# TRS-80® Microcomputer News

Volume 4, Issue 10

November, 1982

Price $1.50

Printer Issue

Color LOGO

Christmas Tree Letterhead

# Fort Worth Scene

This Fort Worth Scene will be devoted to several topics which have come to my attention recently.

First, HAMs—we love you! We didn't mean anything by it.

What I am referring to is Kathleen Keith's eighth commandment, as published in our July/August issue—Thou shalt not let any Ham operate his Amateur Radio Walkie Talkie in thy computer room. We published the list as received. As we understand it, Gilbert Keith (Kathleen's husband) is an active HAM and operates his radio gear along with his Model III and he did not report any problems. It was not our intention to tell people not to use computers and HAM equipment in the same room.

Mr. Ralph Anderson (K0NL) suggested three reasons why you would not use a Ham Walkie-Talkie in the computer room—damage to the computer, damage to the radio, damage to the individual—and then proceeded to show why none of them were valid. We would like to suggest a fourth reason—personal preference. Mr. Gerald Crosby (WB8YUO) sent us a rather long letter, for which we thank him.

### MAGAZINES

We have recently been getting copies of A5, a magazine for Amateur Television, and a quick glance through any issue will show that computers and Amateur Radio/Television are certainly compatible. A5 can be contacted at P.O. Box H, Lowden, Iowa 52255-0408.

Another magazine which has come to our attention recently is two/sixteen. This is a magazine published exclusively for users of TRS-80 Models II and 16. We have heard some good things about this magazine from their readers. For further information, contact:

two/sixteen magazine
131 East Orange Street
Lancaster, Pennsylvania 17602

For Color Computer owners, we have received an issue of East Texas Color Computer Club Newsletter. This "newsletter" was thirty-two pages long and included quite a few program listings. For further information, contact:

East Texas Color Computer Club
2101 E. Main St.
Henderson, Texas 75652

### COLOR LOGO

For this issue we combined part of the Education and Color Computer sections to bring you some expanded information on a new Color Computer Language that I am very excited about. Color LOGO is here!

I have been working with the disk version, and I am very impressed by both the simplicity and the power of the language. Color LOGO is not the kind of language that you will use to write a home accounting system, but you may use it to help someone (especially kids) understand how to solve problems in a logical way, and it will serve as an excellent introduction to the fundamental ideas of programming.

In the Color LOGO RUN mode, each command you enter is performed immediately. Since most of the commands cause the "turtle" to move or change position, it is easy to relate the commands to the action they cause. If you tell the turtle to move "FORWARD 10," you see it happen! Not only do you see the turtle move, but since it draws a line as it moves, it is easy to relate "10" with a distance. From this step, it is natural to wonder about the difference between "FORWARD 10" and "FORWARD 100" (or whatever number comes to mind.)

Two simple commands—FORWARD and RIGHT—are all a child needs to draw pictures. As the child begins to wonder if there aren't easier ways, you can introduce BACK and LEFT. Suddenly drawing becomes easier to do. Add PENUP and PENDOWN and you have the ability to move the turtle without drawing a line. Six commands that allow very complex drawings. With these six commands, a child can go a very long way toward relating the abstraction of a command with the physical result. What has happened? The child has begun to learn problem solving in a fun environment that is completely under control.

I love it!

**A learning environment with the Color Computer and Color LOGO.**

Volume 4, Issue 10 **November, 1982**

Price $1.50

# TRS-80® Microcomputer News

TRS-80 Microcomputer News is published monthly by Radio Shack, a division of Tandy Corporation. A single six month subscription is available free to purchasers of new TRS-80 Microcomputer systems with addresses in the United States, Puerto Rico, Canada and APO or FPO addresses. Subscriptions to other addresses are not available.

The subscription rate for renewals and other interested persons with U.S., APO or FPO addresses is twelve dollars ($12.00) per year, check or money order. Single copies of the of Microcomputer News may be purchased from Radio Shack Computer Centers or Computer Departments for $1.50 suggested retail each. The subscription rate for renewals and other interested persons with Canadian addresses is Fifteen dollars ($15.00) per year, check or money order in U.S. funds. All correspondence related to subscriptions should be sent to: Microcomputer News, P.O. Box 2910, Fort Worth, Texas 76113-2910.

Retail Prices in this newsletter may vary at individual stores and dealers. The company cannot be liable for pictorial and typographical inaccuracies.

Back issues of Microcomputer News prior to January, 1981 are available through your local Radio Shack store as stock number 26-2115 (Suggested Retail Price $4.95 for the set). Back issues of 1981 copies are available as stock number 26-2240 (Suggested Retail Price $9.95 for the set).

The TRS-80 Newsletter welcomes the receipt of computer programs, or other material which you would like to make available to users of TRS-80 Microcomputer systems. In order for us to reprint your submission, you must specifically request that your material be considered for reprinting in the newsletter and provide no notice that you retain copyrights or other exclusive rights in the material. This assures that our readers may be permitted to recopy and use your material without creating any legal hassles.

Material for publication should be submitted on magnetic media (tape, disk, or CompuServe). If you submit material on tape or disk, and it is accepted for publication, we will send you two cassettes or diskettes for each one you sent us. Cassettes will come from our box of mixed blank cassettes. If you submit material on CompuServe, and we think we may use the material, we will extend your Microcomputer News subscription by six months for each article accepted. If you are submitting material over CompuServe, please include your name and address or your subscription number so we can find you. If the material is very short, send it to us in E-Mail. If you have more than a few lines, you need to place the material in the ACCESS area of CompuServe and then let us know it is there by leaving a message on E-Mail.

Material may be submitted by mail to P.O. Box 2910, Fort Worth, Texas 76113-2910, or through CompuServe. The Microcomputer News' CompuServe user ID number is 70007,535.

