79 issue of your excellent Journal (page 28) that the minus sign acts as the logical operator "exclusive OR". This is not quite true, although the example given works perfectly well.

To explain why, let us first introduce the concept of a logical expression; this is an expression that may take on only two values, "true" and "false". On the TRS-80, these values are conventionally represented by the numeric values -1 and 0 (see Level II manual, page 1/5). For example, try PRINT 1=1; believe it or not, it prints -1! Similarly, PRINT 1=0 (or any other false relation) gives 0. In Level II Basic, the logical operators AND, OR and NOT are used to write compound logical expressions like C=(A=1) AND (B=2), which is -1 if both expressions A=1 and B=2 are true, and 0 otherwise. This operator may be simulated, using only arithmetic operators, by writing C=-(A=1)*(B=2). The exclusive OR operator, which is not provided on the TRS-80, may be replaced by C=-ABS((A=1)-(B=2)); the use of the minus sign only, which is suggested in the above mentioned paper, that is writing C=(A=1)-(B=2) results in the value 0 if both expressions are either true or false, -1 if the first one is true and the other false, but +1 in the fourth possible case. This does not matter when the logical expression to be "computed" is used inside an IF statement, because the interpreter only tests whether the arithmetical value is 0 or not. So, the statement IF (A=1)-(B=2) THEN 100 branches to line 100 if the expression is not 0 (which would mean "false"), but it does not care whether it is -1, +1, or any other non-zero value.

The following short program is suggested to test the use of the available operators, in conjunction to the one listed in the Sep-Oct 79 issue:

```
10 INPUT "ENTER A & B";A,B
20 PRINT (A=1) op. (B=2)
30 GOTO 10
```

where op. stands for any operator +, -, *, AND, OR.

Roger Cruon
La Valette, France

In answer to the letter from Rick Coulthurst in the Jan-Feb 80 issue, regarding the lockup by KKeyboard BBounce and the LIST command becoming LLIST, it is very easy to get around the problem. First, a little background information.

In the Level II Reference Manual, on page D/1, you will see that the line printer control block is found in memory at the eight consecutive addresses beginning with 16421 (decimal). Addresses 16422 and 16423 tell BASIC where to find the routine that handles output to the printer; 16422 being the least significant byte of that 16 bit address, and 16423 being the most significant byte. Normally, the control block if pointing to address 058DH which unfortunately is in ROM. Unfortunately, because the ROM routine does not check for the printer being ready *before* sending data to it. This routine is a CALLed subroutine, exited by a RETurn instruction.

Now, there are two ways to get around the lockup problem when no printer is connected. The first, and most complicated, would involve writing a machine language routine to check the status of the printer before trying to talk to it. The address of this routine would have to be POKEd or otherwise put into the pointer 16422/16423; the routine would have to be loaded from tape or disk every time we enter BASIC and it would have to be protected from getting clobbered by using the MEMORY SIZE? to reserve a chunk of memory. The routine would either RETurn to BASIC if the printer were not ready or would junp to 058DH to continue with normal output. This ritual could be a real pain after awhile.

The second way is much simpler. Every time you enter Level II or Disk Basic, before doing anything else, enter this: POKE 16422,67:POKE16423,0. This tells BASIC that the address of the printer driver is now at 0043H; 0043H contains a RETurn instruction. Thus, every time we give a command such as LLIST (or LPRINT), it will appear that nothing is happening because we have totally bypassed the printer routine.

I hope this information will be of help to your readers. This lockup problem is one of the most annoying quirks in the -80, but it can be dealt with. Keep up the good work--and don't change from the bi-monthly format if it will be accompanied by a decline in quality!

Dave Stambaugh
Fountain Valley, CA

Although there have been several suggestions made to this problem, yours is by far the simplest. Thanks.

(Continued on Page 8)

PACKAGE 1

Have you ever backed-up a diskette only to end up killing off a dozen unwanted files. It can get quite tedious typing in all those file names. Well now there's a program for you! DELETE is one of those invaluable programs you'll find yourself depending on more and more. It can be used to kill off unwanted files after a back-up, or you can use it to eliminate development programs once the final copy is finished. DELETE comes with a number of switches, the functions of which are too numerous to mention here. Needless to say, DELETE is quite complete.

NAME has only one function and that is the rewriting of the DOS and date block in your diskette directory. The reasons for wanting to use this program are many, but we will only mention three. First, NAME can protect your diskette from theft by placing your name or initials in the DOS section of the directory. Second, by placing your disk number in the date section you can make finding a program a lot easier. Lastly you can use name to number your diskettes for D.D.M.S., DISKETTE DIRECTORY MANAGEMENT SYSTEM, but that's another program. These two programs will receive constant use, and at only $15.95 you'd be foolish to pass them by.

PACKAGE 2

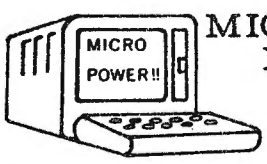
Package 2 consists of three programs. That's right, three! The first is BATCH, a DOS command batch processor. What is a DOS command batch processor? Well, lets suppose you had a certain routine at the beginning of each day which consisted of backing up your previous days data, setting the system clock, finding out the free space on each diskette, turning the clock on, loading BASIC, and executing your program. BATCH could do all that upon power-up in a blink of an eye, well maybe two blinks.

Second is MULF, a user defined key program. In short, MULF lets you specify a function or set of functions for each key, and that includes the <shift> set (ie. <shift> E). However, the best feature is that each key is USER defined, and since the definitions are on a file you can change them anytime you wish. MULF will also allow you to respond to the questions of machine language and BASIC programs, prior to their being asked. There's just no limit to the things that MULF can do! Finally, there's XFER, the data file generator for BATCH and MULF. The price on package 2 is just $15.95

PACKAGE 3

This package contains two of the best tape utility programs available today! DUP and SPEED are a must, and I do mean must! Have you ever had the desire to duplicate a machine language tape, data tape, or BASIC program tape and not been able to do so. Well now you can. It's so simple even a systems analyst could do it. DUP uses two tape recorders and one tape I/O cord. Simply place the play plug in one recorder and the record plug in another, then execute DUP. DUP reads, processes, and outputs your tapes. It's just that simple!

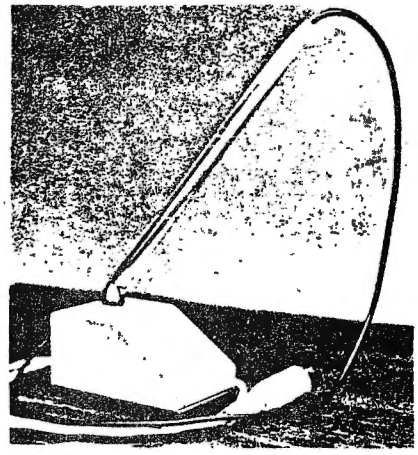
The second program, SPEED, is for all of you who are tired of inputting and outputting your programs and data at the slow rate of 500 baud. Once SPEED is loaded, and you've saved and loaded your programs and data at a 1000 baud you'll never go back to the old rate. What's more, SPEED requires no modification of your computer, thereby preserving your warranty. If you are now using tape and you pass this package by you will be regretting it for as long as you own your computer. The package price on these two programs is only $11.95. Please note speed will not work if your computer has had Radio Shack's tape loading modification installed.



MICRO DATA SYSTEMS

916-334-3817

7275 MEATH WAY NORTH HIGHLANDS, CA. 95660



Micro Data Systems introduces Micro Pen II, the first in a long line of hardware for the TRS-80. With all the light pens on the market you might well ask yourself what makes this pen different. As you can see it looks like a fine piece of office writing equipment not a felt pen on a wire. Because it comes with its own stand and holder it is less likely to get tangled up in a clutter of loose wires. Finally, the Micro Pen II's circuitry has been designed to turn off if the Pen is unplugged, the computer is turn off, or an OUT255,0 is input. Thus prolonging battery life. Micro Pen II comes with two programs. The first is a fully documented light pen scanner routine. This program tells you, step by step, how to write light pen programs. The second is a light pen version of HANGMAN. We have not listed the number of different uses Micro Pen II has, because there are just too many! The $29.95 price includes the pen, programs (on tape), documentation, and battery. The price also includes a limited warranty for the first six months of ownership.

Send to: Micro Data Systems, Dept. B
7275 Meath Way
No. Highlands, Ca. 95660

o Yes, please send info on future items (inclose description of your system)
o Yes, please send me the these items

Title	Price
Calif. Res. add 6% sales tax	
Postage	$ 1.00
Total	$

Name:_____
Address:_____
City, St.
Zip:_____

o Check made to Micro Data Systems
o Master Charge
o VISA

Card No._____
Exp. date_____

_____
Signature

PACKAGE 5

This package is the tape-oriented version of MULF and XFER giving the tape user the same features the disk versions have and one more. MULF will automatically set memory size to protect itself from BASIC. This package is priced right at $14.95.

GRAPHICS MASTER

Have you ever seen Leo Christopherson's bee wary or snake eggs, and wished you had the time or skill to do that type of graphics work. Well with Graphics Master you can! Graphics Master is a disk oriented graphics generator which will allow you to create graphics and alphanumeric screen pages and use them as part of a basic program. This gives you very fast printing of full screen graphic displays, as well as rapid editing without the use of printed graphics charts. The screen pages are stored on disk as an ASCII program file with line numbers, variable name, and subscript ranges assigned by you. These programs may be merged together, and the results are great. While in any of the modes you have full cursor control to go in any direction. The price of GRAPHICS MASTER is only $19.95. For what this program does you'd really expect to pay more.

D.D.M.S.

If you have more than one diskette you need D.D.M.S.! This is a machine language program that catalogs diskette directories. After you have numbered your diskettes and created the librarian data file, this program becomes the greatest tool in finding your files. It can display the directories on file, continuing when you direct, or it can display the directories on file, automatically stopping when you direct. D.D.M.S. can also find files by using the file's name or any part of it's name (ie. You could find LIBRARY/DAT by telling D.D.M.S. to find Y/D). The D.D.M.S. search time is very good. D.D.M.S. can also update and add directories to it's data file. Finally, D.D.M.S. will print the individual directories, or an alphabetized listing of the files. Let's face it, when you need to find a file you don't want to waste anymore time than necessary. At $19.95 can you afford to be without D.D.M.S.? We don't think so!

SYSGEN

This program allows you to restructure NEWDOS 2.1 into a configuration of your choice. Is your disk drive capable of using 77 tracks? Well SYSGEN will change NEWDOS 2.1 into a 77 track DOS. SYSGEN will allow you setup your system so that you can now get at the lower memory addresses you've been kept from copying or examining. SYSGEN will modify NEWDOS 2.1 so that your disk drives can run at one of three speeds, slow (what your running at now), fast (about twice as fast), fastest (seek time is 5 msec). This function is great for people with MPI or Vista disk drives. Why not run them at the speeds they are designed to handle. NOTE: Shugart drives will only go to the fast speed. There are far more features to SYSGEN than can be listed here. SYSGEN is priced at $35.95, and with that price were the only game in town.

NORD DISASSEMBLER

This is the one you've been waiting for. This disassembler creates labels, and an Editor/Assembler readable file. If you have purchased a machine language program only to find it didn't do something you wanted or did something wrong now you can do something about it. Simply use the NORD DISASSEMBLER to create a source file and fix it yourself! This program has taken a year to develop, and the $44.95 price is well worth it!

Some of us are software oriented and could use tutorials about how some of the non-standard hardware is interfaced. For example, how do I get to hear sounds our of my machine without having to record it first. I'm not electronics oriented. Are filters and isolators really required and under what conditions? What are the consequences of using non-Radio Shack disks?

This gives you an idea of some hardware questions I have that seem to be obvious (or overlooked) by the majority of the writers.

Don Cronkhite
Newbury Park, CA

See **Letters** *in our last issue for the filter/isloators question. See also SYSTEM/COMMAND in May-Jun 79 (and LETTERS in Nov-Dec 79) for some discussion on making sounds from your computer. I say "Yea, verily" to non-Radio Shack drives because they are cheaper. But - I would get the first drive from them, the reason being that you get an excellent Manual on TRSDOS, the Disk itself, and last but not least a four drive cable.*

Your editorial (Nov-Dec 79) is well put and well taken. I don't recognize yours or anyone elses role as caretaker of myself. The principle is a general one; not just as applies to Consumer Protection, but in all of life. We are, each of us (supposedly) free and freedom loving, autonomous individuals, responsible for our own acts. - - - -Consumer protection is a meaningless buzz-phrase anyway - why should buying a quart of milk confer on me the title "Consumer": To paraphrase Shakespeare:"The worms are the final consumer". You might have pointed out in your editorial, reference supplier protection, that consumer fraud and theft (including employee theft) equals or exceeds the industrial/business fraud/theft.

Allan Stark
Detroit, MI

Egads! (And I thought I was a Civil Libertarian!)

I have been quietly working here in Lexington KY to solve some of the problems in the TRS-80. Problems such as noise sensitivity (program bombs when you turn on the light in the room), memory sensitivity (some RAMS seem to work better than others), and especially the one that goes "SN ERROR IN LINE XXXX" where XXXX is a non-existant line. This one has caused me much consternation, but now I have the solution. It is a solution that is easy to apply, and I will share it with anyone who will send me a self-addressed, stamped envelope. I will gladly mail a couple sheets of paper from my printer, as long as it only costs paper. I will not refuse a couple of stamps, or what have you, to help compensate for paper cost, but the info is free to anyone who requests it. Please help spread the word.

N Gregory
107 Fairdale
Lexington, KY 40505

The word is out, but why didn't you simply tell us what it was?

I passed up your rag a year ago because it was virtually *all* games and I believe you did not (foolishly) approve of lowercase modification that I have used to write two textbooks on Microwave Engineering!

R M Richardson
Chautaugua, NY

I can't agree with the "all games" idea, but you were right about our disapproval of hardware modifications. Since then, we have changed our attitude towards hardware mods considerably (see our cover story in this issue), and feel that you bought it, it is yours, and if you want to cut it up then be our guest. We plan on having some kind of hardware mod in every issue starting with this one.

Owners of the new Sargon II chess program tape may be interested to know that there is a simple way to put the program on disk, requiring no additional software. First, power up Level II BASIC by holding down the BREAK key while you press the RESET button. Load the Sargon II master tape according to its instructions. When the program comes up running, hold down the BREAK key and push RESET again. Push ENTER for MEMORY SIZE?. Now carefully type in the following, check it and RUN it:

```
1 FORJ=1TO43:INPUTK:POKE30460+J,K:
    FORL=1TO80:NEXTL:NEXTJ
```

The program will prompt you with ? to enter each of the forty-three decimal numbers listed below. Type them in carefully, pushing ENTER after each complete number:

1 253 44 17 0 74 33 0 128 237 176 62 195
50 12 64 62 0 50 13 64 62 80 50 14 64 195
0 80 1 26 45 17 25 173 33 25 119 237 184
195 0 0

After the last one has been entered, the Ready prompt will reappear. If you see that you have entered a number incorrectly, press BREAK and run the program again. Now make sure the disk in the drive has at least 10 grans of free space and no write-protect tab. Still in Level II BASIC, type:

SYSTEM [ENTER]

And at the *? prompt, type:

/30490 [ENTER]

The disk will boot up normally. At "DOS READY", type: DUMP SARGON2/CMD(START=X'8000', END=X'AD19',TRA=X'ACFD') [ENTER]

When the disk stops running, it will have a copy of Sargon II on it. Push RESET, and at DOS READY type:

SARGON2 [ENTER] - The program should come up ready to play.

Roxton Baker
Ellington, CT

Very Clever!

Since you were interested in the Exclusive OR function I thought you might be interested in this one. It is very simple and apparent, but I have not seen it mentioned in any programs or publications.

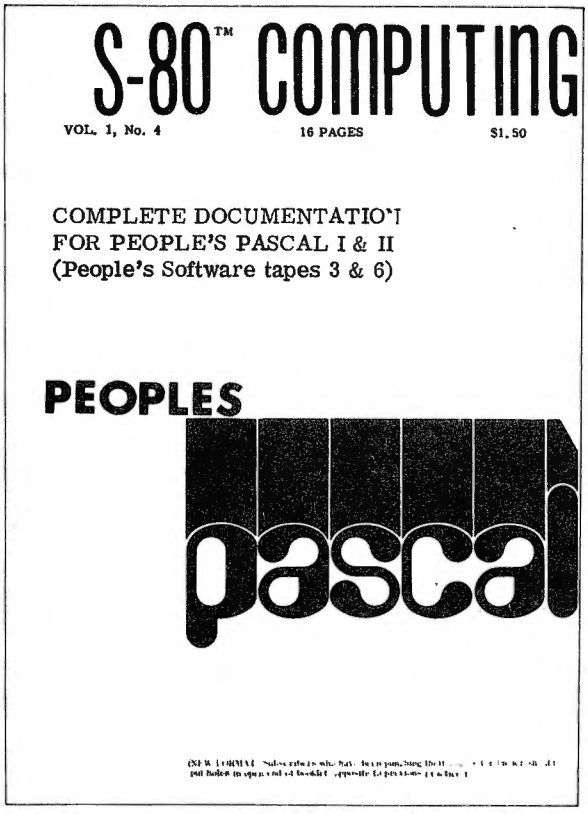
It is that the string functions (CHR$, LEFT$, RIGHT$ and MID$) may be concatenated. For example T$=A$+" "+CHR$(184). This allows addition of arrows and graphic symbols into strings, which can be useful in programming. Concatenation of the other string functions may also prove to be useful.

Also, I would like to add that in the JOURNAL for Nov-Dec 79 page 10, that if you hit the letter O while holding down I and Y the cursor mark will be printed in

(Continued on Page 10)

Announcing-TINY PASCAL FOR TRS-80

Have you heard about the $15 Pascal compiler?



At last your TRS-80 can run Pascal too! The Chung/Yuen "tiny" Pascal is fully implemented for Level II TRS-80, 16K and up. You no longer need be left out of the growing group of Pascal users, because People's Pascal gives you everything you need to write structured Pascal programs:

- tiny Pascal compiler
- complete text editor for writing your programs
- complete tiny Pascal monitor
- sample Pascal programs
- user's manual

(TRS-80 Computing issue 1:4)

People's Pascal is both a powerful, structured language and "CPU expeditor". People's Pascal programs execute at least four times faster than Basic, and often eight-times faster! Special functions open up the complete graphic capability of TRS-80. You now have the means to write those dazzling, impressive, high-speed graphics programs that are great for games, plotting, statistics, etc.

For the serious computerist, side two of People's Pascal II (tape 6) contains a larger compiler and complete source to the compiler, written in Pascal! This means you can re-compile the compiler, making changes, adding features, etc. (but this will take at least 36K RAM and a solid knowledge of programming).

With the complete People's Pascal operating system, you can save and load both source (Pascal) programs, and compiled programs, to or from cassette tape. This means that once you have de-bugged a program, you can save the P-code (compiled program) and thereafter, to run the program, you need only load the super-fast P-code.

Here is a partial list of People's Pascal features:
recursive procedure/functions • for (loop) • case if/then/else • one-dimensional arrays • write • read constant • repeat/until (loop) • "peek & poke" • plot (graphics for TRS-80)

DEALER INQUERIES INVITED

People's Pascal I (tape 3), is written in Basic, implemented for TRS-80 by John Alexander of Berwick Australia. It compiles more slowly and is harder to use, but includes instructions for converting to disk. People's Pascal II (tape 6) is entirely by the Chung/Yuen team and is a further development stage of their concept. Full documentation included.

Other People's Software tapes $8.

TRS-80 trademark
Tandy Corporation

computer information exchange, inc.

box 158 san luis rey ca 92068

amt. _____ encl.

PEOPLE'S SOFTWARE, includes 50¢ per tape postage/handling (via 1st class)

( ) Tape 1 Lev II 34 ass'td programs, or ( ) Lev I, 24, $8; $8.45 in CA
( ) Tape 2 Level II "Common Basic Programs", $8; $8.45 in California
( ) Tape 3 Level II PEOPLE'S PASCAL I, $15.50; $16.40 CA residents
( ) Tape 4 Level I, 17 assorted programs, $8, $8.45 CA residents (tax)
( ) Tape 5 Level II, 28 assorted programs, $8; $8.45 CA residents (tax)
( ) Tape 6 Level II, PEOPLE'S PASCAL II, $23.50; $24.88 CA residents

Make checks payable COMPUTER INFORMATION EXCHANGE

Charge my VISA ( ), MasterCharge ( ) #_____

_____(signed), expires _____

_____
name

_____
street address

_____
city, state, zip

addition to the letter O. This whole series of prints also works when holding the H and X or J and Z keys instead of I and Y. Thus, this takes care of all the non-keyboard characters (ASCII codes 92-95).

Robert A Hood
Bremerton, WA

I just read my first issue of your magazine (Nov-Dec 79) and I like it! I liked all of your articles - except the Stock Market Program - which I couldn't believe. In general, I think putting stuff into computers, and having them return the same stuff without manipulation is useless (like the whole printout in this example). Anyway, though much of this may be redundant, I thought I'd send some hints on cleaning up that program, which are applicable to many other programs.

1. TIME$ is usable for printing date and time. To set it in BASIC, poke the relevant numbers into the following places:

16454 = Month
16453 = Day
16452 = Year
16451 = Hour
16450 = Minute
16449 = Second

Time$ is a meaningful string, and can be manipulated as such.

2. Defining the format of a PRINT USING statement saves space if used more than once or twice, e.g., B$ = "$###,###.##", then LPRINTUSINGB$; whatever.

3. Funny thing about the TRS-80. LPRINTUSINGB$; ATAB(45) returns an error, but, LPRINTA$TAB(45);A will not. Besides the fact that semicolons are not needed before and after tabs, you can put anything you want in a PRINT or LPRINT line, just so *EVERYTHING* after the USING is the correct type of variable for that format.

4. For those of you using the pressure model line printer with continuous feed paper, POKE16425,55: LPRINTCHR$(11), will run the paper up to the tear off point rather quickly. Changing the value POKEd into 16425 changes the number of spaces (55 = 11, 54 = 12, etc.).

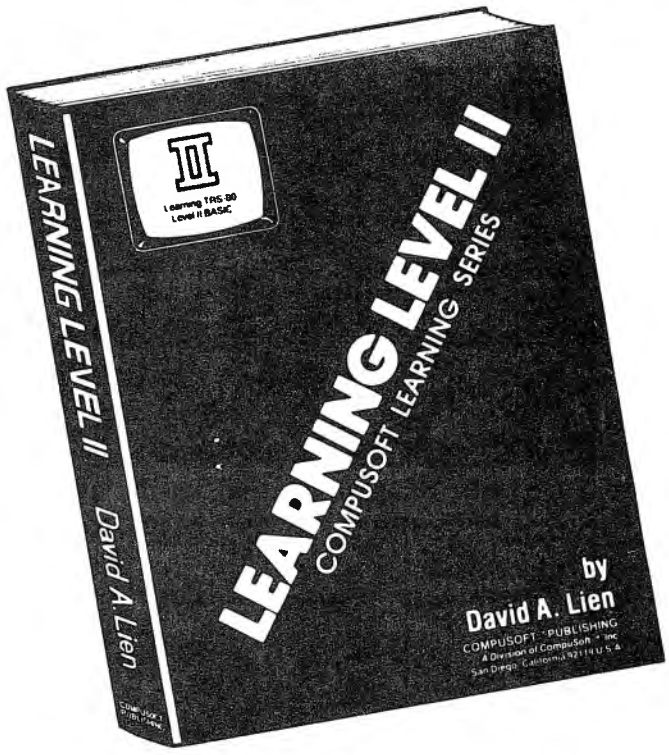
5. If you use a lot of subroutines, put them in the beginning of the program. Jump over them with statement 1 directing a GOTO. While the size of the line number is not relevant in line numbering the program, GOTO5000 uses 3 more bytes than GOTO1, because the computer needs one byte per digit. (Typically re-numbering a large program to start with 1, interval of 1, can save a couple hundred bytes for this reason.

Dan Connors
Harvey, LA

I have sensed, while editing all these letters, a real increase in the amount of useful information that can be exchanged through these columns. Thank you all for your contributions, and keep up the good correspondence!

Mike

Here's the Second Half...



$15.95
(soft cover)

Written by the author of your Level I Users Manual, LEARNING LEVEL II picks right up where the Level I Manual leaves off. It also supplies the changes needed to make the Level I Manual compatible with your Level II TRS-80.

LEARNING LEVEL II covers all Level II BASIC beyond Level I, plus much more. It shows you how to use the Editor, explains what the many error messages are really saying, and leads you thru conversions of Level I programs to Level II.

Dual cassettes, printers, the Expansion Interface with clock and other features are explained in the same easy-to-learn style that made the Level I Manual famous. LEARNING LEVEL II was created specifically for your Level II TRS-80!

Yes, I want to really **learn** how to use Level II!

COMPUSOFT™ PUBLISHING 8643-U Navajo Rd. • San Diego, CA 92119

Please send _____ copies of LEARNING LEVEL II. My check for $15.95 + $1.45 P&H is enclosed. (CA addresses add 6% sales tax).

I understand my order will be shipped promptly and there is a 30 day money-back guarantee.

NAME _____

ADDRESS _____

CITY _____

STATE _____

ZIP CODE _____

ITEMS AT RANDOM

In this issue we are asking you to fill out and return our reader survey. It is supposed to tell us something about you and what you want to see in these pages. (I know you read other magazines, and have been asked the same thing before, but it is the kind of information a publication needs if it is going to keep a finger on the pulse and give its readers what they want). Mostly, the survey should give us an idea of how to allocate editorial content on the various machine configurations. We are too cheap to have postage paid return cards - but, in exchange for your 15 cent stamp, you get three (count them!), three opportunities to get a free subscription (or extension). If you find the need for additional space for comments, please add an extra sheet. We figured that the 15th of April is normally such a bleak day that we would brighten it up some and have our drawing of the lucky winners then. I had intended to give a poster of our own Cathy Shappee in a wet tee shirt as first prize - but she quickly reminded me that this was *not* Playboy, and that it would be appropriate to have the computer do the selection of the winners. Oh well, nothing ventured, nothing gained...

TRSDOS version 1.2 for the MODEL II has been released. It is supposed to handle data transfers 60% faster, correct the "sequential PUT", and fix the backspace in EDIT mode as well as all cursor movements. Like most fixes, this one creates a new problem - 1.2 will not access a 1.1 disk. There is supposed to be a way, and when we understand it, we will pass it on.

CORRECTION! A Cassette Library (Jan-Feb 80 issue). Lines 320 and 1300 should **both** read:
FOR I%=0TOCNT%, and the last statement in line 1030 should read: **CPY%(J%)=CPY%(I%).** These were brought to our attention by several readers as well as the author, Roger Amidon.

Oops, it looks like we gave Digital Equipment Corp credit for having fathered CP/MI Should have been Digital Research (80-U.S. Jan-Feb 80, page 76).

In the **REBOOTS & KKEY-BOUNCES** department we have a correction to make to the Notes on Basic in the Jan-Feb 80 issue. Page 63, left column, program line 540 - near the end of the line there is a funny looking statement which says IF LEN(IN$)X0THEN. That "X" should be a "does not equal" sign. Further down that same column it says that "this can be done on Level II, Disk Basic or Model II Basic". Wrong! Model II Basic does not have PEEK or POKE (perhaps it was wishful thinking on the part of Terry).

Yes, 80-U.S. will be at the West Coast Computer Faire! We will be in Booth 1122C, near the center of Brooks Hall on March 14, 15 and 16. Stop by and see us.

Ex T-PAL READERS!
Welcome aboard!
We hope your stay with us will bc enlightening and rewarding.
(See Editorial, page 2)

RENEW EARLY!
If the 3 letter code on your mailing label ends in 30 you have already expired. If it ends in 50 this is your last issue! If it ends in 70 then the next (May-Jun 80) will be your last issue!

Until now, 80-U.S. has been primarily a software oriented Journal. Until now, that is. Our cover this issue features a six times resolution modification. The complexity of it makes it a case of overkill, but we think it is a nice way to tell the world that we will, from now on, cover hardware construction on a regular basis. No, they won't all be as "heavy" as the first. **Larry Panattoni** has joined the staff as our hardware person. His column (starting soon) will be entitled "Panattoni's Panacea", and for openers, will cover some clever things to do to your 779 Centronics printer.

Jim Crocker is now our Technical Editor, and as you may have noticed, is writing the "View" column, which is being geared towards the relative novice in computing with T-Bug and beginning Assembly Language. Yes, now that we have more pages to play with, we can cater to diverse interest groups within each issue.

It seems that Radio Shack made a fuss about the use of their product name at some recent fairs. The Computer Information Exchange, in their Nov-Dec 79 Bulletin is advocating the usage of **S-80** instead of **TRS-80**. The situation seems to be a replay of the fuss MITS stirred up with vendors claiming to have products compatible with the "Altair Bus". Industry responded by renaming the Altair bus "S-100". I can't see anything wrong with owning a generic trade name - like Kleenex, Kodak or Kotex, it sort of puts you "up front".

Projects, especially at the big business level, have a funny way of unfolding. The following recently came to my attention and it seemed worth passing along:

The 9 phases of a Project

1. Unbridled Enthusiasm
2. Guarded Optimism
3. Cool Objectivity
4. Quiet Confusion
5. Partial dis-engagement
6. Utter disenchantment
7. Search for the Guilty
8. Punishment of the Innocent
9. Awards to the non-participants

The other day I reached for my Level II Manual (for a change) and found things in it I didn't believe existed. Which brought to mind an idea - when it was new, we all probably latched on to those commands and items that worked for us - and ignored some of the other goodies. Now, it is all old stuff and we don't bother to go back and look again. Try it, bet you find something new in there.

Sign in a programmers cubicle:
Think first - code later

Nice days are **made** not **had**.
Tell them you saw it in the **JOURNAL**
Mike

Adventure

A New Type of Game

Welcome to an astonishing new experience! **ADVENTURE** is one of the most challenging and innovative games available for your personal computer. This is not the average computer game in which you shoot at, chase, or get chased by something, master the game within an hour, and then lose interest. In fact, it may take you more than an hour to score at all, and will probably take days or weeks of playing to get a good score. (There is a provision for saving a game in progress).

This game was inspired by the huge Adventure game which has appeared on large mainframe computers the last several years. But there are important differences. Not only will **ADVENTURE** fit into a relatively small computer, but the 'interpreter' is designed so that different Adventures can be created by changing the data base. So look for more Adventures in the future...

In playing the game you wander thru various 'rooms' (locations), manipulating the objects there to try to find 'treasures'. You may have to defeat an exotic wild animal to get one treasure, or figure out how to get another treasure out of a quicksand bog. You communicate thru two-word commands such as 'go west', 'climb tree', 'throw axe', 'look around'.

Six Different Adventures

ADVENTURELAND (by Scott Adams) - You wander through an enchanted world trying to recover the 13 lost treasures. You'll encounter WILD ANIMALS, MAGICAL BEINGS, and many other perils and puzzles. Can you rescue the BLUE OX from the quicksand? Or find your way out of the maze of pits? Happy Adventuring.....

CS-3007 TRS-80 16K Level II (Machine language)	$14.95
CS-3506 TRS-80 32K Disk (Includes Pirate Adventure)	$24.95
CS-1009 PET 24K (Includes Pirate Adventure. In Basic)	$14.95
CS-5003 SORCERER 16K (Machine language)	$14.95
CS-9003 8" CP/M 48K Disk (Includes Pirate Adventure In Microsoft Basic	$24.95

PIRATE ADVENTURE (by Scott Adams) - "Yo Ho Ho and a bottle of rum..." You'll meet up with the pirate and his daffy bird along with many strange sights as you attempt to go from your London flat to Treasure Island. Can you recover LONG JOHN SILVER's lost treasures? Happy sailing matey.....

CS-3008 TRS-80 16K Level II (Machine language)	$14.95
CS-3506 TRS-80 32K Disk (Includes Adventureland)	$24.95
CS-1009 PET 24K (Includes Adventureland. In Basic)	$14.95
CS-5004 SORCERER 16K (Machine language)	$14.95
CS-9003 8" CP/M 48K Disk (Includes Adventureland. In Microsoft Basic.	$24.95

ORIGINAL ADVENTURE (by Crowther, Woods, Manning and Roichel) - Somewhere nearby is a collosal cave where others have found fortunes in treasures and gold, but some who have entered have never been seen again. You start at a small brick building which is the wellhouse for a large spring. You must try to find your way into the underground caverns where you'll meet a giant clam, nasty little dwarves, and much more. **This Adventure is Bi-Lingual**—you may play in either **English or French**—a language learning tool beyond comparison. Runs in 32K CP/M system (48K required for SAVE GAME feature). Even includes SAM76 language in which to run the game. The troll says "Good Luck."

CS-9004 8" CP/M 32K Disk	$24.95

MISSION IMPOSSIBLE ADVENTURE (by Scott Adams) - Good Morning, Your mission is to... and so it starts. Will you be able to complete your mission in time? Or is the world's first automated nuclear reactor doomed? This one's well named, its hard, there is no magic but plenty of **suspense**. Good luck.....

CS-3009 TRS-80 16K Level II (Machine language)	$14.95
CS-3507 TRS-80 32K Disk (Includes Voodoo Castle)	$24.95
CS-5005 SORCERER 16K (Machine language)	$14.95

VOODOO CASTLE (by Scott Adams) - Count Cristo has had a fiendish curse put on him by his enemies. There he lies, with you his only hope. Will you be able to rescue him or is he forever doomed? Beware the Voodoo Man.....

CS-3010 TRS-80 16K Level II (Machine language)	$14.95
CS-3507 TRS-80 32K Disk (Includes Mission Impossible)	$24.95
CS-5006 SORCERER 16K (Machine language)	$14.95

THE COUNT (by Scott Adams) - You wake up in a large brass bed in a castle somewhere in Transylvania. Who are you, what are you doing here, and WHY did the postman deliver a bottle of blood? You'll love this Adventure, in fact, you might say it's LOVE AT FIRST BITE.....

CS-3011 TRS-80 16K Level II (Machine language)	$14.95

To order, send payment plus $1.00 shipping or bankcard number to **Creative Computing Software,** P.O. Box 789-M, Morristown, NJ 07960. Or call toll-free 800/631-8112 (in NJ 201/540-0445).

sensational software

creative computing software

New PRODUCTS

1700 BAUD LOADER

ABS Suppliers, PO Box 8297 Ann Arbor, MI 48107 have introduced their 1700 baud loader. It allows you to Load & Save as well as Verify BASIC or SYSTEM tapes 3 times faster. It eliminates 2/3 of your wasted tape I/O time. They have included a New Visual Loading Reference (allows you to see the best tape volume setting), checksum error routines, load & go option, automatic preloader and provision to give your BASIC program a six character name. Price is $25.00, postage paid.

RACET GSF FOR MOD II

A Generalized Subroutine Facility (GSF) is available now for the TRS Model II. Machine Language functons available through USR calls include USR PEEK and POKE capability, both byte and word. It will compress and un-compress data, move blocks of data and propagate across arrays, among other things. It sorts 1000 elements in six seconds! 5000 elements in 26 seconds! GSF for the Model II is available from RACET Computes, 702 Palmdale, Orange, CA 92665 (714) 637-5016 for $50.00 on your DOS diskette.

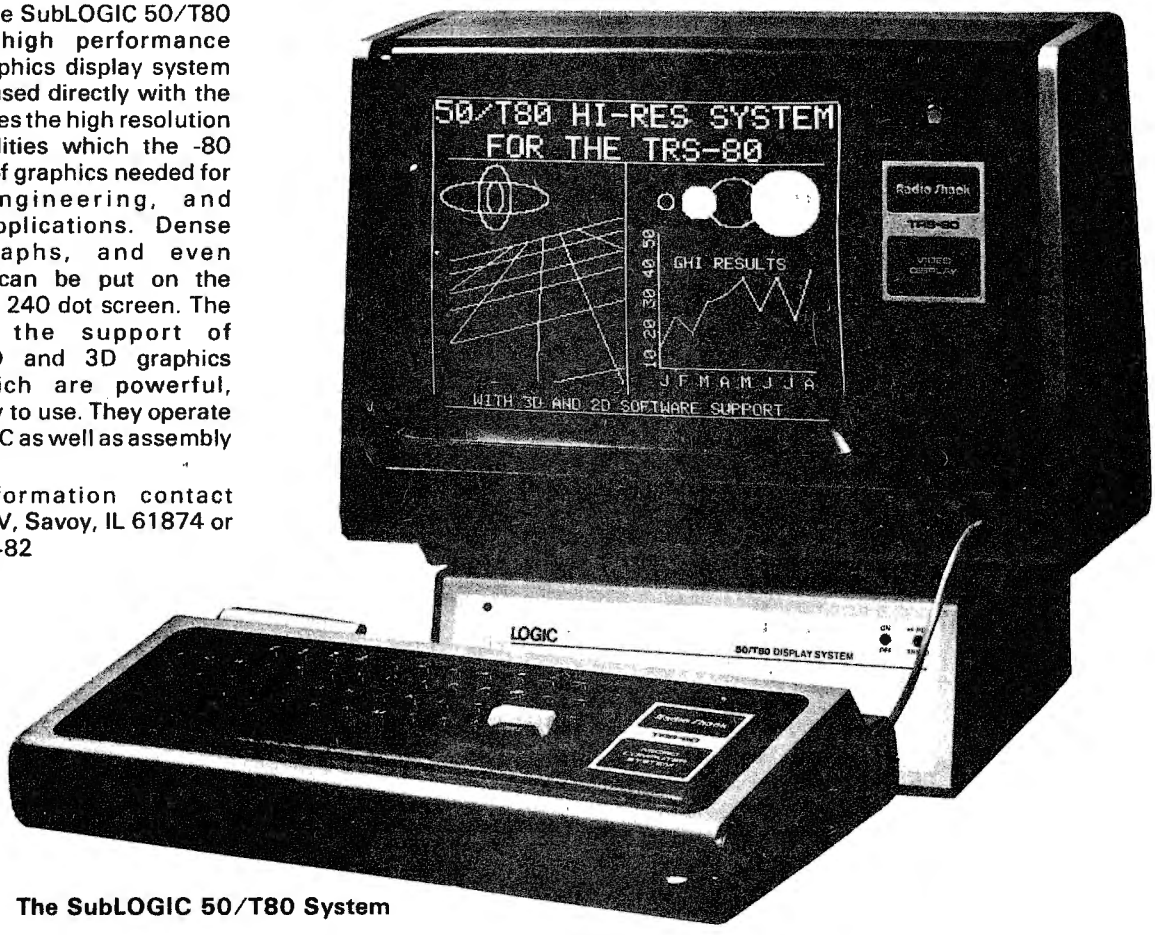
KEYPLUS INTEGRATED UTILITY

Keyplus is a powerful collection of utilities that can be enabled directly from keyboard. Choose from seven different keyboard entry modes: Basic shorthand (2 modes), direct graphic character input (3 modes), typewriter style input, and standard TRS-80 keyboard entry. Keyplus routines may be enabled or disabled in just two key strokes. Keyplus is marketed by SJW Inc, PO Box 438, Huntingdon Valley, PA 19006. The Level II version is available for $14.95, an even more powerful disk version (32 or 48K) sells for $19.95. PA residents add 6% tax.