## Notes to Program Users:

Programs published in the Microcomputer News are provided as is, for your information. While we make reasonable efforts to ensure that the programs we publish here work as specified, Radio Shack can not assume any liability for the accuracy either of the programs themselves, or of the results provided by the programs.

Further, while Microcomputer News is a product of Radio Shack, the programs and much of the information published here are not Radio Shack products, and as such can not be supported by our Computer Customer Service group. If you have questions about a program in the Microcomputer News, your first option is to write directly to the author of the program. When possible, we are now including author's addresses to facilitate communications. If the address is not published, or if you are not happy with the response you get, please write us here at Microcomputer News. We will try (given the limited size of our staff) to find an answer to your question and, in many cases, will publish the answer in an upcoming issue of Microcomputer News.

### Trademark Credits

| | |
|---|---|
| CompuServe™ | CompuServe, Inc. |
| Dow Jones™ NEWS/RETRIEVAL Service® | Dow Jones & Co., Inc. |
| VisiCalc® | VisiCorp, Inc. |
| Program Pak™ | Tandy Corporation |
| SCRIPSIT™ | Tandy Corporation |
| TRSDOS™ | Tandy Corporation |
| TRS-80® | Tandy Corporation |

## Contents:

### Color Computer

### Computer Customer Service

### Data Bases

### Education

### Model I/III

### General Interest

### Model II

### Peripherals

### Pocket Computer

**Prices shown in TRS-80 Microcomputer News are in U.S. Funds.**

# View from the 7th Floor

**by Jon Shirley**
**Vice President**
**Radio Shack Computer Merchandising**

No, I did not get fired, break a leg or take a long vacation. I just got very busy and did not have time to write. During the month of August, we travel around the country going to 6 meetings with our store managers. Each meeting has about 850 attendees. We tell them about our new product line for the next year and they tell us about what is lacking from that line. It's one of the many ways we try to listen to what you, the customer, want from us. It's a lot of fun, but it is also very tiring and takes a great deal of preparation. Hence my absence from the last issue.

I am told this issue is about printers. I love being involved in the purchasing of printers as their technology is being advanced even faster than CPUs. For years dot matrix printers were expensive, slow, unreliable and noisy. Then Japan discovered the American microcomputer boom and although they have not been able to sell computers here, they did see an opportunity in printers. Printers combine electronic and mechanical technology, an area where the Japanese excel. Just look at VCRs as an example. Today most of the printers for microcomputers are made in the Orient as are most of ours.

I am very proud of our printer line this year. It is all new, except for our best selling Daisy Wheel II. All the new ones are faster than the ones they replace and offer more features. The DMP-200, 400 and 500 are really worth a look if you are printer shopping. They have multiple type fonts including a very high quality "correspondence" font that is fixed pitch and makes very nice looking letters. They all have adjustable tractors and cut sheet handling. One of the neatest features is a user accessible rotary switch to set the power up print style default. For example with a DMP-200 there is a condensed type style to print 132 columns on an 8 inch page. On most printers to activate this you would have to tell the printer in software to switch from the normal 10 pitch to the condensed pitch every time you turned on the printer. With the DMP-200 all you do is set the switch and it goes into condensed mode on power up. Another set of user accessible dip switches allow setting other power up defaults.

Our new Daisy Wheel, the DWP-410, is also a great item. In addition to the low price, it is very quiet. In fact, it is the quietest daisy type printer on the market. Yet it is just as rugged and trouble free as the Daisy Wheel II which is the most reliable daisy wheel in the world.

At the low end, we have a pretty neat product in the Color Graphics Printer. If you do not own a printer and have a limited budget this $249.95 beauty is for you. At that price any printer is a good deal but this one prints in four colors and has graphics. It also has both parallel and serial interfaces so you can connect it to just about anything. I did not mean to turn this into a commercial, but we have spent a lot of time working on these printers. We really do believe that they are superior to any on the market.

I get some very interesting letters about what people are doing with our equipment and I would like to pass on two to you. The first is of interest to anyone with a mainframe or large mini who is also using TRS-80s. A company called Warren Glimpse & Co., 105 Oronoco St., Alexandria, Virginia 22314, is offering a service to convert 9 track computer tape to/from 8 inch diskettes. They claim they can take a 9 inch, 1600 BPI tape and convert to Model II readable diskettes. Tapes can be EBCDIC or ASCII with block sizes of 80 to 2048. They also can do diskette to diskette file conversion.

They use Scripsit, Profile Plus and other Radio Shack software. They can take, for example, a list of names and addresses from tape and put it into Profile format. They will also take a Census Bureau tape and put it on diskette. Their charges for this service seem reasonable. Needless to say I cannot verify their service, but I certainly urge anyone who needs this type of help to give them a call.

The other item I received was a photoletter from Mr. Rohn Engh. Rohn lives in Osceola, Wisconsin where he uses a Model II with four drives in his barn to handle his business. His business is photography and he uses the Mod II for communications with his publishers which allows him to live in the country. He also publishes a twice-monthly newsletter that tells photographers who is in the market for specialized photos. He uses the Mod II to write the newsletter and he even has an electronic version available via NewsNet. If you are interested in selling photos write to Rohn at the Photoletter, Osceola, WI 54020.

Finally to all of those who waited patiently and to those who were not so patient for SuperScripsit and Profile III Plus. To those of you who waited, thank you. I received several letters within the first two weeks of their availability extolling one or both as the best Model III software offered by anyone. I hope those of you who did wait agree the product was worth the delay, and that having seen them, you might understand why they took so long to develop and debug. To those of you who did not wait, my apologies. And I hope that you will at least go to a store and try them out. They are really superior products.

Until next month.

# What is an Operating System?

Whenever we have a program which requires disk access, there are really two ways to do the disk access. We can write into the program all the steps required to perform the access, or we can call upon the Disk Operating System (DOS) to provide those services for us. If we are writing the routines into the program, we must be very consistent, and must set up a means to keep track of whatever is on the disk in our program. If, however, we choose to use an operating system approach, we have the consistency of the system, and can save considerable time and effort in programming by calling upon the system to do all of our routine "housekeeping" functions.

Let's look at the difference between a tape operating system and a disk operating system. Don't be silly, I hear someone cry. There's no such thing as a tape operating system. Who ever heard of TRSTOS? Well, you're right. However, there are many similarities between how you handle tape program and data files and how a disk operating system handles disk files. For example, if you want to load the best little program to do whatever it is you want to do, on a tape system you would first find the tape by looking through your shoe box full of tapes until you find the cassette on which you think this program resides. Up to this point, I must admit, it's pretty much like finding the right disk. However, with the disk, we just shove it into the drive and type "DIR" to see what's on it. With the tape, we have to look at the label and see if we can make out what WE wrote on it. I don't know about you, but my computer prints better than I do. Anyway, after locating the tape, we then must try to use the tape counter to locate the specific section of tape that we want. Of course, we could use the old tried and true method of listening to the recorder's output, but that's not much more fun. With the disk and its operating system, we just type in something like "BASIC PROG1 -F:3" at TRSDOS Ready. This tells the Computer (by way of the Operating System) that we wish to load Disk BASIC (which allows us to use many features of the operating system), load a program called PROG1 and allow buffer space for three files. The operating system then carries out our wishes. (Did we lose you on file buffers? If we want to have disk access from BASIC for data files, we must have BASIC reserve space in memory to accumulate what will be written to disk. This way we can write a "chunk" to disk at one time. This area of memory is known as a buffer.)

The only hardware requirement is that the disk drive be able to find and read a "boot track" which will then tell it where to find the rest of the required information. Disk Operating Systems are usually on the Disk where they will be used. This makes it relatively easy to change the way the operating system works, if that should become necessary.

A notable exception to the DOS being on the disk is our own Color Computer. In this case, the DOS is in a ROM-PACK. Some advantages to this are that you only need one copy of the operating system, not a copy on each disk that will be used in Drive 0 as is required on the Models I, II, III and 16. Also, the space normally occupied by the operating system is now available for you to store programs and data on the disk. Also, it cannot be damaged as easily as the information on a disk can be.

Now we will try to answer some of your questions about Operating Systems, particularly about TRSDOS 4.1 for the Model II/16.

**Question:** How can I get a directory of a 'saved' diskette?
**Answer:** ⟨RESTORE 0 [DIR]⟩
**Question:** How can I get a directory of a 2.0 format diskette?
**Answer:** ⟨FCOPY 0 [DIR]⟩
**Question:** How do you use wild cards?
**Answer:** Wild cards are a means of defining a group of files. The TRSDOS-II commands which allow use of wild cards are DIR, FCOPY, SAVE, RESTORE, FILES, MOVE and KILL. For example, all 3 disk General Ledger files start with GL as the first two characters. To see a directory of just the General Ledger files, type **⟨DIR GL*/*⟩**. The asterisk means 'any sequence of characters here'. You could be even more specific by using **⟨DIR GL*/DTA⟩** which will give you a directory of just the data files.
**Question:** Explain what the IND option for the SAVE is.
**Answer:** The IND option allows you to specify a list of files to be saved. For example, say you have several Profile databases on hard disk along with some 3 drive accounting packages. Rather than saving the entire contents of the hard disk to floppy, it would be much faster and more convenient to save just the data files. The following example illustrates the use of the IND option: (User input inside ⟨ ⟩)

```
Step 1. TRSDOS-II READY
⟨BUILD TRF⟩ (Note:TRF is an example
filename.)
ENTER COMMAND LINE (1-80)
⟨*/DTA⟩
Store Line? (cr/esc)
⟨ENTER⟩
*** LINE STORED IN FILE ***
ENTER COMMAND LINE (1-80)
⟨PRO1*/*⟩ (Note: PRO1 is a Profile
database name.)
Store Line? (cr/esc)
⟨ENTER⟩
*** LINE STORED IN FILE ***
ENTER COMMAND LINE (1-80)
  ⟨PRO2*/*⟩ (Note: PRO2 is another Profile
  database name.)
  Store Line? (cr/esc)
  ⟨ENTER⟩
  *** LINE STORED IN FILE***
  ENTER COMMAND LINE (1-80)
  ⟨ENTER⟩
  **************EDITCOMPLETE**************
  TRSDOS-II READY
  ⟨SAVE TRF:4 TO 0 [IND]⟩
```

This save command tells TRSDOS that TRF is not actually the file to be 'saved' but a file which contains a list of files to be 'saved'. (Notice the use of wild cards.)

**Question:** I have a Model II and a Daisy Wheel printer. Sometimes I need to type a quick envelope. How do I do that?

**Answer:** Place a blank sheet of paper in the printer. At TRSDOS READY, type ⟨**DUAL ON**⟩. At TRSDOS READY, type ⟨**ECHO**⟩. Now remove the sheet of paper, insert the envelope and start typing. When you are through, re-insert the paper and press ⟨**BREAK**⟩. At TRSDOS READY, type ⟨**DUAL OFF**⟩.

**Question:** How can I run a program without getting printouts?

**Answer:** Use the command ⟨**FORMS D**⟩. To return to the normal mode, use ⟨**FORMS N**⟩.

**Question:** Using the HOST command, can I run Profile and Visicalc from remote locations?

**Answer:** No. These programs use 'cursor positioning' to write information at selected locations on the screen. Currently HOST is not capable of transmitting the codes necessary for that positioning to a remote terminal.

**Question:** What are the 'Supervisor Calls' used for?

**Answer:** They are used by Assembly Language programmers as programming aids to shorten development time. SVCs provide standard entry points to TRSDOS which tend not to change from version to version.