HI-RES GRAPHICS

(See photo) The SubLOGIC 50/T80 system is a high performance professional graphics display system designed to be used directly with the TRS-80. It provides the high resolution graphics capabilities which the -80 lacks - the kind of graphics needed for scientific, engineering, and educational applications. Dense drawings, graphs, and even alphanumerics can be put on the 50/T80's 256 X 240 dot screen. The 50/T80 has the support of SubLOGIC's 2D and 3D graphics packages, which are powerful, proven, and easy to use. They operate easily from BASIC as well as assembly language.

For more information contact SubLOGIC, Box V, Savoy, IL 61874 or call (217)359-8482



The SubLOGIC 50/T80 System

Please excuse the lack of Reader Inquiry Cards in this issue. That space was taken by the Reader Survey. Reader Service Cards will return next issue.

DISK I/O for Editor/Assembler

DISK*MOD is the latest TRS-80 software from MISOSYS. It is a machine language program that will patch the Radio Shack Ed/Assembler to provide disk I/O for the text buffer as well as support object files written directly to disk. Full tape operation is retained. The disk-modified Editor/Assembler functions under TRSDOS 2.2, NEWDOS or VTOS 3.0. A number of additional features also provided including Block Move and Global change commands. 32K and 1 drive are required. DISK*MOD is priced at $20.00 - MISOSYS, 5904 Edgehill Drive, Alexandria, VA 22303

SWITCHABLE ISOLATOR

Electronic Specialists announces the SWITCHABLE ISOLATOR as the newest addition to their popular ISOLATOR filter line. Featuring input spike/surge suppression together with three individually switched and filtered sockets, it answers requests from many microprocessor customers. Total ISOLATOR load capability is 1875 watts Max, with each switched, filtered socket capable of handling a 1 KW load. Each switch has an associated pilot light. Model ISO-6 Switchable Isolator (see photo) $119.95 - from Electronic Specialists Inc, 171 S Main St. Natick, MA 01760 (617) 655-1532

COMMON PILOT

Lords Corp, PO Box 99, Port Angeles, WA 98362 (206) 457-3064 has introduced COMMON PILOT which is written in machine code and requires 1 drive and 32K. It allows a full range of floating point, scientific and array operations, string manipulation and answer processing features. It is an ideal CAI language which allows an instructor to write a program after only a few hours. Common Pilot with an 85 page manual and a Disk oriented Text Editor sells for $99.95. The manual alone may be purchased for $12.00 postpaid.

STATISTICAL ANALYSIS

Now available from Radio Shack is a system of computer programs designed for the analysis of data in business, education, medicine, government administration and other fields. It may be used with Level II Basic or Disk Basic on a 16K TRS-80. The Advanced Statistical Analysis system consists of 13 computer programs stored on cassette tapes, and a comprehensive manual which takes the user through each program step-by-step. Included in the system are ten programs for describing data sets and conducting statistical data analysis; two utility programs for preparing, updating, and listing data files stored on tape or disk; and a program to aid in selecting data samples. Advanced Statistical Analysis is available from Radio Shack Computer Centers and participating Radio Shack stores and dealers, nationwide, priced at $39.95

QUICK REFERENCE GUIDES

Murnane & Associates, 1056 Metro Circle, Palo Alto, CA 94303 have made available 3 Quick Reference Guides for programmers of the TRS-80. Level I Basic Guide ($0.95 each), Level II Basic Guide ($1.95 each) and a Microcomputer BASIC Quick Reference Guide ($0.95 each). Add 50 cents for handling and postage. CA residents add 6% tax.

The Switchable Isolator

INFO STORAGE & RETRIEVAL

Information Storage & Retrieval (ISAR) is a data base management system designed and priced to accomodate personal applications for persons desiring to use TRSDOS random file structures. The primary ISAR system consists of six modules which allow creation of any number of new files, accurately define all elements within each file, and manipulate each file to your needs.The package includes a complete source listing, documentation, potential recovery techniques and suggested personal applications with sample documentation. ISAR is available on cassette for $13.95 or diskette for $16.95. For further information contact The Alternate Source, 1806 Ada St. Lansing, MI 48910 or phone (517) 487-3358

"THE EIGHTY"

Softside Publications, publishers of SOFTSIDE and PROG 80 Magazines announces the publication of "The Eighty". "The Eighty" will be a consumer oriented publication filled with the latest in new products, a classified marketplace, a consumer forum and much more. It will have a controlled circulation of 15,000

persons who have subscribed to magazines specifically for the TRS-80, or who have purchased or requested information on products for the TRS-80. It will be issued monthly. For editorial and Advertising information call (603) 673-0585/0586. Softside Publications, PO Box 68, Milford, NH 03055

CP/M SYSTEM FOR MOD II

A fully compatible operating system for the new Radio Shack Model II computer has been announced by MPU in San Carlos, CA. The new operating system will work with CBASIC and all other CP/M programs, requiring absolutely NO changes to the operating codes. Source and object files will both work on the system, and programs from MPU and other CP/M code suppliers will be fully compatible. The entire system sells for $249.95. For more information write MPU at PO Box 808, San Carlos, CA 94070 or call toll free anytime (800) 824-7888 and ask for Operator 883 to take your name and address and information will be sent to you (In CA call 800 852-7777)

KEYWORD INDEXING PACKAGE

A TRS-80 based keyword indexing system for searching and accessing data or text records stored on disk has been developed by NORTHEAST MICROWARE. The package consists of a series of programs that enable the user to create a disk data file, build an index of all occurances of "keywords" in the file, and inquire into that file using any combination of keywords. It can be used for a large variety of applications where searching of data or descriptive information is required. The package requires 2 disks and 32K of memory. It is available for $39.95 from NORTHEAST MICROWARE, PO Box 6153, Syracuse, NY 13217

BASEX COMPILER

BASEX COMPILER is a powerful, easy-to-learn language that runs up to 20 times faster than BASIC and makes very compact code (only 2K runtime overhead). It is an 8K interactive compiler and features arrays, strings, 16 bit math, block move and search, subroutines with multiple arguments, fast graphics and tape I/O. Cost is $25. plus $8 for the 97 page manual, from: Interactive Microware Inc, PO Box 771. State College, PA 16801 (814) 238-8294

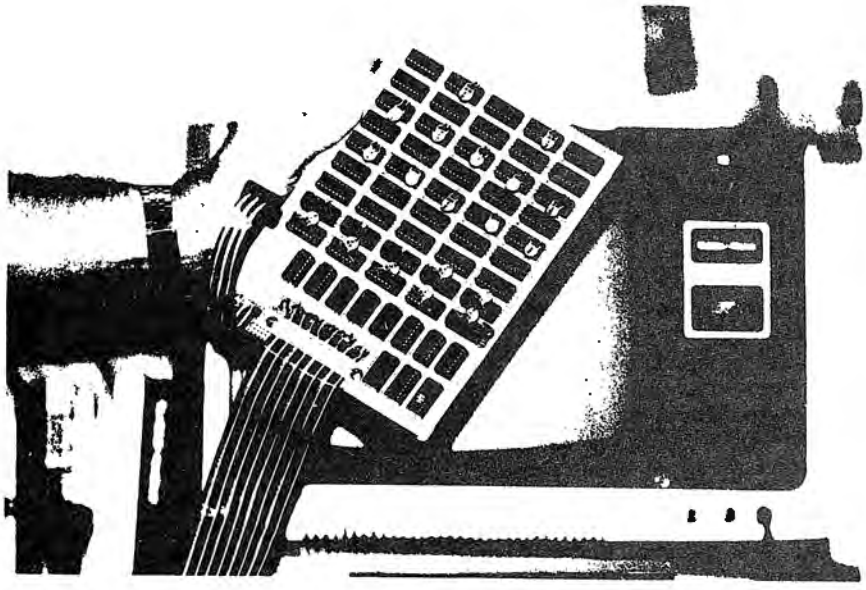
RENT A COMPUTER

Connecticut Information Systems will now rent TRS-80 (Models I and II), Apple, Pet and Atari computers. In addition, the company will also rent add-on accessories and software for these computers. Their policy is to let the user rent for any length of time, and return the item or buy it at his option. They charge 15% of manufacturer's list price for the first month's rent, and 10% for each subsequent month. Associate Director Phil Brotman says that the rental policy is being implemented to give prospective buyers a chance to see if computer products are really right for them, without wasting the full purchase price and getting "stuck". Contact Connecticut Information Systems, 218 Huntington Road, Bridgeport, CT 06608 (203) 579-0472

MODEL I GRAPHICS

Larry S Panattoni, Yakima, WA

(Improved 6 Times!)



The Author's Prototype Board

The graphics of the TRS-80 Model I leave something to be desired, even though programmers like Leo Christopherson and others seem to do wonders with a resolution of only 6144 dots!

Just think what they could do with a resolution of 36864 dots.

This article will show that such resolution can be obtained. At the same time you can retain access to all of the 36864 dot locations with the original commands SET, RESET, POKE, PEEK, PRINT@ or CHR$( ). This will allow you to use the string packing technique with the higher resolution. Furthermore, this modification will not affect the size or shape of any character, such as letters, numbers or punctuation. Only the graphics characters are affected by becomming smaller and greater in number.

It would be helpful if you have a copy of the "TRS-80 Microcomputer Technical Reference Handbook" which is sold by Radio-Shack. Also, a

background in digital electronics is almost a must, for without it, I would recommend you not break the seal on the bottom of your keyboard.

This modification consists of more than 50 added integrated circuits in an out-board box, as well as an additional power supply to power them. It requires cutting traces and adding wires inside the keyboard. The added chips, for the most part, consist of additional video memory. There are no changes to existing ROM, and the access to these additional video memory cells is through the BASIC "OUT" statement.

The complete modification, with drawings, schematics and text consists of between 30 and 50 pages. Only an overview will be presented here. This project is not intended for the novice in electronics; you should know what you are getting into before attempting it. This article should show what can be done, as well as help you decide if the project is within your capability.

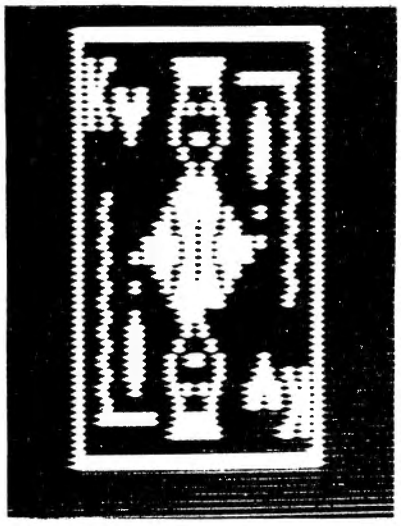
HOW IT IS NOW

The screen, as you know, is divided into 1024 locations, 64 horizontally on each of 16 lines vertically (See Fig 1). Each of these 1024 locations are capable of displaying one character (letters, numbers or punctuation), or a graphic character (a combination of six squares). The graphics character is shown in Figure 2.

If you light up only one of the six squares of the graphics character on your screen, you will note that it consists of four horizontal scanning lines (we'll call these "rows"). It also consists of three dots per row. You cannot see the separation between these three dots, but they are there. Therefore, since there are two of these squares side by side in each graphics character, there are a total of six dots (columns) and a total of twelve rows per graphics character.

Figure 3 shows how a letter (the letter 'S') occupies the same location that a graphic character could occupy. All letters, numbers and punctuation are of a 5 X 7 dot matrix arrangement. Notice the sixth dot of each row is always blank. That column is used for the separation between characters on the same line. The separation of each of the sixteen lines is achieved by blanking the top line, and the last four scanning lines of the graphic character location.

In the Technical Handbook, page 108, schematic sheet 2, you will notice Z29 (character generator) and Z8 (graphic generator), each being fed from the video memory through latch Z28. Each of these generators feed a parallel in - serial out shift register (Z10 and Z11), of which the serial out becomes the modulation for the video screen. It is important to note that only one of the shift registers is turned on at a time. Now, if we modify only the input to shift register Z11 for the graphic character, we will not affect any letter, number or punctuation displayed on the screen.



WHAT WE ARE TRYING TO DO

If we were to obtain individual control of each of those six dots (columns) across the graphics character, (which is easily done) we would have three times the resolution as normal. However, that would give us a high, rectangular shape, so we must cut the height down some. This can be done by allowing the graphic character to remain on the screen for only one half the scanning rate; instead of four scanning lines per graphic square - only two. It is a little more troublesome to achieve this, but is can be done.

Figure 4 shows what we would have - one column of two rows. We can now light up and control any one of 36 dots in each of the 1024 graphic character locations on the screen, giving a total resolution of 36864.

As Figure 4 shows, when in the higher resolution, the normal six square graphic character will occupy the upper left six squares of this new 36 square character. This allows the retention of all normal graphic commands.

Figure 5 shows a simplified drawing of the modification necessary to the input of the graphic register Z11. There are six memory block sections at the top of Figure 5. Each memory block consists of six memory units (21LS02 1K static memory chips). The six units in memory block 1 are the normal screen memory units in your keyboard. As shown in the schematic (sheet 2, page 108 of the Technical Handbook) they are Z62,Z61,Z45,Z46, Z47 and Z48.

USING HIGH RESOLUTION

The command OUT 254,x (x is from 0 through 6 and sets the graphics mode) is used to select which part of the former graphics character we will be lighting up. You will be allowed to draw on the screen six different times, because there are now six times more dots than normal. In high resolution mode "one" (OUT 254,1), all graphic commands will print at normal screen location, but only affect the upper left portion of the new 36 square graphic character (see Fig 4).

When you select mode "two" (OUT 254,2), you will have all the screen locations to draw on again. Only this time your graphic commands will affect only the upper middle portion of the new graphics character. You can continue to modes 3, 4, 5 and 6. Each time you will regain all screen locations to draw upon; but each of the graphics commands will be affecting only that portion determined by the mode you have selected.

The screen display will not forget what you have drawn in a previous mode. It will continuously display them on the screen, even though you are now in a different mode. And, they will remain there until you go back to that mode and change them. The graphics at a particular location can be removed and replaced by a letter or number, regardless of the mode you are using. Letters, numbers and punctuation are always stored in memory section 1 (the normal screen memory). They take precedence over all graphics.

When in mode "zero" (OUT 254,0), the normal six square graphic characters are utilized, but the graphic for them are the same as the graphics stored in mode "one". The memory used for mode "one" is the same memory as that used for the normal graphics in mode "zero".

Be aware you may have to program the graphics on the screen six different times, depending upon the nature of your drawings - but then, you have six times as many graphics dots with which to program!

Note: Complete documentation, including drawings, schematics and text is available from the author for $15.00. Address requests to: Larry S Panattoni,216 South 18th Ave., Yakima, Washington 98902 (price includes postage) ●

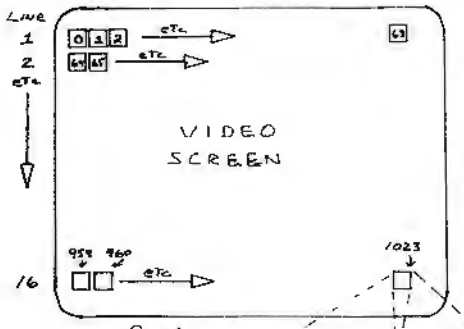


fig 1

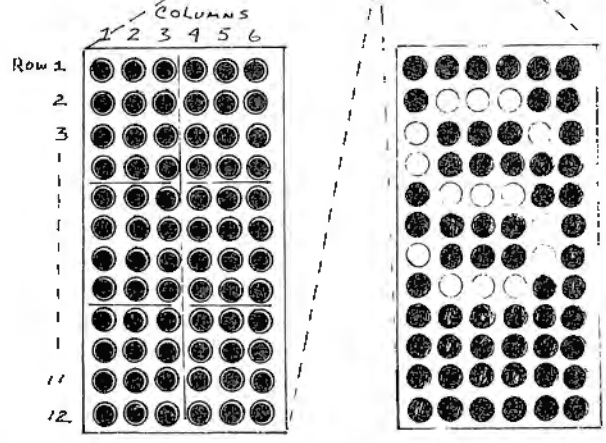


fig 2

fig 3

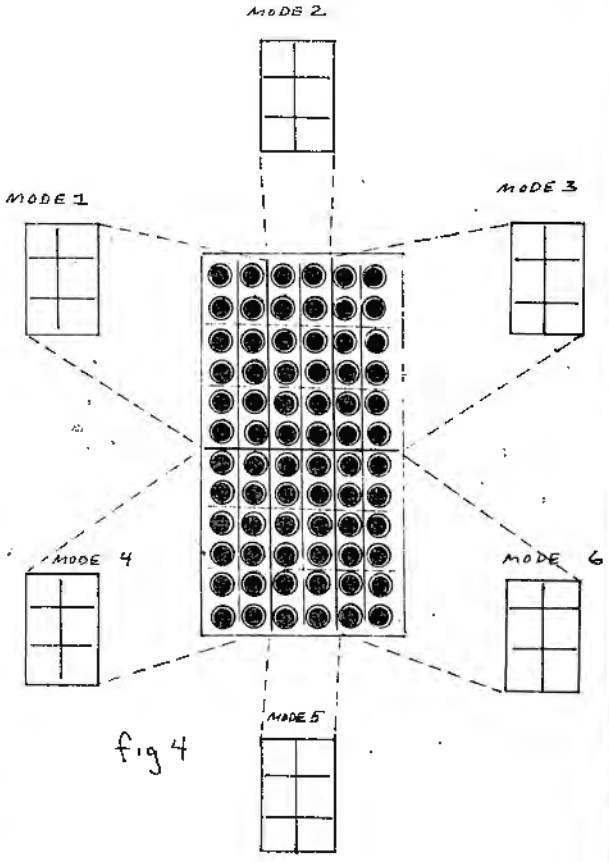
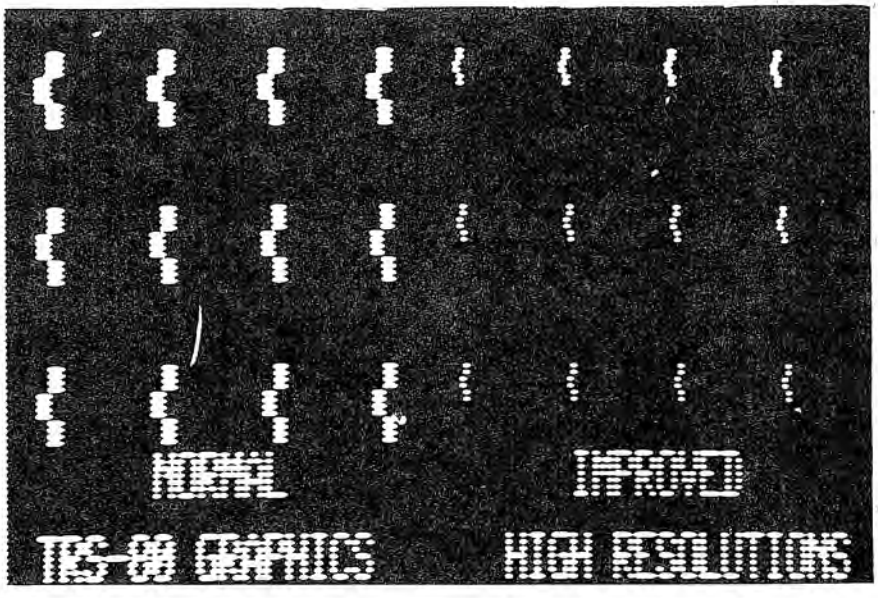


fig 4



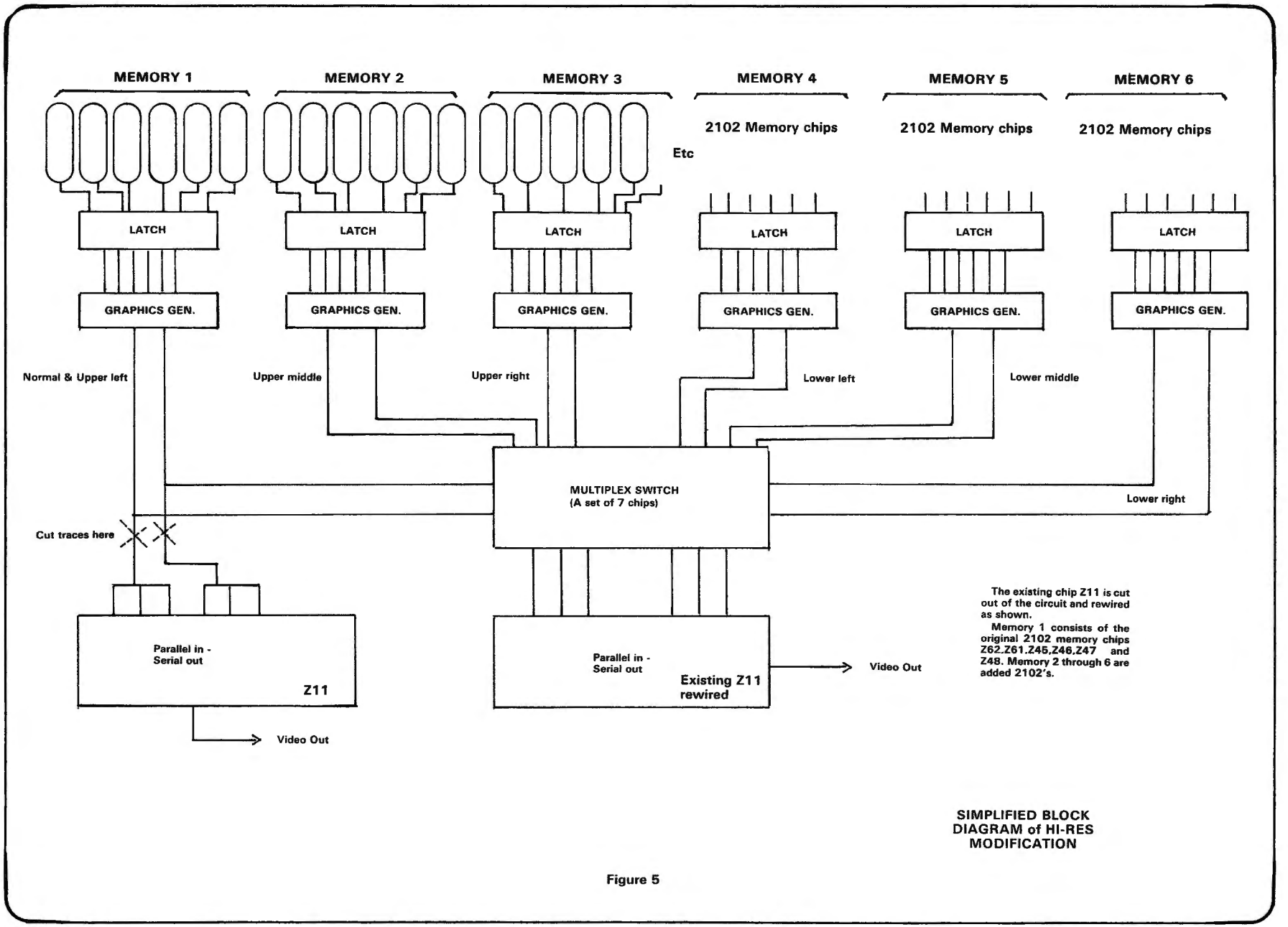

MEMORY 1
MEMORY 2
MEMORY 3
MEMORY 4
MEMORY 5
MEMORY 6
2102 Memory chips
2102 Memory chips
2102 Memory chips
Etc
LATCH
LATCH
LATCH
LATCH
LATCH
LATCH
GRAPHICS GEN.
GRAPHICS GEN.
GRAPHICS GEN.
GRAPHICS GEN.
GRAPHICS GEN.
GRAPHICS GEN.
Normal & Upper left
Upper middle
Upper right
Lower left
Lower middle
Lower right
MULTIPLEX SWITCH
(A set of 7 chips)
Cut traces here
Parallel in -
Serial out
Z11
Video Out
Parallel in -
Serial out
Existing Z11
rewired
Video Out
The existing chip Z11 is cut out of the circuit and rewired as shown.
Memory 1 consists of the original 2102 memory chips Z62.Z61.Z45,Z46,Z47 and Z48. Memory 2 through 6 are added 2102's.
SIMPLIFIED BLOCK DIAGRAM of HI-RES MODIFICATION

Figure 5

PARTS LIST

QUANITY	PART NUMBER	DESCRIPTION
30	2102LHPC	1024 X 1 bit static RAM
4	74LS157	switch
2	74LS93	counters
2	74LS08	quad 2 input NAND gates
5	74LS32	quad 2 input OR gates
1	74156	multiplexer
6	74LS367	hex buffers
2	74LS368	hex buffers/inverters
2	7404	hex inverters
5	74LS174	data latches
5	74LS153	graphics generators
1	74LS132	quad 2 input schmitt NAND gates
1	74LS04	hex inverters
1	74LS30	8 input NAND gate
1	74LS175	data latch
16	-------	14-pin wire-wrap IC sockets
62	-------	16-pin wire-wrap IC sockets
1	2037	40-pin EDGE-CARD CONNECTOR w/cable
5	-------	16-pin DIP connector cable
1	10K OHM, 1/4 WATT	resistor
1	68 OHM, 1/4 WATT	resistor
1	.33 OHM, 2 WATT	resistor
1	560 OHM, 1/4 WATT	resistor
1	1.2K OHM, 1/4 WATT	resistor
1	3.3K OHM, 1/4 WATT	resistor
1	1K OHM, 1/4 WATT	potentiometer
1	4.7K OHM, 1/4 WATT	resistor
1	220 micro-farad, 16VDC	capacitor
1	10K micro-farad, 16VDC	capacitor
1	.001 micro-farad, 16VDC	capacitor
1	10 micro-farad, 16VDC	capacitor
1	47 micro-farad, 16VDC	capacitor
29	.1 micro-farad, 16VDC	capacitor
1	MJE 2955	transistor
1	2N3904 (2N2222)	transistor
1	6.2 VDC	ZENER diode
1	full-wave bridge rectifier	
1	6.3 volt 3 amp	transformer
1	723C	voltage regulator

micellaneous board, wire-wrap wire, heat-sink, etc.

NOTE: all integrated circuits, sockets, and 40-pin socket (with cable) are available from:
HOBBY WORLD
19511 BUSINESS CENTER DRIVE
NORTHRIDGE, CA 91324
You may call toll-free:
1-800-423-5387
In CA call 1-800-382-3651
Their local number is (213) 866-9200

Notes on BASIC

It has been said that it is difficult to set MEMORY SIZE from inside a BASIC program. To a certain extent that is true, but in a very special case you can.

The Case where you can is at the very beginning of a program, within the first few executable lines. Here, before the program has generated any strings, you can change MEMORY SIZE.

To see why this works, we have to look at how the high end of memory works. Location 16561 contains the address of the highest memory available to the BASIC program. The interpreter will start putting newly created strings (except those wholly defined within a program line), into the space cleared by a CLEAR instruction starting at this location.

If you are careful not to create any strings until you have changed the MEMORY SIZE, then there are no references to this area and nothing will be messed up when you work up there. An example of a possible use for this area is to decide whether or not to put a machine language program in memory, compute where it should be located, memory protect for it, and then POKE it in.

In order to see how it works, let's step through the process. First, execute a CLEAR 1000 to clear the high memory and remove all pointers. Then locate the top of available memory as a double precision number

```
SM# = PEEK(16561 +PEEK(16562)*256
```

Now, subtract off the number of bytes that the machine language routine will take (let's say 100 bytes).

```
SM# = SM# - 100
```

This will be the memory protect location. We now have to put it back. Since this is a number larger than 256, we have to take it apart into two bytes so we can POKE it back into memory. To take it apart, we execute the following statements:

```
X2 = INT(SM#/256): X1 = SM# - X2*256
```

The new values X1 and X2 are the two byte representation of what we want for a MEMORY SIZE. To set MEMORY SIZE, we do:

```
POKE(16561),X1: POKE(16562),X2
```

After this is done, we can POKE our machine language routine in without fear. Executing another CLEAR will set aside memory for strings starting at our new MEMORY SIZE. Be careful, if you don't CLEAR before setting the MEMORY SIZE, it dosen't work and if you don't CLEAR after MEMORY SIZE you may have a string that contains your machine language program.

Multiple command lines can be very useful, and they can be used in command mode as well as in programs. The command line

```
CSAVE:CSAVE:CSAVE
```

will save a program to tape three times in a row. Not all commands will work this way, since they return control directly to BASIC. The line

```
CLOAD?:CLOAD?:CLOAD?
```

will only verify the first program on a tape and not the first three.

Need to have the month of the year available easily? The simplest way is to store a string with the names of all the months, and then use a MID$ function to get the correct one. If you are interested in having only the first three letters of the month name, then you can use the string.

```
MONTH$ "JANFEBMARAPRMAYJUNJUL
       AUGSEPOCTNOVDEC"
```

If you know the number of the month (for instance from inputting the date as 6/23/79 and breaking down the string for the month number - M = 6) then the correct month name can be taken from the string with the following call:

```
MID$(MONTH$,(M-1)*3+1,3)
```

This says that the first character of the month name is in position (M-1)*3+1 in the string MONTH$, and the name is 3 spaces long. For M 6 (June), the first character should be in position 16. In fact, we have

```
MID$(MONTH$,16,3) , "JUN"
```

which is what we want!

This isn't limited to just months. You can also use this technique to form Hexadecimal numbers as strings by forming a string.

```
HEX$ "0123456789ABCDEF"
```

and pulling the correct character out of the string for each position of the HEX number as you compute it.

Clearing memory is always a problem and sometimes you just can't tell how much memory will be available in a particular system. It is legal however, to use a statement like this:

```
CLEAR 2*MEM/3
```

To clear two-thirds of the available memory for strings. In this way, you do not have to specify a number for the CLEAR that may or may not fit into the target system.

Did you know that the statement:

PRINT STRING$(60,"*")

will have the same effect as.

PRINT STRING$(60,42) ?

Do you need to know what line a program is in during execution? Location 40A2H and 40A3H (16546 & 16547) hold the current line number. Try the following during a program.

PRINT PEEK(16546)+256*PEEK(16547)

It should print out the line number. This information can also be used from within a USR routine to tell where (in a program) control has been transferred *from*.

A frequent source of confusion for the new computer owner is the distinction between Bits, Bytes, and Words. There is a simple explaination, the definitions are different depending upon who you read and what system you are talking about.

Everyone agrees that a "Bit" is simply the smallest unit in a computer. It is a single 1 or 0, an ON or OFF in a memory location. A "Byte" on the other hand, is a group of eight Bits together as a unit.

We can refer to an "eight bit byte" which is a number that ranges from 0 through 255 (decimal), or 0 through FF (Hex), or in terms of actual bits, 00000000 through 11111111 (binary). What is so special about a Byte? Well, it happens to be a fundamental unit for many computers since. 1. Many microprocessors handle 8 bits at a time, and/or 2. Data coding is generally in 8 bit bytes (ASCII for example can be coded in 8 bits).

Have you heard of the "NIBBLE"? It is a group of 4 bits taken as a unit. Each Hexadecimal digit can be represented by a NIBBLE.

A Word is more loosely defined. Generally a computer Word is the largest unit that the processor can handle at one time. For example, in the TRS-80, most normal work is done with an 8 bit word size (although some computations can be done with 16 bits), while the Control Data 6400 computer works with 60 bits at a time, a 60 bit word size!

Here is another ROM routine to try in your Model I. If you set the HL register to point to an ASCII number, then a call to 1E5A will return the Hex value in DE (register pair BC is used in the process). Try this short routine using DEBUG (or another monitor such as TBUG) and single step through:

CD 5A 1E

You can put this anywhere in memory. In assembly language, this is just "CALL 1E5A". In order to see how it works, use the register command for your monitor to set HL to a memory location, then set the PC register to the address of the first byte of our little program, then single step and look at DE. You might want to try putting some ASCII codes into a few memory locations after the 3 byte program.

ROM location 0049 is the start of a keyboard scan routine that scans the keyboard until a key is pressed and then returns the ASCII value of the key in the A register.

ROM location 0033 displays the byte in the A register on the screen at the current cursor location. ●

KNOW YOUR COMPUTER

Mike Schmidt, Editor

"The Facts do not necessarily reflect the Truth"
Kung Fu, circa 1970

A walk through the inards of your machine, and a simplified look at what goes on there.

Are you intrigued by what goes on inside your computer? Would it help you to program it if you did know? How about all the fancy jargon you see spread all over the pages of this and other publications; terms like HEX, ASCII, Machine Code, Interpreter, Compiler etc? Where do all these terms come from and why do we need them?

To answer just one of the questions above we can say that you really do not need to know the internal workings of your computer to program it - but it sure helps! If you are the type of person who can operate comfortably on "blind obedience", just following the rules and never wondering about them, this may not be for you. On the other hand, if you are not ---

Let's look at figure 1. This is a functional block diagram of your computer system. There are five main blocks including the power source.

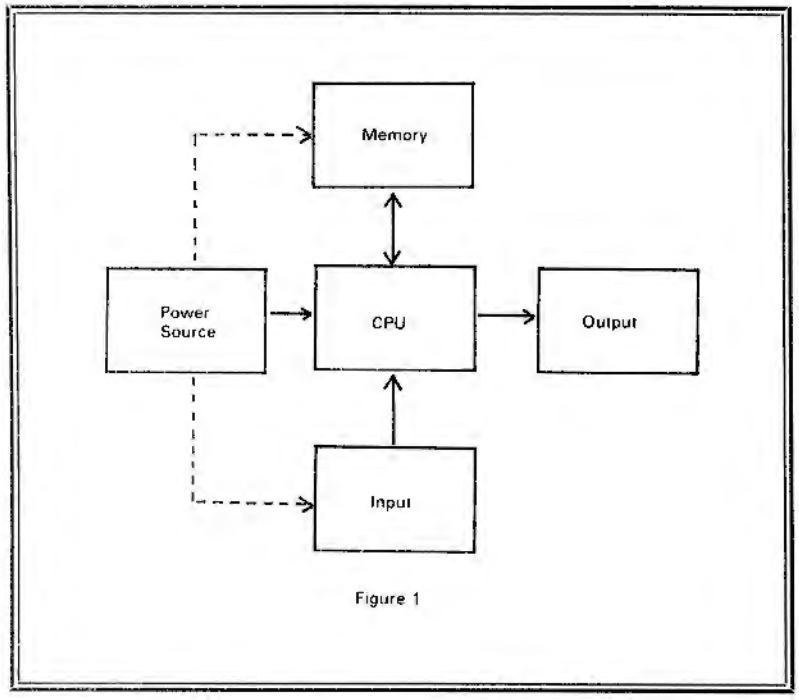


Figure 1

Since the power source is taken for granted, it is sometimes left off of such diagrams.