**Question:** What is the purpose of VERIFY?

**Answer:** Verify has 2 purposes on TRSDOS 2.0.

VERIFY ⟨**ON/OFF**⟩: (Defaults to ON) causes the computer to verify each write. We recommend that you leave this option on at all times.

VERIFY DETECT ⟨**ON/OFF**⟩: (Defaults to ON) causes the computer to check for disk swap before each read or write. For applications like PROFILE, turning off verify detect will speed up the program tremendously.

On TRSDOS 4.1, VERIFY DETECT is no longer an option.

**Question:** How can I have an ASCII text file list to the screen in text format?

**Answer:** At TRSDOS READY, type:

```
⟨FORMS D⟩
⟨PRINT filename, V⟩
⟨FORMS N⟩
```

## Computer Customer Services Address and Phone Number

8AM to 5PM Central Time
Computer Customer Services
400 Atrium, One Tandy Center
Fort Worth, Texas 76102

| | |
|---|---|
| Model I/III Business Group | (817)390-3939 |
| Model II/16 Business Group | (817)390-3935 |
| Languages and Compilers Group | (817)390-3946 |
| Color and Pocket Computer Group | (817)390-3944 |
| Hardware and Communications Group | (817)390-2140 |
| Educational Software | (817)390-3302 |
| Games, Books, and News Product Info | (817)390-2133 |
| Newsletter Subscription Problems | (817)870-0407 |

# Budget Management and the LP VIII

**Ruth Shackelford**
**Route 1, Box 261**
**Yale, OK 74085**

Here are some line changes that I have made to your Budget Management Program in order to have the program work with my Line Printer VIII.

Thank you for the many wonderful articles that you have been publishing in the Microcomputer Newsletter. I have found them to be of great assistance in my learning to program.

This works with the Model III, 16K.

```
3110 T$="   NO.      NAME              CURRENT
     BUDGET    PERCENT    REMAINING"
3120 F$=" ##### %                 %  #####.##
     #####.##    ###.##     #####.##-"
    : RETURN
3130 T$="DATE          DESCRIPTION        AMOUNT"
    : F$="##/##    %          %       ######.##-
3140 IF TY > 1 AND TY < 4 THEN T$=
    "   NO.     " + T$
    : F$=######     " + F$
    : ELSE T$=STRING$(10," ") + T$
    : F$=STRING$(10," ") + F$
3150 IF TY > 1 AND TY < 5 T$=T$+"   EXP. ACCT"
    : F$=F$ + "  ######"
3160 IF TY=3 THEN T$="  NO.   WRTN   CAN
     DESCRIPTION          AMOUNT  EXP. ACCT"
    : F$=######  ##/## ##/##  %                  %
     ######.##   ######"
    : RETURN ELSE RETURN
3530 GOSUB 3900
    : LPRINT TAB(25); "BUDGET MANAGEMENT REPORT" +
     STRING$(15," ");
    : LPRINT USING "PAGE ##"; PG
3540 LPRINT TAB(25) "** "H$" REPORT**"
    : LPRINT TAB(30)"DATE: ";
    : D=DT
    : GOSUB 500
    : LPRINT USING "##/##/##"; D1; D2; D3
3900 LPRINT STRING$(80,"*")
```

# Megawars

*Editor's Note: The CompuServe Information Service is one of the largest information and entertainment services available to owners of personal computers and computer terminals. With each issue of TRS-80 Microcomputer News, various features of CompuServe will be discussed. The CompuServe Information Service is sold at Radio Shack stores nationwide and in Canada.*

## NOW YOU CAN BE IN CHARGE OF YOUR OWN KRYON OR COLONIST SPACE ODYSSEY

By day you can be a mild-mannered newspaper writer, receptionist or company president, but by twilight you are ZELTAR, Kryon king. You are the dictator of the brave Colonists who have been enslaved under your reign for 700 years. You fight the Colonists and they fight you. As Kryon king you have immense power but you might lose that power if you are not careful.

Megawars is CompuServe's intergalactic battle game located on the games menu page. It's a noble game for one to 10 players at a time. Choose either to side with the Colonists and champion the Coalition or become a Kryon whose allegiance is to the Empire. The game can be played "interactively" by people throughout the country.

Whether Colonist or Kryon, both warriors' objective is to destroy all enemy bases and ships and capture all enemy planets before your enemy beats you to it.

As an intergalactic fighter you are in control of your destiny by commandeering your own spaceship from your computer keyboard. You are free to enter and leave battles whenever your objectives are complete. Of course you will have a rank which will be decided when you choose your battlefield.

New recruits enter the clash as Cadets who are assigned a fighter class ship with sufficient powers until they learn the ways of destroying the enemy.

Upon accumulating a battle score of 10,000 units, you will be eligible for promotion to Lieutenant where you will be given the opportunity to command a Scout or Fighter ship. A Scout ship is a reconnaissance model and a Fighter is a workhorse of sorts. Thirty thousand points promotes you to Captain and the option of using a Scout, Fighter or a Miner class spaceship which is a converted merchant ship used for retrieving Photon Mines. For those inherent leaders who score 100,000 points, the rank of Admiral becomes yours and you are then permitted to command any class of ship you desire including a Flagship which is a movable Starbase.

As you begin your intrepid journey in your Flagship, Fighter or Miner, you must always remember that your enemy and his ship are equal to you and yours and your survival depends on your skill as a commander. You must be aware of the locations of certain space inhabitants. An immobile starbase will surely blow you to bits if you stray too near, but is harmless if left to itself.

Enemy planets are weaker but still not to be ignored. You must be careful of neutral planets which will fire at you unless you win them over to your side. Another fearful enemy to watch is the dreaded Acheron. This devious enemy is concealed by a cloaking device which hides it from all wanderers and star fighters alike, and whose existence it seeks to terminate.

In your perilous voyage you are armed with various devices which keep you in flight. In case of attack, your crew is able to repair any damages while still in flight. You also are equipped with back-up engines, for when warp engines are down, as well as a deflector shield, life support system, subspace radio, a tractor beam for the towing of fellow ships in trouble and a ship computer used for firing, computation of movement and phaser control.

You are now ready to travel out into the galaxy of battle. Go and be well and may the forces be with you.

## SHOPPING AT HOME WITH DICK AND JANE

Dick and Jane are sitting in their living room when Jane turns to Dick and says:

"Dick, now that we have some new furniture, we need that new stereo system we've been discussing. It would be nice to have for company. I think we should do something about it."

"We can't afford to buy a stereo so we'll just have to sit on the new furniture and listen to each other sing."

"Why don't we shop through Comp-U-Store. Their prices are 10 to 40 percent less than other places."

"What is Comp-U-Store?"

What IS Comp-U-Store? Well, Dick and Jane, it used to be Comp*U*Star, the CompuServe Information Service's shop-at-home offering. Now it's Comp-U-Store, the same thing but with a name change and available for the standard CIS rates. CompuServe subscribers can access Comp-U-Store now for $5 an hour for standard service weekday evenings and all day on weekends and $22.50 an hour for prime service (weekdays).

Comp-U-Store was originally only a shop-by-phone service operated by Comp-U-Card of America in Stamford, Conn. Then, Comp-U-Card linked with CompuServe to provide home computer users with the same shop-at-home service but with an even more versatile method of ordering merchandise than using the telephone.

To view that better method, we'll follow along as Dick and Jane access CompuServe on their home computer.

After logging onto CompuServe and proceeding to the main menu, they choose Home Services, then item 4, the shop and bank at home section. They then request a description of Comp-U-Store, and are told that they, as members of Comp-U-Store, have access to information on product features, model numbers, price quotations and can order from a selection of over 30,000 brand name products including appliances, cameras, crystal, musical instruments, video equipment and even stereo equipment.

Dick and Jane like the service because it guarantees these products, and they can also comparison shop between items. When they've decided to order, they can pay by credit card or check and then have it delivered right to their door. And if they're not satisfied with their purchase once it arrives, they're entitled to a full refund.

Running through the service, Dick and Jane find the stereo they wish to examine and perhaps purchase. Item 4 is a full stereo system. They type in the stereo system designator (STSY) under stereo equipment and then a brand name. Looking through that category's offerings they decide on one set and then key in for a closer look. In doing this, Dick and Jane compare the systems by their description, price, list price and amount saved by buying through Comp-U-Store.

They agree on one system but have one other area to check. This month's bargain in the Databasement, a section offering merchandise reduced even further than the Comp-U-Store discounts, is stereo equipment. Going through the Databasement they find a system they like even more and decide to purchase it.

Dick and Jane then proceed to the ordering section where they give all their ordering specifications at the appropriate prompts and then review the information for accuracy.

They then pay by their Master Card or no fee VISA offered through Comp-U-Store, and their new stereo will be sitting in their living room in a matter of weeks. Now Dick and Jane can entertain, don't have to sing to each other and have saved enough money so that they can buy a new toy for Sally and a vacuum cleaner to clean up after Spot.

## COMPUSERVE'S TODAY MAGAZINE: A GROWING MAGAZINE FOR A GROWING INDUSTRY

All CompuServe Information Service subscribers receive an issue of our publication, TODAY magazine. TODAY is a color magazine which gives our customers a more in-depth view of our services, new products, CompuServe Information Providers, customers and what they think of the service and special announcements and information.

Since TODAY is a publication TRS-80 computer users see quite frequently, we thought we'd like to give them and others a little background on the magazine, including when it was born and how it's growing up.

CompuServe decided there was a need for a more in-depth communications vehicle; something which we could use to send to our customers which would highlight our services and discuss our offerings in more detail. So, in July of 1981, CompuServe Information Service debuted its first edition of TODAY magazine as a communicator to its CIS customers and as an informational piece to people who had responded in some way to CompuServe advertisements.

Since that July 1981 debut, TODAY magazine has blossomed from an idea to a circulation of 60,000. Over the past year, we've steadily modified our coverage to include the reporting and analyses of the videotex industry and also modified our production schedule to make TODAY a bi-monthly rather than a quarterly publication. In this way we feel we can be an even more valuable information tool for companies and organizations who have products and software to advertise to the growing computer user and CIS customer market.

TODAY magazine offers regular features such as industry watch, I. P. Spotlights, Profiles, and a column on the future.

Special features have included bank-at-home-services, the electronic Better Homes and Gardens magazine, CompuServe on Qube Cable television, new games and various financial services, CompuServe's network expansion, electronic hobby groups and clubs, new menu structure, schools for computer literacy, a look at electronic mail, health and medical information, a congressman who uses EMAIL to communicate with his constituents and an example of what vacations may be like in the future.

CompuServe's TODAY magazine is available with the purchase of a subscription to the CompuServe Information Service at Radio Shack outlets or by mailing in the subscription offer card in each TODAY magazine.

*Questions and comments about the CompuServe Information Service can be sent to Richard A. Baker, editorial director, or Jacquie A. Farthing, assistant editor, CompuServe Information Service, 5000 Arlington Centre Blvd., P. O. Box 20212, Columbus, Ohio 43220 or through Feedback, main menu item 5, CompuServe User Information.*

# More Quotes and a Whole World of News

*Editor's Note: With more than 45,000 subscribers, Dow Jones News/Retrieval is the leading provider of on-line business and financial information. We will keep you up-to-date on new software, new data bases and new prices in upcoming issues of TRS-80 Microcomputer News. Dow Jones News/Retrieval is sold at Radio Shack stores nationwide and in Canada.*

News/Retrieval users are benefitting from two more recent additions to their service—vastly expanded general news coverage and an enhanced historical quotes data base that will extend back an entire year.