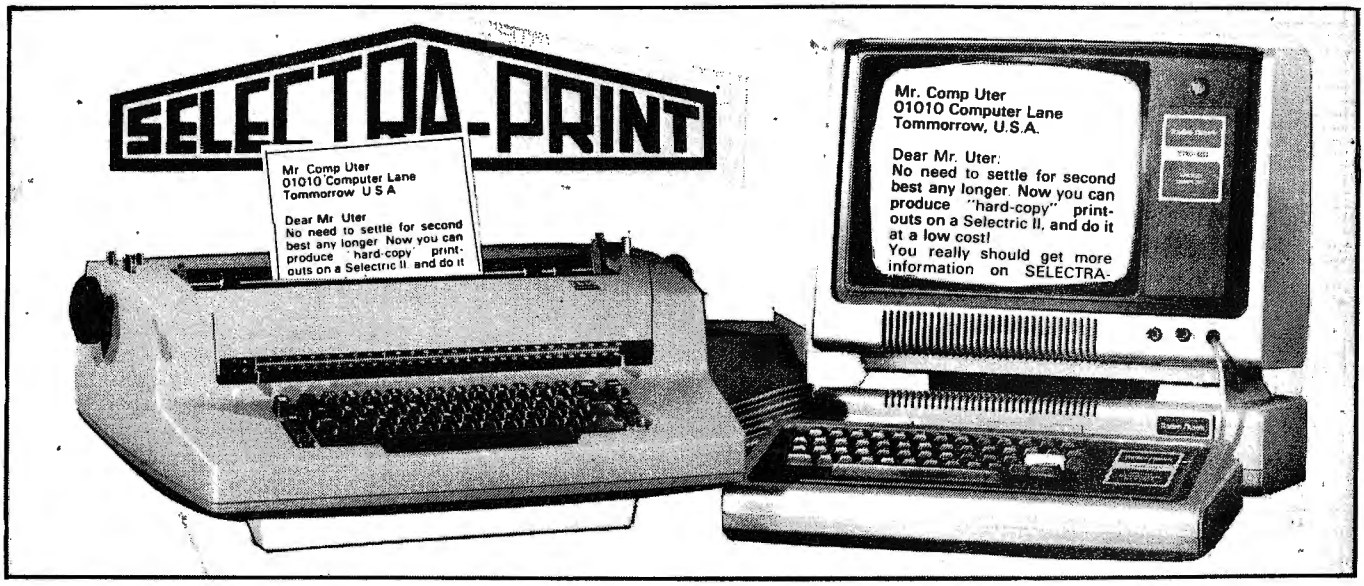
Note that all lines lead to the Central Processing Unit which in our case is the Z80 microprocessor chip. Just about everything that goes on in your computer either goes through, or is controlled by, that Central Processor (CPU).

We need a means of getting information into the CPU, so there is an Input block. This is generally the keyboard, but it can also be the cassette recorder or a modem. The input block can also represent such things as joy-sticks, digitizers, light pens and other assorted paraphenalia.

The output block obviously represents our video screen. It can also represent a line printer, an X/Y plotter, the modem again, or the cassette recorder (during CSAVE or PRINT# operation). It can also represent other things which can be controlled by the computer.

As you can see from the above, even though the functional blocks are

(Continued on Page 59)



IF YOU OWN A TRS 80 *.....

TRS-80 is a product of Radio Shack, A Tandy Corporation

SELECTRA-PRINT PUTS YOU JUST A CABLE LENGTH AWAY FROM A "HARD—COPY" (Typewriter Quality: clean, clear, high-fidelity) Word Processing System!

. . .just a simple hook-up with the supplied cables and your *SELECTRA-PRINT* is ready to run.

For Word Processing on a TRS-80, for example, just command "LPRINT" and *SELECTRA-PRINT* automatically outputs clear, clean, high-fidelity, hard copy. . . . and of course you can use it to print-out any other information you might need.

SPECIAL NOTE:
***SELECTRA-PRINT* is a Selectric II typewriter and although it has been modified for computer print-out, it may still be used as a standard office typewriter.**

PRICE: $1,850.00* *TRS-80 Version $2025.00

OPTIONS:

Dual Pitch	$195.00
Correction Feature	$195.00
Noise Reduction Feature	$65.00
RS-232 Interface	$195.00

* ***SELECTRA-PRINT is VERSATILE!***
SELECTRA-PRINT is compatible with most micro- and mini-computers including ***PET ● Apple ● Heath H8 ● IMSAI ● Cromemco ● Alpha Microsystems ● Space Byte ● North Star Horizon ● SWTP ● Vector Graphic ● Sol ● Polymorphic ● Digital Group ● Ohio Scientific ● Altair ● Sorcerer Xitan ● Rex ● KIM ● EXORcisor . . . .***

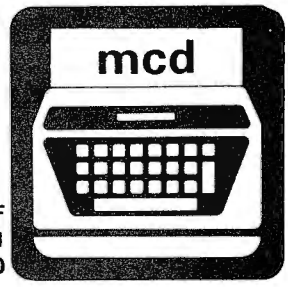
***"From now on,* PRINT-OUT takes on a new meaning"**

Direct International Sales:
International Sales Division
17648 Orna Drive
Granada Hills, CA 91344, USA

Discounts Available to
EDUCATIONAL ACCOUNTS
Contact **Delores Sun P.O. Box 8394**
***Ann Arbor, MI 48105* (313) 665-8514**

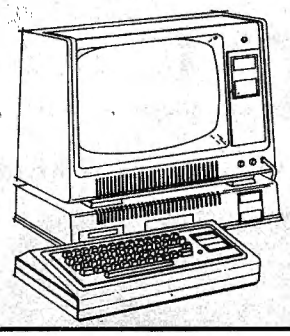
M30
micro computer devices inc.

960 E. Orangethorpe, Bldg. F
Anaheim, California 92801
Telephone (714) 992-2270



"Innovators to the Microcomputer Industry"

From ACCESS UNLIMITED...



TRS-80 Owners Save on Equipment & Software!

We buy in quantity. You buy at discount. Brand new factory-warranted products from Apparat, Percom and other best-rated companies.

Order today, call toll-free: 1-800-527-4196†

Percom Mini-Disk Systems

These mini-disk systems from Percom store more data, are more reliable. TFD-100™ units store 102K bytes of formatted data on *either disk surface*. TFD-200™ drive systems provide 197K bytes of formated data on-line. (35-track Tandy drives store only 86K bytes.) Access times are fastest possible with your Expansion Interface. Heavy duty power supply — included with each drive — runs cooler, lasts longer. Low noise three-wire ac power cord is safer than RS two-wire cord. Multiple drive systems are mounted in a single enclosure. Enclosures are finished in compatible silver enamel. Prices:

TFD-100™ (40-track)
One-Drive Add-On ..... $ 374.95
Two-Drive Add-On 749.95
Three-Drive Add-On 1124.95

TFD-200™ (77-track)
One-Drive Add-On ..... $ 634.95
Two-Drive Add-On 1268.95
Three-Drive Add-On 1903.95

Price includes Percom PATCH PAK™ described elsewhere in this ad.

Disk System Interconnecting Cables

Improved design places drive 0, which includes the cable termination, at the end of the cable — not in the first position like the RS design — to eliminate the reflected noise of an unterminated cable and thereby improve data and control signal integrity. Prices:

Two-Drive Cable ..... $ 24.95
Four-Drive Cable 34.95

Power Line Filter

Eliminates switching spikes and other noise caused by equipment sharing the TRS-80* ac power line. Also reduces TV interference caused by the TRS-80.* 115/250 V, 50-400 Hz. Instructions for easy installation in standard aluminum mini-box chassis.
Price ..... $ 19.95

Media

Minidiskettes:
10 Disks in a convenient plastic organizer box $34.90
Single Disk ..... $ 3.49

Cassettes:
These data cassettes provide orders-of-magnitude improvement in data integrity over ordinary audio cassettes. Features include a pilon-coated pressure pad which eliminates lint and minimizes erratic tape motion, and an energy-absorbing foam pad that is superior to a leaf-spring-mounted pad which tends to oscillate and cause fluttering. Five-screw case virtually precludes deformation during assembly.
Pilon-10™ ..... $1.95 ea
Pilon-30™ ..... $1.99 ea

Disk System Software

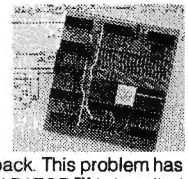
MICRODOS,™ an advanced, easy-to-use operating system for TRS-80s* using Percom disk storage systems, *works entirely with Level II BASIC commands*. MICRODOS™ replaces primitive TRSDOS*, is more powerful and faster, yet resides in only 7K bytes of RAM. Up to 10 user-defined functions may be added to existing functions, and hexadecimal constants may be used in expressions. MICRODOS™ is supplied on a system minidiskette that also includes three self teaching BASIC language programs: (1) a file management program, (2) a disk utility program — illustrating how disk utilities may be written with only a few BASIC statements, and (3) a sample application program. The application program is an expandable, user accessible "notebook" of information about MICRODOS™ BASIC statements. Price $29.95

PATCH PAK™ is supplied free (on disk) with the purchase of a Percom TFD-100™ or TFD-200.™ PATCH PAK™ extends TRSDOS* to accommodate 40- and 77-track drives. It also deglitches TRSDOS* 2.1. With the patch applied to TRSDOS* 2.1, interference with disk operations from the TRS-80* "heartbeat" pulse is eliminated, as is premature stopping of the drive motor during operation — a problem sometimes referred to as "silent death." PATCH PAK™ is applied to your system disk "on the fly" using two drives. Application instructions are included.
Price (if sold separately) $10.00

new DOS, new DOS+ — Disk operating system corrects and improves TRSDOS* 2.1 Fixes keyboard bounce. Fixes APPEND, LOC and VERIFY. Fixes SYS3 bugs which crash the disk directory, and bugs which cause "Lost Data" errors. Enhancements include: machine language line renumbering program • faster LOAD & SAVE functions • space saving allocation of minidiskette granules • password disable/re-enable capability • validity check of output to disk — and more. new DOS+ also includes these exceptional utilities: Editor/Assembler with tape & disk I/O and output to line printer • disk-based disassembler with output to line printer • SUPER ZAP — Apparat's hex dump utility • Level I ROM located in Level II RAM • LMOFFSET for Tape/Disk transfers — and more. Prices:

new DOS $49.95
new DOS+ $99.95

New! the SEPARATOR™



This PC board plug-in adapter for the TRS-80* virtually eliminates data read errors (CRC error — Track locked out!) which occur on high density inner disk tracks when clock and data bits are not reliably separated during playback. This problem has plagued many TRS-80* systems. The SEPARATOR™ is installed in the Expansion Interface without modifying the host system. Caution: Opening the TRS-80* Expansion Interface may void the limited 90-day warranty. Price $29.95

How to Order

Order by calling Access Unlimited toll-free on 1-800-527-4196†.

Mail orders also accepted. Orders may be charged to a VISA or Master Charge account, or paid by a cashier's check, certified check or money order. We accept COD orders with 25% deposit. Sorry, we cannot accept personal checks. We pay shipping and insurance charges on orders over $1,000.00. Add approximate insurance and shipping charges for under $1,000.00. If in doubt about these charges, ask when you call in your order. Texas residents include 5% sales tax. Minimum order: $20.00. Allow 2 to 4 weeks for delivery.

†Texas residents call (214) 494-0206

Prices and specifications subject to change without notice.

ACCESS UNLIMITED

315 N. Shiloh · Ste. D1 · Garland, TX 75042

(214) 494-0206

™ trademark of Percom Data Company, Inc.

*RADIO SHACK and TRS-80 are trademarks of Tandy Corporation

NINE TAC TOE

Roy Groth, Brookline, MA

FOR 16K Level II and UP..

Most people know that Tic Tac Toe is a trivial game. With perfect play, every game will end in a draw.

Playing nine games of Tic Tac Toe at the same time against the computer to see who can win the most of the nine may not be trivial though. The first player seems to have some advantage, but even when going first, it is difficult to beat this program. Going either first or second, you stand a chance of losing at least half the games.

The board consists of nine Tic Tac Toe grids. Every space contains a number between one and twenty-seven. Each of these numbers appears in three different spaces. The positions of these numbers is not random. It is helpful to study the locations of the numbers before commencing play.

You are always "X", and the computer is always "0", but a flip of the random number generator determines who goes first.

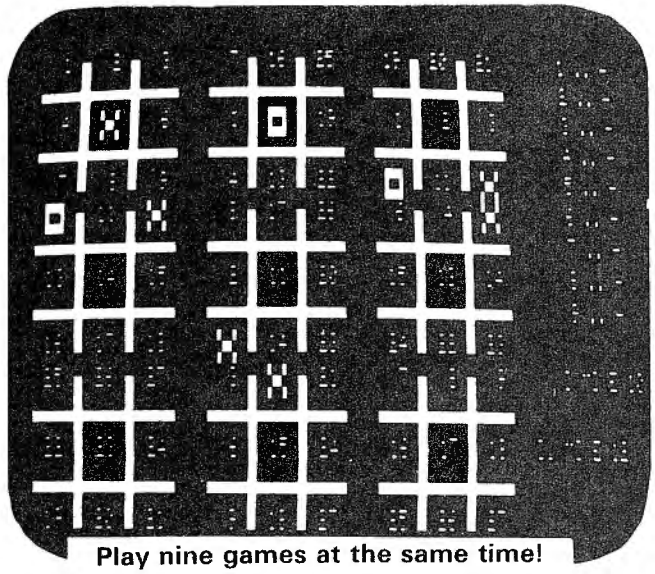
You take your turn by selecting a number that has not previously been chosen. Your symbol will then mark in the three places where your number appears.

Play proceeds in this manner, each player trying to form straight rows of three numbers (horizontal, vertical or diagonal) in one or more of the nine games. When a row of three in any game has been formed, that game is won. There is only one winner in each game. Once a game has been won, no more numbers in that game may be marked, although the same numbers can be selected if they appear in other unfinished games.

A totally filled game in which no one has three in a row is a draw. Play continues until the winner of all games has been decided. The player who has won the most of the nine is the winner. A draw occurs if both have won an equal number of games.

When you run the program, there will be a delay of a few seconds to read and process some data. Once that is done, the game board will appear.

The nine Tic Tac Toe grids are on the left of the screen. In the upper right hand portion of the screen the numbers one through nine appear, each followed by a dash. Each number corresponds to one of the nine games. The top row of games, from left to right, correspond to the numbers one through three, middle row, four through six and the bottom row, seven through nine.



Play nine games at the same time!

The dashes mean that the corresponding game is unfinished. If a game becomes a draw, the dash changes to a "C". If either player wins a game, the dash changes to that player's symbol.

When "YOUR MOVE" appears in the lower right of the screen, you enter the number you wish to take, if the move is legal, "X's" will be placed where appropriate.

If the computer instead displays "YOUR MOVE" again, then the move you just made was not a legal one, and you try again.

When the computer is "thinking" about its move, a graphics segment will work its way down the right side of the screen. This not only keeps you occupied while waiting, it gives you an idea of how much longer the computer is going to take. At times, the computer will change its mind, and will let you know that it has.

At the beginning of the game, the computer will take about 25 seconds to decide on a move. Near the end of the game, the time taken is reduced to less than 10 seconds.

When all nine games are finished, the final result is displayed. The computer then asks if you wish to play again. Most assuredly you will, to, in order to avenge your lo[illegible] ●

```
10 CLS
20 DEFINTA-Z
30 DIMBD(8,2,2),X(26,2,2),WI(8),BM(8)
40 REM     **********************************
50 REM     *          NINE TAC TOE          *
60 REM     *    PROGRAMMED BY ROY GROTH     *
70 REM     *      CONCEPT COURTESY OF:      *
80 REM     *         GAMES MAGAZINE         *
90 REM     **********************************
100 PRINT"ONE MOMENT PLEASE....THERE'S A LOT OF DATA TO READ."
110 FORI=0TO26:FORJ=0TO2:FORK=0TO2:READX(I,J,K):NEXTK:
    BD(X(I,J,0)-1,X(I,J,1)-1,X(I,J,2)-1)=I+1:NEXTJ,I
120 DATA 1,1,1,2,1,2,3,1,2,1,2,1,3,2,2,5,1,2,1,3,1,3,3,2,8,1,1,
    1,1,2,2,1,3,6,2,1,1,2,2,5,1,3,6,3,1,1,3,2,6,1,1,8,1,2
130 DATA 1,1,3,2,1,1,9,1,1,1,2,3,5,1,1,9,2,1,1,3,3,8,1,3,9,3,1,
    2,2,2,3,1,3,4,1,1,3,2,3,4,2,1,5,2,2,3,3,3,4,3,1,8,2,1
```

```
140 DATA 2,2,3,4,1,2,6,2,2,4,2,2,6,3,2,5,2,3,4,3,2,6,1,2,8,2,2,
    2,2,1,4,1,3,9,1,2,4,2,3,5,2,1,9,2,2,4,3,3,8,2,3,9,3,2
150 DATA 2,3,2,3,1,1,7,1,3,3,2,1,5,3,2,7,2,3,3,3,1,7,3,3,8,3,1,
    2,3,3,6,2,3,7,1,2,5,3,3,6,3,3,7,2,2,6,1,3,7,3,2,8,3,2
160 DATA 2,3,1,7,1,1,9,1,3,5,3,1,7,2,1,9,2,3,7,3,1,8,3,3,9,3,3
170 GOSUB690
180 IFRND(4)<3GOTO450
190 SV=-32767:MS=0:SB=1
200 FORMV=0TO26
210 SET(127,MV)
220 IFX(MV,0,0)=0GOTO310
230 SC=0:FORNU=0TO2:GOSUB240:NEXT:GOTO280
240 BD=X(MV,NU,0)-1:X=X(MV,NU,1)-1:Y=X(MV,NU,2)-1
250 IFWI(BD)<>0GOTO270
260 ONX+Y*3+1GOTO600,610,620,630,640,650,660,670,680
270 RETURN
280 IFSC<SVGOTO310
290 IFSC=SVANDRND(SB)<>1SB=SB+1:GOTO310
300 SV=SC:MS=MV:SB=2
310 RESET(127,MV):NEXTMV
320 PRINT@693,"I MOVE";MS+1;"  ";:MV=MS
330 FG=0
340 FORNU=0TO2:BD=X(MV,NU,0)-1:X=X(MV,NU,1)-1:Y=X(MV,NU,2)-1
350 IFWI(BD)<>0GOTO390
360 FG=1
370 BD(BD,X,Y)=-1:X(MV,0,0)=0:SC=0:GOSUB260:IFSC>500WI(BD)=-1:
    GOTO390
380 BM(BD)=BM(BD)+1:IFBM(BD)=9WI(BD)=2
390 NEXT
400 IFFG=0PRINT@693,"NO, I DON'T";:X(MV,0,0)=0:GOTO190
410 GOSUB700
420 FORI=0TO8:IFWI(I)=0GOTO450
430 NEXTI:GOTO1130
440 GOTO520
450 PRINT@817,"YOUR MOVE?";CHR$(95);:A$=INKEY$
460 B$=INKEY$:IFB$=""GOTO460
470 IFASC(B$)=13GOTO500
480 IFASC(B$)>57ORASC(B$)<48GOTO460
490 PRINTCHR$(8);B$;CHR$(95);:A$=A$+B$:GOTO460
500 PRINTCHR$(8);:MV=VAL(A$)-1:IFMV<0ORMV>26GOTO520
510 IFX(MV,0,0)<>0GOTO530
520 PRINT@817,STRING$(19,32);:GOTO450
530 FORNU=0TO2:BD=X(MV,NU,0)-1:X=X(MV,NU,1)-1:Y=X(MV,NU,2)-1
540 IFWI(BD)<>0GOTO570
550 BD(BD,X,Y)=-4:X(MV,0,0)=0:SC=0:GOSUB260:IFSC<-135WI(BD)=1:
    GOTO570
560 BM(BD)=BM(BD)+1:IFBM(BD)=9WI(BD)=2
570 NEXT
580 GOSUB700:PRINT@817,"YOU MOVE";MV+1;"    ";:FORI=0TO8:
    IFWI(I)=0GOTO190
590 NEXTI:GOTO1130
600 GOSUB910:GOSUB960:GOSUB1030:GOTO270
610 GOSUB940:GOSUB960:GOTO270
620 GOSUB950:GOSUB960:GOSUB1010:GOTO270
630 GOSUB910:GOSUB990:GOTO270
640 GOSUB940:GOSUB990:GOSUB1010:GOSUB1030:GOTO270
650 GOSUB950:GOSUB990:GOTO270
660 GOSUB910:GOSUB1000:GOSUB1010:GOTO270
670 GOSUB940:GOSUB1000:GOTO270
680 GOSUB950:GOSUB1000:GOSUB1030:GOTO270
690 CLS
700 PRINT@0,"";
710 FORI=0TO8:Q=INT(I/3)*3
720 FORJ=QTOQ+2
730 FORK=0TO2
740 IFBD(J,K,I-Q)<=0GOTO890
750 PRINTUSING"###";BD(J,K,I-Q);
760 PRINT" ";:IFI-Q=0PRINTCHR$(188);:GOTO790
770 IFI-Q=1PRINTCHR$(191);:GOTO790
780 PRINTCHR$(143);
790 NEXTK:PRINTCHR$(8);"    ";
800 NEXTJ
810 PRINTCHR$(26);CHR$(29);:IFINT((I+1)/3)=(I+1)/3GOTO840
820 FORH=1TO3:FORG=1TO3:PRINTSTRING$(4,140);CHR$(191);:NEXT:
    PRINTCHR$(8);"    ";
830 NEXTH:PRINTCHR$(26);CHR$(29);
840 NEXTI
850 FORI=0TO8:PRINT@I*64+53,I+1;".. ";:IFWI(I)=0PRINT"-";:
    GOTO880
860 IFWI(I)=1PRINT"X";:GOTO880
870 IFWI(I)=2THENPRINT"C";:ELSEPRINT"O";
880 NEXTI:RETURN
890 IFBD(J,K,I-Q)=-1PRINTCHR$(32);CHR$(183);CHR$(187);:GOTO760
900 PRINTCHR$(32);CHR$(153);CHR$(166);:GOTO760
910 QQ=0
920 SD=0:FORII=0TO2:IFBD(BD,QQ,II)<0SD=SD+BD(BD,QQ,II)
930 NEXTII:GOTO1050
940 QQ=1:GOTO920
950 QQ=2:GOTO920
960 QQ=0
970 SD=0:FORII=0TO2:IFBD(BD,II,QQ)<0SD=SD+BD(BD,II,QQ)
980 NEXTII:GOTO1050
990 QQ=1:GOTO970
1000 QQ=2:GOTO970
1010 SD=0:FORII=0TO2:IFBD(BD,II,2-II)<0SD=SD+BD(BD,II,2-II)
1020 NEXTII:GOTO1050
1030 SD=0:FORII=0TO2:IFBD(BD,II,II)<0SD=SD+BD(BD,II,II)
1040 NEXTII
1050 ON-SDGOTO1080,1090,1100,1070,1110,1110,1110,1120,1110
1060 IFSD<>0SC=SC-800:RETURN
1070 SC=SC+1:RETURN
1080 SC=SC+25:RETURN
1090 SC=SC+50:RETURN
1100 SC=SC+2000:RETURN
1110 RETURN
1120 SC=SC+60:RETURN
1130 SC=0:FORI=0TO8:IFWI(I)=2GOTO1150
1140 SC=SC+WI(I)
1150 NEXTI
1160 IFSC=0PRINT@757,"TIE";:GOTO1190
1170 IFSC>0PRINT@757,"YOU WIN";:GOTO1190
1180 PRINT@757,"I WIN";
1190 PRINT@817,"PLAY AGAIN";:INPUTA$:IFLEFT$(A$,1)="Y"RUN
1200 CLS
```

MISHMASH

R D Boozer, Inman, SC

Step-by-step Program Breakdown

3 Sets aside 1000 bytes string space. (Poke instruction in following line prevents unintentional restoration of data caused by loading program off of a cassette.

5 Provides a seed number for subsequent random number generation.

20 Determines number of words.

30 Value I represents position of the word to to be scrambled (I=1 for 1st word, etc.

40 Reading of word to be scrambled and its definition.

50-90 Routine which dissassembles the word into its constituent letters.

100-170 Routine which reshuffles letters into disorder Secondary routine 120-150 makes sure that no letter will appear more times in the scrambled word than it did in the original word.

180-210 Assembles the disordered letters into the scrambled word.

213-215 Makes sure a word is not reassembled into its original form (or anything similar).

220 Prints on the screen, the designation number of the word, the scrambled word, and its definition

240-300 Displays answers upon request

500-540 Data statements containing the words you want scrambled and their definitions

```
3 CLEAR1000
4 CLS:POKE16553,255
5 RANDOM
10 PRINT:PRINT"********************* MISHMASH *************************"
20 K=10
30 FORI=1TOK
40 READA$(I),B$(I)
50 NUM=LEN(A$(I))
60 FORJ=1TONUM
70 O$(J)=MID$(A$(I),J,1)
90 NEXTJ
100 FORQ=1TONUM
110 X=RND(NUM):G(Q)=X
120 IFW=1THEN160
130 FORZ=Q-1TO1STEP-1
140 IFG(Q)=G(Z)THEN110
150 NEXTZ
160 R$(Q)=O$(X)
170 NEXTQ
180 NU$=""
190 FORU=1TO NUM
200 NU$=NU$+R$(U)
210 NEXTU
213 IFLEFT$(NU$,4)=LEFT$(A$,4)THEN 60
215 IFRIGHT$(NU$,4)=RIGHT$(A$,4)THEN60
220 PRINTI;" ";NU$;"-";B$(I)
230 NEXTI
240 PRINT:INPUT"PRESS ENTER FOR ANSWERS";J$
250 CLS:PRINT:PRINTTAB(25),"* ANSWERS *"
260 FORM=1TOK
270 PRINTM," ";A$(M)
280 NEXTM
300 END
500 DATA"ABOVE","UPWARD","PRIME","FIRST CLASS"
510 DATA"PLUMMETING","RAPIDLY FALLING","DISCUSS","TO CONVERSE"
520 DATA"VALUE","WORTH","INVENTION","AN INNOVATION"
530 DATA"FOLLOW","TO GO BEHIND","MANUAL","BY HAND"
540 DATA"REVOLUTION","A TURN OR CYCLE","PIANO","KEYED INSTRUMENT"
```

Mishmash is an educational word puzzle generator which may have potential as a teaching aid for vocabulary building.

Stated briefly, the program scrambles the letters in each word of a group of ten, and it is up to the player to figure out what each word is by examining its garbled letters and reading its definition The program also displays the answers upon request.

Although this game is part of a book of programs written and copyright by the author, this particular program is given into public domain, to be reproduced or used in any way by readers of 80-U S. It is meant as an aid to spur student interest in vocabulary building.

FOR 4K Level II and UP...

realsoft™ PROGRAM STORE

(202) 337-4691

4200 Wisconsin Ave NW P.O. Box 9609 Washington D.C. 20016

'This TRS-80* fascinates my whole family!'

50 unique programs for all ages

A package of fifty different game, quiz, instructional, educational, financial and fun programs are available for any TRS-80. Every member of your family will enjoy the TRS-80 with this set of programs called "The Fascinating Fifty." And it is reasonably priced at $24.95.

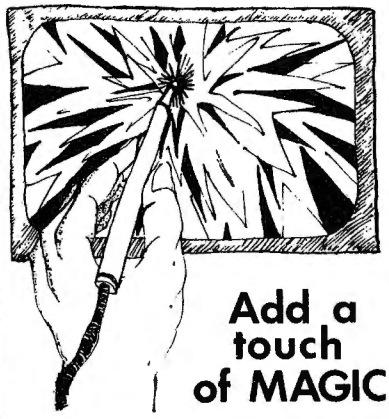
Designed for the whole family, these programs are fascinating. Have fun playing MANCALA or ZOO BREAK, guessing WORLD RECORDS or U.S. PRESIDENTS, learning MOON PHASES or WORD COMPLETION and using FINANCIAL TIMETABLE or METRIC CONVERSION. Other programs include: Rocket, Bingo, Dice, Reverse, World Population, World Area, States, Air Distance Between Cities, World Population by City, Grade Averages, Number Base, Accounts Receivable, Loan Amortization, Interest From Savings, Business Ledger, Calendar, World Clock, Blackboard I, II, III & IV, Maze, Letter Recognition, Numeral Recognition, Right Reading, Word Recognition, How Many, Add On, Take Away, Learn To Spell, Sounds Like, Story Telling, Multiply, Divide, Fractions, State Capitals, Foreign Capitals, Cub Reporter, Spelling - Grade 4, 5 & Jr. High, and Test Your Vocabulary.

If you would like every member of the family to enjoy your TRS-80, today order your copy of "Fascinating Fifty" by Neal Jensen for only $24.95.

Galactic Empire

by Douglas G. Carlston

This is fantastic! "Galactic Empire" opens up a three dimensional universe for exploration and conquest. While travelling through space you command the Imperial Forces of Galactica, your ship's senior officers and its computer. You must plan you construction of additional support craft and their arrival at nearby planets. Increase your strength by taxing and enlisting additional troops from the varied and unique planets you conquer. Besides maneuvering in a three-dimensional universe which changes from game to game, your use of time to travel between planets is very important. Good graphics and the sophisticated strategy make this game much different from any "Star Trek" or "Star Wars." Explore the galaxy for $14.95. You'll love it!



Add a touch of MAGIC with a Light Pen.

With only a few lines of BASIC, any program can accept inputs by touching the light pen on the video screen. Hundreds of applications! For only $19.95 you receive the light pen, instructions on using it in your programs, several sample programs, and a cassette of three light pen programs. This is the standard pen which several software companies support.

DUNJONQUEST

Datestones of Ryn

from Automated Simulations

In this microquest you are Brian Hammerhand on a mission into the Haunted Mountains and Underground lair of Rex the Reaver and his band of thieves. A perfect introduction into the Dunjonquest series which started with "The Temple of Apshai." 16k $14.95.
Also available "Temple of Apshai." $24.95.

Invader

from Level IV

Machine language arcade game with sound. $14.95

Adventure

by Gordon Letwin from Microsoft

The complete Adventure is the ultimate fantasy game which allows you to explore the "Colossal Cave." This is the same as originally written for the DEC PDP-10. It fills an entire disk and requires 32k. For just $29.95

Video Speed Reading

from Instant Software

Using the scientific principle behind the tachistoscope, a mechanical device used to flash characters or words on a screen, this three-part package will train your eyes and mind to quickly recognize numbers, letters, words and phrases. $7.95

Personal Inventory

from Southern States Systems

Designed to allow you to develop an inventory of your personal possessions for insurance purposes, you record for each item its description, serial number and value. Permits sorting, modifying and other record manipulations with storage on cassette tape. $12.50.

TRS 80 Disk & Other Mysteries

by H.C. Pennington

We don't usually list books, but this one is so unique that we thought you would want to know about it. There are over 100 pages about how DOS works, how a disk is organized, and how to recover from errors. This is THE technical backup for NEWDOS+ with great illustrations. $19.95.

Disk*Mod

by Roy Soltoff from Misosys

This machine language program modifies your copy of the Radio Shack Editor/Assembler for use with your minidisk and any disk operating system. You can save and load both text source and assembled object files. Unlike the NEWDOS+ version you can read the directory and the space used and available while in the EDTASM. You can also kill files. It is a complete disk modification for one or more drives.

Other capabilities are also added which are not found on NEWDOS+. The block move command relocates a section of text to any other area. The global change command permits, for example, changing a label throughout the text. The pagination feature provides hardcopy on 8 1/2 by 11 pages on either single sheets or continous paper. In addition, high memory can be reserved, like in BASIC, for machine language routines like printer drivers. You can also display the amount of memory remaining.

The <CLEAR> key is functional, the symbol table is sorted alphanumerically and output 5-across, the scroll up/down allows 15 lines on the screen, and the 'DEFM' assembly is improved. Lower case input is now permitted and you can branch to any address. Plus, it also corrects the errors in the Radio Shack tape version.

Save your time and make full use of your disk system by upgrading your Editor/Assembler today. $19.95

System Doctor

from The Bottom Shelf

Assure yourself that your TRS-80 is working properly. The causes of hardware problems are often hard to detect, but you can now do a thorough diagnostic check of your entire computer. It checks the ROM, the RAM, disk drives, video memory and display and cassette recorder. Also provided is a special test tape and a 12-hour test with a recording of results to find those occasional errors. $28.50.
On disk $38.50

THE PROGRAM STORE
4200 Wisconsin Ave NW
PO Box 9609 Dept KR6
Washington, D.C. 20016

YES please send me these TRS-80 programs:

title	price
postage:	$ 1.00
total:	

name: _____

address: _____

city, state & code _____

☐Check payable to The Program Store
☐MASTERCHARGE mc bank code:
☐VISA exp date:
card number:
signature:

realsoft™ THE PROGRAM STORE

(202) 337-4691

4200 Wisconsin Ave NW P.O.Box 9609 Washington D.C. 20016

TRS-80
a trademark of Tandy Corporation

THIS IS A SMALL SAMPLE OF THE PROGRAMS WE HAVE AVAILABLE. A CATALOG IS INCLUDED WITH EVERY ORDER.

taipan
by Art Canfil from Cybernautics
This is an intriguing game set in the 1860's on the wild China Coast. George Blank said in 80-US, that he likes "it more than any of the other single player games." And Ramon Zamora of Recreational Computing found that it "is so intriguing that I played for over six hours when I first received a copy." $9.95.

pyramid of doom
by Scott Adams from Adventure International
Adventure #8 is now available for you to explore. Will you be able to collect the treasures from the recently exposed pyramid? $14.95

adventure sampler
by Scott Adams from Adventure International
This is a scaled down version to give you a taste for adventure. $5.95

final approach
by K. H. Littlefield from Level IV
You fly a multi-engined plane in for a ground controlled landing. Quite challenging. $7.50

owl tree
by James Talley from 80-US
Can you fill the tree with owls by shooing away the bats? A logical, animated puzzle with sound, like Android Nim. $9.95

great race
by Scott Carpenter from 80-US
One or more players compete with the computer in a road race. Outstanding sound effects and great graphics. $9.95

lying chimps
by Roy Groth from 80-US
An animated version of the card game "I Doubt It" except that you play with three cheating chimps. Includes sound effects. $9.95.

concentration
by Richard Taylor from 80-US
This version of the game has sound effects and prizes like a line printer or a disk. $9.95.

scramble
by Richard Taylor from 80-US
Two players guess each other's words. You ca use either the program's words, or select your own. Includes sound. $9.95.

opera
by Richard Taylor from Acorn
Choose from five operatic selections featuring "The William Tell Overture." Let your TRS-80 play music classics! $9.95.

word challenge
by Richard Taylor from Acorn
Guess a popular phrase, letter by letter. One or two players can use either the program's phrases or select their own. Includes sound. $9.95

****

SPAINISH
by Lindy and Andrew Bartorillo
The advanced, language teaching program contains in excess of 500 phrases, 800 word vocabulary and 1600 verb conjugation forms. Switch between Spanish to English and English to Spanish. Print multiple choice question and answer test. $19.95 on diskette.
(German and French may be ready by the time you read this.)

CHECKBOOK PLUS
PERSONAL FINANCE MANAGER
by Andrew Bartorillo
This highly sophisticated checkbook program for disk can handle 700 checks per year. It sorts into categories and classifies for ease in tax preparation by family or small business. $19.95

SYSTEM SAVERS
by Tom Stibolt from Acorn
If you ever use the SYSTEM command, you should buy this two program package. These programs allow you to save any system tape on tape or disk, plus offer several features for machine language programmers.

In Computer Cassettes Magazine, Robert Purser said that a tape duplication program "should be in everyone's library." We agree and beleive that System Savers is the perfect answer.

With FLEXL, which is one of the two programs, you can make back-up copies of any system format tape. Most often a cassette that you make will load easier than an orginal. This also protects the valuable copies of your system programs.

Plus you also find the filename on any system tape because it is displayed on the screen. If you ever forget to save a program file name, just use FLEXL to recover it. And at anytime you can stop the reading of the tape by simply pressing BREAK.

For any machine language programmer, FLEXL offers the advantage of producing more efficient tapes than the object files from the assembler. It is also written to interface directly with the Small Systems RSM-1 and other monitor programs. And machine language tapes can be merged following the prompting of the program.

Disk drive owners can use TDISK to save any system format tape onto disk. Sargon I, Adventure, Airaid, Editor/Assembler and other programs cannot normally be loaded to disk. Now, TDISK allows you to save these programs onto disk. After DOS READY you will be able to simply type the filename and be up and running. It even loads non-contiguous tapes. TDISK will greatly increase the benifit of owning a disk drive.

And as a FREE BONUS, Acorn Software Products, Inc. also provides complete instructions on how to load MicroChess 1.5 onto disk. A short BASIC program will let you play chess directly from disk.

Complete your system with the routines not found in either Level II or DOS for only $14.95. Order your System Savers, today!

DISASSEMBLER
by Roy Soltoff from Misosys & Acorn
This two pass Z-80 disassembler produces symbolic lables with output to either the video monitor, printer or tape. Radio Shack's Editor Assembler can load the tapes. If you own the Editor Assembler, complete the package with this program. Program on tape for two different memory locations. TRS Level II $19.95

BASIC TOOLKIT
by F. Barry Mulligan from TBS
If you program in BASIC, you can use this package. This easy-to-use machine language utility will save you time and and effort.

The use of variables in BASIC programming can often get rather complicated. While you can keep track of where each varialbe is used, why not let the computer do that work? Well, this program lets the computer do just that. It produces a "map" alphabetically listing the variables in any of your BASIC programs and gives the line number where they are used.

In addition, the program displays a listing of each line number jumped to followed by the lines from which the jumps are made. This includes all GOTO's, GOSUB's and RESUMEs. This can be especially important when trying to limit the number of lines in a program or before making any changes to a subroutine.

Esides these two 'tools' you can also merge BASIC programs from cassette, recover any programs 'lost' by an accidental NEW, and test the computer's memory. And all of these functions are available from a single keystroke while you are working on your BASIC program.

For these features and more, order your 'Toolkit' today! $19.80.

PACKER
by Cottage Software
This is the ultimate editing tool for BASIC program lines. There are five commands which allow easier reading of BASIC programs and more efficient execution by the computer.

The 'unpack' command breaks multiple statement lines into single statement lines with extra spaces for easier reading and editing. The 'short' command deletes any unnecessary words like LET and all REMarks.

The 'pack' commmand compresses lines into multiple statement lines up to the maximum length you specify while maintaining complete program logic. This can easily reduce the memory requirement by more than 33%! As you can imagine this also speeds up execution of a program, saves time in loading a program from either tape or disk and saves disk space.

With the 'renumb' command you can renumber your BASIC lines. And the 'move' command allows you to move any section of your program to a new location.