### THE WORLD REPORT

News/Retrieval now provides national and foreign news in a new service called the News/Retrieval World Report. The data base is compiled from the wires of United Press International, which for years has been recognized as a leader in the field of gathering and distributing news to newspapers, radio and television.

Throughout the day, the News/Retrieval editorial staff monitors and rewrites the news wire to produce a concise, coherent, continuously updated picture of the world and national scene in an easy-to-use electronic package designed specifically for busy subscribers.

Instead of putting all UPI material directly on-line, stories are selected and ranked according to importance by journalists in News/Retrieval's Princeton, N.J. newsroom.

Minutes after events are reported, they are included in the World Report. The editing process continues throughout the day, so that major stories appearing in the morning may be replaced by updated versions or by other late-breaking stories in the afternoon or evening. Typically, stories in the World Report run three pages, although breaking events of major importance may run longer.

The World Report is continuously updated from around 8 a.m. to about 8 p.m. (Eastern time), Monday through Friday. It is one of the General News and Information Services data bases, and costs standard subscribers 60 cents per minute in prime time or 30 cents per minute in non-prime times. It can be accessed by typing //NEWS.

After subscribers enter //NEWS to get into the World Report a copyright statement will appear at the top of the page, followed by headlines of the five most important national and foreign stories access codes. This is the World Report "front page".

Subscribers can press the N key to view a list of other national news headlines and the access codes to obtain the stories behind the headlines. Or, they can press the F key to view a list of additional foreign news headlines and story access codes.

In addition to the five "front page" stories, there are approximately 25 national stories and 25 foreign stories, with the most important headlines at the top and lesser ones farther down.

### DAILY HISTORICAL QUOTAS EXPANDED

The daily historical quotes, a data base, is being expanded to cover a full year of historical stock quotes, and quotes are available in composite form or by individual exchange.

News/Retrieval will enhance the Historical quotes data base in two other important ways:

- Composite historical quotes for monthly and quarterly requests
- Daily historical quotes accessible by specific date

To retrieve daily historical quotes for a full year use the same method that you currently do for daily quotes, e.g. for the last 12 days of Coca-Cola quotes type KO P1. For the preceding 12 days type KO P2. The new system extends back to P22. By typing KO P22, you would access price and volume data for days 253 through 264.

In the new system KO P1 will retrieve composite quotes for Coca Cola. To get historical quotes on an individual exchange, type the number of the exchange before the company code, e.g. 1KO P1. This will retrieve quotes from the New York Stock Exchange. Composite quotes will be retrieved automatically if an exchange number is not used.

To retrieve monthly and quarterly composites do not use an exchange, e.g. KO 82 M or KO 82 Q. To get quotes by exchange, use the same method as for daily historical quotes.

To retrieve historical quotes by date type the date, including month, day, and year, e.g. for June 30, 1982, type KO 6/30/82.

DOW JONES
NEWS/RETRIEVAL®

### TO GET THAT QUOTE . . .

The following tables will help you determine which code should be entered to retrieve information on the trading days you desire, and which exchange code should be entered to get specific stock exchange information rather than the automatically retrieved composite information. For example, if you're interested in the trading days from 157 to 168 days ago, type P14 after the desired company code. And if you want information from the Pacific exchange instead of composite quotes, type 3 before the desired company code.

| | Accesses | | Accesses | | |
|---|---|---|---|---|---|
| | Trading | | Trading | | |
| Code | Days | Code | Days | Code | Exchange |
| P1 | 1-12 | 12 | 133-144 | 1 | New York |
| 2 | 13-24 | 13 | 145-156 | 2 | American |
| 3 | 25-36 | 14 | 157-168 | 3 | Pacific |
| 4 | 37-48 | 15 | 169-180 | 4 | Midwest |
| 5 | 49-60 | 16 | 181-192 | - | Composite |
| 6 | 61-72 | 17 | 193-204 | | or OTC |
| 7 | 73-84 | 18 | 205-216 | | |
| 8 | 85-96 | 19 | 217-228 | | |
| 9 | 97-108 | 20 | 229-240 | | |
| 10 | 109-120 | 21 | 241-252 | | |
| 11 | 121-132 | 22 | 253-264 | | |

# microRESEARCH Problem #4

**Richard V. Andree**
**Department of Mathematics**
**University of Oklahoma at Norman**

A cryptarithm is an arithmetic problem in which the digits have been replaced by letters. Each letter represents a distinct digit and no digit has more than one letter replacement.

The cryptarithm

I * LOVE = TANDY

has 4 * 7039 = 28156 as one possible solution, but there are a dozen other solutions. Find them.

Possible extensions:

FIRE + BURNS = PAPER (2 solutions)
FEAR * FEAR = FASTSTAR
RUT * RUT = CAREER

### A SOLUTION FOR microRESEARCH PROBLEM #3

Here is a simple program to find the four three-digit whole numbers where the sum of the cubes of each of the three digits of the number is equal to the three digit number.

```
100 FOR N=100 TO 999
110 X=INT(N/100)
120 Y1=N-X*100
125 Y=(INT(Y1/10))
130 Z=Y1-(Y*10))
160 T=INT(X^3)+INT(Y^3)+INT(Z^3)
170 IF T=N THEN PRINT T
180 NEXT N
```

# Binding Fan Fold Paper

**Earl R. Kooi**
**2196 Albright Ave.**
**Upland, CA 91786**

To bind program listings, etc., LLISTed on 9$\frac{1}{2}$ by 11 inch fan-fold paper where punching holes after removal of the tear-off margins would punch through the left edge of the printing, place a strip of $\frac{3}{4}$ inch transparent tape on the left side of the page over the pin holes and tear-off line. Then punch the holes and tear off the right margin. This gives a rugged binding edge and leaves all the printing intact.

# Tape Mailing List and Two Across Labels

**Randy Blinn**
**717 Ellwood-Zelienople Road**
**Ellwood City, PA 16117**

I was using a Line Printer VII with single across labels, but they did not work very well because of the thickness of the labels. But when I went to the dry gum labels (26-1456) my program would only print one across. I have Radio Shack Tape Mailing List 26-1503, so I did some changing and I came up with these changes. I thought others might be interested so I am sending it to you.

### TAPE MAILING LIST

```
2Ø15 INPUT"ARE THE LABELS TWO ACROSS < Y OR N >"; ZZ$
19ØØØ K=1
19ØØ1 IF MID$(FI$(I,Ø),K,1)=";" THEN IF K=1 THEN
       RETURN
     : ELSE LPRINT MID$(FI$(I,Ø),K+1);
     "  "; LEFT$(FI$(I,Ø),K-1);
     : IF ZZ$="Y" THEN GOTO 191ØØ ELSE LPRINT
     : GOTO 19ØØ5
19ØØ2 K=K+1
     : IF K <= LEN(FI$(I,Ø)) THEN 19ØØ1 ELSE LPRINT
      FI$(I,Ø);
     : IF ZZ$="Y" THEN GOTO 192ØØ ELSE LPRINT
19ØØ5 IF E=1 THEN LPRINT FI$(I,1);
     : IF ZZ$="Y" THEN LPRINT CHR$(16); "45";
      FI$(I+1,1) ELSE LPRINT
19Ø1Ø LPRINT FI$(I,2);
     : IF ZZ$="Y" THEN GOTO 1925Ø ELSE LPRINT
     : LPRINT FI$(I,3);"     ";FI$(I,4); "-"; FI$(I,5)
     : LPRINT CHR$(138)
     : LPRINT CHR$(138)
19Ø15 IF E=Ø THEN LPRINT CHR$(138)
     : ELSE IF FI$(I,1)="" THEN LPRINT CHR$(138)
19Ø2Ø RETURN
191ØØ IF B$(Ø)="" THEN 1912Ø
1911Ø IF B$(Ø)=LEFT$(FI$(I+1,S), LEN(B$(Ø))) THEN
      1912Ø
1912Ø K=K-1
192ØØ IF MID$(FI$(I+1,Ø),K,1)=";"THEN IF K=1 THEN
      RETURN
     : ELSE LPRINT CHR$(16); "45"; MID$(FI$(I+1,Ø),
     K+1); "  "; LEFT$(FI$(I+1,Ø),K-1)
     : GOTO 19ØØ5
1921Ø K=K+1
     : IF K <= LEN(FI$(I+1,Ø)) THEN 192ØØ ELSE LPRINT
      CHR$(16); "45"; FI$(I+1,Ø)
     : GOTO 19ØØ5
1925Ø LPRINT CHR$(16); "45"; FI$(I+1,2)
     : LPRINT FI$(I,3); "  "; FI$(I,4); "-"; FI$(I,5);
193ØØ LPRINT CHR$(16); "45"; FI$(I+1,3);
     "     "; FI$(I+1,4);
     : I=I+1
     : LPRINT FI$(I,5)
     : LPRINT
     : GOTO 19Ø15
2ØØØØ DATA A, R, S, L, P, W
```

# PROFILE III+: Optimizing Storage Capacity

**Profile**
**The small Computer Company**

Written for the users of PROFILE II, PROFILE+ and PROFILE III+

**PROFILE Editor**
**P.O. Box 2910**
**Fort Worth, TX 76113-2910**

**By Ivan Sygoda, Director, Pentacle**

For your PROFILE III+ database to be as large, efficient and flexible as possible, it has to be made to "fit" well in a number of places—on the diskette(s) where it lives, so to speak, and on the screens, reports and labels where it interacts with the outside world. Optimizing usage takes a bit of juggling and maneuvering on the user's part, but the effort pays dividends in increased system capacity.

We can't do the juggling for you, because only you can decide what fields to include, their relative lengths, and where they are to be arrayed on various screens, reports, labels and form letters. However, we can help you with the decision process, showing you how to avoid inefficient solutions.

## WORKSHEETS FOR DEVELOPING A DATABASE

We recommend that you develop your entire system on paper first, rather than on the computer. Although you can't set up screens, reports and labels without a working set of fields, you may regret decisions concerning field lengths unless you anticipate how the fields will fit on these screens, reports, and labels. You should plot out likely formats on Radio Shack's video worksheets (cat. no. 26-2105). Reports and labels require special care because their size and layout is so confined.

## LABELS AND REPORTS

For instance, the second line of a label format might include both "Title" and "Department." Suppose you had allotted 12 characters for the title and 30 characters for the department name: 12 + 30 + 1 space between the two fields = 43 characters. But typical address labels are less than four inches wide—at 10 characters per inch (c.p.i.) the line might require 4.3 inches. A miss is as good as a mile.

Shave three characters off the department field, or two off department and one off title, or any feasible combination. What you're doing is trading off an occasional awkward abbreviation against misprinted labels. The time to make such adjustments is before you define fields.

Suppose you're setting up a mailing list file and know that report printing will be an important function. Say that you had defined your fields as follows: Last Name = 16, First Name = 12, Address = 24, City = 20, State = 2, Zip = 10. You'd also need spaces between fields, a total of five in this case. The grand total is 89 characters, which makes it impossible to print one-line reports on your 80-character-wide printer. Solution: Shave one character off the last name field, two off the address field, one off the city field, and reduce zip code to five (the nine-digit code seems mainly to be for business-reply mail . . . so far). Most of your data will still fit, and you can do efficient one-line reports. Don't forget that PROFILE III+ lets you include the record number (indicated by @; leave five spaces) on reports and labels. Since accessing records for updating by record number is so efficient and easy, you might consider including this field in your formats.