So if your programs need more memory, or you need more time, order your 'packer'! 16k, 32k and 48k versions supplied on two cassettes for $29.95.

CREDIT CARD CALLERS MAY PHONE US 24-HOURS A DAY.
OR CLIP THIS COUPON AND MAIL YOUR ORDER TODAY!

THE PROGRAM STORE
4200 Wisconsin Ave NW
PO Box 9609 Dept KR5
Washington, D.C. 20016

☐ YES please send me these TRS-80 programs:

title	price
postage:	$ 1.00
total:	

name: _____

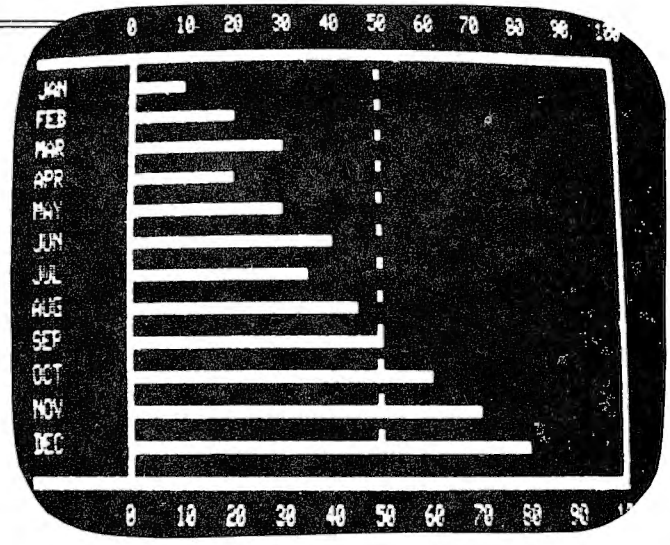
address: _____

city, state & code _____

☐Check payable to The Program Store
☐MASTERCHARGE mc bank code:
☐VISA exp date:
card number:
signature:

TWO Graph Plotting Routines

Rod Hallen/Truman Krumholz



FOR 4K LEVEL II and UP...

Displaying data on the video screen as a bunch of names and numbers can get very confusing at times. On the other hand, presenting that same data in the form of a graph usually helps to clear up that confusion. Graph Plotter was designed to allow the display of a graph containing up to 12 variables along with the names of each.

As it is written in the listing, Graph Plotter will give a brief description of it's operation, ask for a scale value, and then ask for the name and value of each variable. When all twelve items or 0,0 has been entered, a horizontal graph of the items will be displayed.

The scale value will be printed at the top and bottom of the graph and a vertical dotted line will divide the graph in half so that the comparative value of each variable is easily seen.

While Graph Plotter as written is intended for direct input, it can be modified for use in a variety of plotting situations. In order to use it with another program, you would want to set up the scale value divided by 100 in the variable S (S = Scale Value / 100), set up the names of the items in the string array N$(0), N$(1), N$(2), ... N$(11), and set up the values in a numeric array X(0), X(1), X(2), ... X(11). After that was done, a GOTO 380 would draw the graph.

While this is a relatively simple program, it does show what can be done with the TRS-80 graphics characters. Plotting titles, game layouts, and other visual displays will help to make your programs look more professional.

Program 1

```
100 REM *GRAPH PLOTTER  (C) COPYRIGHT 1979*
110 REM *BY ROD HALLEN  BOX 73  TOMBSTONE, AZ 85638*
120 CLS:CLEAR200:DIMN$(12),X(12):PRINTTAB(24)"GRAPH PLOTTER":PR
    INT
130 PRINTTAB(21)"(C)  COPYRIGHT 1979":PRINT:PRINTTAB(24)"BY ROD
     HALLEN"
140 PRINT : PRINT
150 PRINT"YOU CAN ENTER UP TO 12 DIFFERENT  NAMES AND ITEMS.  F
    OR INSTANCE";
160 PRINT"IF YOU WERE PLOTTING  YOUR USE OF ELECTRICITY OVER TH
    E PERIOD OF";
170 PRINT"A YEAR,  ITEM #1'S NAME  WOULD BE JANUARY,  ITEM #2'S
     NAME WOULD";
180 PRINT"BE FEBRUARY,  ETC.  tHE VALUE FOR EACH ITEM WOULD BE
     THE NUMBER";
190 PRINT"OF KILOWATT  HOURS  USED  THAT MONTH.  THE ITEM NAME
     AND VALUE";
200 PRINT"SHOULD BE ENTERED CONSECUTIVELY SEPARATED BY A COMMA.
    ":PRINT
210 PRINT:INPUT"PRESS ENTER TO CONTINUE ";A$:CLS:PRINT:PRINT
220 PRINT"mAXIMUM  SCALE  VALUE  IS A NUMBER  LARGER THAN  THE
    LARGEST"
230 PRINT"VALUE TO BE ENTERED.  fOR INSTANCE, IF YOU WERE PLOTT
    ING THE"
240 PRINT"INDIVIDUAL  WEIGHTS  OF A GROUP  OF PEOPLE  AND THE H
    EAVIEST"
250 PRINT"WEIGHED 240 POUNDS, mAXIMUM SCALE VALUE SHOULD BE SET
     AT 250."
260 PRINT"SCALE VALUES FROM 1 TO 999 CAN BE USED BUT EVEN 50S-
    SUCH AS"
270 PRINT"50, 100, ETC. ARE BETTER.  IF YOU HAVE LESS THAN 12 I
    TEMS TO"
280 PRINT"ENTER,  0,0 WILL END INPUT."
290 PRINT:PRINT"USE YOUR BREAK KEY TO EXIT THE GRAPH."
300 REM *INPUT SCALE, NAMES, AND VALUES*
310 PRINT:INPUT"MAXIMUM SCALE VALUE ";S:S=S/100
320 CLS:PRINT:FORZ=0TO11
330 PRINT"ITEM #";Z+1;:INPUT"- NAME, VALUE ";N$(Z),X(Z)
340 IFLEN(N$(Z))>9THENPRINT"NAMES LIMITED TO 9 CHARACTERS ":GOT
    O330
350 IFX(Z)>100*STHENPRINT"VALUES LIMITED TO A MAXIMUM OF";100*S
    :GOTO330
360 IFN$(Z)="0"THENZ=Z-1:GOTO380
370 NEXTZ
380 REM *DRAW OUTLINE*
390 CLS
400 FORI=0TO120
410 SET(I,4)
420 SET(I,43)
430 NEXTI
440 FORI=6TO41STEP3
450 SET(69,I)
460 NEXTI
470 FORI=4TO42
480 SET(120,I)
490 SET(19,I)
500 NEXTI
510 REM *PRINT SCALE & NAMES*
520 FORI=0TO50STEP5
530 PRINT@I+08,INT(20*I*S)/10;
540 PRINT@I+968,INT(20*I*S)/10;
550 NEXTI
560 FORI=0TOZ
570 PRINT@128+I*64,N$(I);
580 NEXTI
590 REM *PLOT GRAPH LINES*
600 FORJ=0TOZ
610 FORI=1TOX(J)/S
620 IFX(J)=0THEN640
630 SET(19+I,7+J*3)
640 NEXTI
650 NEXTJ
660 GOTO660
```

Program 2

```
10 REM: GRAPH - BY T. KRUMHOLZ -
20 CLEAR100:DIMA(18):DIMB(18):CLS:PRINT" *  BAR GRAPH  *"
30 PRINT:PRINT"THIS PROGRAM CREATES A BAR GRAPH."
40 PRINT:PRINT"THE MAXIMUM NUMBER OF POINTS ALLOWED IS 18."
50 PRINT:PRINT"THE POINTS MUST BE BETWEEN 0 AND 999999."
60 PRINT:INPUT"SUBJECT OF THE GRAPH";X$:IFLEN(X$)>38PRINT:PRINT
   "TITLE TOO LONG.":GOTO60
70 PRINT:INPUT"NUMBER OF POINTS TO BE PLOTTED";K:IFK>18PRINT:PR
   INT"18 IS THE LIMIT.":GOTO70
80 PRINT:FORG=1TOK:PRINT"ENTER POINT NUMBER";G;:INPUTA(G):IFA(G
   )>999999PRINT:PRINT"TOO LARGE":GOTO80
90 B(G)=A(G):NEXT:CLS:FORI=1TOK-1:FORJ=I+1TOK:IFB(J)>=B(I)THEN1
   10
100 B1=B(I):B(I)=B(J):B(J)=B1
110 NEXTJ,I:X1=INT(B(K)):X2=0:X3=X1:IFX3<=10THENX3=10:GOTO140
120 FORM=1TO5:IFX3<10↑(M+1)THENX3=INT((X3+10↑M)/10↑M)*10↑M:GOTO
   140
130 NEXT
140 X3=INT(X3):X4=X3/10
150 PRINT@8,"----- C O M P U T E R I Z E D  B A R  G R A P H
   -----";
160 FORX=16TO125:Y=35:SET(X,Y):SET(X,Y-32):NEXT
170 X=16:FORY=3TO35:SET(X,Y):SET(X+109,Y):NEXT
180 F=64:FORX=X3TO0STEP-X4:PRINT@F,X;:PRINT@F+7,CHR$(95);:PRINT
   @F+63,CHR$(95);:F=F+64:NEXT
190 E=9:X=767:FORY=1TOK:PRINT@X+E,Y;"  ";:E=E+3:NEXT
200 A=19:FORG=1TOK:Z=A(G):Z=INT(Z/(X3/30)):Z=35-Z
210 FORX=35TOZSTEP-1:Y=A:SET(Y-1,X):SET(Y,X):SET(Y+1,X):NEXTX
220 A=A+6:NEXTG:PRINT@842,"GRAPH SUBJECT:  ";X$
230 PRINT:INPUT"  PRESS ENTER TO PLOT ANOTHER GRAPH";Z$:CLS:GOT
   O60
```

Another Way To Do It

Here is yet another way to create a graph. This one, by Truman Krumholz, draws the graph vertically. It does not require setting of a maximum value, and up to 18 items may be graphed. As with the first graph, this too may be used as a subroutine in a larger program.

Omikron transforms TRS-80* into a powerful business system.

STANDARD DRIVES 8" Drives give you 5 times the speed and 3 times the storage of your mini drives! Our system provides a standard Shugart interface so you can use either your 8" drives or ours.

SOFTWARE CP/M* is the most popular operating system for microcomputers. But many high-level languages and advanced business programs cannot run with the special CP/M* designed exclusively for the TRS-80*. The Omikron MAPPER with standard CP/M* allows you to expand your software capability to go beyond the few available TRS-80 compatible packages. TRS-80* with Mapper outperforms systems costing $1000 or more.

The MAPPER I and MAPPER II are plug-in modules. They don't require any circuit changes, are easy to install, and they don't interfere with the normal operation of your TRS-80*. All your original software will still run properly. Omikron products require a minimum of 16K memory and the TRS-80* Expansion Interface.

★ ★ ★

MAPPER I is a memory management unit which adapts your TRS-80* to run standard CP/M*. The user can choose either CP/M* or TRS-80* DOS through keyboard control. The package includes CP/M* software on 5" diskette and documentation. Specify memory size when ordering. $199.

MAPPER II is a disk adapter module which enables the TRS-80* to run both 5" and 8" drives. It will interface to the MAPPER I for CP/M* operation, or can be used alone with our modified TRS-80* DOS software. Files can be transferred between the different size drives. Specify cable requirements when ordering. $99, plus $10 per cable connector.

SYSTEMS—Omikron's complete systems feature Shugart 8" drives mounted in a dual drive cabinet with heavy duty power supply, MAPPERS I and II, cable and CP/M* software. Dual drives—$1795; Single drive $1195.

WARRANTY—6 months parts and labor. Satisfaction guaranteed. Dealer inquiries invited.

Call for details on Omikron's TRS-80* DOS package.

*CP/M is a TM of Digital Research. TRS-80 is a TM of Tandy Corporation

INVADERS

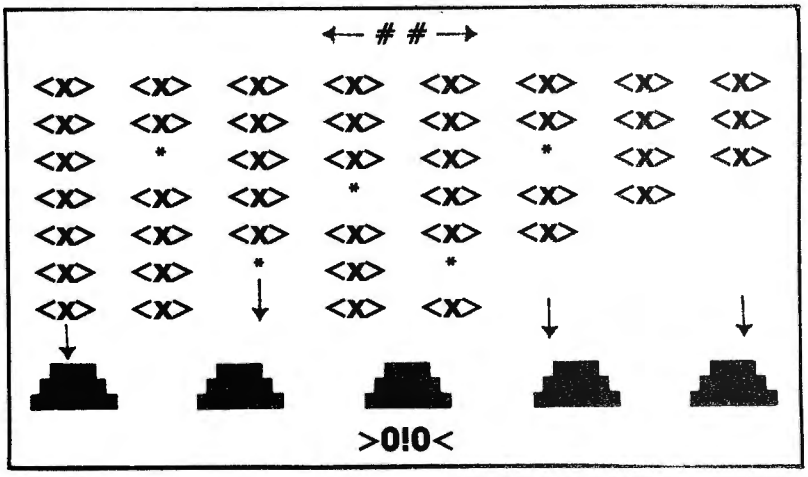
"Alien **INVADERS** from deep space are now approaching our planet in great swarms to colonize on Earth. The invaders feel they must destroy all life forms . . ." With SOUND!

This Machine Language Arcade Game Will be the Best in Your Library!

5 Levels of Play

Will You Survive?

ONLY $14.95 Cassette



from
LEVEL IV PRODUCTS, INC

32238 Schoolcraft
Livonia, MI 48154

Michigan (313) 525-6200
Outside Michigan 1-800-521-3305

Hours: Tues through Sat 7 AM to 11 PM

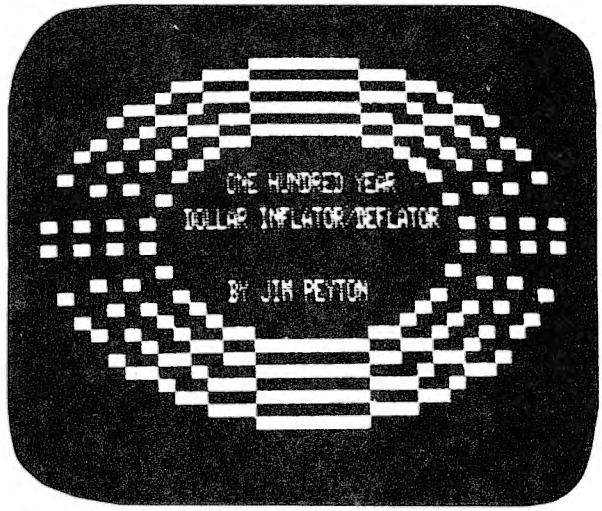
MASTER CHARGE/VISA ACCEPTED

100 YEAR *Dollar Inflator/Deflator*

Jim Peyton,

Georgetown, KY

$$$$$$$$$$$$$$$$$$$$



This program compares dollar values and prices between any two years of the century spanning 1881 to 1980. Input is through INKEYs. The program accepts dollar amounts from $1 to $999,999,999.00. Years or dollar amounts outside these ranges return an error message and allow the user to redo.

Upon input of years and amounts, a dollar value comparison is returned in the form of a two bar graph, and the user is given a choice between comparing price differences for those years or comparing other years. If the user chooses to compare price differences, and if the years are from 1950 to 1980 (good basic conditional), he is presented with a menu allowing selection from (1) food, (2) housing, (3) utilities, (4) automobiles, (5) medical or (6) total costs. Otherwise, he is presented with a total cost comparison. (No break out of item costs was available for the years prior to 1950).

Data Source: The data used as a basis for the program is the Consumer Price Index. The sources used were *Historical Statistics of the United States: Colonial Times to 1970, p 210-211, United States Department of Commerce, Bureau of the Census, US Printing Office, 1975;* and *Statistical Abstracts of 1978, p 490, Bureau of Labor Statistics, US Printing Office, 1978.* The information used is in the public domain. Data for housing costs in the years 1950-1952 was not available and had to be estimated as did the entire Consumer Price Index for 1980. Consumer Price Index data for the year 1979 are estimated based on the first seven months and amount overall to an 11.1 percent increase over the previous year. The estimate for 1980 is a 9.2 percent increase. Both of these reflect the Goldman and Sacks Econometric Forecast.

FOR 16K Level II and UP..

```
30          BY: JIM PEYTON
                ROUTE 2
                GEORGETOWN, KY 40324
                APRIL 1979
                10.1K
40 '
50 CLS:FORR=10TO17STEP2:FORA=-RTOR:X=R*R-A*A:Y=SQR(X):Y=INT(Y-.
   5)
60 SET((A*3)+60,23+Y):SET((A*3)+61,23+Y):SET((A*3)+62,23+Y)
70 SET((A*3)+60,23-Y):SET((A*3)+61,23-Y):SET((A*3)+62,23-Y):NEX
   TA,R
80 GOSUB210:FORR=1TOLEN(T1$):A$=LEFT$(T1$,R):PRINT@415-LEN(T1$)
   /2,A$;:NEXT
90 GOSUB220:FORR=1TOLEN(T2$):A$=LEFT$(T2$,R):PRINT@479-LEN(T2$)
   /2,A$;:NEXT
100 FORR=1TO750:NEXTR
110 CLEAR600
120 PRINT@531,"NEED INSTRUCTIONS? (Y/N)";
130 Z$=INKEY$:IFZ$=""THEN130ELSEIFZ$="N"THEN230ELSEPRINT@531,ST
   RING$(24,32);
140 X=0
150 P=598:N=0
160 X=X+1:IFX>3GOTO230
170 ONXGOSUB180,190,200:FORT=1TOLEN(A$):N=N+1:B$=MID$(A$,N,18):
   PRINT@P,B$;:FORR=1TO50:NEXTR,T:GOTO150
180 A$=STRING$(17,32)+"... THIS PROGRAM WAS DEVELOPED WITH MUCH
    DIFFICULTY YET WITH GREAT JOY AND SERENITY BY JIM PEYTON OF GEO
   RGETOWN, KENTUCKY ... I HOPE YOU FIND IT WORTHWHILE  ... ":RETUR
   N
190 A$=STRING$(17,32)+"... THE PROGRAM CONVERTS DOLLAR VALUES A
   ND PRICES FROM ONE YEAR TO ANOTHER BASED ON THE CONSUMER PRICE I
   NDEX.  VALID YEARS ARE FROM 1881 THROUGH 1980 ... ":RETURN
200 A$=STRING$(17,32)+"... JUST TYPE IN THE YEARS AND AMOUNTS A
   S DIRECTED  (NO NEED TO PRESS ENTER)  AND WAIT A BIT.  YOU MAY B
   E BOTH ENLIGHTENED AND DISMAYED AT THE RESULTS ... ":RETURN
210 T1$="ONE HUNDRED YEAR":RETURN
220 T2$="DOLLAR INFLATOR/DEFLATOR":RETURN
230 CLEAR100:DIML(100,7)
240 RESTORE:FORI=1TO100:IFI<32THENFORJ=1TO7ELSEFORJ=1TO2
250 READL(I,J):NEXTJ,I
260 CLS:GOSUB470:PRINT@400,CHR$(188)STRING$(32,140)CHR$(188);:P
   RINT@462,STRING$(2,143)CHR$(191)CHR$(32)STRING$(30,143)CHR$(32)C
   HR$(191)STRING$(2,143);:PRINT@528,STRING$(34,131);:A$="":B$="":C
   $=""
270 PRINT@603,"$$$$$$$$$";:GOSUB960:FORI=1TO4
280 Z$=INKEY$:IFZ$=""THEN280ELSEIFASC(Z$)<48ORASC(Z$)>57THEN280
   ELSEA$=A$+Z$:PRINT@465," ";A$;:NEXT:A=VAL(A$):PRINT@470," <<<<<<
   <<<<>>>>>>>>>> ";"      ";:GOSUB580:GOSUB970:FORI=1TO4
290 Z$=INKEY$:IFZ$=""THEN290ELSEIFASC(Z$)<48ORASC(Z$)>57THEN290
   ELSEB$=B$+Z$:PRINT@492,B$;" ";:NEXT:B=VAL(B$):GOSUB590:GOSUB980:
   FORI=1TO10
300 Z$=INKEY$:IFZ$=""THEN300ELSEIFZ$="."THENI=10ELSEIFASC(Z$)<4
   8ORASC(Z$)>57THEN300ELSEC$=C$+Z$:PRINT@604," ";C$;"      ":NEXT
310 C=VAL(C$):GOSUB600
320 C$=C$+Z$:PRINT@604," ";C$;"       ";:NEXT:C=VAL(C$)
330 IFA<BTHENN=(1980-A)+1ELSEN=(1980-B)+1
```

```
          1 0 0   Y E A R
      DOLLAR INFLATOR / DEFLATOR
           1881 - 1980


 VALUE IN
YEAR 1950 ..............................     $100.00
YEAR 1980 .........       $30.46


   TO COMPARE COST DIFFERENCES.......TYPE 1
   TO COMPARE OTHER YEARS............TYPE 2



          SELECTED  COST  ITEMS

     FOOD COSTS..................TYPE 1
     HOUSING COSTS...............TYPE 2
     UTILITY COSTS...............TYPE 3
     AUTO COSTS..................TYPE 4
     MEDICAL COSTS...............TYPE 5
     TOTAL COSTS.................TYPE 6
```

```
          1 0 0   Y E A R
      DOLLAR INFLATOR / DEFLATOR
           1881 - 1980
              -3-
          U T I L I T I E S
          - - - - - - - - -
 COST IN
YEAR 1950 .........       $100.00
YEAR 1980 ..............................        $319.83


   TO COMPARE COST DIFFERENCES.......TYPE 1
   TO COMPARE OTHER YEARS............TYPE 2
```

```
340 FORI=1TON
350 IFL(I,1)=A,D=L(I,2)
360 IFL(I,1)=B,E=L(I,2)
370 NEXTI
380 F=1:G=D/E:H=60:GOTO410
390 G=E/D:CLS:GOSUB470:PRINT@394,"COST IN";:PRINT@457,"YEAR";A:
    PRINT@521,"YEAR";B:GOTO420
400 CLS:GOSUB470:PRINT@394,"COST IN";:PRINT@457,"YEAR";A:PRINT@
    521,"YEAR";B:ONXGOSUB750,790,830,870,910,950:GOTO420
410 CLS:GOSUB470:PRINT@394,"VALUE IN";:PRINT@457,"YEAR";A:PRINT
    @521,"YEAR";B:GOTO420
420 AA$="$$###.#":AB$="$$###,###.##"
430 IFF>GTHEN500ELSE540
440 FORI=1TO500:NEXT:GOSUB480:PRINT@780,D$:PRINT@844,E$
450 Z$=INKEY$:IFZ$="1"ANDN>31THEN390ELSEIFZ$="1"THEN630ELSEIFZ$
    ="2"THEN260ELSE450
460 GOTO460
470 PRINT@24,"1 0 0   Y E A R";:PRINT@83,"DOLLAR INFLATOR / DEF
    LATOR";:PRINT@154,"1881 - 1980";:RETURN
480 D$="TO COMPARE COST DIFFERENCES.......TYPE 1"
490 E$="TO COMPARE OTHER YEARS...........TYPE 2":RETURN
500 Y=21:FORX=38TO97:SET(X,Y):SET(X,Y+1):NEXT:IF(F*C)>999999PRI
    NT@497,USINGAA$;(F*C*.000001);:PRINT" MIL";:GOTO520
510 PRINT@497,USINGAB$;F*C;
520 Y=24:FORX=38TO97-H*(F-G):SET(X,Y):SET(X,Y+1):NEXT:IFG*C>999
    999PRINT@562-(H/2)*(F-G),USINGAA$;G*C*.000001;:PRINT" MIL";:GOTO
    440
530 PRINT@562-(H/2)*(F-G),USINGAB$;G*C;:GOTO440
540 Y=21:FORX=38TO97-H*(1-F/G):SET(X,Y):SET(X,Y+1):NEXT:IFF*C>9
    99999PRINT@498-(H/2)*(1-F/G),USINGAA$;F*C*.000001;:PRINT" MIL";:
    GOTO560
550 PRINT@498-(H/2)*(1-F/G),USINGAB$;F*C;
560 Y=24:FORX=38TO97:SET(X,Y):SET(X,Y+1):NEXT:IFG*C>999999PRINT
    @564,USINGAA$;G*C*.000001;:PRINT" MIL";:GOTO440
570 PRINT@564,USINGAB$;G*C;:GOTO440
580 IFA<1881ORA>1980THEN610ELSERETURN
590 IFB<1881ORB>1980THEN610ELSERETURN
600 IFC<1ORC>1000000000THEN610ELSERETURN
610 CLS:PRINTCHR$(23):PRINT@266,"YEAR OR AMOUNT";:PRINT@336,"OU
    T OF RANGE
620 PRINT@406,"PLEASE REDO";:FORR=1TO1500:NEXT:GOTO260
630 CLS:PRINT:PRINT:PRINTTAB(21)"SELECTED  COST  ITEMS"
640 PRINT:PRINTTAB(14)"FOOD COSTS.................TYPE 1
650 PRINTTAB(14)"HOUSING COSTS.............TYPE 2
660 PRINTTAB(14)"UTILITY COSTS.............TYPE 3
670 PRINTTAB(14)"AUTO COSTS................TYPE 4
680 PRINTTAB(14)"MEDICAL COSTS.............TYPE 5
690 PRINTTAB(14)"TOTAL COSTS...............TYPE 6
700 X$=INKEY$:X=VAL(X$):ONXGOTO720,760,800,840,880,920
710 GOTO700
720 FORI=1TON:IFL(I,1)=A,D3=L(I,3)
730 IFL(I,1)=B,E3=L(I,3)
740 NEXTI:F=1:G=E3/D3:H=60:GOTO400
750 PRINT@222,"-1-";:PRINT@284,"F O O D";:PRINT@348,"- - - -";:
    RETURN
760 FORI=1TON:IFL(I,1)=A,D4=L(I,4)
770 IFL(I,1)=B,E4=L(I,4)
780 NEXTI:F=1:G=E4/D4:H=60:GOTO400
790 PRINT@222,"-2-";:PRINT@281,"H O U S I N G";:PRINT@345,"- -
    - - - - -";:RETURN
800 FORI=1TON:IFL(I,1)=A,D5=L(I,5)
810 IFL(I,1)=B,E5=L(I,5)
820 NEXTI:F=1:G=E5/D5:H=60:GOTO400
830 PRINT@222,"-3-";:PRINT@279,"U T I L I T I E S";:PRINT@343,"
    - - - - - - - - -";:RETURN
840 FORI=1TON:IFL(I,1)=A,D6=L(I,6)
850 IFL(I,1)=B,E6=L(I,6)
860 NEXTI:F=1:G=E6/D6:H=60:GOTO400
870 PRINT@222,"-4-";:PRINT@276,"A U T O M O B I L E S";:PRINT@3
    40,"- - - - - - - - - - -";:RETURN
880 FORI=1TON:IFL(I,1)=A,D7=L(I,7)
890 IFL(I,1)=B,E7=L(I,7)
900 NEXTI:F=1:G=E7/D7:H=60:GOTO400
910 PRINT@222,"-5-";:PRINT@281,"M E D I C A L";:PRINT@345,"- -
    - - - - -";:RETURN
920 FORI=1TON:IFL(I,1)=A,D=L(I,2)
930 IFL(I,1)=B,E=L(I,2)
940 NEXTI:F=1:G=E/D:H=60:GOTO400
950 PRINT@222,"-6-";:PRINT@277,"T O T A L   C O S T S";:PRINT@3
    41,"- - - - -  - - - - -";:RETURN
960 FORI=1TO50:NEXT:PRINT@780,">  TYPE IN FIRST OF YEARS TO BE
    COMPARED":RETURN
970 FORI=1TO200:NEXT:PRINT@844,">  TYPE IN SECOND YEAR":RETURN
980 FORI=1TO200:NEXT:PRINT@908,">  TYPE IN DOLLAR AMOUNT  (END
    WITH A PERIOD)":RETURN
990 RETURN
1000 DATA1980,2367,2574,2461,2597,2314,2599,1979,2168,2357,2254
    ,2378,2119,2380,1978,1954,2114,2104,2160,1855,2194,1977,1815,192
    2,2049,2134,1766,2024,1976,1705,1808,1917,1890,1646,1847,1975,16
    12,1754,1817,1696,1498,1686
1010 DATA1974,1477,1617,1632,1458,1366,1505,1973,1331,1414,1467
    ,1264,1215,1377,1972,1253,1235,1401,1205,1175,1325,1971,1213,118
    4,1337,1147,1166,1284,1970,1163,1149,1285,1073,1111,1206
1020 DATA1969,1098,1089,1160,1028,1065,1134,1968,1042,1036,1057
    ,1009,1030,1061,1967,1000,1000,1000,1000,1000,1000,1966,972,991,
    963,996,975,934,1965,945,944,927,994,963,895,1964,929,924,908,99
    4,947,873
1030 DATA1963,917,912,890,994,934,856,1962,906,899,879,994,930,
    835,1961,896,891,869,994,913,814,1960,887,880,863,986,906,791,19
    59,873,871,844,947,911,764,1958,866,885,835,924,874,732,1957,843
    ,849,817,893,847,699
1040 DATA1956,814,822,783,884,801,672,1955,802,816,770,875,789,
    648,1954,805,828,763,853,803,634,1953,801,830,750,842,824,614,19
    52,795,843,740,826,808,593,1951,778,828,730,815,758,563,1950,721
    ,745,725,812,725,537
1050 DATA1949,714,1948,721,1947,669,1946,585,1945,539,1944,527,
    1943,518,1942,488,1941,441
1060 DATA1940,420,1939,416,1938,422,1937,430,1936,415,1935,411,
    1934,401,1933,388,1932,409,1931,456,1930,500,1929,513,1928,513,1
    927,520,1926,530,1925,525,1924,512,1923,511,1922,502,1921,536
1070 DATA1920,600,1919,518,1918,451,1917,384,1916,327,1915,304,
    1914,301,1913,297,1912,290,1911,290,1910,280,1909,270,1908,270,1
    907,280,1906,270,1905,270,1904,270,1903,270,1902,260,1901,250
1080 DATA1900,250,1899,250,1898,250,1897,250,1896,250,1895,250,
    1894,260,1893,270,1892,270,1891,270,1890,270,1889,270,1888,270,1
    887,270,1886,270,1885,270,1884,270,1883,280,1882,290,1881,290
```

TSE "TRS-80 USERS! Your Software Supermarket"

GAMES

Warfare I 4-game anthology, Level II, 16K $7.95

Backgammon by Scott Adams.Level II, 16K $7.95

X-Wing II by Chris Freund. Level II, 16K $9.95

Taipan by Art Canfil. Level II, 16K $9.95

Sargon Chess by Dan and Kathe Spracklen. Level II, 16K $19.95

Chess Companion by Michael Kelleher. Level II, 16K $7.95

Three D Tic Tac Toe by Scott Adams. Level II, 16K $7.95

Concentration by Lance Micklus. Level II, 16K $7.95

Amazin' Mazes by Robert Wallace. Level II, 16K $7.95

Time Bomb by David Bohlke. Level II, 16K $4.95

Life Two by Leo Christopherson. With sound -Level II, 16K $14.95

Android Nim by Leo Christopherson. With sound - Level II, 16K $14.95

Cubes by Leo Christopherson Level II, 16K $9.95

Mastermind II by Lance Micklus $7.95

Mastermind II - Source List -$20.00

Robot/Breakaway Game duo by Lance Micklus. Level II, 4K $7.95

Tycoon by David Bohlke. Level II, 16K $7.95

Slalom by Denslo Hamlin. level II, 16K $7.95

9 Games for Preschool Children by George Blank. Level II, 16K $9.95

Ten Pin by Frank B. Rowlett, Jr. Level II, 16K $7.95

Atlantic Balloon Crossing by Dean Powell. Level II, 16K $9.95

Space Battles by Level IV, Level II, 16K Tape or 32K Disk, Tape -$14.95, Disk - $19.95

Star Trek III.3 by Lance Micklus Level II, 16K $14.95

Dog Star Adventure by Lance Micklus. Level II, 16K $9.95

Safari by David Bohlke. Level II, 16K $7.95

Treasure Hunt by Lance Micklus. Level II, 16K $7.95

'Round the Horn by George Blank. Level II, 16K $9.95

Pork Barrel by George Blank. Level II, 16K $7.95

Kamikaze by Russell Starkey. Level II, 16K $7.95

All Star Baseball by David Bohlke. Level II, 16K $7.95

Barricade by Small Systems Software. Machine Language -$14.95

Journey To The Center Of The Earth by Greg Hassett. Level II, 16K Tape $7.95

Pentominoes by James Garon. Level II, 16K $7.95

Snake Eggs by Leo Christopherson. With sound Level II, 16K $14.95

Now In Stock!

SARGON was the BEST
SARGON II is MUCH BETTER!

Seven levels of play . . .
faster, better end game . . .
randomized opening.

$29.95

NEW!!!

Understandable! Indexed library with 200 Assembly Language Routines.

Z80 Software Gourmet Guide And Cookbook

from Scelbi. $14.95 + $1 postage.

BOOKS

Sargon Handbook by Don & Kathe Spracklen. $14.95 plus $1.00 shipping and handling.

The Basic Handbook by Dr. David A. Lien. $14.95 plus $1

Z80 Instruction Handbook by Scelbi Publications. $4.95

The Little Book Of BASIC Style by John Nevison. $5.95 plus $1

TRS-80 Assembly Language Programming by William Barden, Jr. $3.95 plus $1

Introduction to TRS-80 Graphics by Don Inman. $7.95 plus $1

Learning Level II by David A. Lien. $15.95 plus $1

SPECIAL PURPOSE

Calculator by R. W. Robitaille, Sr. Level II, 4K $2.95

Moving Signboard by Circle Enterprises. Level II. 4K $9.95

Histograph/Scattergram by Gary S. Breschini. Level II, 16K $9.95.

Simple Simon by George Blank. Level II. Written in BASIC. $4.95

Math Drill by K. L. Brown. Level II, 16K $4.95

RPN Calculator by Russell Starkey. Level II, 16K $9.95

Ham Radio by Michael Kelleher. Level II, 16K $9.95

Ham Radio ARS I.1 (32K disk) $24.95

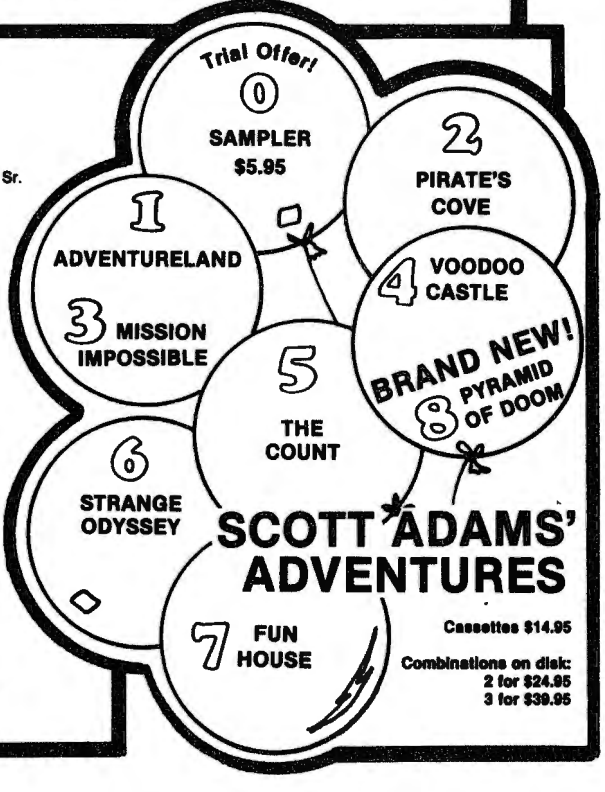
Electronics Assistant by John Adamson. Level II, 16K $9.95

Preflight by Stephen Hebbler. Level II, 16K $20.00

Basic Statistics by Steve Reisser. Level II, 16K $20.00

Drill Masters by Computer Graphics - specify title desired. Level II, 16K $7.95 each. German, Russian, Italian, Spanish, or Music Theory

Keyboard-80 by John Adamson. Level II, 16K $9.95



ST·80D
by Lance Micklus

The STANDARD
in Smart Terminal
Programs for the TRS-80

(32K Disk)

$79.95

IMPORTANT

No sales tax •

All C.O.D.'s or special • delivery orders are a minimum of $5 for special handling

When ordering Percom • please add $5 each, packaging and handling fee.

PRICES DO NOT • INCLUDE SHIPPING

BUSINESS

Inventory II.2 by M. Kelleher and R. W. Robitaille, Sr. 16K disk systems $59.95

Inventory System II.3 by M. Kelleher. Improved version, $79.95

Inventory 'S' by Roger W. Robitaille, Sr. Level II, 16K Tape -$24.95; 32K Disk -$39.95

Payroll by Stephen Hebbler. For disk systems. 32K $59.95

Accounts Receivable II by S. Hebbler. 32K disk systems $79.95

Appointment Log by Michael Kelleher. Level II, 16K $9.95 Disk version, $19.95

General Ledger I by M. Kelleher, requires 32K Disk, $79.95

Mail List II by R. W. Robitaille, Sr. 32K disk systems $99.95

Small Business Bookkeeping by R. W. Robitaille, Sr. Level II, 16K. With journal -$22.00; Without journal - $15.00

UTILITIES

NEWDOS by Apparat $49.95

NEWDOS+ by Apparat $99.95

Machine Language Monitor by Small Systems Software. Level II, 16K $26.95

Three Monitors for Disk by Small Systems Software. Disk for 16 through 48K (all in one) $29.95

KVP Extender by Lance Micklus. Tape - $29.95; Disk - $34.95

KVP 232 by Lance Micklus - KVP adapted for the TRS-232. Tape - $29.95

ST80 Smart Terminal Level II, 16K $49.95

ST80D Smarter Terminal for disk systems. $79.95

Micro Text Editor by Don Coons. Level II, 4K or 16K $9.95

Text-80 by Frank B. Rowlett, Jr. For 32K disk systems $59.95

8080-Z80 Conversion Level II, 16K $15.00

Renumber by Lance Micklus. Level II, available in 16 through 48K (specify when ordering) $7.95

Renumber source listing $20.00

Electric Pencil by Michael Shrayer. Powerful machine language word processing system. Level II, 16K tape - $100; Disk version - $150

Level III BASIC by Microsoft. $49.95

Level I in Level II by Apparat. Level II, 16K $15.00

Fortran by Microsoft. 32K - 2 Disks. New low price $195.00

PERSONAL

Typing Tutor by Roger W. Robitaille, Sr. Level II, 16K $19.95

Secrets of the Tarot by John T. Phillipp. Level II, 16K $9.95

Biorhythms by Frank B. Rowlett, Jr. Level II, 4K $4.95

Personal Finance by Lance Micklus. Level II, 16K $9.95

Advanced Personal Finance by Lance Micklus for 32K disk systems $24.95

Home Financial Management by Michael Kelleher. Level II, 16K $9.95

HARDWARE ACCESSORIES

Cassettes boxes of ten each. C-10 - $6.50 plus $1.00 shipping
C-20 - $7.50 plus $1.00 shipping

Diskettes Dysan, (premium quality) box of 5 -$24.95 plus $1.00 shipping; nationally known brand, box of 10 - $34.95 plus $1

Diskette Storage Box $5.00

Floppy Armour™ Protective envelope for shipping floppy disks, 5-pack - $4.95 plus $1.00 shipping and handling

NEWDOS

Eliminate 90% of the hassle of a disk system by replacing your TRS-DOS with NEWDOS! faster, more reliable, many more features $49.95.

or

Add Superzap, Directory checks, other utilities. NEWDOS+ $99.95

PACKAGE PRICE $150

SEPARATELY:
MACRO ASSEMBLER $80.
FORTRAN $80.

HARDSIDE™

WANTED

Used TRS-80 equipment! We buy and sell used equipment. Call or write for details.

SPECIAL

$669. TRS-80
16K, LEVEL II
NO KEYPAD.

SAVE $167.

TRS-80 expansion interface with our 16K RAM, single PERCOM disk drives with cable, and NEWDOS operating system. $830.

MAGAZINES

SoftSide(monthly)

$18. 1 yr.,Bulk Rate
$30. 1 yr., Overseas Airmail
$25. 1 yr., First Class or Overseas Surface
$38. 6 mos. w/cassette $68. 6 mos. w/disk

PROG/80(bi-monthly)

$15. 1 yr., Bulk Rate
$22. 1 yr.,First Class or Overseas Surface
$27. 1 yr. Overseas Airmail

SoftSide's AppleSeed(monthly)

$15. 1 yr., Bulk Rate
$22. 1 yr., First Class

FREE!!

For more detailed descriptions of our software and accessories, send for the "TSE" catalog...it's FREE!

TRS 232 by Small Systems Hardware - $49.95

Percom Disk Drives Single or dual, for TRS-80's. Single drive -$399.00; Dual drive - $795.00; Cable required - $29.95

ASK ABOUT OUR FREE HARDSIDE CATALOG

TO ORDER (9AM - 5:30 PM, EST)

TOLL-FREE 1-800-258-1790

The Software Exchange

6 South Street, Box 68, Milford, NH 03055 603-673-5144

* A trademark of Radio Shack and Tandy Corp.

Anatomy of the Program

(100 Year $ Inflator/Deflator)

R C Bahn

I. SUMMARY

The program compares dollar values and prices between any two years of the century spanning 1881 to 1980. The program provides numerous examples of more complex BASIC Statements such as DATA, RESTORE, GOSUB, ON...GOSUB, INKEY$, MID$, CHR$, VAL, PRINT@,PRINTTAB and PRINT USING. There are several examples of useful video display graphics including a concentric circular design, a moving banner display, numeric data input panels and bar graphing.

II. LINE BY LINE COMMENTARY

10-40 Program identification

50-70 Generate concentric circular pattern with set statements

80-90 Print title of program by "left to right scrolling" routine without destroying prior graphics. Note use of PRINT@ and trailing semicolon.

100 Simple timing loop

110 Clear string space for "moving banner"

120 Print instruction query

130 INKEY$ loop. Note that double quotes ("") represent a 'null string', or one with a length of zero.

140-150 Initialize "moving banner" routine.

160 Increment counter (X) for the number of messages (3) defined in lines 180,190, and 200.

170 Fetch next message (A$), form series of substrings of length 18 (B$) and display at P. R is a timer.

180-200 Strings for "moving banner".

210-220 Strings for "scrolled" titles displayed in lines 80-90.

230 Initialize and dimension for remainder of program

240-250 Reset start of data read, read data, update timing display.

260-270 Routines to generate "graphic data input panels" for years and dollars. Note use of graphics symbols (128-191)

280-320 Data input routine which accepts years and dollars by INKEY$ statements, progressively modifies graphics, branches to limit checking routines (580-620) and prints prompting for sequential entry of A$, B$,and C$. Note concatenation of string variable by A$= A$+Z$ for keyboard entry.

330 Set up timing delay for intervals greater than 9 years

340-370 Find values of D and E.

380 Find inflation/deflation ratio (G).

390 Print "cost" labels to be used following statement 470.

400 Print "cost" label to be used following statements 750, 790, 830, 870, 910, and 950.

410 Print "value" label.

420 Define formats for subsequent PRINT USING statements.

430 Branch to appropriate bar graphing routine

440 Print next options on video screen.

450 Accept next data from keyboard.

470 Routine for printing heading label.

480-490 Video display of next options.

500-570 Bar graphing and row labels.

580-620 Keyboard data input error routines. Note use of CHR$(23) for large letters. This command is turned off by the CLS in line 260

630-710 Print itemized cost menu and accept selection from keyboard.

720-740 Routine for food costs. All of the following routines are constructed similarly. Initialize data read, read data, define appropriate D3 and E3, define G, return to line 400

750 Print food label.

760-780 Routine for housing costs.

790 Print housing label

800-820 Routine for utility costs.

830 Print utility label.

840-860 Routine for auto costs.

870 Print auto label

880-900 Routine for medical costs.

910 Print medical label.

920-940 Routine for total costs.

950 Print total costs label

960-980 Subroutines for display of video input prompts.

1000-1050 Data statements. All data is stored in the array L(I,J). Records for individual years are listed in descending order for the years 1980 to 1880 Every record consists of at least two numbers the year and total economic index number. In the interval 1950 to 1980.each record contains five additional numbers related to the sub category indices arranged in the order in lines 640-680.

1040-1080 Data, 1980 - 1950, 7 entries per year

1050-1120 Data 1949-1880; 2 entries per year

MICROSOFT CONSUMER PRODUCTS CONTINUING THE MICROSOFT TRADITION

Microsoft set the standard in microcomputer system software. We know more about the structure and capabilities of today's microcomputers than anyone else. And now we're using that power in a whole new way!

Announcing Microsoft Consumer Products. Distinctive software packages backed by the Microsoft name. Each is created by a top-notch programmer and comes to you fully documented, at a cost you can afford.

Microsoft Editor/Assembler-Plus.™ Now get every feature of Radio Shack's Editor/Assembler and T-Bug all in one package. PLUS—many "big computer" features to simplify your programming, editing and debugging. All in a low cost cassette package. Don't waste time creating both source and object tapes—Assembler-Plus assembles directly into memory. Supports macros and conditional assembly, too. Editor-Plus simplifies editing with extra commands like Substitute, Move, Copy and Extend. And Z-Bug,™ the most powerful debugger ever available for the TRS-80, has single step execution, direct execution in calculator mode and symbolic references. And, you can use up to 8 breakpoints at a time, with no need to remove a breakpoint before proceeding. For the 16K, Level II, cassette TRS-80. Priced at $29.95.

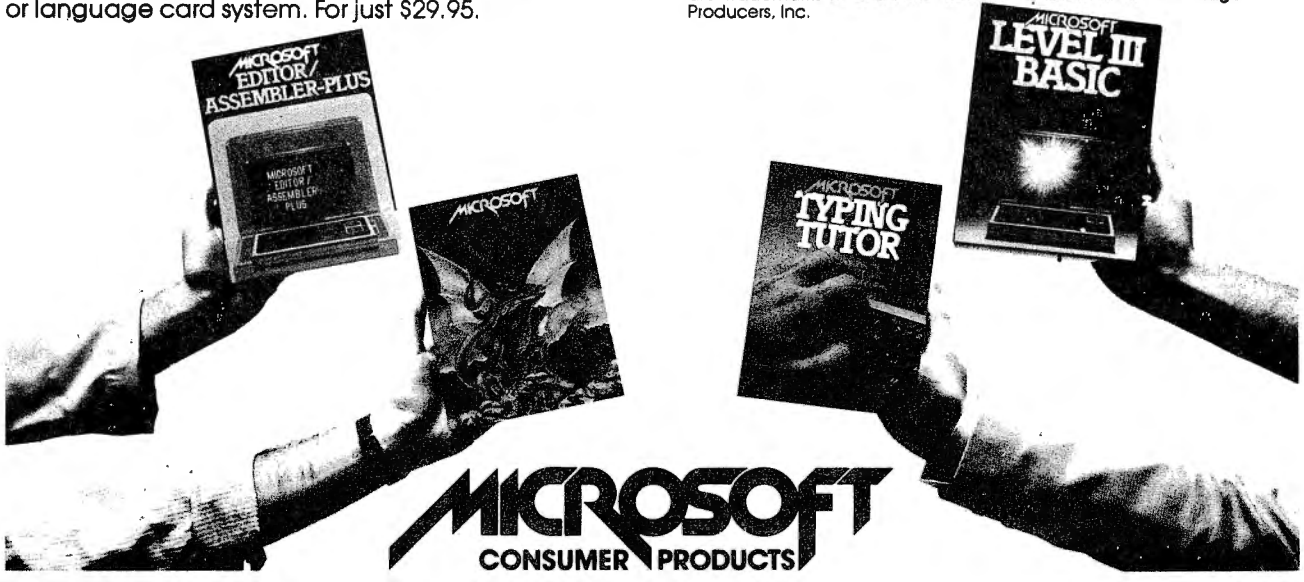
Microsoft Adventure. Only Microsoft offers Adventure complete, as originally written for the DEC PDP-10, now implemented on personal computers. The ultimate fantasy/logic game, Adventure allows you to explore the depths of the "Colossal Cave," collecting treasures and magic, solving puzzles, avoiding hazards and adversaries—including the dreaded killer dwarves. Don't be fooled by imitation or incomplete versions. Only Microsoft has it all. Adventure fills an entire disk with everything you need for your exploration. Written by Gordon Letwin, of SOFTWIN, Associates. Adventure for the TRS-80 requires a single-disk, 32K system. For the Apple II,** a single-disk, 32K system with either the standard disk or language card system. For just $29.95.

Microsoft Typing Tutor. There's no easier way to master your keyboard! Faster and more efficient than any other teaching method, Typing Tutor helps you if you're starting from scratch or simply building speed. The secret lies in Typing Tutor's exclusive TRM™ or "Time Response Monitoring" software. TRM monitors your keyboard 20 times per second so the computer can evaluate your skill. Your speed. Your errors. Your weakest keys. Typing Tutor tells you where you stand then automatically adjusts itself to help you improve. Written by Dick Ainsworth and Al Baker of the Image Producers, Inc. For the Apple II with 16K and Apple BASIC or the TRS-80 with 16K and Level II BASIC. Priced at $14.95.

Microsoft Level III BASIC. Upgrade your Level II TRS-80 and increase your programming efficiency without additional hardware. Microsoft Level III loads from cassette tape on top of the Level II ROM. It gives you every feature of Disk BASIC except disk file commands. But that's not all—Level III's high-speed graphics turn your TRS-80* into a virtual electronic drawing board. And there's program renumbering, long error messages, quick shift-key entries, time-limit INPUT statements and many more features. System requirements: Level II BASIC and 16K. Occupies 5.2K RAM. Priced at $49.95.

Where To Buy. Microsoft Consumer Products are sold by computer retailers nationwide. If your local computer store doesn't have them, call us. Phone (206) 454-1315. Or write Microsoft Consumer Products, 10800 Northeast Eighth, Suite 819, Bellevue, WA 98004.

*TRS-80 is a trademark of Radio Shack Corp. **Apple II is a trademark of Apple Computer, Inc. ™ Editor/Assembler-Plus and Z-Bug are trademarks of Microsoft. TRM is a trademark of The Image Producers, Inc.



MICROSOFT
CONSUMER PRODUCTS

A *SEARCH* PROGRAM

Using DOS Random Files, with use of Hash Codes

K W Burrowes, Millbrae, CA

Those of you familiar with the common disk-based data base management systems are aware that one of their major drawbacks is that knowledge of a record number is required to locate a particular record on the disk, or a sequential (and often lengthy) search through the file for some characteristic field must be made. This program solves this problem by allowing the user to specify a unique number of attributes in lieu of a record number, whereupon all the records having these characteristics will be displayed and printed by direct random access.

The particular application shown is for the author's personal stereo record collection, but conversion to other applications is simple and is described in more detail below. For the time being, however, we will demonstrate the program's capabilities using this example. Actually, four separate disk files are involved: File "INDEX" is automatically updated after each run of ATRISRCH/BAS is ended (ATRIbute SeaRCH is the main program), and is read again when the next run is initiated. It always contains the current status of the main data base ("RECORDS"). "INDEX" is initially created when program "DATABASE/BAS" is run, whereafter DATABASE/BAS is not needed again. The main program, ATRISRCH/BAS first reads "INDEX" and then proceeds to open the Database File "RECORDS" for use and displays the following menu on the video:

```
SELECT ONE OF THE FOLLOWING:
LOOK UP A RECORD          =1
ADD A RECORD              =2
DELETE A RECORD           =3
END                       =4
INPUT YOUR CHOICE (1, 2, 3 or 4)?
```

We will now proceed to discuss each option.

ADDING TO THE DATABASE

Assuming we are just starting, we would take option 2, whereupon the following three attribute groups would be sought:

```
GROUP 1:        CATEGORY OF MUSIC:
        CLASSICAL SOLO PRINCIPAL    =1
        CLASSICAL NON-SOLO          =2
        LIGHT BACKGROUND MUSIC      =3
        JAZZ                        =4
        WEST INDIAN MUSIC           =5
        OTHER                       =6
```

Let us suppose our first record is classical non-solo, so we enter 2. Similarly for "Music type" we are asked to select one of the following:

```
GROUP 2:        TYPE OF MUSIC:
        VOCAL                       =1
        PIANO                       =2
        VIOLIN                      =3
        GUITAR                      =4
        OTHER                       =5
```

Our record is orchestral, so we enter 5. Finally, we are questioned about record quality:

```
GROUP 3:        RECORD QUALITY:
        GOOD                        =1
        POOR                        =2
```

We enter 1, whereupon we are now asked the following specifics about the record:

```
RECORD TITLE (40 char)?
COMPOSER (15 char)?
PLAYED OR SUNG BY (20 char)?
QUALITY OF RECORD (5 char)?
```

After we answer each request and the record has been written to disk, the main menu is again displayed.

LOOK UP RECORDS

Assuming we have now repeated the above routine a number of times so that our database file "RECORDS" contains 100 entries, and we now wish to find all our records which are: Light Background Music, Played on the Guitar, with a Good Record Quality. First we select Option 1 from the main menu and then 3, 4 and 1 from the subsequent attribute questions, whereupon the first record to have these attributes is displayed on the video. At this point, the program asks if a modification of the record is desired. From now on we say "NO" or "N", and the record is printed on the line printer. This cycle continues until the last record with these attributes is printed and the main menu returns. Had the answer to the request for modification been "YES" or "Y", the program would have prompted the changes and re-saved the record on disk.

DELETE A RECORD

Each combination of attributes has a unique attribute number associated with it. (We will describe below how this is calculated). When the "look up" option is used, note that the attribute number is given just below the attribute descriptions. To delete a record, we first select 3 from the main menu and the video display will show a request for the attribute number of the file we wish to

delete. Say we enter 19 (which corresponds to attributes 2, 5, 1), the video will display each record having these attributes in sequence, each time asking if a deletion is desired. A "YES" or "Y" reply will remove that subrecord from the data base but leave the remainder of the file unchanged.

PROGRAM FEATURES

- Size of Directory. When DATABASE/BAS is run, the user must input the directory size. In general, the size chosen depends on the total number of entries planned for file "RECORDS". A good rule of thumb is to use a number equal to 1½ to 2 times the number of entries anticipated. (The larger the directory size, shorter the access time needed to find the initial record, using the Hashing Function described below). Of course, the DIM statements (DATABASE/BAS - line 20 and ATRISRCH/BAS - line 30) should be appropriately changed if necessary (i.e., if size is greater than 300).
- Attribute Number - Line 8010 shows the equation used to calculate a unique attribute number for each combination of attributes for this application. In general, if we have N1 possible attributes of type A1, N2 possible attributes of type A2, and so forth,

I1 = (N2*N3*N4....)(A1-1)+(N3*N4*...)(A2-1)+...+*AN*

In our case, **N1=6, N2=5, N3=2, A1=C, A2=L and A3=B,** so **I1 = (5*2)(C-1)+2(L-1)+B**
or **I1 = 10C+2L+B-12**
This equation would be modified to suit the application being considered.

- Hashed Array Search Routine - subroutine 8000 uses a hashed search routine for rapid access to the first sub-record corresponding to a given attribute number. Once the first sub-record is found, it links to the second, the second links to the third, and so forth, thus enabling the program to rapidly find all the sub-records. (A sequential search would be very slow by comparison).

CHANGES TO ATRISRCH/BAS FOR OTHER APPLICATIONS

The following changes can be made to adapt the program to other applications:

LINE 30: Change size of index as described above to 2 times the number of entries anticipated in "RECORDS".

LINES 1120-1150, 6020-6050, 6140-6170, 9000-9190, and 10000-10180: Modify these lines to pertain to the attributes of the new application.

LINES 3030, 3130, 5000, 5020, 5030: Change to fit the new application's field lengths and subrecord definition (for further discussion see te Radio Shack TRSDOS & Disk BASIC REF Manual, pages 7-73 and 7-74).

LINE 3050: Assumes that the first field length exceeds 3. If less than 4, use**IF LEFT$(A$,1) <> " "...**

LINE 8010: Change equation as described above.

NOTE: Retain E$ as the final item in the field statement Line 5030 with length 4 and do not add any new uses for the variable E$ beyond those already used. The reason is that E$ is used by the program to "Link" one sub-record to the next one with the same attribute number of signify the end of the "chain" ●

FOR 32K DOS and UP...

DATABASE

```
10 CLS
20 DIM F(300),I(300),K(300)
30 PRINT"PROGRAM TO SET UP DATA BASE FOR ATTRIBUTE INDEX"
40 PRINT"INDEX FILE NAME IS 'INDEX'"
45 INPUT"SIZE OF DIRECTORY (300 MAXIMUM)";N
50 OPEN "O",1,"INDEX"
60 PRINT#1,N;000
70 FOR J=1 TO N
80 F(J)=-1:I(J)=0:K(J)=-1
90 PRINT#1,F(J);I(J);K(J)
100 NEXT J
110 CLOSE
120 PRINT"DATA BASE PREPARED"
```

```
10 'PROGRAM TO SEARCH A FILE FOR CERTAIN ATTRIBUTES
20 CLEAR 1000
30 DIM F(300)-I(300),K(300)
40 CLS:PRINT"ATTRIBUTE SEARCH PROGRAM FOR RECORDS":PRINT:PRINT
50 OPEN "I",1,"INDEX"
60 INPUT#1,N,V1
100 FOR J =1 TO N
110 INPUT#1,F(J),I(J),K(J)
120 NEXT J
125 CLOSE
130 OPEN "R",1,"RECORDS"
135 CLS
140 PRINT"SELECT ONE OF THE FOLLOWING :"
141 PRINT:PRINT"LOOK UP RECORD       = 1"
142 PRINT"ADD A RECORD          = 2"
143 PRINT"DELETE A RECORD       = 3"
144 PRINT"END                   = 4"
145 PRINT:INPUT"YOUR CHOICE (1, 2, 3, OR 4)";Z
148 ON Z GOTO 150,200,300,990
149 GOTO135
150 ' LOOK UP RECORD
155 GOSUB 9000
160 F1=-1
165 GOSUB8000
170 IFR<>0 THEN 178
175 INPUT"RECORD NOT FOUND. CONTINUE";Z$
```

```
999 CLOSE:END
1000 'FIND SUBRECORD BY KEY% ,MODIFY OR DELETE & LPRINT
1030 GOSUB5000
1080 CLS:PRINT"KEY NO.";KEY%
1090 PRINT"PHYSICAL RECORD NO.";PR%;TAB(30)"SUBRECORD NO.";SR%
1100 GOSUB 6000:GOSUB6100
1110 LPRINT"KEY NO. = ";KEY%
1120 LPRINT:LPRINT"RECORD TITLE:"TAB(25);A$
1130 LPRINT:LPRINT"COMPOSER:"TAB(25);B$
1140 LPRINT:LPRINT"PLAYED OR SUNG BY:"TAB(25);C$
1150 LPRINT:LPRINT"QUALITY OF RECORD:"TAB(25);D$
1160 LPRINT:LPRINT:LPRINT
1170 RETURN
3000 'ADD NEW  RECORDS TO DATAFILE
3020 PR%=LOF(1)
3030 KEY%=(PR%-1)*3+1
3040 GOSUB5000
3045 IF PR%>LOF(1) THEN3060
3050 IF LEFT$(A$,4)<>"     "THEN KEY%=KEY%+1:GOTO3040
3060 PRINT"BLANK SUBRECORD EXISTS IN PHYSICAL RECORD";PR%
3070 PRINT"SUBRECORD NUMBER";SR%
3075 PRINT"KEY NUMBER";KEY%
3080 INPUT"DO YOU WISH TO ADD A NEW SUBRECORD";Z$
3090 IF LEFT$(Z$,1)="N" THEN RETURN
3100 IF LEFT$(Z$,1)="Y"THEN3120
3110 PRINT"ANSWER YES OR NO":GOTO3080
3120 GOSUB6140
3130 KEY%=KEY%+1:GOTO3040
5000 PR%=INT((KEY%-1)/3)+1
5020 SR%=KEY%-3*(PR%-1)
5030 FIELD 1,((SR%-1)*85) AS STARTHERE$,40 AS A$,15 AS B$,20 AS
     C$,5 AS D$,4 AS E$
5040 GET 1,PR%
5050 RETURN
6000 PRINT:PRINT"RECORD DETAILS:"
6010 PRINT
6020 PRINT"RECORD TITLE:"TAB(25);A$
6030 PRINT"COMPOSER:"TAB(25);B$
6040 PRINT"PLAYED OR SUNG BY:"TAB(25);C$
6050 PRINT"QUALITY OF RECORD:"TAB(25);D$
6090 PRINT:PRINT:RETURN
6100 INPUT"DO YOU WISH TO UPDATE THIS DATA";Z$
6110 IF LEFT$(Z$,1)="N"THEN RETURN
6120 IF LEFT$(Z$,1)="Y"THEN6140
6130 PRINT"ANSWER YES OR NO":GOTO6100
6140 LINEINPUT"RECORD TITLE (40 CHAR)";A1$:LSET A$=A1$
6150 LINEINPUT"COMPOSER (15 CHAR)";B1$:LSET B$=B1$
6160 LINEINPUT"PLAYED OR SUNG BY (20 CHAR)";C1$:LSET C$=C1$
6170 LINEINPUT"QUALITY OF RECORD (5 CHAR)";D1$:LSET D$=D1$
6180 IF F1>0 THENLSET E$=MKS$(-1)
6210 PUT 1,PR%
6220 RETURN
8000 'HASHED ARRAY SEARCH SUROUTINE
8010 I1=10*C+2*L+B-12
8020 Q1=INT(I1/N):R1=INT(I1-Q1*N)
8030 IFQ1>0 THEN8050
8040 Q1=1
```

```
177 GOTO135
178 GOSUB10000
180 KEY%=K(R)
182 GOSUB1000
184 KEY%=CVS(E$)
186 IF KEY%<=0 THEN 135 ELSE 182
200 'ADD A RECORD
205 GOSUB 9000
210 F1=1
215 GOSUB8000
220 IF R=0 THEN 275
225 V1=V1+1:KEY%=V1
227 GOSUB5000
230 GOSUB6140
235 IF I(R)=I1 THEN250
240 K(R)=V1:I(R)=I1:F(R)=1
245 GOTO135
250 KEY%=K(R)
253 GOSUB 5000
255 IF CVS(E$)>=0 THEN KEY%=CVS(E$):GOTO253
260 LSET E$=MKS$(V1)
262 PUT 1,PR%
265 GOTO135
275 INPUT"DIRECTORY FULL REORGANIZE. READY";Z$
280 GOTO135
300 'DELETE A RECORD
310 INPUT"ATTRIBUTE NUMBER OF RECORD";I1
320 F1=-1
330 GOSUB8020
340 IF R<>0THEN370
350 INPUT"RECORD NOT FOUND. CONTINUE";Z$
360 GOTO135
370 KEY%=K(R)
375 K1=-1
380 GOSUB5000
390 GOSUB6000
400 INPUT"DO YOU WISH TO DELETE THIS RECORD";Z$
410 IF LEFT$(Z$,1)="Y" THEN460
420 IF LEFT$(Z$,1)="N" THEN440
430 PRINT"ANSWER YES OR NO.":GOTO400
440 IF CVS(E$)<0 THEN 135
450 K1=KEY%:KEY%=CVS(E$):GOTO380
460 N1=CVS(E$)
490 IF N1<0 AND KEY%-K(R)<.01 THEN F(R)=-1:I(R)=0:K(R)=-1:GOTO1
    35
500 IF K1>0 THEN KEY%=K1 ELSE 540
510 GOSUB5000
520 LSET E$=MKS$(N1):PUT 1,PR%
530 GOTO135
540 K(R)=N1
550 GOTO135
990 CLOSE
992 OPEN "O",1,"INDEX"
994 PRINT#1,N;V1
996 FOR J=1 TO N
997 PRINT#1,F(J);I(J);K(J)
998 NEXT J
```

```
8050 R=R1+1:Q=Q1
8060 'REPEAT
8070 FOR J=1 TO N-1
8075 IF F(R)>0 THEN 8100
8080 IF F1>0 THEN8500
8090 R=0
8095 GOTO8500
8100 'EXIT OR TRY AGAIN
8105 IF I(R)=I1 THEN 8500
8110 R=R+Q1:Q=INT(R/N):R1=INT(R-Q*N):R=R1+1
8115 NEXT J
8310 R=0
8500 RETURN
9000 'INPUT ATTRIBUTES
9010 CLS:PRINT"INPUT ATTRIBUTES":PRINT:PRINT
9015 PRINT"CATEGORY OF MUSIC":PRINT
9020 PRINT"CLASSICAL SOLO PRINCIPAL      = 1"
9030 PRINT"CLASSICAL NON SOLO            = 2"
9040 PRINT"LIGHT BACKGROUND MUSIC        = 3"
9050 PRINT"JAZZ                          = 4"
9060 PRINT"WEST INDIAN MUSIC             = 5"
9070 PRINT"OTHER                         = 6"
9080 PRINT:INPUT"CATEGORY NUMBER";C
9090 PRINT:PRINT:PRINT"TYPE OF MUSIC":PRINT
9100 PRINT"VOCAL      = 1"
9110 PRINT"PIANO      = 2"
9120 PRINT"VIOLIN     = 3"
9130 PRINT"GUITAR     = 4"
9140 PRINT"OTHER      = 5"
9150 PRINT:INPUT"TYPE NUMBER";L
9160 PRINT:PRINT:PRINT"QUALITY OF RECORD":PRINT
9170 PRINT"GOOD  = 1"
9180 PRINT"POOR  = 2"
9190 PRINT:INPUT"QUALITY";B
9200 RETURN
10000 'LPRINT TITLES
10005 LPRINT"CATEGORY OF MUSIC: ";
10010 ON C GOSUB 10060,10070,10080,10090,10100,10110
10015 LPRINT"TYPE IF MUSIC: ";
10020 ON L GOSUB 10120,10130,10140,10150,10160
10025 LPRINT"QUALITY OF RECORD: ";
10030 ON B GOSUB 10170,10180
10040 LPRINT"ATTRIBUTE NUMBER";I1;" (";C;",";L;",";B;")"
10050 LPRINT:LPRINT:RETURN
10060 LPRINT"CLASSICAL SOLO PRINCIPAL":RETURN
10070 LPRINT"CLASSICAL NON SOLO":RETURN
10080 LPRINT"LIGHT BACKGROUND MUSIC":RETURN
10090 LPRINT"JAZZ":RETURN
10100 LPRINT"WEST INDIAN MUSIC":RETURN
10110 LPRINT"OTHER":RETURN
10120 LPRINT"VOCAL":RETURN
10130 LPRINT"PIANO":RETURN
10140 LPRINT"VIOLIN":RETURN
10150 LPRINT"GUITAR":RETURN
10160 LPRINT"OTHER":RETURN
10170 LPRINT"GOOD":RETURN
10180 LPRINT"POOR":RETURN
```

The 8th in a series on Assembly Language Applications.

SYSTEM/ COMMAND

Phil Pilgrim
Discovery Bay Software Co
Port Townsend, WA

For Model I, Level II 16K or 32/48K DOS

Selectively scroll any portion of your video screen, while holding the remainder stationary!

Before launching into this issue's project, some words are in order about the last one (Keyboard Macros, Jan/Feb'80, page 52) Some of you were confused by the use of the word ENTER in that column, and with good reason. there was no way to distinguish the sequence of letters E-N-T-E-R from the word *ENTER* meaning "hit the *ENTER* key" The same was true of *BREAK* and *SHIFT*. As it happened, the italics were dropped somewhere between manuscript and finished page, and I apologize for the confusion So henceforth in this column any word like *ENTER* or *SHIFT* appearing in italics means to hit (or hold) that key, it does not mean to type out the word Now, on with the new business

This issue's project is lengthy code-wise, but easy to describe. It's purpose is to allow the definition of rectangular subregions (zones) on the screen into which all text subsequently PRINTed or LISTed will be displayed. It's like having multiple screens within a screen which can be set up and dismantled at will, with the regions outside a given zone protected from overwriting. It's applications are many, including multiple and independent columns for numerical output, split screen effects wherein LISTings of different program segments can be displayed and compared side-by-side, and screen protection, which is handy in data entry systems Your imigination, no doubt, will produce many other uses for this utility.

With ZDSPLY activated, the display routine behaves in normal fashion until the control code 1, CHR$(1) from BASIC is received. It then accepts two more parameters, specifying the locations of the upper-lefthand and lower-righthand corners of the new display zone. Next, it homes the cursor in the upper-lefthand corner of the zone, and returns. Henceforth, any characters sent to the display routine will be displayed in the defined zone, the entire zone scrolling when necessary, just like a regular screen. The zoned display may be exited by one of four methods Hitting the *CLEAR* button, a CLS instruction, PRINTing a CHR$(31)[Clear to the end of the frame], or PRINTing a CHR$(0). In any of the above occurances,the screen is returned to it's normal (unzoned) display.

The parameters specifying the corners of the zones are derived from the PRINT@ positions (see page E-1 of the Level II manual). Since the CHR$ command requires a parameter between 0 and 255, we must convert these PRINT@ positions to something that can be printed. To do this, simply take the INTeger of the result of the PRINT@ position divided by 4. The statement

```
PRINT CHR$(1),CHR$(INT(35/4)),
CHR$(INT(575/4));
```

Will cause a zone to be defined whose upper-lefthand corner is PRINT@ position 35, and whose lower-righthand corner is PRINT@ position 575. At that time, anything PRINTed or LISTed will stay within those boundries, just as though the rest of the screen never existed

While in the zoned mode, certain control characters are not recognized. These are CHR$(23) *convert to 32-character mode*, CHR$(24)-(27) *cursor movement*, CHR$(29)*move to the beginning of the line*, and CHR$(30) *erase to the end of the line*. Two control codes have different meanings from those given in the Level II manual. They are. CHR$(28)*home, or return cursor to print position 0* now returns the cursor to the upper-lefthand corner of the zone, and

CHR$(31)*clear to the end of the frame* returns to the unzoned mode and clears the entire screen. Hitting *CLEAR* will always return thus to the unzoned mode. Note that you **must** return to unzoned mode before defining a new zone. If you want to keep the data already displayed, use PRINT CHR$(0).

To use ZDSPLY, key the program as shown into EDTASM. For memory sizes other than 16K, change the ORG to 0BEC2H (32K) or 0FEC2 (48K), and use MEMORY SIZEs of 48834 and 65218, respectively. DOS 2.2 users will want to subtract 64 (40H) from the ORG and MEMORY SIZE to accomodate the high-memory scratchpad. In addition, all DOS users will want to change the JP 1A19H in the START block to JP 402DH (return to DOS), so ZDSPLY can be saved as a '/CMD' file to be loaded and initiated from DOS instead of BASIC. Once the program is entered, assemble it, make a SYSTEM tape (or /CMD file) and load it in the manner appropriate to your system. Once the START block has been executed (by typing */ENTER* from SYSTEM or loading as an autostarting command file), the program will perform as described.

Now a few words about the program itself. The START block links ZDSPLY into the display calling sequence, then exits. ZDSPLY, when called, may be in one of four modes, starting with mode 0 (regular, unzoned display). A CHR$(1) takes it to mode 1 (ready to accept ULHC parameter), and each subsequent character bumps the mode by one until mode 3 (zoned display mode) is reached. Depending on the mode, calls to ZDSPLY are fielded by RMODE, GULC, GLRHC, and ZONE, respectively.

The section of the program starting with ZONE is the real meat of ZDSPLY. ZONE is divided into two major parts: the section which puts the characters on the screen (before CNTRL), and the section which handles control characters (from CNTRL onward). Throughout ZONE calls are made to locations in the Level II ROM. These CALLed locations are 0458H (with the carry flag set) which does nothing but return the cursor position in HL and the character previously "covered" by the cursor in the A, 0480H, which is the universal return point and which replaces the cursor at it's new position; 04B8H, which turns on the cursor; 04BDH, which turns it off, and 01C9H, which executes the CLS (clear screen) command.

ZONE also calls several internal subroutines. The most important of these, CCHECK, checks to see if the cursor address (in HL) is inside the defined zone. If not, it returns with the carry flag set. LJUST just returns the cursor (HL again) to the left edge of the zone on the same line. SCROLL scrolls all the characters in the zone up one line, blanks the bottom line, and places the cursor at the beginning of the bottom line. CLRROW (part of SCROLL) just blanks the current line.

As it stands, ZDSPLY is just ripe for modification. You may even wish to implement the missing control codes. To do so, just put the necessary CP's in the CNTRL section, with JR's to your routines. A RET at the end of each routine will automatically jump to 0480H in ROM to fix up the cursor. But modified or not, you should find ZDSPLY a useful addition to your program library. ●

Side-by-side Listing of *Zone Display*

```
7EC2            00100          ORG     7EC2H       ;16K MEM SZ=32450
7EC2 21CB7E     00110 START    LD      HL,ZDSPLY   ;LINK INTO CALLING SEQ.
7EC5 221E40     00120          LD      (401EH),HL  ;       .
7EC8 C3191A     00130          JP      1A19H       ;RET TO L2/DOS USE 402DH
7ECB 3AFF7F     00140 ZDSPLY   LD      A,(MODE)    ;GET MODE
7ECE B7         00150          OR      A           ;IS IT ZERO?
7ECF 2845       00160          JR      Z,RMODE     ;  YES:REGULAR MODE
7ED1 3D         00170          DEC     A           ;IS IT ONE?
7ED2 2827       00180          JR      Z,GULHC     ;  YES:GET UPPER LH CRNR
7ED4 3D         00190          DEC     A           ;IS IT TWO?
7ED5 2047       00200          JR      NZ,ZONE     ;  NO:DO ZONE DSPLY
7ED7 69         00210 GLRHC    LD      L,C         ;GET LOWER RH CRNR IN L
7ED8 260F       00220          LD      H,0FH       ;0F=3C AFTER SHIFTING
7EDA 29         00230          ADD     HL,HL       ;SHIFT LEFT
7EDB 23         00240          INC     HL          ;ADD ONE
7EDC 29         00250          ADD     HL,HL       ;SHIFT LEFT
7EDD 23         00260          INC     HL          ;ADD ONE
7EDE 29         00270          ADD     HL,HL       ;SHIFT LEFT
7EDF 29         00280          ADD     HL,HL       ;SHIFT LEFT
7EE0 22FB7F     00290          LD      (LRHC),HL   ;SAVE SHIFTED LRHC ADDR.
7EE3 ED5BF97F   00300          LD      DE,(ULHC)   ;GET ULHC ADDR.
7EE7 7D         00310          LD      A,L         ;GET COLUMN PORTION
7EE8 93         00320          SUB     E           ;SUB COL. PART OF LRHC
7EE9 387B       00330          JR      C,ZMODE     ;NO. COLS <= 0: FORGET IT
7EEB 1F         00340          RRA                 ;RESHIFT NO. COLS
7EEC 1F         00350          RRA                 ;       .
7EED 3C         00360          INC     A           ;AND INCREMENT
7EEE 32FE7F     00370          LD      (NCOLS),A   ;SAVE IT
7EF1 7C         00380          LD      A,H         ;GET ROW PORTION OF LRHC
7EF2 92         00390          SUB     D           ;SUB ROW PORTION OF ULHC
7EF3 3871       00400          JR      C,ZMODE     ;NO. ROWS <= 0:FORGET IT
7EF5 3C         00410          INC     A           ;INCREMENT IT
7EF6 32FD7F     00420          LD      (NROWS),A   ;AND SAVE
7EF9 180D       00430          JR      MODINC      ;BUMP MODE TO 3
7EFB 69         00440 GULHC    LD      L,C         ;GET UPPER LH CRNR IN L
7EFC 260F       00450          LD      H,0FH       ;0F=3C WHEN SHIFTED
7EFE 29         00460          ADD     HL,HL       ;SHIFT HL LEFT
7EFF 29         00470          ADD     HL,HL       ;  AGAIN
7F00 22F77F     00480          LD      (HOME),HL   ;SAVE UPPER LH CRNR ADDR.
7F74 19         01080          ADD     HL,DE       ;       .
7F75 CDEB7F     01090          CALL    LJUST       ;LEFT JUSTIFY
7F78 3AFE7F     01100          LD      A,(NCOLS)   ;RIGHT JUSTIFY
7F7B 3D         01110          DEC     A           ;       .
7F7C 85         01120          ADD     A,L         ;       .
7F7D 6F         01130          LD      L,A         ;       .
7F7E CD9D7F     01140          CALL    CCHECK      ;IN ZONE NOW?
7F81 DA997F     01150          JP      C,GOHOME    ;  NO:HOME THE CURSOR
7F84 3620       01160 BSPOUT   LD      (HL),20H    ;BLANK CURSOR POSITION
7F86 C9         01170          RET                 ;AND RETURN TO 0480H
7F87 014000     01180 CR       LD      BC,64       ;CARRIAGE RET: BUMP LINE
7F8A 09         01190          ADD     HL,BC       ;       .
7F8B CDEB7F     01200          CALL    LJUST       ;LEFT JUSTIFY
7F8E CD9D7F     01210          CALL    CCHECK      ;IN ZONE?
7F91 DCBD7F     01220          CALL    C,SCROLL    ;  NO:DO SCROLL
7F94 AF         01230          XOR     A           ;CLEAR THE ROW
7F95 CDDB7F     01240          CALL    CLRROW      ;       .
7F98 C9         01250          RET                 ;AND RETURN
7F99 2AF77F     01260 GOHOME   LD      HL,(HOME)   ;GET ULHC ADDR. IN HL
7F9C C9         01270          RET                 ;RET TO 0480H
7F9D E5         01280 CCHECK   PUSH    HL          ;CHECK IF CURSOR IN ZONE
7F9E 29         01290          ADD     HL,HL       ;SHIFT HL LEFT
7F9F 29         01300          ADD     HL,HL       ;AGAIN
7FA0 3819       01310          JR      C,CCHOUT    ;OUT OF BOUNDS:RETURN
7FA2 ED5BF97F   01320          LD      DE,(ULHC)   ;GET ULHC ADDR.
7FA6 7D         01330          LD      A,L         ;GET COLUMN PORTION
7FA7 BB         01340          CP      E           ;CURSOR TOO FAR LEFT?
7FA8 3811       01350          JR      C,CCHOUT    ;  YES:RETURN
7FAA 7C         01360          LD      A,H         ;GET ROW PORTION
7FAB BA         01370          CP      D           ;CURSOR TOO HIGH?
7FAC 380D       01380          JR      C,CCHOUT    ;  YES:RETURN
7FAE ED5BFB7F   01390          LD      DE,(LRHC)   ;GET LRHC ADDR.
7FB2 7B         01400          LD      A,E         ;GET COLUMN PART
7FB3 BD         01410          CP      L           ;CURSOR TOO FAR RIGHT?
7FB4 3805       01420          JR      C,CCHOUT    ;  YES:RETURN
7FB6 7A         01430          LD      A,D         ;GET ROW PART
7FB7 BC         01440          CP      H           ;CURSOR TOO FAR DOWN?
7FB8 3801       01450          JR      C,CCHOUT    ;  YES:RETURN
7FBA AF         01460          XOR     A           ;RESET CARRY
7FBB E1         01470 CCHOUT   POP     HL          ;RESTORE CURSOR
7FBC C9         01480          RET                 ;AND RETURN
7FBD ED5BF77F   01490 SCROLL   LD      DE,(HOME)   ;SCROLL:GET ULHC ADDR.
7FC1 3AFE7F     01500          LD      A,(NCOLS)   ;GET NO. OF COLS. IN BC
7FC4 4F         01510          LD      C,A         ;       .
7FC5 0600       01520          LD      B,0         ;       .
7FC7 3AFD7F     01530          LD      A,(NROWS)   ;GET NO. OF ROWS
7FCA 3D         01540          DEC     A           ;  MINUS ONE
7FCB 280D       01550          JR      Z,SCDONE    ;JUST ONE ROW:DONE
7FCD 214000     01560 NXTROW   LD      HL,64       ;GET ADDR. OF NEXT ROW
7FD0 19         01570          ADD     HL,DE       ;       .
7FD1 E5         01580          PUSH    HL          ;SAVE IT
7FD2 C5         01590          PUSH    BC          ;SAVE COLUMN COUNT
7FD3 EDB0       01600          LDIR                ;MOVE NXT ROW TO THIS ONE
7FD5 C1         01610          POP     BC          ;RESTORE COLUMN COUNT
7FD6 D1         01620          POP     DE          ;NEXT ROW IS NOW THIS ROW
7FD7 3D         01630          DEC     A           ;DECREMENT ROW COUNT
7FD8 20F3       01640          JR      NZ,NXTROW   ;ZERO? NO:DO AGAIN
7FDA EB         01650 SCDONE   EX      DE,HL       ;  YES:HL=BOTTOM ROW
```

```
7F03 29       00490          ADD     HL,HL       ;SHIFT HL LEFT
7F04 29       00500          ADD     HL,HL       ;  AGAIN
7F05 22F97F   00510          LD      (ULHC),HL   ;SAVE ULHC IN ROW/COL FORM
7F08 3AFF7F   00520 MODINC   LD      A,(MODE)    ;INCREMENT MODE
7F0B 3C       00530          INC     A           ;       .
7F0C 32FF7F   00540          LD      (MODE),A    ;       .
7F0F FE03     00550          CP      3           ;TIME FOR ZDSPLY?
7F11 C0       00560          RET     NZ          ;  NO:RETURN
7F12 0E1C     00570          LD      C,28        ;  YES:HOME CURSOR IN ZONE
7F14 1808     00580          JR      ZONE        ;       .
7F16 79       00590 RMODE    LD      A,C         ;REGULAR DSPLY:GET CHAR
7F17 FE01     00600          CP      01H         ;ZONE DSPLY PARS COMING?
7F19 C25804   00610          JP      NZ,0458H    ;  NO:GOTO ROM DSPLY PROG.
7F1C 18EA     00620          JR      MODINC      ;  YES:INCREMENT MODE
7F1E DA5804   00630 ZONE     JP      C,0458H     ;CALLER JUST INQUIRING
7F21 37       00640          SCF                 ;SET CARRY
7F22 CD5804   00650          CALL    0458H       ;GET CURSOR POS. IN HL
7F25 77       00660          LD      (HL),A      ;REPLACE CURSOR CHAR.
7F26 118004   00670          LD      DE,0480H    ;RET'S GOTO 0480H
7F29 D5       00680          PUSH    DE          ;       .
7F2A 79       00690          LD      A,C         ;GET CHAR IN A
7F2B FE20     00700          CP      20H         ;CNTRL CHAR?
7F2D 3813     00710          JR      C,CNTRL     ;  YES:TAKE CARE OF IT
7F2F FE80     00720          CP      80H         ;GRAPHICS?
7F31 3006     00730          JR      NC,CPUT     ;  YES:BRANCH AROUND
7F33 FE60     00740          CP      60H         ;LOWER CASE?
7F35 3802     00750          JR      C,CPUT      ;  NO:BRANCH AROUND
7F37 D620     00760          SUB     20H         ;  YES:CHANGE TO UPPER
7F39 77       00770 CPUT     LD      (HL),A      ;PUT CHAR ON SCREEN
7F3A 23       00780          INC     HL          ;INCREMENT CURSOR POS.
7F3B CD9D7F   00790          CALL    CCHECK      ;STILL IN ZONE?
7F3E D0       00800          RET     NC          ;  YES:RETURN
7F3F 2B       00810          DEC     HL          ;  NO:TREAT LIKE CR
7F40 1845     00820          JR      CR          ;       .
7F42 B7       00830 CNTRL    OR      A           ;CHAR ZERO?
7F43 2821     00840          JR      Z,ZMODE     ;  YES:ZERO THE MODE
7F45 FE08     00850          CP      8           ;BACKSPACE?
7F47 2822     00860          JR      Z,BSP       ;YES:TAKE CARE OF IT
7F49 FE0A     00870          CP      10          ;CARRIAGE RETURN?
7F4B D8       00880          RET     C           ;  NO, A 9:FORGET IT
7F4C FE0E     00890          CP      14          ;CARRIAGE RETURN?
7F4E 3837     00900          JR      C,CR        ;  YES:TAKE CARE OF IT
7F50 CAB804   00910          JP      Z,04B8H     ;  NO:TURN ON CURSOR
7F53 FE0F     00920          CP      15          ;TURN OFF CURSOR?
7F55 CABD04   00930          JP      Z,04BDH     ;  YES:GO DO IT
7F58 D61C     00940          SUB     28          ;HOME CURSOR?
7F5A 283D     00950          JR      Z,GOHOME    ;  YES:DO IT
7F5C D603     00960          SUB     3           ;CLEAR?
7F5E C0       00970          RET     NZ          ;  NO:JUST DROP IT
7F5F 32FF7F   00980 CLEAR    LD      (MODE),A    ;  YES:ZERO MODE
7F62 D1       00990          POP     DE          ;POP RET TO 0480H
7F63 C3C901   01000          JP      01C9H       ;DO CLS ROUTINE
7F66 AF       01010 ZMODE    XOR     A           ;ZERO THE MODE
7F67 32FF7F   01020          LD      (MODE),A    ;       .
7F6A C9       01030          RET                 ;       .
7F6B 2B       01040 BSP      DEC     HL          ;DECREMENT CURSOR
7F6C CD9D7F   01050          CALL    CCHECK      ;STILL IN ZONE?
7F6F 3013     01060          JR      NC,BSPOUT   ;  YES:FINISH UP
7F71 11C1FF   01070          LD      DE,-63      ;  NO:BACK UP ONE LINE
7FDB 47       01660 CLRROW   LD      B,A         ;A WAS 0, SO B=0
7FDC 3AFE7F   01670          LD      A,(NCOLS)   ;GET NO. OF COLUMNS
7FDF 3D       01680          DEC     A           ;  LESS ONE
7FE0 4F       01690          LD      C,A         ;  IN BC
7FE1 E5       01700          PUSH    HL          ;SAVE CURSOR POS.
7FE2 E5       01710          PUSH    HL          ;AND BOUNCE INTO DE
7FE3 D1       01720          POP     DE          ;       .
7FE4 13       01730          INC     DE          ;ADD ONE
7FE5 3620     01740          LD      (HL),20H    ;BLANK FIRST POSITION
7FE7 EDB0     01750          LDIR                ;AND THE REST OF THE LINE
7FE9 E1       01760          POP     HL          ;RESTORE CURSOR POSITION
7FEA C9       01770          RET                 ;AND RETURN
7FEB 3AF77F   01780 LJUST    LD      A,(HOME)    ;GET LSB OF HOME ADDR
7FEE E63F     01790          AND     3FH         ;JUST COLUMN PORTION
7FF0 47       01800          LD      B,A         ;SAVE IN B
7FF1 7D       01810          LD      A,L         ;GET LSB OF CURSOR POS
7FF2 E6C0     01820          AND     0C0H        ;BLANK COLUMN PORTION
7FF4 B0       01830          OR      B           ;OR IN HOME COLUMN
7FF5 6F       01840          LD      L,A         ;AND PUT IN L
7FF6 C9       01850          RET                 ;ALL DONE: RETURN
0002          01860 HOME     DEFS    2           ;ADDR OF ULHC
0002          01870 ULHC     DEFS    2           ;ULHC IN ROW/COL FORM
0002          01880 LRHC     DEFS    2           ;LRHC IN ROW/COL FORM
0001          01890 NROWS    DEFS    1           ;NO. ROWS IN ZONE
0001          01900 NCOLS    DEFS    1           ;NO. COLS. IN ZONE
7FFF 00       01910 MODE     DEFB    0           ;MODE, INITIALLY ZERO
7EC2          01920          END     START       ;AUTOSTART AT START
00000 TOTAL ERRORS
```

```
BSP     7F6B 01040   00860
BSPOUT  7F84 01160   01060
CCHECK  7F9D 01280   00790 01050 01140 01210
CCHOUT  7FBB 01470   01310 01350 01380 01420 01450
CLEAR   7F5F 00980
CLRROW  7FDB 01660   01240
CNTRL   7F42 00830   00710
CPUT    7F39 00770   00730 00750
CR      7F87 01180   00820 00900
GLRHC   7ED7 00210
GOHOME  7F99 01260   00950 01150
GULHC   7EFB 00440   00180
HOME    7FF7 01860   00480 01260 01490 01780
LJUST   7FEB 01780   01090 01200
LRHC    7FFB 01880   00290 01390
MODE    7FFF 01910   00140 00520 00540 00980 01020
MODINC  7F08 00520   00430 00620
NCOLS   7FFE 01900   00370 01100 01500 01670
NROWS   7FFD 01890   00420 01530
NXTROW  7FCD 01560   01640
RMODE   7F16 00590   00160
SCDONE  7FDA 01650   01550
SCROLL  7FBD 01490   01220
START   7EC2 00110   01920
ULHC    7FF9 01870   00300 00510 01320
ZDSPLY  7ECB 00140   00110
ZMODE   7F66 01010   00330 00400 00840
ZONE    7F1E 00630   00200 00580
```

BUSINESS COMPUTING

T R Dettmann, Associate Editor

A close look at the Model II Microcomputer

Last year, Radio Shack announced that it would be releasing a new TRS-80 computer, not a successor, but a big brother to the Model I. We waited a long time, and it finally came.

Most of you have seen pictures of it in magazine advertisements. You may have read about what it can do. For those of you who are not overly familiar with it, some of its features are:

2 Z80-A Processors (one for the computer and one for the video display) operating at 4 Mhz

Powerup and RESET are handled by a small ROM routine which is electronically switched out and replaced with RAM when the routine is complete.

A full 80 character by 24 line display screen with FULL ASCII (Upper & lower case).

The keyboard has an LSI controller so that it is independent of the Z80-A operation. It also has standard typewriter layout and a host of normal computer keyboard keys including CTRL, LOCK, CAPS, REPEAT, HOLD etc.

An 8 inch double density disk drive is built into the system which can hold a half megabyte. Three more drives can be added in an optional external cabinet.

A parallel I/O port is provided for printers and two serial I/O ports for printers, modems, etc

Everything including powerup and RESET can be accomplished from the front of the unit

An AC line filter is built into the power supply.

On looks alone, the Model II is a nice unit. However, the test of the computer is in its use.

USER REPORT

The first thing that strikes you is the bulk of the machine. It is bigger than the Model I, but everything is there.

Turning power on is simply a matter of pushing up the front panel power switch. The computer tells you to insert your system disk (DON'T insert it before you power up or you will lose it!)

On power up, the system checks memory and functions and lets you know right away if there is any problem with the hardware. If not, the new Tandy Computer (or is it Tandy Corp?) logo comes up on the screen and you are asked for the date (this must be input correctly) and the time (optional).

Once you are in the DOS READY command mode, you can start playing with a wealth of system level commands. The first impulse is to type DIR for a directory check. My first mistake was typing it in lower case. This causes a system error because the system command interpreter understands only upper case!

Getting the system error so soon points out an interesting problem: The error messages are all error codes, and are not written out. So just what is an ERROR 13?. A little reading in the manual can tell you, but the designers provided a handy method for finding the error message. Just type *ERROR 31* and the system returns the error message "PROGRAM NOT FOUND". That is, it found the letters "dir" in lower case, couldn't interpret them as a command, and so went on to try and execute them as a program. Oh well.--

It is best to run in DOS READY mode with the CAPS key on. This automatically capitalizes all letters from the keyboard so your commands come out right. One nice feature of this and the LOCK key is that both have a small light in the key which lets you know they are on.

Unlike Model I, the keyboard is connected to the main unit with a flexible cable which moves into and out of the main unit. With the flexible cable, you can set the Model II as the side of your desk angles toward you,

and then bring the keyboard to you when you need it.

SYSTEM LEVEL OPERATION

Once you have overcome the initial problems with letting the DOS know what you want, you can find a very flexible command structure compared to the MODEL I DOS. It incorporates many features that do not exist except in more advanced operating systems.

My attention was drawn to the Command file ability. Using the DOS command BUILD, you can create a file of DOS commands which will automatically be executed by issuing a single command to "DO" the filename of commands. With this, you can build a powerup AUTO sequence that will take care of loading all support software, initializing all hardware drivers (the printer and RS232's have to be set up before use at the command level), and then deliver the user to the menu of a BASIC program with everything ready.

This one capability will make the Model II a perfect system for "Turnkey" type applications, where a vendor supplies the computer, the software, and provides the maintenance. The end-user interacts with the computer only via a menu driven program.

This can be done on the Model I, but is nowhere near as convienient as the Model II, since the software has to be written by the vendor to accomplish the task.

I particularly like the FREE disk space command, since it actually displays a map of the disk, rather than just putting the numbers on the screen. It makes it clear how everything really is placed on the disk. Overall, I was satisfied with the DOS, but think that after a few months, someone (Apparat maybe?) will come out with a better DOS for this system.

Before I tell you some of the things I didn't like, let me say first that these are mostly "nit-picking" items and are only a minor bother. Overall, the system is easy to use and better than it might have been.

Some of the nit-picks are that (1) When changing diskettes, you have to do an "I" command at the system level or the directories are all fouled up. Apparently the system keeps a portion of the directory in memory and so you have to re-initialize a new disk or the system gets confused.

A very depressing problem, is (2) the DEBUG system is built to prevent you from getting a look at the system. I don't know if the Radio Shack people are getting paranoid in their old age, but anyone with a little machine language programming can easily get around this restriction. Nit (3) is that there is no override on the AUTO power up feature. If you set auto on a command file that goes into some kind of infinite loop, you can really be in trouble.

MACHINE LANGUAGE

Model II dosen't forget the machine language programmer (though just now there is no adequate system development software around. I hear that RACET Computes now offers a Model II Assembler, but haven't seen it yet). At present, you have two real options for writing assembly language programs: (1) put them in a byte at a time with DEBUG (2) assemble them on Model I and upload them to the Model II (More about this later).

Either option isn't particularly the best. But since the machine does not have PEEK and POKE in BASIC, you have to make do with this. Why no PEEK or POKE? The official word is

that it wasn't needed for this system. Whether or not that was the real driving force behind it, most of us are sad to see them not available.

BASIC still has the USR call to machine language subroutines, so calls to them can work once you get the machine language in. If you want to work exclusively in machine language though, writing your own assembler may be the way to go (unless you can wait).

The Model II is no slouch in the machine language department. The technical information section in the DOS manual details calling procedures for 97 special "Supervisory Routines" (SVC's). These are system routines which can be called directly by a machine language program to accomplish something the DOS programmers have already done (re-inventing the wheel?).

Providing special calling sequences for the routines without giving away the routines themselves (including their locations in memory) makes the system safe from the Radio Shack side, since they are not compromising their software, but still gives us access to the programs. I would have liked to have seen the actual programs made available, but that is probably too much to ask (and it takes the sport out of disassembling DOS).

The PARSER for the system is accessible to the programmer. This makes the job of development of special programs (such as compilers) much easier. Unfortunately, this also makes these programs VERY system dependent.

BASIC PROGRAMMING

The Model II's BASIC, designated *LEVEL III* by Radio Shack, has significant improvements over their earlier BASIC's. Microsoft added and deleted features as specified by Radio Shack so that the system is in most respects compatible with programs written on the Model I.

The lack of a PEEK or POKE facility has to be the major drawback to the system. However, we might forgive them since the whole BASIC interpreter is much larger.

One of the things that BASIC on the Model II can do is access the system commands while in BASIC. For example, SYSTEM"DIR" will print a disk directory and then return to BASIC (just like the CMD function in Apparat's BASIC).

A major problem in business programs is that sorting on the Model I takes a long time because of memory management. We covered how to fix that for Model I earlier (*80-US Journal, Nov-Dec 79*), but that fix relies on PEEK and POKE. The Model II provides us with the SWAP command, which apparently changes the pointers to the variables without having to physically change the strings. So far, I haven't seen any memory management during sorting using SWAP.

File handling is an even more impressive improvement. Some capabilities latent in the Model I DOS such as variable length records are now fully operational. We can now create random access (now called *direct access*) files truly tailored to the application.

UPLOADING TO MODEL II

Once you get the software (available from Radio Shack) to upload programs from your Model I, you can use software you have already developed to run your new system. You need the following hardware to do this: RS232 in the Model I and an Interconnect Cable with DB25 connectors (1 male, 1 female).

Once everything is connected, you can start sending programs over, but don't expect them to go fast. The software checks and rechecks and sends only one block at a time. This can be pretty tedious for large programs, but it goes well once started.

After you get the programs on the Model II, there will still be rather extensive changes needed to make them work right. Screen displays for one will need to be reset. Screens that use TAB's and PRINT's will look OK and work right, though they will be rather small. Displays which use PRINT@ extensively will have to be modified more severly, since the larger screen will displace your display enough to make it unrecognizable.

Any programs that rely on PEEK and POKE for their effects will have to be re-written to somehow use other techniques. In many cases, where the PEEK or POKE was used for greater speed in the Model I, the 4 Mhz clock speed in the Model II will make up for it.

Graphics displays will have to be redone completely since the Model II graphics are not the same as those on the Model I. They are less extensive and use different figures for the graphics characters.

A less obvious change in many programs will be the use of control codes. In the Model I, executing the statement: "PRINT CHR$(23)" causes the display to shift to large characters, twice the normal size. On the Model II, this same statement would erase to the end of the current line and leave the cursor where it was. Many programs that use control codes don't clearly identify where they are or what they are supposed to accomplish. That means that unless you know the program well, changing control codes to the new machine could be a tricky job.

IS MODEL II FOR YOU?

That question is one you really have to answer for yourself. But assuming our impressions are roughly in line with what you would think, (had you played with the system), then they might help.

The Model II has quite a few things going for it: Greater processor speed, larger screen display, more storage, expandable electronics (does Radio Shack have something up their sleeve?). The whole unit is built better than the Model I and so should be less susceptible to the kinds of small faults that have plagued the earlier machine.

If you are investing in a computer for the first time, the Model II will be the better business machine, without a doubt. At least for the time being, the Model I will certainly hold the lead in Games, if that is your only interest, and - you can get into a Model I for a lower price at the start.

You should compare prices however. Getting a Model II is not that much more expensive than getting a full Model I Business System. If you are going to go that far anyway, the Model II is the better choice.

Still another thing in favor of the Model II is the memory. Since memory is clear after system boot, you have more flexibility in terms of programming languages. You don't have to waste 12K of memory with a ROM package that isn't used for the language of your choice!

From the talks I've had with a number of people, much good software is going to start hitting the market soon for the Model II. CP/M is already available. Microsoft Inc will have FORTRAN, COBOL, and a BASIC compiler soon, and most Business system houses are beginning to turn out Model II versions of their software.

Radio Shack has already released a General Ledger package, an Inventory Management System, a Mailing List, Payroll and Accounts Receivable. The future for the Model II owner look' bright, particularly if he is a busines' user ●

The 9th in a series of tutorials on Machine Language Programming...

View from the Top of the Stack

Jim Crocker, Technical Editor

Last issue, we discussed T-BUG, and some of the uses of a MONITOR program. In this issue, we will discuss the Radio Shack Editor/Assembler program, and some of it's uses.

As before, a definition is needed. Since the EDTASM is really two programs, we will provide two of them. The first part is the EDITOR. An editor is simply a progam that accepts input from a device (in our case, the keyboard or the cassette), and allows the user to make changes in the text that was input. Last but not least, the editor translates the text that you input into a format that is acceptable to another program.

That other program in our case is the ASSEMBLER. An assembler takes an input that is in a specific format, and translates the abbreviations and code words (called source) into something called an OBJECT. Object is just a fancy word that means 'machine code that is acceptable to the loader'. The loader that we are concerned with is the 'SYSTEM' command in our TRS-80, so the object output form the EDTASM is directly compatable with our SYSTEM command.

There are a few mere terms that we should become familiar with. One of these is 'OPCODE'. An opcode is to the EDTASM like our BASIC keywords (like CLS, DEFINT, etc.) are to Level II BASIC. In simpler terms, the opcodes are simply a shorthand that allows us to pack more

information into a smaller space. It also makes the job of the assembler easier, since there is no room for ambiguity in the very precise world of machine language programming.

Another word we should understand is FIELD. Fields were first developed in the wayback when the only viable input device was the punched card (also called the IBM card). These cards used holes punched in a cardboard card to represent letters and numbers. The holes were arranged into 80 columns (isn't it amazing how the printers of that time were 80-column devices?). When you wrote a program to input from a card reader, you arbitrarily decided that the first xx columns would stand for one thing, the next xx columns would stand for something else, and so on. We don't use punched cards, but the idea of columns has survived. The fields that we use are defined by the TAB (right-arrow) key. The first 6 columns are called the LOCATION or SYMBOL field. Tab once, and you are in the OPCODE field. Tab again, and you are at the OPERAND field.

The LOCATION or LABEL field is (you guessed it) where the LABELS live. A label is just what it says it is, a tag we put on something so we can recognize what it is later. In the case of the EDTASM, the labels are used to 'tag' addresses in memory or numeric values. For example, if we define the symbol 'CHARLIE' as being 5000H (H stands for HEX), we don't have to do anything with address 5000 anymore. Instead, we may load, store, jump, or whatever to CHARLIE and the assembler will figure out that we mean 5000. There is another thing that may be done with labels. Say, for example that you are typing in a program that is 1000 bytes long. We can simply put a symbol in the right place, and the assembler will figure out what it represents for us.

Opcode is a contraction of the term 'operation code'. There are a total of 158 opcodes that can be executed by the Z-80. Included in these 158 codes are all 78 of the opcodes that can be executed by the 8080. This provides what is known as 'upward compatibility' (now you know what that term means). For all you 8080 'old hands', the opcodes are in Zilog format, so you will have to learn them to use this assembler. The opcodes (or mnemonics, as they are sometimes called) are 'LD', 'ADD', 'SBC', etc, and comprise the first half of the term that tells the computer what we want to do.

The second half of this word lives in the OPERAND field. If the opcode tells the computer what to do, the operand tells it what to do it to. Or, if the opcode tells the computer to go somewhere, the operand tells it where to go. A 'LD' opcode tells the computer to LOAD something. A 'LD A,B' tells the computer to load the A register from the B register. Adding this second half of the word brings our total number of operations to somewhere around 700! This is where some of the subtle power of machine language programming (and the Z-80) comes into view.

Not all opcodes require an operand to be a complete operation. 'EXX', for example, tells the computer to exchange the 6 'primary' registers (B, C, D, E, H, and L) for the 'alternate' registers (B', C', D', E', H', and L').

The rest of the line of text is called the COMMENTS field. Comments are simply the little notes we use to tell ourselves and others what we are doing (not unlike the REM in BASIC).

The comment field is assumed to be preceeded by a semicolon, and as a result we can put a semicolon in the first character of the line, and the entire line becomes comment. Comments, by the way, are completely ignored by the assembler.

When we assemble a program, there are three separate outputs. They are the OBJECT, SIDE-BY-SIDE LISTING, and the SYMBOL TABLE. The object was discussed before. The SIDE-BY-SIDE LISTING is a combination of the addresses and hexcode generated by the assembler, and the source that is being assembled. They are placed beside one another, hence the name. A typical line of SIDE-BY-SIDE might look like this.

```
6000 3A0050   CHARLIE LD   A,(5000H)
```

The address is 6000, the hex code is 3A0050, and the rest is word-for-word from the source.

The SYMBOL TABLE is generated to tell us what symbols we used, and what hexadecimal address or value they represent. Some versions of the EDTASM arrange the symbol table in alphabetical order, others arrange them in the order in which they occur (I have one that dosen't arrange them, it just puts them in whatever order it feels like. I think it's reverse polish notation, but I'm not sure). The symbol table can be very useful when making changes to a program, or when writing one program what overwrites part of another.

Any of the three outputs from the assembler may be disabled with a SWITCH. Switches are hold-overs from the larger machine which have rows of switches that can be read by the computer, and acted upon by the program. Since we don't have any switches on our computer, we simulate them with letters. The command to assemble a program with no listing (side-by-side), no object output, and no symbol table would be 'A/NL/NO/NS'. The slash means that 'this is a switch', and the two letters tell which switch it is. Assembling a program like this will result in nothing being printed except error messages, which is the most important use of the switches. A message is printed:

00000 TOTAL ERRORS

BE YE AWARE that ANY errors in the assembly of a program will most likely result in a program that will not work!

An important part of any assembler are the 'PSEUDO-OPS'. A pseudo-op is used like an opcode, except that it does not assemble into machine code. Instead, they are used to give the assembler information. the pseudo-ops that are supported by the Radio Shack EDTASM are: ORG, EQU, DEFL, DEFB, DEFW, DEFS, DEFM, and END.

ORG is short for 'Origin', and tells the assembler where to put the program (or data) that follows.

EQU means 'equates to' or 'equals'. The primary use of the EQU pseudo-op is to define a label that isn't otherwise defined in the program.

DEFL means 'define label'. For the purposes of this discussion, we will assume that DEFL means the same thing as EQU, since for the most part it does.

DEFB means 'define one byte'. You can define a byte as a decimal number (DEFB 23), an octal number (DEFB 230), a hexadecimal number (DEFB 23H), or an ASCII character (DEFB 'A').

DEFW means 'define a [16-bit] word'. The syntax is the same as the DEFB. Note that if you DEFW'A', the resulting hex word is 4100.

DEFS is sort of like the CLEAR statement in BASIC. It simply reserves a 'string' of memory to be used later. In reality, it is just a block of memory that is left untouched by the assembler. You can reserve from 1 to 65535 bytes of memory, but there isn't much need for that much in a TRS-80.

DEFM is a very powerful pseudo-op, and means 'define memory as the string'. In other words, DEFM 'THIS IS A TEST' would result in the ASCII for 'THIS IS A TEST' being stored starting at the specified address. You can specify from 1 to 63 bytes of ASCII using this command.

Appropriately enough, END is the last pseudo-op we will discuss. Not unlike the END in BASIC, it means 'this is the END of my source, so stop!'. You MUST include an address or symbol after the END pseudo-op, as this is the address that represents the start address to the SYSTEM command. If you don't include one, the SYSTEM command won't shut off the tape, since it is looking for one. The address you give is the one that is used by the 'autostart' function (that's what is used when you answer "/ ENTER" to the second "*?".

There are a few other things about the EDTASM we should discuss. One of the things that it can do for us is to perform simple math and logical operations on our operands. Consider Fig 1, borrowed from the example in the EDTASM manual. The EDTASM will compute VIDEO+1 and 3FFFH-VIDEO for us, and will translate 191 into BF hex for us.

```
VIDEO   EQU   3C00H              ;start of VIDEO RAM
        ORG   7000H              ;start of program
START   LD    HL,VIDEO           ;HL now equals 3C00
        LD    DE,VIDEO+1         ;DE now equals 3C01
        LD    BC,3FFFH-VIDEO     ;BC now equals 400
        LD    (HL),191           ;graphics character
        LDIR                     ;fill screen
```

Figure 1.

There is some special syntax about entering hexadecimal numbers that is mentioned in the EDTASM manual, but it is very easy to miss. If the assembler finds a number in the operand field, it assumes a number. If it finds a letter, however, it assumes a symbol. Therefore, if you try to enter the value F0H, the assembler will assume you mean the symbol called 'F0', and will give an 'undefined symbol' error. To use a value of this type, precede the number with a zero. So '0F0' will generate the proper value.

That brings us to the final process, assembly. The assembler will do it's best to produce a useable output from whatever you give it, but sometimes circumstances prohibit it. 99 44/100% of the time it's because of an error in your source, and you can use the editor to fix it. Sometimes, however, there is simply nothing that can be done. These are called 'terminal errors', and consist of things like 'SYMBOL TABLE OVERFLOW'. That's the EDTASM's equvilant to an out of memory error. Unless you are trying to assemble a new interpreter in 16K, you shouldn't have any experience with that one.

Now that all that's out of the way, we will leave you now to go and find yourself an EDTASM (they are available at any Radio Shack store) and play with it a while. There are some examples in the manual, and some simple routines in the back of the book. READ those portions of the manual that explain how to use it. READ the 'introduction to instruction types on page 111. READ the explanations of the error messages on page 125.

Next time, we will take you step by step through the process of writing and assembling a machine-language program. We are on our way! ●

KNOW YOUR COMPUTER (from page 25)

separate, one physical unit can share both the input and output blocks, notably the cassette recorder and the modem.

The last block in our diagram is the memory. Sometimes called the "store" or the "core". In it, we can store information processed by the CPU, instructions to tell the CPU what to do and when, and intermediate results which the CPU will need at a later time. In our case, the memory is physically located in the keyboard unit. If you have an expansion interface unit then more memory may be located in it. Both electrically and functionally, it still looks like one block of memory to the CPU.

Now lets talk about how "dumb" the computer really is. Yes, dumb! You may think that with all the high-level words we use in speaking "computerese", that it must be a really intelligent device. It is and it isn't. It really does not know or understand anything about machine language or editors, compilers, HEX, ASCII or any of that. What it does know well is simply the presence or absence of a single voltage. The voltage, when present, is usually plus five volts. The absence of this voltage results in zero volts. Another bit of knowledge the computer possesses is "time". Time in the computer is generated by a voltage which turns on and off at regular intervals.(which is the same as saying that it has frequency and period). All operations in the computer are synchronized in one way or another with this "clock" or time.

When it comes to mathematical functions the computer only knows one thing: Addition. (Yes, that is a period after the word "addition"). It knows how to look at two voltage levels (in a given slice of time) and produce a zero if both the voltage levels were zero, a plus five volts if either of the levels were plus five volts, and a zero with plus five volts transferred to the next element if both levels were plus five volts.

Notice that the computer knows nothing about "Binary". Binary is a term we humans use. We interpret the plus five volts and zero volts as two "states". Since there are only two states, we can use binary (counting in a base of two) to define the state of the elements inside the computer. It is we humans who attribute the terms "high and low", "one and zero", "true and false", "active and inactive" to the two states the computer elements can take.

Subtraction in the computer can be accomplished by adding the compliment. The compliment of a series of elements with two states is simply changing all the "highs"to "lows" and all the "lows" to "highs". The process is still one of addition. Multiplication is simply repeated addition, and division is a series of successive subtractions which are complimented addition. So, the computer gets the job done, but it still only knows how to add - and that is all.

Lets define some of those "elements" we speak of above. One of the most elementary elements in the computer is the "flip-flop". The flip-flop is a storage device in that it can *hold* a high or low voltage level at its output. A "trigger pulse" (a zero level going to plus five volts and back to zero again) will change the "state" of the flip-flop, which means that if it previously held a high at its output, then after the trigger pulse it will hold a low there (even after the trigger has gone away). Flip-flops may be connected so that they represent a "counter", so that every time one changes from high to low for example, it will create a "carry" pulse to the next flip-flop in line. Thus, it becomes a "binary counter" and will, after the input of a series of trigger pulses, contain highs and lows such that they can be interpreted as a binary number. This number will usually represent the number of trigger pulses applied. It is important to remember that this string of flip-flops (or counter, or register) will continue to hold that count until it is reset, more pulses are applied, or the power is turned off. Additionally, most flip-flops have two outputs. These two outputs are always opposites. If one of the outputs is high the other must be low and vice versa. It is easy to see from this where the complement of a binary number comes from, as all we would have to do is read the "other" output of a series of flip-flops. Series of flip-flops are called "counters", or "registers". Although flip-flops possess the ability to "remember" or hold a count, they are not to be confused with the main memory in your machine. Your main memory is somewhat similar, but because of size and cost is considerably different.

Another element to consider is the "Inverter". This element is "forgetful", i.e., it does not remember and is not a memory type element. It simply inverts whatever voltage level is at its input to the opposite at its

(Continued on Page 60)

KNOW YOUR - (from Page 59)

output. If the level going in is zero with an occasional high-going pulse on it, the output will be high, going low each time the input goes high. Simply stated, it makes a "one" out of a "zero", or a "zero" out of a "one". (Another way incidentally, to get a complement).

So far, we haven't yet seen any real "logic". The element called a "gate" can show us some smarts though. The "Gate" element, in its simplest form, has two inputs and one output. Since there are two inputs, each capable of having two states, there are four possible combinations of input. Both inputs may be high (plus five volts), in which case the output is high. Neither input may be high, in which case the output is low. Either of the inputs may be high with the other input being low, in which case the output is low. If we label the inputs A and B and the output C, then we can say that we have an "and" gate, because it takes A "and" B to get a high at C. Now, are you ready for some confusion? The gate we just considered can be an "or" gate if we consider "lows" as the active signal. -- Lets try it: A low on A "or" B (or both) will produce a low at C. Again, the interpretation of this is human, since the machine is still operating on the presence or absence of voltage and doesn't know the difference between "And" or "Or".

Connecting gates, flip-flops and inverters, along with the clock, we can create some extremely complicated logic trees (sort of like "spit in the ocean" in poker, where one eyed jacks and the queen with the axe - only on Saturday etc etc).

There are variations on these elements. A gate may have more than two inputs for example. A gate may have an inverter built right into it, in which case the "and" gate becomes a "nand" (not and) gate, or an "or" gate becomes a "nor" (not or) gate.

Lets take a closer look at the CPU, since it is the heart of the system. First off, it is (speaking now of the Z80) physically about 2 1/2 inches long, 3/4 inch wide and about 1/8 inch thick. It has 20 pins along each side, and can be plugged into an appropriate socket on a printed circuit board. The price of such a chip these days is about $11.00. Through the wonders of micro-miniaturization this little package is crammed full of flip-flops, inverters and gates. It is composed primarily of registers (strings of flip-flops connected together) and many gates. These gates

(Continued on Page 62)

KNOW YOUR - (from page 60)

route signals (high and low voltage levels - or ones and zeros if you prefer) around to the various registers at appropriate times.

Functionally, the CPU contains an Arithmetic/Logic Unit (ALU), the CPU registers (there are 14 general purpose registers plus index, stack and program counter registers), an Instruction register, Data Bus controls and Address Bus controls.

It is possible through the use of gates to connect various registers so that all the voltage levels in one register are transferred at the same time into the second register. It is also possible through the use of gates to connect two registers end to end, thus increasing the length of the register (which is just what happens when you go to double precision in BASIC).

Without getting too involved, lets just say that the instruction register, upon receiving various combinations of highs and lows, sets up gating to cause the highs and lows in other registers to be manipulated into what we can call "addition" if the instruction was to add. Or to enable gates so that the Address Bus can sense a certain memory location (which also only contains zero volts or plus five volts) and let whatever is in that memory location be "felt" through the data bus to the ALU. Note that data is said to be "moved" to memory or from memory. It is not so much "moved" as it is "felt", "sensed" or "impressed".

At this point let us assume we have a front panel on our block diagram computer in figure 1. This front panel will become our "input" device. Lets say also that it has a row of 16 lights on it with a button under each light. It also has a rotary switch which is marked with all the registers in the CPU, so that by selecting register A for example, the plus five volts in each flip-flop of register will light the light on the panel. Now the lights which are on (and those which are off) will present us with a pattern of ones and zeros which we can read as a binary number. Lets say further that at the end of our row of front panel lights there is a button marked "clear". When we push "clear" all the lights go out, and register A is cleared. Now we can push the individual buttons under each light and "load" some binary value into register A. We can now program our computer in the most primative way, but one which the computer understands best. We can rotate our switch so that we can see the address register, then enter some memory location. Next we can switch to the data register and enter some value there (in binary of course). Then we can switch to the instruction register and enter the appropriate binary number which stands for "store in memory". It will, and this is exactly the way that early computers were programmed. If we give it a series of meaningful (possible) instructions and set the program counter to the right place and tell it to go, it would probably step through all the instructions and execute them assuming the program counter is incremented. After this, we can rotate our switch to the various registers and see what the results look like.

Words cannot describe the drudgery of programming this way. When is the last time you answered to : 1101100011107 (I did, just last week, but the answer is unprintable!).

What do we need now? Well, it sure would be nice if we could speak to the computer in something other than all of those ones and zeros! Lets play a little mind game and instead of talking in ones and zeros, lets group them into threes. Since three binary bits can count from zero through seven, let us mark off the lights on our front panel in groups of three starting from the least significant light. Lo and behold! We can now speak to the computer in Octal, or base 8 (a decided improvement over binary, at least the numbers look a little more like what we are used to). Now instead of writing 1100101001101100, we can call it 145154.

Using three fingers now we can poke those buttons with numbers that are a little easier to understand, since it is easy to learn to count up to seven in binary but more difficult after that. As you can see, the computer has not gotten smarter at all, has it? In fact, it is still chugging merrily along with its zero volts and plus five volts. We changed the whole picture by the way *we* perceive the number.

Time passes. You are now so fluent in octal it is second nature - you even dream in octal, a definite sign you have arrived in the land of eight. You suddenly realize that you are capable of counting to 16 if you really had to. Lets see - if we can group the lights in groups of four we can make things even simpler. But - four places in binary go all the way from zero through 15 in decimal - how are we going to designate the numbers above 9? How about A, B, C, D, E and F? (Can't you just see George Carlin on stage: "Like WOW man, its HEXadecimal!!").

Has the computer learned anything? No. It is our human interpretation which became a little more refined, thats all. (The computer is STILL handling only zero volts and plus five volts).

We are still punching buttons under lights on our front panel - but it is a little easier now. Especially when we try to communicate to other computer freaks. Earlier, we had a binary number 1100101001101100 (ugh) which we could speak of in octal as 145154. Now, all of a sudden, this same number in hexadecimal looks like CA6C - now, isn't that nicer? (The computer - still dumb - is forcing *us* to think. Maybe it is not as dumb as we thought!)

Every time we turn on our computer from figure 1 (with the added front panel - lets call it the Model Figment 1) we have to start from scratch. There is no program inside that will start things going - nothing. Lets say we have written a nice little program and punched it into memory. It is a useful little program, and we want to keep it, so we hook up an output device (cassette recorder) and include a section of code that will read the correct part of memory and save our program on tape. Outstanding! Turn off the power and go to bed. -- How are we going to get that program back in?.

Those kind of thoughts at 3 AM are not uncommon with computer buffs. Obviously, there is nothing in the freshly turned on computer to help us load the tape - after all, it needs some instructions to do that. So, we punch in a short routine that will act as a "loader". This is what is generally called a "bootstrap" loader, ie., it picks itself up by its own bootstraps. Now the loader loads our tape, after which we can execute our program (which was probably much longer than the loader.) It works - but humans are never satisfied.

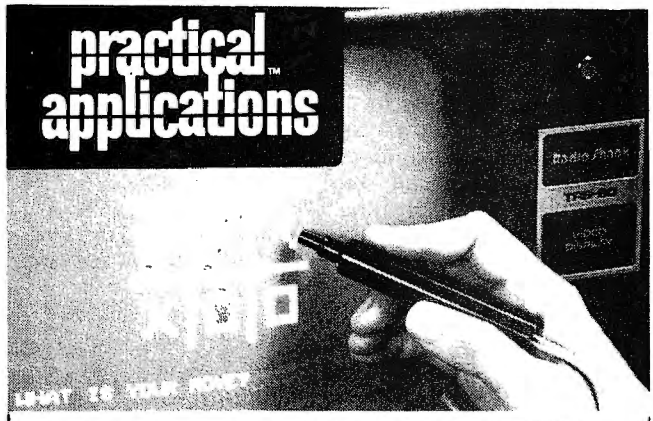
Punching in a loader each time you power up is a bore. There really ought to be a way to have the loader in the machine all the time, so that when you power up it would automatically execute itself. But how? Lets see - if we take a whole bunch of flip-flops and put plus five volts on either the preset or reset and leave it there all the time -- and then on power up find a way to force the program counter to read these flip-flops - thats it! We just made ourselves a Read Only Memory (ROM) ! Well, its not really that simple,

(Continued on Page 64)

Talked about at NCC '79, the Cover Feature of the August '79 CREATIVE COMPUTING, highly rated by COMPUTER CASSETTE Magazine and others, ADVENTURE by Scott Adams has rapidly become a classic. Each ADVENTURE is a 16K machine language program designed for you, the Armchair Adventurer! See for yourself what everyone is talking about by ordering one of our seven different Adventure tapes today from our many fine dealers, or order directly from:

ADVENTURE INTERNATIONAL
Box 3435, Dept. X
Longwood, Florida 32750
(305) 862-6917

Each Adventure, only $14.95, Visa and Mastercharge accepted, Send For Free Flyer!
Dealer Inquiries Encouraged
Available for: 16K TRS-80, 16K SORCERER



LIGHT-PEN FOR YOUR TRS-80

PLUGS RIGHT IN! Exclusive design includes two sample programs and complete documentation so you can write your own programs in Basic. Long life from standard 9-volt battery. **A bargain at only $24.95!**

PRACTICAL APPLICATIONS™ (415) 573-8217
Post Office Box 4139, Foster City, CA 94404

☐ Please send me _____ TRS-80 Light Pens ($24.95 each enclosed. Calif. residents add tax).
☐ Send your catalogs.

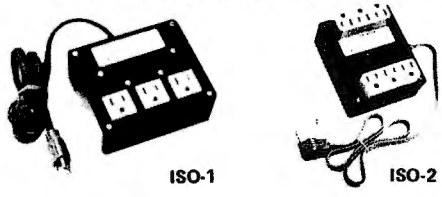
Name_____

Address_____

City_____State_____Zip_____

TRS-80 is a trademark of Tandy Corp. US779

DISK DRIVE WOES? PRINTER INTERACTION?
MEMORY LOSS? ERRATIC OPERATION?
DON'T BLAME THE SOFTWARE!



ISO-1 ISO-2

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

***ISOLATOR (ISO-1A) 3 filter isolated 3-prong sockets; integral Surge/Spike Suppression; 1875 W Maximum load, 1 KW load any socket $54.95**
***ISOLATOR (ISO-2) 2 filter isolated 3-prong socket banks; (6 sockets total); integral Spike/Surge Suppression; 1875 W Max load, 1 KW either bank $54.95**
***SUPER ISOLATOR (ISO-3), similar to ISO-1A except double filtering & Suppression $79.95**
***ISOLATOR (ISO-4), similar to ISO-1A except unit has 6 individually filtered sockets $93.95**
***ISOLATOR (ISO-5), similar to ISO-2 except unit has 3 socket banks, 9 sockets total . . . $76.95**
***CIRCUIT BREAKER, any model (add-CB) Add $ 6.00**
***CKT BRKR/SWITCH/PILOT any model (-CBS) . . . . . Add $11.00**

PHONE ORDERS 1-617-655-1532

ESP Electronic Specialists, Inc.
171 South Main Street

10 A$="

SUPER GRAPHICS

Don't you wish graphics were easy? Well, now they are! **PICTYPE** lets you type graphic characters into your BASIC program one rectangle at a time. No need to look up the character codes. Moreover, statements containing graphics can be LISTed and EDITed like normal statements. Graphics show up as graphics—not garbage. **PICTYPE** loads easily and saves as part of your program, giving you fast-printing, memory-efficient graphics every time you run it. So get **PICTYPE**, and program action-packed graphics like a pro!

PICTYPE on cassette for TRS-80 Level II/Disk BASIC, with instructions, postpaid: **$19**

DISCOVERY BAY SOFTWARE CO.
P.O. Box 464 Port Townsend, WA 98368

Bank Cards Welcome. Dealer Inquiries Invited.

KNOW YOUR- (from page 62)

but almost. Has the computer got more smarts? No, it just looks like it has.'

We're really having fun now folks! We have a computer which automatically reads in its own loader. Now we can write all sorts of neat things, save them on tape and load them whenever we want. But - we are still pushing buttons on the front panel in "four bit at a time" hex code. Yes, we have very comprehensive knowledge of the Z80 operation codes - the book is dog-eared and coffee-stained by now. Isn't there a better way? How about building a program that will take just one combination of bits and then automatically go through a sequence that will replace all the finger poking? Sure, it can be done. Then how about a way of setting breakpoints, so that execution can be stopped and the register displayed on the lights? That too, can be done. And if you put this program in a specific place in memory, reserved just for it - (here comes Carlin again, "Like WOW man, he's built a MONitor!!"). Yes it sounds almost like TBUG, in fact, if you look closely---

Shall we check the "smart quotient" of our computer again? No, don't bother - it hasn't really changed.

(Insert an interlude of "elevator" music here to denote the passage of time).

Well, time has sure passed fast, (glad they finally turned off the Muzak!). What? You are STILL not satisfied with your computer?? What do you want it to do; talk to you in plain English? Mmmm - why not? The Monitor idea was good, it worked, so why not build a real sophosticated monitor that will really allow us to give it plain language instructions? That way we won't have to fool around with these HEX codes. Excellent idea of course, this will really make the computer smart *(or will it)*. But how are we going to talk plain english to a bunch of buttons on the panel? And how to read plain English from a bunch of lights?

The most obvious way to input English would be a keyboard like typewriter - hey!, they use coded keyboards on teletype equipment, why not use one of them? And, they already have an 8 bit code set up to determine which key is which, it's called the American Standard for Information Interchange (or something like that) anyway, its ASCII for short - pronounced "Ask Key". We still need something in our computer that will check to see if a key is pushed, and if so, decode the code and do something with it. Why not put it into the new high-level monitor we are about to program? That was almost too easy--

Now to the output. There are two ways to go here - we can either use the same teletype as an output device, or make a Video Monitor from an old TV. Using the teletype would be easy, just shape the output back into ASCII and send it to the teletype. But that would be too easy - we like the challenge, right?

OK, lets build the video screen - what do we need? First we have to find a Character Generator ROM somewhere. This is necessary to get the ASCII bits changed into a character of the alphabet or a number or punctuation. Fine, now we have to find some memory that will hold the information which goes to the screen. Why? Because simply connecting the character generator to the main memory would cause all sorts of garbage on the screen. We need a separate section of memory, just for the screen, that can be connected to the main memory *at will* to update the screen. That way, the characters on the screen will remain stationary, instead of flying around.

Now we have to build an output section into our new monitor, and a control section (do we dare call it a Device Control Block?). Lets do it! Throughout all this it is becoming painfully apparent that our new monitor is going to have to be something extraordinary. We have given it a lot of chores to do, and it is also apparent it has to do a lot more yet!

What more? Well, for openers, if we are going to give it commands in plain English, it has to have built into it a "command table". Then it needs a way to take your input from the keyboard, decode it and check it against that table before it can take further action. And what happens if you misspell a word? Does the whole thing die? Or are we going to include something to check for syntax, so it can throw the ball back into our court? Sure, it needs all of that. How is it going to execute what we tell it? Wouldn't line numbers with each statement be ok? Lets let it execute by line number, in ascending order.

Interesting things happening here - all of a sudden we realize that if we incorporate a command "GOTO", followed by a line number, we can make the program loop!

Well, by now you are getting the idea and there is no point in holding back any longer - our monitor has just become a BASIC interpreter! *(Extremely simplified)*. In our figment 1 computer we have to load it into memory and go from there - but it works. (By the way, do you think our computer has more smarts than before? - No, it is hung up on zero volts and plus five volts - but *we* sure have come a long way!)

Now lets put our whole BASIC interpreter into a ROM and let our power-up sequence read that ROM into memory, so that we will automatically be in BASIC on power-on. It saves valuable memory space for other things doesn't it?

BASIC is sure nice, but did you notice how slow it is? With all the syntax checking and look up tables, and the moving things to memory and getting them back again - punching buttons was a chore, but at least it was fast!

Now that we have a nice way to input letters and numbers, lets see if we can write a different kind of program. One that will punch the buttons for us when we input symbolic code like "LD A" for "load A". (You're way ahead of me, aren't you?) That way we have the advantage of machine language speed (button punching directly into the registers) and still being able to talk in somewhat plain language. Since it is going to "assemble" the code for us, lets call it an "Assembler". We also need a means to make additions, deletions and corrections to this code, so lets build in an editor, so that now we have an Editor/Assembler!

But how can you use the Editor/Assembler if every time you power up you automatically go into BASIC? Shall we disconnect the ROM (with BASIC in it) with a switch? We could do that (but you know we won't). We are going to include a new command in BASIC that will do it for us - it will be called "SYSTEM", and is going to drop us out of BASIC so we can load and use our Editor/-Assembler. (Ever read SYSTEM COMMAND in that journal with the funny name, - oh yes, its called 80-US or something like that - anyway, there's a guy called "Pilgrim" in there who really comes across with the assembly language stuff).

All together now - has our computer got any new smarts? No - it still only knows nothing except zero volts and plus five volts. Then again, it made *us* think it had some real intelligence, and you have to give it some credit for that! ●

BOOK Reviews

Introduction to TRS-80 Graphics
by Don Inman
Dilithium Press
Portland, Or 1979
$8.95, 133 pages

Since the introduction of the TRS-80, we have seen a number of virtuoso craftsmen appear in the field of TRS-80 graphics. Most of us envy the apparent ease with which Leo Christopherson generates superb graphics characters and animates them.

It is no surprise then that we find a new book available for the beginner to start in this fascinating area. Make no mistake, this is primarily a beginner's book but even the more experienced can gain from reading it.

The book is based on the TRS-80 Level I graphics and so uses SET and RESET instructions to generate displays. It does not cover advanced graphics such as those found in the more popular games programs. There are no machine language controlled displays, no music, and no super fast graphics.

Before you take this as a reason NOT to read the book, look at what it can give you. The most elementary techniques of screen display, including SET, RESET, and PRINT AT are clearly explained and their use is illustrated with real programs that work!

Some particular discussions of interest were in chapter 7 ("Sitting Ducks") where a shooting gallery program is developed from start to finish, including techniques for generating moving ducks on the screen.

Chapter 6 talks about drawing curves, including making a figure with PRINT's and TAB's. The book even covers such techniques as three dimensional figures, graph axes, and making letters out of graphics.

The beauty of the book is that it carries you along well with programs you can run and play with to learn the techniques it is trying to teach.

If you want a solid grounding in graphics, suitable for Level I machines (but workable on Level II), this book is a good place to start.

Introduction to TBUG
by Don & Kurt Inman
Dilithium Press
Portland, Or 120 pages

As TRS-80 users become more sophisticated in the ways of the computer, more of them are turning to machine language programming to achieve the speed that is available as well as for the challenge. Being able to say to friends who were impressed by a program, "No, I programmed that myself, in Machine Language!", has become a goal for many.

As expected, a growing number of books have become available dealing with Assembly Languages for the Z80 or with the TRS-80. But, to this point, no one has produced a book that takes you through the most important tools of the Assembly Language programmer, those tools in his Machine Language Monitor program. No one, that is, until this book appeared.

Machine Language programming is a difficult and time consuming task. You must pay careful attention to everything that is happening in memory and keep track of hundreds of individual instructions. The new programmer might think that this is handled in much the same way as he normally handles a BASIC program. For most people that means program, test run, re-program, test run, etc. But this dosen't have to be the case.

When I first went in to get an Editor/Assembler, the store clerk tried to tell me that it was all I needed. He said that TBUG was just a poor cousin of the Editor/Assembler, How wrong he was!

TBUG (or any Debugging Monitor such as D-BUG in Disk systems) is a powerful tool that lets you find what is wrong with a Machine Language program or test out new sections of code in a controlled manner. You can look at what you are doing, step by step, set stop points where control is returned to you (called breakpoints) and much more.

Having justified the need for TBUG, it is clear that we need to know how to use it effectively. The manuals which come with it are reasonably clear, but they are just what they are written to be: manuals. To really learn the use of TBUG, you need more. This book provides it.

The book is written to work with either Level I or Level II systems, and it assumes little or no prior knowledge of Machine Language programming. It is not a text in programming though, it is a learning aid for the use of TBUG.

The book is arranged in "problems" instead of chapters. Each problem treats some programming technique and how TBUG is used in each. The first problem is "Talking to the Computer". It shows you how to get characters from the keyboard (using ROM routines) and display them on the screen.

The second problem is displaying data from memory. Then there is "Using the recorder", "Gaming", "Drawing your own Graphics", "Games with Graphics", and finally "Debugging with TBUG".

The book is very readable, and most important, it includes a large number

(Continued next page-)

of example programs with which to work. Most of the programs are written so that you could use an Editor/Assembler or TBUG to get them into memory. Either way, you will learn much about the system.

If you are serious about Machine Language programming, you should have a Debugging Monitor. TBUG is a good choice. If you have a Disk System, you already have D-BUG which will work as well. In fact, some of the exercises were done with TBUG, and some with D-BUG when I worked through the problems. The programs worked, and the debugging and other techniques from the book worked. Only the terminology of the commands was different.

If you still think that an Editor/Assembler is all you need to do Machine Language programming, then by all means read this book, work with TBUG or some other monitor, and you will wonder how you ever got along without it.

Learning Level II
by David A Lien
Compusoft Publishing
San Diego, Ca
$15.95 plus postage
352 pages

The Level I user's manual which came with your TRS-80 and was written by the same author is a classic. It is a unique book and served well, teaching owners of a very complex technical device how to master it and put it to work.

But who stayed with Level I for any length of time?

Levell II BASIC opened new doors for us. It increased the capability of our computers. With Level II we could do meaningful things. But the manual which came with Level II left a whole lot of us scratching our heads.

The information is there. It is simply not explained or elaborated upon. The power of Level II was locked up in the manual, instead of in our heads and imaginations.

Now, *Learning Level II* is available (the second half of the Level I user's Manual). It takes over where Level I left off. It begins by covering every update needed to make the Level I Manual fully compatible with your Level II computer. "Old timers" can skim these changes to come up to speed. New owners simply adjust

their Level I manual as they go, a chapter at a time. When finished, both sets of readers go right on to the rest of *Learning Level II*. It's simple and it really works since the same author wrote both of them.

As the name implies, it concentrates on Level II BASIC. Every inportant BASIC capability is explored in detail with the emphasis on learning how to use it. Also, you will learn how to use the Editor - a powerful tool for changing and correcting Basic programs both while learning and when writing those special programs. Dual cassette operation, using the expansion interface, the realtime clock, printers and other perhipheral devices are covered.

A special section deals with converting Level I programs to Level II - and you can understand it!

This is not just a book you read. You prop it up by your computer and follow through with "hands on" the computer (the only way to learn). Even if you think you have a good working knowledge of Level II, this book is bound to teach you something you didn't know (or forgot about) that will make programming easier for you.

It is a reference book which takes you through, step by step, all the fundamentals and nuances of Level II Basic. Since it has an index, you can refer back later for refresher study as needed.

The book contains 25 chapters, starting with a Level II Overview, through the Editor, Chasing Bugs and Errors, and AUTO line numbering. Part II goes into advanced concepts of *Learning Level II* and includes a discussion of the ASCII set, Strings in General, (LEN, DEFSTR, CLEAR and DIM), Search and Sort, VAL($), INKEY$, What Price Precision?, PRINT USING, the math functions, PEEKS & POKES, and Multi Dimension Arrays - just to name some.

Almost any chapter can be worth the price of the entire book.

The only thing wrong with this book is - Where was it when Level II first came out? And why didn't Radio Shack carry on with the fine start they had and include it with Level II?

Well, no use crying over spilt milk, it is here now, and is worth your while to look into. *Learning Level II* is available from *Compusoft Publishing, 8643 Navajo Road, San Diego, CA 92119* for $15.95 plus $1.45 postage and handling. If you own a Level II, you really should get this book

The Incredible Secret Money Machine by Don Lancaster Synergetics $6.95 159 pgs Howard W. Sams & Co

Don Lancaster is my choice for the patron saint of all the Walter Mittys in computerdom. He is an absolute genius at taking some big bucks project like a TV Typewriter and doing the same thing with a few dollars worth of unlikely parts. One of his current projects is a $25 pneumatic hardcopy interface. His prototype, using an aquarium pump for an air compressor, costs $16.00 and will drive an IBM Selectric typewriter at 160 words per minute!

In **The Incredible Secret Money Machine,** he does to big business what he did to video terminals with his $35. TVT-6 a few years ago. With his suggestions and your own interests, you can have your own small business and find independance, set your own working hours, and maybe even make a good living as well!

The book is outrageous, provocative, entertaining, and downright informative. From a countercultural standpoint, he tells how to start your business without capital, how to pay minimum taxes, get others to pay you for enjoying yourself, and even how to invest your profits. He details the traps to avoid with unforgettable rules like "don't mess with the eagle". Translation: Never tangle with the federal government, as they have a long memory, and unlimited time and money to make life miserable for you. For example, don't let the FCC know that you have clipped a home-built TV game to the antenna of a television set without paying the $3100. type approval fee. There is an excellent section in the book on how to write and get published. The sections on strategy and tactics will split your sides with amusement, but ignoring the information presented may just split your business into nothingness.

The Incredible Secret Money Machine is $6.95 postpaid from Synergetics, Box 1077, Thatcher AZ. 85552, and comes with a money back guarantee - a safe one, for you won't want to part with it. 159 pages, paperback, published by Howard W. Sams & Co.

Software Reviews

Temple of Apshai
Automated Simulations
16K L2 Cassette $24.95
Invasion Orion
Automated Simulations
16K L2 $19.95

The Temple of Apshai is on both sides of one tape. It comes with a fifty-four page book of instructions, referred to as the "Book of Lore", and two pages of additional instructions, all in a plastic envelope.

This game is roughly based on the game of Dungeons and Dragons, except the computer version requires only one player. You are first introduced to the Innkeeper, a burly, tough old bird who will help you choose a character and then sell you the equipment for your trip. All the while he will be trying to talk you out of as much of your silver as he can. After you have finished bargaining, you are asked to select a level of the temple you wish to explore. There are four levels, and each level makes it harder for you to survive. The treasure increases in value with increasing levels of play. It varies from magic arrows (useful against Antmen), to copper, gold and silver ingots. After fighting off such incredible monsters as Antmen, Centipedes, and Jellies, you pick up your treasure and must fight your way back through the maze of rooms to the Inn. There you are presented with a careful tally of your wealth, and given the chance to continue your adventure, or take the money and run.

The various commands at your disposal, plus the four levels of difficulty make this a demanding game for the beginner and hardened adventurer alike. Even after many games, one can easily become absorbed, and spend several hours slashing his way back through the dank and terrible rooms of the temple. Even after securing the treasure, the challenge is not gone. The return trip to the Inn is always more difficult since you are burdened by the weight of all of your treasure and the wounds you have suffered.

During your travels you often need to refer to the Book of Lore for an account of the room you are in. While you are trying to find the correct page, a monster may suddenly spring upon you before you have a chance to do anything about it. You can avoid this problem by hitting SHIFT@, which will freeze the display until you are ready to go on.

Even though the price seems a little steep, it is a very good game. With a trusty Broadsword, and a magic talisman or two, it is possible to conquer the world. The game is a great escape, and is well worth having.

C Quante

Invasion Orion
Automated Simulations
16K L2 $19.95

In our May/June issue we reviewed Starfleet Orion. This is the single player version of the same game. It shares many features with Starfleet Orion, including the same worksheet tablet, an attractively bound and thorough instruction booklet, a series of programs, including the game program, the game builder program, a graphics cover, and data for two of the ten new scenarios all of which are on the same cassette tape.

The scenarios constructed with the Starfleet game and the Invasion game will each work in the other game, expanding to a library of 22 games plus the capacity to do your own. Each of the scenarios is given a story line, and while this is well done in the first version, it is even better in the single player game.

I had two complaints about the first version, and they still exist in the second. The graphics consist only of lighted blocks on the screen with a number or letter beside them. However, I find them a little more pleasing to the eye in Invasion because a line has been added at the bottom of the screen for a reference point, the ships have been spaced better in the scenarios, and a larger display has been added for a planet, giving some improvement. My other complaint was about pacing, and this is also somewhat improved, as the computer enters its strategy a little faster than a human player.

The computer player is not as sophisticated as a human player and needs the advantage of a more powerful fleet for a fair game. However, this gives the human beginner a chance to play some easy games to learn, as the computer will play either side.

One particularly nice feature of the new version is the explanation of the computer strategy in the back of the book. It helps you to understand what is happening and to plan your own strategy and new scenarios.

Invasion Orion and Starfleet Orion are more than just good computer games. They are the first really successful blending of the war gaming hobby with the computer. War gamers have built up an elaborate game system over the past twenty years, and the benefit of that experience is clearly seen in Orion. A significant factor in the game's quality is that it is a team effort designed by Jim Connelley and Jon Freeman and tested by a number of war gamers. Jim also did the programming, Jon wrote the battle manual. Both are to be highly commended.

I like Invasion Orion even better than Starfleet Orion, though I have been told that the graphics changes will be made to the original version. Ratings:

Instructions:	Understandable
Documentation	Excellent
Challenge:	Very Good
Graphics:	Fair
Pacing:	Moderate to slow

Recommendation: Probably the best single person war game currently available for a 16K TRS-80. Buy it!

Invasion Orion is available from Automated Simulations, Dept 8U, P.O. Box 4232, Mountain View, CA. 94040. The price is $19.95 and it comes for Level II only. California residents add sales tax. Starfleet Orion, for two players, sells for $16.95.

G Blank

Periodical Cross Reference
by Dave Stambaugh
The Software Exchange
Milford, N.H. $19.95 Disk
$14.95 Cassette

Have you ever sifted through a stack of magazines trying to find that article on how to implement super graphics, or how to tune up your Subaru, or the strengths and weaknesses of that company whose stock you were considering as an investment? You know it is there somewhere - was it in 80-U.S. or Kilobaud? Was it in the April issue? No, it must have been in.....

Well, now that you have your own computer system, why not use it to do the work for you?

Periodical Cross-Reference Program by Dave Stambaugh uses your TRS-80 to catalog magazine articles by subject, publication, date, page and title. You can build a file using any subjects or publications you like. You can search through a file and list (to the Video or Printer) only those articles under a particular subject or publication, or you can list all the entries. You can edit, delete or sort the entries. You can load a file, make corrections or additions, and save it back on disk or tape. The disk version also includes a tape-to-disk conversion program for your data tapes in case you upgrade to disk at a later time, and a sample data file.

The two versions of Periodical Cross-Reference and the minimum system configuration for each are:

Cassette Version - 16K Level II
(approximately 120 entries per file)
Disk Version - 32K single disk
(approximately 250 entries per file)

If you have more memory, the program will allocate space for more entries.

This was an interesting program to review. I put it through it's paces and tried every option. The program is menu-driven throughout, and when you have no printer it ignores LPRINT, thus preventing lock-outs.

It sorts a file by date, publication, page and will alphabetize the subject classes and publications. The video display is "paged", so that files longer than 16 lines will remain on the screen.

If publications are not your bag, this program can easily be modified (simply change the heading names) to keep virtually any type of reference list.

The program falls into the class of "meaningful" things to do with your computer, and the price makes it an exceptional value.

IMS

Space Battle
by R. Papo
Level IV Products
Livonia, Mi
16K L2 Cassette $14.95

Since I have already reviewed 7 space war games in previous issues of 80-U.S., it would take a good game to bring another review. This is another space war game review, and you may safely conclude that I have found another good one.

In many ways Space Battle, from Level IV Products, is like Time Trek, the top rated Star Trek game in my Jul-Aug 79 80-U.S. review. The display routines are in machine language, it is real time, it is tough, and the enemy maneuvers on you during battle. Actually, it is not quite as good in the battle routines themselves, largely because action alternates between the player and the enemy, while in Time Trek your ship is always under your control. The extra control makes Time Trek a better action game.

The unique value of Space Battle lies in the extra dimensions of the game. Instead of cruising the galaxy with an unlimited supply of taxpayer financed weapons, supporting services and energy, you are a mercenary. In Space Battle, you must pay for the energy you use, the torpedoes you fire, repairs accomplished at the space station, even funeral expenses for crew members killed in battle or dead of radiation sickness. You receive a bounty for each enemy ship destroyed to pay for these expenses, and hopefully make a profit. As if you did not have enough to worry about between fighting the enemy and making a profit, there is a further consideration. If you lose too many of your crew, you may not be able to fully replace the lost members, as the potential recruits at the space station will be afraid to serve under you.

Actually, it is not too hard to make a profit. What is tough is staying alive. In the battle routines, the enemy gets first shot, and they shoot pretty well. One good hit and it is all over for you, sometimes before you even get a chance to fire. Successful strategy is based on the fact that the enemy has a difficult time locating you in the four corners of the screen. If, instead of charging boldly into a quadrant with your hyperdrive, you ease into one of the corners quietly with your reaction drive rockets, you have a much better chance of survival. The only way I have won the game was by never entering a quadrant containing an alien without such advance planning.

In order to keep your mind on what you are doing, you must also remember that attempting to go too far can damage or destroy your drive units, putting too much power into your laser can damage that, and if your shields use more energy than your ship has left, the reactor blows up, killing you. In short, this is a challenging game with a lot of factors to create interest. Do not expect to become expert at it easily.

The graphics routines are nothing special, showing only your ship and the enemy in uninspired profiles, with the exception of the star base routine. If you move into a quadrant with a star base, the huge space station opens up and draws you inside with a tractor beam. It can be lots of fun to watch.

The game is supplied in BASIC, and loads its own machine language subroutines. You must protect memory prior to play. It is available for 16K Level II on cassette.

Ratings:

Challenge	Tough
Graphics	Fair (Star Base is excellent)
Pacing	Fast
Content	Excellent
Value	Good to very good
Animation	Good

Cost: Level II Cassette (for 16K) $14.95

Order from: Level IV Products 32238 Schoolcraft, Livonia, MI 48154, OR from The Software Exchange, PO Box 68, Milford, NH 03055, OR from 80-US Software, 3838 S. Warner St, Tacoma, WA 98409

Recommendation: One of the best space war games available.

G Blank

A User Report

Racet Computes

GSF/DOSORT

TR Dettmann, Associate Editor

The never ending search for better utilities is an important one for most programmers. Particularly on Microcomputer systems; good utilities are necessary because of limited memory and speed.

Even "good" large system programmers often write bad programs for Microcomputers because they are not accustomed to the low speed that is normal with a Micro.

In one case I know of, a business program was developed for a firm with a straight bubble sort programmed in to handle several hundred items. On a Mini (or Maxi) computer, the businessman would have been done in minutes. On the TRS-80, the sorting could take hours if the list gets over about 200 items. Sorting improvements *(see 80-US Nov-Dec 79)*can help, but using a better sort to start with is more important.

The people at **RACET COMPUTES** *(702 E Palmdale, Orange, CA 92665)* may have given us the fastest in-memory sort possible for the TRS-80. It makes *any* BASIC program sort slow by comparison. For our article *(Super Sorting, 80-US, Nov-Dec 79)* we sorted a file of 450 names and addresses in about eight and one half minutes using VARPTR techniques. This was good, considering that previously the program had taken about 4 hours.

Running the same file with the RACET sort facility in their GSF (*Generalized Subroutine Facility*), we get it sorted in about 3 seconds!

Are GSF and DOSORT worth the $24.95 price tag? What are they? - Let's take a look at both packages.

GSF

The Generalized Subroutine Facility is just what the name implies, a generalized library of in-memory routines for use with your programs through USR calls. There are seven basic capabilities of the package:

(1) Display Screen Control
(2) Drawing Lines
(3) Duplicating Memory
(4) Moving Data
(5) Compressing & Uncompressing Data
(6) Reading and Writing Tape Data
(7) In-Memory Sorting

These capabilities reside in high memory, in memory protected machine language programs.

Let's see what they can do. For display screen control, we can scroll the screen up, down, left or right, and reverse the video if we want. This is not a complete reversal, since you can specify those sections of the screen where reverse video is desired.

The line drawing routines draw either vertical or horizontal lines incredibly fast. Lines can be made to flash at you on the screen so fast you won't believe it is really your TRS-80! The memory duplication routines can be used to set variables to zero, fill in large blocks of a screen display, or anything where you want some portion of memory to quickly take on the same value.

The combination of the moving data routines with the data compression and de-compression routines make a powerful programming tool. Screen displays can be stored in memory, either in packed or unpacked form, and swapped with another similar display. You can save a whole screen in pieces or several screens and bring them back as you want them.

Imagine the possibilities for really good text editors using this type of screen behavior.

The tape writing and reading routines are designed to rapidly perform dumps to and from tape without tape leaders between the data items. We have already indicated how fast the in-memory sort is, but we didn't mention that you can do this super fast sort either on strings or on multiple variable fields (including several different data types).

DOSORT

A major difficulty with any type of business application is that files often get larger than can be easily handled in available memory. What do you do when you can only hold 450 names on your 4000 name mailing list in memory at a time?

The normal answer to that question is to split the file down by zip codes (or other logical breaks). If that gets to be too much, then split each division further into halves, quarters or whatever. Clearly, this will eventually make for an unwieldy file. DOSORT was made to prevent this.

DOSORT is a series of 5 BASIC programs that can be modified to suit by the user (within some rather clearly spelled out limits) that will allow sorting and merging of really large, multi-disk files. This is the kind of capability programmers on large systems are accustomed to.

The DOSORT programs use the GSF routines (GSF is part of the DOSORT tape, so you get both packages when you get DOSORT). Most particularly, for the in-memory part of the sorting process, DOSORT uses the GSF in-memory sort.

DOSORT is actually a very complex SORT-MERGE facility that will break any file down into manageable chunks, sort them, store them temporarily, and then merge them into one sorted file at the end. Sorts may be done on either single strings or multiple variable fields with keys selected by the user, either ascending or descending.

Evaluation

I was rather skeptical of the claims for GSF, most particularly the sort (450 items in 3 seconds, come on now! *The Ads couldn't mean that!*). But once I started using the routines, their performance quickly convinced me.

The GSF tape has two sample programs which show off the capabilities of the system. The graphics displays are quick and clean; better than I have seen with most such systems.

The most impressive display was the sorting demonstration. I modified the routine for string sorting to read in a mail list file of 430 names and then sort it. Before and after the sort, I had the system print the time on the screen. Only three seconds elapsed between the first and last print.

The sorting time didn't vary appreciably with the size of the file. Cutting the file in half might at best cut the sort time down by a second or so - I won't quibble about it.

DOSORT seemed less impressive at first. After loading GSF and executing the first of the five SORT programs, you are asked a number of questions to specify the file for the program as well as the sort you would like.

Unless you will be using the type of file the designers had in mind when they wrote the program, you will have to change some dimensions, some INPUT statements, and some PRINT statements. These changes are pointed out in the manual for the user, and it is not hard to do. Any other changes in the program require a considerable amount of care because of the structure of the program and its memory usage.

If you specify your file closely and it is small enough, the whole sort can be done in memory. If the file is fairly large, then sub-files will be created and a sort-merge operation will take place. This takes time (about four to five minutes for one file of long strings with about 450 entries). Even this is fast though, compared to what many sorting programs are capable of with in-memory sorts.

DOSORT really begins to shine when you get "impossible" sorts, i.e., files that are too large to fit into memory at one time. DOSORT then uses a series of work disks as well as the main disk to sort the big file into a series of smaller ones, and then merge them back together.

When sorting with work disks, the program allows you to name each disk arbitrarily and then prompts you to load the correct disk by name as it needs it. This makes the DOSORT almost (but not 100%) foolproof. It is still possible to make the program mess up, but it is thouroughly error trapped and so you should not lose your data except by some fluke.

Do you need GSF/DOSORT?

Most users won't really get the full benefit of either package. The number of users who MUST sort large, multiple disk files is relatively small. Still, I would highly recommend the package to anyone who wants to have a fast, efficient sort or good machine language graphics.

For most users, GSF is quite enough. The DOSORT package is overkill and is far less efficient than simply calling the GSF routines directly from your own program. But, if you have to do complicated, large file sorting and merges, then by all means, use DOSORT.

How easy is it to use?

I have already mentioned that DOSORT is really just a matter of answering questions about your files. There is very little more programming to be done. Some users may be afraid of accessing machine language routines in GSF from their BASIC programs. There is little need for this fear. Once loaded in memory, a call to a USR routine is simply a matter of specifying the routine number, and then calling with the parameters required. (Of course, you have to use DEFUSR to tell the system where to find the routines, there may be a little POKEing for some initialization, but that is minor).

As an example of how simple a call can be, to do an in-memory sort of a character string array (say a list of names), you have to pass:

the GSF routine number (18)
the location of the first item in the array to be sorted (VARPTR)
the starting array index for the sort
the ending array index for the sort

These parameters are passed in seperate calls like this:

1428J=USR(18) OR USR(VARPTR(S$(0)) OR USR(100) OR USR(VARPTR(IE(0))

When control returns from the subroutine, nothing else need be done, the array is sorted.

Simple calls are the major feature of this package, and place it above the rest. ●

BASE CONVERSION

James Ranney, Las Vegas, NV

If you do assembly language programming or need base 2, 8 or 16 numbers, this conversion program should be a great time saver for you. The program has two main parts, the first part checks to see that you have entered a valid number, and the second part does the conversion. The program will run in Level II in less than 4K of memory.

You may enter any number from 0 to 1111111111111111 in binary, 0 to 177777 in octal, 0 to 65535 in decimal or 0 to FFFF in hexadecimal. You must follow the number you enter with a letter to designate its base. B for binary, O for octal, D for decimal and H for hexadecimal.

Lines 190 through 260 check to see if you entered too large a number, left off the trailing letter or entered an invalid number. An error message is printed out if you made an error and you are then prompted to re-enter the number. The reason for the size limit of the number is that the computer will print out any number larger than those above in scientific notation in the binary answer, also the numbers are the largest that can be entered in 16 bits and equal the highest memory location.

If the number you entered was not a decimal number, then lines 640 to 780 convert it to decimal, and lines 790 to 960 convert it from decimal to the other bases. The numbers are entered as strings and converted to integers as needed to perform the necessary math. Lines 1020 to 1080 are necessary to print out the hexadecimal equivalent without blanks between the alphanumeric characters.

Lines 570 to 620 demonstrate a way to make use of the INKEY$ statement. It has the same effect as using the input statement, except there is no cursor displayed and the screen does not scroll upward when the correct character is entered. (This may be used in a graphics display without destroying the graphics due to scroll up). ●

For Level II 16K and UP

```
2 'BY JAMES RANNEY, LAS VEGAS, NEVADA
10 CLEAR 150:CLS:DIM B(16),B$(16)
20 A=10:B=11:C=12:D=13:E=14:F=15
30 DATA A,B,C,D,E,F
40 PRINT:PRINT
50 PRINT" ENTER ANY WHOLE NUMBER YOU WANT TO CONVERT FROM 0 TO
```

```
60 PRINT" 1111111111111111 BINARY OR 0 TO 177777 OCTAL OR 0 TO
70 PRINT" 65535 DECIMAL OR 0 TO FFFF HEXADECIMAL FOLLOWED BY A
80 PRINT" LETTER TO INDICATE THE TYPE OF NUMBER: B FOR BINARY,
90 PRINT" O FOR OCTAL, D FOR DECIMAL AND H FOR HEXADECIMAL.":GO
   TO 140
100 PRINT:PRINT" NUMBER TOO LARGE":GOTO 130
110 PRINT:PRINT" LETTER MISSING OR WRONG":GOTO 130
120 PRINT:PRINT" NUMBER FORMAT WRONG
130 PRINT:PRINT" ERROR - REDO
140 PRINT:INPUT I$
150 I=LEN(I$)
160 II=I-1
170 II$=LEFT$(I$,II)
180 LT$=RIGHT$(I$,1)
190 IF LT$<>"B"AND LT$<>"O"AND LT$<>"D"AND LT$<>"H"GOTO 110
200 IF LT$="B"AND II<=16 THEN Y=49:GOTO 240
210 IF LT$="O"AND VAL(I$)<=177777 THEN Y=55:GOTO 240
220 IF LT$'D"AND VAL(I$)<=65535 THEN Y=57:GOTO 240
230 IF LT$="H"AND II<=4 THEN Y=72:GOTO 240  ELSE GOTO 100
240 FOR N=1 TO II
250 IF LT$<>"H"AND MID$(II$,N,1)>="0"AND MID$(II$,N,1)<=CHR$(Y)
    GOTO 280
260 IF LT$="H"AND MID$(II$,N,1)>="0"AND MID$(II$,N,1)<="9"OR LT
   $="H"AND MID$(II$,N,1)>="A"AND MID$(II$,N,1)<="F"GOTO 280
270 GOTO 120
280 NEXT
290 IF Y=49 GOTO 330
300 IF Y=55 GOTO 390
310 IF Y=57 GOTO 450
320 IF Y=72 GOTO 510
330 'BINARY
340 L=2:GOSUB 640
350 L=8:GOSUB 790  :O$=G$
360 L=16:GOSUB 790  :H$=G$
370 CLS:PRINT:PRINT:PRINT" THE BINARY NUMBER YOU CHOSE WAS: ";T
   AB(41)II$
380 GOSUB 990  :GOSUB 1000 :GOSUB 1010 :GOTO 570
390 'OCTAL
400 L=8:GOSUB 640
410 L-2:GOSUB 790  :B$=G$
420 L=16:GOSUB 790  :H$=G$
430 CLS:PRINT:PRINT:PRINT" THE OCTAL NUMBER YOU CHOSE WAS: ";TA
   B(41)II$
440 GOSUB 980  :GOSUB 1000 :GOSUB 1010 : GOTO 570
450 'DECIMAL
460 D$=II$:L=2:GOSUB 790  :B$=G$
470 L=8:GOSUB 790  :O$=G$
480 L=16:GOSUB 790  :H$=G$
490 CLS:PRINT:PRINT:PRINT" THE DECIMAL NUMBER YOU CHOSE WAS: ";
   TAB(41)II$
500 GOSUB 980  :GOSUB 990  :GOSUB 1010 :GOTO 570
510 'HEXADECIMAL
520 L=16:GOSUB 640
530 L=2:GOSUB 790  :B$=G$
540 L=8:GOSUB 790  :O$=G$
550 CLS:PRINT:PRINT:PRINT" THE HEXADECIMAL NUMBER YOU CHOSE WAS
   : ";TAB(41)II$
```

```
560 GOSUB 980  :GOSUB 990  :GOSUB 1000
570 PRINT:PRINT" IF YOU WANT TO ENTER ANOTHER NUMBER HIT 1, IF
    NOT
580 PRINT" HIT 0 TO RETURN TO BASIC.
590 M$=INKEY$
600 IF M$="0"GOTO 630
610 IF M$="1"GOTO 10
620 IF M$<"0"OR M$>"1"GOTO 590
630 END
640 'CONVERT BINARY,OCTAL OR HEXADECIMAL TO DECIMAL
650 D1$=MID$(II$,1,1):D1=VAL(D1$)
660 FOR P=10 TO 15
670 READ P$
680 IF D1$=P$ THEN D1=P
690 NEXT:RESTORE
700 FOR N=1 TO II-1
710 IF II=1 GOTO 780
720 D2$=MID$(II$,N+1,1):D2=VAL(D2$)
730 FOR P=10 TO 15
740 READ P$
750 IF D2$=P$ THEN D2=P
760 NEXT:RESTORE
770 D1=D1*L+D2
780 NEXT:D$=STR$(D1):RETURN
790 'CONVERT DECIMAL TO BINARY,OCTAL OR HEXADECIMAL
800 G$=" ":NI=0
810 BL=VAL(D$)
820 FOR N=1 TO 16
830 IF BL=0 THEN NI=1
840 BI=INT(BL/L)
850 B(N)=BL-BI*L
860 FOR P=10 TO 15
870 READ P$
880 IF B(N)=P THEN B$(N)=P$
890 NEXT:RESTORE
900 BL=BI
910 IF B(N)<=9 THEN B$(N)=STR$(B(N))
920 IF NI=1 GOTO 940
930 NEXT
940 FOR NN=N-1 TO 1 STEP-1
950 G$=G$+B$(NN)
960 NEXT
970 RETURN
980 PRINT:PRINT" IT'S BINARY EQUIVELANT IS: ";TAB(40)VAL(B$):RE
    TURN
990 PRINT:PRINT" IT'S OCTAL EQUIVELANT IS: ";TAB(40)VAL(O$):RET
    URN
1000 PRINT:PRINT" IT'S DECIMAL EQUIVELANT IS: ";TAB(40)D$:RETUR
    N
1010 PRINT:PRINT" IT'S HEXADECIMAL EQUIVELANT IS: ";TAB(41);
1020 IF H$=" " PRINT USING"#";VAL(H$):RETURN
1030 Q=LEN(H$)
1040 FOR N=1 TO Q
1050 HH$=MID$(H$,N,1)
1060 IF HH$>="0"AND HH$<="9"THEN PRINT USING"#";VAL(HH$);
1070 IF HH$>="A"AND HH$<="F"THEN PRINT HH$;
1080 NEXT:PRINT TAB(62)" ":RETURN
```

FOR 4K Level II and UP...

HEXDUMP

Louise Frankenberg, Pasadena, MD

Here is a handy routine for TRS-80 Level II or Disk BASIC (or other Microsoft BASIC's). It prints a nice hexadecimal memory dump which has all sorts of neat uses, especially for those with a printer, such as:

- compact hard copy of a machine-language program for your files (invaluable if tape or disk programs must be restored);
- an easy way to pass along a program to a friend, without all the fuss of copying source code;
- an easy way to pass along a program to an enemy (he'll tear his hair out trying to relocate it without the source code);
- for non-disk users, a quick way to examine chunks of memory in order to locate programs, addresses that need changing, bad bytes, etc. - much faster than one-byte-at-a-time TBUG;
- for disk owners, a quick hard-copy snapshot of memory areas found of interest under DEBUG.

HEXDUMP's output is similar to that of DOS's DEBUG. Each line starts with a divisible-by-16 hex address and is followed by the contents of that and the following 15 addresses; the dump thus may include up to 15 extra bytes on the first and last lines. This format was chosen in order to make possible the easy copying of HEXDUMP'ed programs into memory by the use of DEBUG.

A typical HEXDUMP run is shown in Figure 1. Starting and ending addresses may be specified in either hex or decimal, and the dump may be sent either to the screen or printer.

The following HEXDUMP routines may be of use to readers in other programs:
Lines 250-320 and subroutine 560-580 convert a decimal address in AD to a four digit hex address in AD$. In order to convert the least significant digit, line 310 should be changed to read

```
N=AD-(INT(AD/16)*16):
GOSUB 570:AD$=AD$+D$
```

Lines 370-430 plus the same subroutine (drop the '+X' in 370) convert decimal number 'A' (when less than 256) to a two digit hex number in ME$.

Subroutine 490-540 converts a four digit hex number in N$ to a decimal number in N. ●

```
H E X  D U M P

DO YOU WANT HARD COPY (Y OR N)? Y
WILL YOU GIVE ADDRESSES IN HEX (H) OR DECIMAL (D)? H
STARTING ADDRESS? 68BA
ENDING ADDRESS? 68FA_

68B0     00 00 00 00 00 00 00 00 00 00 E5 68 64 00 93 20
68C0     2A 20 48 45 58 20 44 55 4D 50 20 42 59 20 4C 4F
68D0     55 49 53 45 20 48 20 46 52 41 4E 4B 45 4E 42 45
68E0     52 47 20 2A 00 EB 68 6E 00 93 00 05 69 78 00 84
68F0     3A B2 22 48 20 45 20 58 20 20 44 20 55 20 4D 20
```

Sample RUN

```
100 REM * HEX DUMP BY LOUISE H FRANKENBERG *
110 REM
120 CLS:PRINT"H E X  D U M P":PRINT
130 INPUT"DO YOU WANT HARD COPY (Y OR N)";Y$
140 INPUT"WILL YOU GIVE ADDRESSES IN HEX (H) OR DECIMAL (D)";A$
150 INPUT"STARTING ADDRESS";ST$
160 INPUT"ENDING ADDRESS";EN$
170 CLS
180 REM * CONVERT ADDRESS TO DECIMAL IF GIVEN IN HEX *
190 IF A$="D" THEN ST=VAL(ST$): EN=VAL(EN$):
    ELSE N$=ST$: GOSUB 490: ST=N+1: N$=EN$: GOSUB 490 :EN=N+1
200 REM * CALCULATE 1ST & LAST ADDRESS TO BE PRINTED *
210 FR=INT(ST/16)*16: LA=INT(EN/16)*16
220 REM
230 REM *PRINT ADDRESS FOLLOWED BY CONTENTS OF 16 MEM LOCATIONS
240 FOR AD=FR TO LA STEP 16
250   AD$=""
260    REM* CALCULATE HIGHEST TO LOWEST HEX DIGITS
270   N=INT(AD/4096): GOSUB 570: AD$=AD$+D$
280   N=INT((AD-(N*4096))/256): GOSUB 570: AD$=AD$+D$
290   N=INT((AD-(INT(AD/256)*256))/16): GOSUB 570: AD$=AD$+D$
300    REM* RIGHT DIGIT ALWAYS ZERO
310   AD$=AD$+"0"
320   IF Y$="Y" THEN LPRINT AD$;"     ";ELSE PRINT AD$;"     ";
330    REM* CONVERT TOO-LARGE NUMBERS FOR PEEK OPERATION
340   IF AD>32767 THEN A=(-1)*(65536-AD) ELSE A=AD
350    REM* CONVERT 16 MEM LOCATIONS AT A TIME TO HEX AND PRINT
360    FOR X = 0 TO 15
370     ME=PEEK(A+X)
380     ME$=""
390     REM* LEFT DIGIT
400     N=INT(ME/16): GOSUB 570: ME$=ME$+D$
410     REM* RIGHT DIGIT
420     N=ME-(N*16): GOSUB 570: ME$=ME$+D$
430     IF Y$="Y" THEN LPRINT ME$;" "; ELSE PRINT ME$;" ";
440     NEXT X
450     IF Y$="Y" THEN LPRINT ELSE PRINT
460 NEXT AD
470 END
480 REM * ROUTINE TO CONVERT HEX ADDRESS TO DECIMAL
490 N=0
500 FOR X = 1 TO 4
510     X$=MID$(N$,X,1)
520     IF ASC(X$)<=57 THEN V=VAL(X$) ELSE V=ASC(X$)-55
530     N=N+(V*16↑(4-X))
540 NEXT X
550 RETURN
560 REM * ROUTINE TO CONVERT DECIMAL 10-15 TO HEX A-F
570 IF N>9 THEN D$=CHR$(N+55) ELSE D$=RIGHT$(STR$(N),1)
580 RETURN
```

DATA BASE MANAGER IDM-IV $69
You can use it to maintain a data base & produce reports without any programming. Define file parameters & report formats on-line. Features key random access, fast multi-key sort, field arith., label generator, audit log. MOD-II version with more than 50 enhancements $199.

ACCOUNTS RECEIVABLE ACCT-III $69
One or more drives. Order entry calculates sales tax, shipping, amount for multiple items. Credit checking, aging, sales analysis, invoices, statements and reports. As opposed to most other A/R, ours can be used by doctors, store managers, etc. MOD-II version $149.

WORD PROCESSOR 16K $39. 32K $49. MOD-II $49.
First word processor specifically designed for the TRS-80 that uses disk storage for text. Written in BASIC. No special hardware and text limit. Use for letters, manuals & reports. 32K version features upper/lower case without hardware change and multiple input text files.

MAILING LIST advanced MAIL-V $59.
Fast sort by any field. Multiple labels and reports. 4-digit selection code, new zip code ext., screen input, live keyboard, powerful report writer. MOD-II $99.

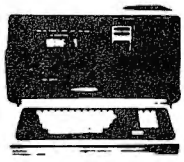
INVENTORY INV-V $99.
9-digit alphanumeric key for fast key random access. Reports include order info, performance summary, etc. Calculate E.O.Q. Powerful report writer. MOD-II $149.

All programs are on-line, interactive, random access, virtually bug free, documented and delivered on disks. MOD-I requires 32K, DOS. We challenge all software vendors to offer low cost manuals so you can compare and avoid those high-priced undocumented, 'on-memory' programs. Send $5 for a MOD-I manual and $10 for MOD-II.

MOD-II programs are extensively modified, guaranteed to run with 1 year newsletter & updates. 10% off for ordering more than 1 MOD-II programs.

MICRO ARCHITECT
96 Dothan St., Arlington, MA 02174

Radio Shack DEALER
COMPUTER SPECIALISTS



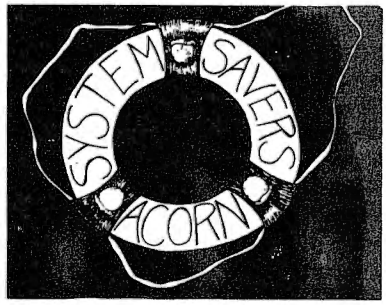
15% Discount on TRS-80's - I AND ACCESSORIES

FAST 100 cps Centronics 730 PRINTER. . $800.00
HIGHLY RELIABLE LOBO DRIVES. . $375.00

MICRO MANAGEMENT SYSTEMS
DOWNTOWN PLAZA SHOPPING CENTER
115 C SECOND AVE. S.W.
CAIRO, GEORGIA 31728
912-377-7120

TAPE OR DISK WITH

Complete your TRS-80* with these routines not found in either Level II or DOS.



SYSTEM SAVERS

by Tom Stibolt

If you ever use the SYSTEM command, you can use this two program package. These programs allow you to save any system format program on tape or disk, plus offer several features for machine language programmers.

With FLEXL, which is one of the two programs, you can make back-up copies of any system format tape. Most often a cassette that you make will load easier than an original. Plus you can find the filename on any system tape because it is displayed on the screen.

Disk drive owners can use TDISK to save any system format tape on disk. *"Air Raid", "Editor/Assembler"* and other programs cannot normally be loaded to disk. Now TDISK allows you to save these programs onto disk. After DOS READY you will be able to simply type the filename and be up and running. It even loads non-contiguos tapes. TDISK will greatly increase the benefit of owning a disk drive.

Acorn produces several other utility programs for the TRS-80. These include *"Aterm"* and *"Numbering"* by Tom Stibolt; and *"Disassembler", "Tape Utility"* and *"Disk Utility"* by Roy Soltoff. All are available for less than $20.00. Ask for these and other quality Acorn programs at your local computer store.

*TRS-80 is a trademark of Tandy Corp.

DEALER INQUIRES INVITED.

Acorn
Software Products, Inc.

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

80-U.S. Software

NEW! CAT2/XFR by Don Fielding! CAT2 is a disk directory program that will read, sort, organize and store your directories on an index disk. It gives you alphabetized listings of what is on your disks. Simply give the program a disk ID and put the disk to be read into the drive. CAT2 will do the rest. **BONUS!** Along with CAT2 you get **XFR**, a basic program which will selectively transfer only those programs you choose from one disk to another with the computer unattended! Has the option of deleting the original files from the source disk to keep things tidy. **Both programs come on one disk with documentation for just $24.95 first class postage paid. They require a minimum of 32K, 2 drives and NEWDOS.**

107 Owl Tree - by James Talley $9.95

Can you fill the OWL-TREE with Owls by shooting out the BATS? Easy? Careful, when you shoot a BAT it scares away OWLS! A super challenge, with great animated graphics and SOUND!

109 The Great Race - by Scott Carpenter $9.95

Try to finish this 600 mile race before your opponents, or before they stop you with flat tires, wrecks etc. Patterened after a popular board game, the computer plays too, (unless it dosen't like the name you give it!). Outstanding graphics, with real life-like car sounds!

111 Lying Chimps - by Roy Groth $9.95

The old game of "I doubt it" or "Liar", only you play with four chimps who love to cheat! Excellent animated graphics with SOUND.

113 Concentration - by Richard Taylor $9.95

The game of Concentration on your TRS-80! The prizes change places with each game. Win the Tandy Corp, or maybe a 48K TRS-80 (or a bad check!). With excellent SOUND effects.

110 Scramble - by Richard Taylor $9.95

A word guessing game for two players. Use the words in the computer or enter your own for your opponent. Excellent scoring routine, with SOUND. If you wait too long, you lose points.

103 Snake Eggs w/sound - by Leo Christopherson $14.95

This version of "21" has talking snakes, who argue with each other Try to avoid "scrambled eggs", they lose!

117 Function Grapher/Bluffit - by Roy Groth $9.95

FUNCTION GRAPHER: A new math program which allows you to graph a math function, then compress, expand, and find the root using 6 different methods!
From 80-US, Nov 79 issue, also on this tape from the Mar 79 issue, BLUFFIT, a card game of bluff and counter-bluff.

116 Psych/Lifeboat $9.95

From 80-US in May 79, Psych is the program which tells you if you are Analytical or Intuitive, and to what degree. Provides for screen or printer output. LIFEBOAT, from the Jul 79 issue, is a game of survival.

Please order by number using the order form in the centerfold. All programs are on cassette and run in 16K Level II or above. Note: Beewary, Android w/sound, Snake Egg w/sound and Lifetwo w/sound do not run reliably when transferred to disk using DOS 2.2, they do function OK with 2.0, 2.1 and NEWDOS. All others work with most current DOS, including 2.2.

108 TRS-80 OPERA - by Richard Taylor $9.95

A SOUND extravaganza! Hear the William Tell Overture in intricate detail and clear sound (that's the theme from the Lone Ranger, remember?). Contains four other excellent operatic selections. The SOUND is superb!

112 Challenge - by Richard Taylor $9.95

Guess the hidden phrase, but if you guess vowels wrong you lose 10 points! Use the phrases in the program, or enter your own. Careful! The short ones are the hardest! With fast graphics and SOUNDS to match.For 2 players

106 Beewary w/sound - by Leo Christopherson $14.95

Brilliant graphics and fantastic sound enhance this challenging game matching a persistent Bee with a cunning Spider in a duel to the death! Leo has done it again.

104 Lifetwo w/sound by Leo Christopherson $14.95

Conway's game of Life at an astounding 100 generations per minute! Plus Leo's "talking" animated creatures playing the "Battle of Life" in one 16K L2 program.

114 Space Battle - by R Papo $14.95

A super speed "Trek" type game, said to be one of the best so far.

101 Original Android Nim - by Leo Christopherson $9.95

The Original TRS-80 animated graphics game. This is the one that started it all. It does not have sound, but has been the most popular animated game to date!

105 Cubes - by Leo Christopherson $9.95

Gives the solution to "Instant Insanity" ® or numbered blocks. Watch the computer try all the possible combinations.

115 Biorythm 1 & 2 $9.95

Two Biorythm games (published in Nov 78 80-US), one with actual day-by-day data including day of week, the other a graphic display of all three curves on the screen at one time. Includes "day of week" and "days between dates" functions.

102 Android Nim w/sound - by Leo Christopherson $14.95

The TRS-80's first animated graphics winner, now with more animation and sound.

All software is sent POSTPAID during the week in which orders are received. Any malfunctioning programs will be replaced free of charge.
We also accept VISA and Master Charge orders for software, call (206) 475-2219 during normal business hours (Pacific Time Zone).

Dealer Inquiries Invited

Unclassified ADS

$ 2.50 per half inch and $ 2.50 for each additional half inch per insertion . Send printed or typed ad with correct amount to 80-U.S. Journal, 3838 South Warner St. Tacoma, Wa. 98409

TELETYPE USERS: Unique solid state time delay relay. Reduces energy and maintenance costs. Info 50¢, plans $5.00, with PC $10.00. Keith Ryan, Box 3103-J, Ottawa, Ontario, Canada K1P 6H7 (050)

ADVENTURE FOR TRS-80 $14.95 per adventure or send SASE for flyer 7 adventures currently available. M Scott Adams, PO Box 3435, Longwood, FL 32750 (050)

OWNERS or USERS of TRS-80 L2 or DOS systems interested in doing community/humanitarian service, drop a postcard with name, address & type equip to Mike Freeman, 946 Alder St., Tacoma, WA 98406 (0X0)

KISS IS A MONEY-MAKING PROGRAM for schools or large clubs. A 16K Level II machine will hold 20 responses for 400 surveys in this computer dating simulation. List, doc., sample survey, admin instrs for $7. D Bohlke, Coggon, IA 52218 (050)

TRS-80 4K L1 LOAD & RUN SOFTWARE Recreational, Educational, & Personal Management programs...3 Game tape, Programmer's Newsletter, & Software Catalogue all for $6.00 Zapata Microsystems PO Box 401483 Garland, TX 75040 (050)

DISK DATA BASE REPORT GENERATOR system 32K. No user programming. Full functions. Random access by key or rec# Define file parameters & report formats on-line. Report features: select, filter, sort, arith, summary. Multi-keys. Audit log. Blocking, hashing, buffering. Full documents. Ridiculously priced at $49 so a perfect package can be shared by all. Micro Architect. 96 Dothan St., Arlington, MA 02174 (070)

INCREDIBLE MICROPOSTER! The Brain Cell of your TRS-80 blown up to poster size in full color. Amazing print shows intricate details of quarter inch chip. $4. plus 75¢ shipping. Also MICROPOSTER in BLUE. 8080 CPU poster in beautiful blue. $3. plus 75¢ shipping. Both sent first class in mailing tube. J Shell, Box 304, Falls Church, VA 22046 (030)

MATH-PAK-1---MATH-PAK-2. Math Drill Programs for addition, subtraction, multiplication, and division with many features not found in other programs of their type Order MATH-PAK-1 for whole numbers, MATH-PAK-2 for fractions. 80-US says of MATH-PAK-1 "An excellent program-makes math practice both fun and .easy-recommended for all school age children" Available for $14.95 Ea(L2 16K)
CASSETTES-CASSETTES-CASSETTES Top quality C-20 cassettes with Norelco boxes-only $10.95/doz+$1 50 shipping. Shipping free with software orders (MA residents add 5% tax).
CASSETTE DUPLICATING SERVICE-small or large quantities-quick turnaround-guaranteed loadable. Write for rates EDU-WARE, PO Box 336 Maynard, MA 01754 (030)

ASR-33 with 4 ROLLS PAPER, 10 ROLLStape. Completely serviced Nov 79 by Teletype Corp $600. complete You pay shipping. Pick-up preferred Steve Larson 1525 S Lansing St., Aurora, CO 80012 (303) 752-3768 (030)

UNIQUE TRS-80 SOFTWARE/BOOKS Morse code transmit/receive program 5-25 WPM, no ancillary devices reqd $15. disk/cassette ppd Disassembled Handbook all Basic CALLS $10 ppd TV Satellites az-el-distance ur location. $5 ppd Decimal/Binary/Hex/Split Decimal/Split Hex both-way conversion pgm to 65K $10. disk/cassette ppd. Richcraft Engineering, Box 1065, Chautaugua Lake, NY 14722 Phone (716) 753-2654 for COD orders. (011)

SOFTWARE HOUSES ONLY-1st Year German Course, 17 programs, L2 16K, unpubl., text-indep., sold by arr. w/author. Jeff Collins, 5 Greenwood Ave., Wheeling, WV 26003 (304) 242-1890 (030)

FOR SALE: TRS-80 32K, CTR, EXPANSION interface, 2 Disc Drives, Centronics 779 w/tractor feed, stand & system table, Software. Best offer over $2000 (415) 824-6191 (030)

CONVERT DECIMAL/HEX/OCTAL or Binary to Decimal/Hex/Octal or Binary! A must for all programmers. Only $2.00. Dean R Zimmerman, 444 North Grove Drive. Alpine, UT 84003 (030)

PARENTS, TEACHERS, MANAGERS. Micro-Gnome's CAIWARE sets the pace in authoring and using Computer Assisted Instruction on the 16K TRS-80 Suitable for wide range of subjects at any learning level. Programs in Basic II and assembly guide lesson designer with six well-designed prototype questions and text screens. Lessons saved on tape. Features include: duplication from preceding screens; programmed instruction; weighted scoring; true/false, multiple choice or completion questions; keyword and alternate answer recognition $24.95. SUPER-CAI includes CAIWARE + extensions: runs faster. $44 95. Manual $9.00. MD res. add 5% tax. $2.00 shipping. MC/VISA/Check. Fireside Computing Inc., 5843 Montgomery Rd. Elkridge, MD 21227 (030)

TRS-80 EDUCATOR TEST PKG $29.95 Disk Level II Cassette $25.95 3 fully documented, self-directing programs, MAKETEST, GIVETEST, GRADES. Any subject or class, no programming, immediate score, feed-back. Easy to use. Assigns score, grade, avg., and std dev. for class (a test, a series or a term). Written by/for teachers. M G Manchester, Phd Box 4188 Spokane, WA 99202 (509) 535-8970 (050)

GET THE MOST FROM L2, 4K: Graphic Slot Machine & Loan Repayment/Return on Investment programs Both $4 95 w/program listings. Listings only $2. Experts at debugging and compressing L1, L2, and Disk programs & systems. Send tape/disk for free estimate Custom programs/systems designed. Write for estimate Micro Business Systems, 2320 Hampton, Harvey, LA 70058 (504) 362-6574 (030)

UPGRADING MY TRS-80 SYSTEM. SELLING some software and hardware in excellent condition. Send SASE for list. Robert Lee, 25 Amaryllis Ave., Waterbury, CT 06710 (030)

MEDHELP 1.2 TRS-80 48K 3 Disks CPT Code ICDA Codes Standard Health Care Form, Medicare and 3rd party billing $495.00 M Data Systems 205 Simmon St. Spencer, WV 25276 (304) 927-1255 (030)

MEDHELP 2.1 TRS-80 MODEL II CPT, ICDA, Practice Analysis A/P - G/L - Payroll - Patient & 3rd party billing. M Data Systems 205 Simmon St. Spencer, WV 25276 (304) 927-1255 (030)

I WANT YOUR NAME FOR MY MAILING LIST TRS-80 Text Editor $4.95 on disk. M Data Systems 205 Simmon St. Spencer, WV 25276 (030)

4K LEVEL 1 GRAPHICS GAME! Star Trek 80. Listing & instructions $4.50 Jon Waples, 70 Ives Road, East Greenwich, RI 02818 (030)

MODEL II Software. The BEST! Field Tested on Mod I for a year, and many enhancements. Data Base Manager features 2 levels security, search command, error trapping, statistics, full field arith. MAIL turns your terminal into an intelligent form processor. Handle subscription, reuse sort buffer, rotate name, SHELL sort. Come in for a demo or send $10. for a manual. Micro Architect, 96 Dothan St , Arlington, MA 02174 (110)

DISK DATA BASE REPORT GENERATOR SYSTEM. 32K. No user programming. Full functions. Random access by key or rec#. Define file parameters & report formats on-line. Report features select, filter, sort, arith, summary Multi-keys. Audit log. Blocking, hashing buffering. Full documents. Ridiculously priced at $69. so a perfect package can be shared by all. Micro Architect, 96 Dothan St., Arlington, MA 02174 (070)

ADVERTISER INDEX

NOW... Exclusively Yours from ACS

TRS-80® DOS 3.0

Disk Operating System by the Original Author

☐ DOS 3.0
EFFECTIVE SEPTEMBER 1
ONLY **$49.95**

Check these DOS 3.0 features . . . that go far beyond 2.1

NO KEY BOUNCE • UPWARDS COMPATABLE FROM 2.1 • CHECK EOF
SEEK • RE-READ • RE-WRITE • LOC • VARABLE LENGTH RECORDS
SKIP • DISK LOGGING OF MESSAGES • BOOT • CHAIN • WRITE EOF
PURGE • SET • RESET • ROUTE • RUN & LOAD FOR 1 DRIVE SYS.
XFER • FORMAT W/O ERASE • DIR FROM BASIC • PATCH • LINK
USER DEFINED KEYS • KEY AUTO REPEAT • UPPER AND LOWER
CASE DRIVER • SHIFT LOCK • RS-232-C DRIVERS
•PAUSE •USER DEVICE CONTROL BLOCKS
•MULTI PROTOCOL COMMUNICATIONS

HARDWARE FOR TRS-80®

☐ **Pertec Disk Drives FD-200** ..... **$375.00 ea.**
These are 40-track Drives that are completely compatible with the TRS-80® and Radio Shack Drives. 3.0 DOS included. Will allow Turning Diskette over and Write on other side.

☐ **4-Drive Cable for Pertec Drives** **$35.00**

☐ **DECwriter III, 132 Character** **$2500.00**
110 to 9600 band EIA tractor feed keyboard printer. This is truly the nicest printer available. (30 day delivery)

☐ **701 Centronics TRACTOR FEED Bidirectional Printer** **$1500.00**
2½ times as fast as the Radio Shack 779 Printer, has full size 132 Char. Carriage Bell tone. Complete with Cable plug in and use. Shipped Freight COD.

☐ **200 ns 16K Dynamic Memory Clips for Keyboard or Expansion Interface, Lifetime Guarantee, complete** **$110.00**
Lifetime Guarantee. Complete with Instructions and Jumper Blocks.

☐ **10 Key Numerical Keypad Kit** **$79.95**

☐ **TRS-80® Level II – 16k** **$750.00**
- ☐ **Expansion Interface** **$275.00**
- ☐ **RS-232-C Interface** ..... **$ 89.00**

ORDER NOW AND SAVE

Just list the items you want and mail this convenient coupon.

SOFTWARE BY ACS

☐ **Monitor No. 3** **$29.95**
Complete Machine Language Monitor for TRS-80® features: Find, EDIT, Relocate, Symbolic Dump to Tape, etc.

☐ **Monitor No. 4** **$49.95**
All of the commands that reside in Monitor No. 3, plus: RS-232 I/O, Disk Program I/O, Symbolic Dump to Disk for Loading into Disk Editor/ASM., Track & Sec I/O for modification.

☐ **PCLEND** **$15.95**
Will Patch ASCII files of Basic Programs or text or DATA FILES so that they may be loaded into the Disk Version of the Electric Pencil for Editing purposes comes on Cassette that will automatically create a Disk file of PCLEND.

☐ **MAKE TAPE AND MAKE DISK for Cassette Dealers** **$69.95**
These are two programs that will allow you to take any type of Program from Disk and store it on tape for mailing purposes. When the user receives the program in the mail on cassette, it is loaded into the computer which will automatically make a Disk file of the program.

☐ **CP/M & C BASIC for the TRS-80®**
CP/M Includes: MOVCPM, STAT, PIP, Dump, DDT, ASM (8080), ED, plus 6 user manuals.
CP/M **$150.00**
C Basic-2 Includes: XREF2, CBAS2, and manuals.
C BASIC 2 ..... **$99.95**

☐ **G2 LEVEL III BASIC for TRS-80®** ..... **Special $39.95**

☐ **TELCOM – Telecommunications for the TRS-80®** **$29.95**
Telecommunications for the TRS-80® allows one TRS-80® to communicate with another through the RS-232-C over the phone line.

Orders received by 6:00 p.m. shipped next day on Master Charge, Visa, Certified Check or Money Order. Personal Checks require 14 days to clear. No C.O.D. Collect calls not accepted. All Hardware warranted for 90 days except Radio Shack equipment which is warranted through Radio Shack. Software guaranteed for replacement only. Prices subject to change without notice.

AUTOMATED COMPUTER SOFTWARE SERVICE
(615) 244-2798
Division of
Computer World INC.
625 Main Street • Nashville, TN 37206

Send Check or Money Order payable to –
SOFTWARE • P.O. Box 60097 • Nashville, TN 37206

Quan.	Description	Unit Price	Total
	HANDLING CHARGE $1.50 TENN. RES. ADD 6% SALES TAX **TOTAL**		

☐ Check
☐ Money Order
☐ MasterCharge
☐ Visa
Card No. _____
Exp. Date _____

Name _____
Address _____
City _____ State _____ Zip _____

The 80-U.S. Journal
3838 South Warner Street
Tacoma, Washington 98409

Application to
MAIL
at 2nd Class
Postage Rates
Pending at
Tacoma, WA