Work out report formats on the coding side of the video worksheets. Counting the space separating the variables list, there are 80 characters across the sheet, perfect for planning 80-character-wide reports. You will often discover that a little judicious shaving of field lengths will increase the number of fields that will fit in a report line or make it possible to group fields more logically. The PROFILE III+ manual discusses file definition and the use of data fields.

## ALLOCATING SEGMENTS TO DRIVES

When you've finally determined your fields and their lengths, and decided which are key fields, your next task is to array these fields across the four segments available. The variables are: 1) the number of disk drives you have or plan to have; 2) the disk space available on each of these drives. Again, you have to juggle things a bit. There's a spot of math involved, but it's easily done with a calculator.

The TRSDOS directory utility will tell you the amount of free space in granules (1 gran = 768 bytes or characters) remaining on any particular diskette. To see a directory, at TRSDOS type: DIR :n ("n" is the drive number) and press ⟨ENTER⟩. Remember, however, that your data files aren't alone in taking up space on the diskettes. You also have to decide which diskettes will hold the screen, label and report formats, user menus, math formulas, et cetera. Since each of these formats takes up disk space, it will be useful to predict how much.

## HOW MUCH SPACE ON A DISKETTE?

A 5¼ inch data diskette with 40 tracks, 18 sectors per track and 256 bytes per sector (which is the same as 40 tracks at 6 grans each) can contain 184,320, or 180K, bytes

(180 x 1024) of information. The directory occupies track 17 (6 grans, or 4608 bytes).

A system diskette's operating system occupies an additional 21 sectors or seven grans. Also taking up space on a system diskette are: system information (not shown on the directory) at 49 grans, BASIC at 7 grans, CONVERT at 4 grans, and XFERSYS at 2 grans, for a total of 62 grans. The Runtime diskette programs delivered with the PROFILE III + system (EFC7, EFC8, EFC9, EFCA, EFCB, EFCC, EFCD and the Runtime menu M) take up another 76 grans.

So a data diskette will have approximately 171K free space for user data, and a system diskette—your Runtime drive 0 diskette—will have only 95K free space. Much of this 95K could easily be used up by the format files.

## SPACE USAGE

User-generated files or programs are of two sorts: 1) formats (screen, report, label, SCRIPSIT and VisiCalc selections, plus user menus, math formulas and BUILD files) whose lengths are relatively constant and independent of the number of records currently in the data base; and 2) the actual data segments and the index and merge files, all of which grow in size as the user's database grows.

You can anticipate that formats will occupy space as follows—the grand totals indicate the amounts of space you'd use if you defined all available formats:

| | |
|---|---|
| Screen formats | : [yourfilename]/PM1-/PM5 @ 2 grans each = 10 grans total |
| Report formats | : [filename]/PR1-/PR5 @ 3 grans each = 15 grans total |
| Label formats | : [filename]/LB1-/LB5 @ 3 grans each = 15 grans total |
| SCRIPSIT formats | : [filename]/SL1-/SL5 @ 3 grans each = 15 grans total |
| VisiCalc formats | : [filename]/VC1-/VC5 @ 3 grans each = 15 grans total |
| User menus | : each menu occupies 6 grans |
| BUILD files | : used with menus; occupy 1 gran each |
| Math formulas | : [filename]/MTH occupies 2 grans |

Data, index and merge files, as mentioned above, grow in size as you expand files. The length in grans of any data file may be computed as follows:

(LRL x #REC)/768 = #GRANS (round to next higher integer)

LRL = Logical Record Length
#REC = number of records in the file
768 = number of bytes in one gran.

All these factors can be taken from the directory listing.

It is just as easy to solve for the number of records which will fill a known quantity of grans:

#REC = (#GRANS x 768)/LRL

These two equations give you the information you need to calculate the space proposed data segments will occupy.

The LRL of an index file (filename/IX1) is the length of the sort field plus 2 characters. The LRL of a SCRIPSIT merge file (filename/SR1-/SR5) is a single character; the number of records, on the other hand, is highly variable, and the space necessary must be evaluated on a case-by-case basis.

Note: "Bytes" in PROFILE III + are equivalent to "characters." While you're defining your fields on paper, you can get a rough idea of the number of bytes per segment by totalling the numbers of characters being used for the fields. The approximate Logical Record Length is the number of characters being used for all fields in a segment.

## DISTRIBUTING THE DATA

As you can see, at their theoretical limit of complexity, PROFILE III + Runtime programs and formats can fill the system diskette. However, it is not necessary to group these files together on the same diskette.

Fact: Any PROFILE III + Runtime program or any user-generated format program or data segment can be moved at any time to any disk drive which has room for it. PROFILE III + will find the program when it needs it by scanning the directories. This gives you a lot of flexibility in arranging disk usage for maximum storage capacity.

You have, therefore, two basic ways to go. One, you could array your data segments across the available drives as per the principles explained in the PROFILE III + user manual. Any screen, label, etc. format could be fit in around these segments like mortar around uneven bricks. Or, two, you could reserve the system (drive 0) diskette for Runtime and format programs and relegate all data segments to drives 1 to 3.

This latter method has certain advantages. When your data base outgrows the capacity of your system, you can easily add a disk drive or, less expensive, divide your data base by some relevant criterion: geography—by zip, state or area code; alphabetically—by name, A-M and N-Z. Then you can switch data disk sets, using the same Runtime diskette for each data set—you'll have one less diskette to back up at the end of each session!

## A GOOD SENSE OF PROPORTION

If you do elect to reserve the drive 0 disk for Runtime programs and formats, the chart in Figure 1 will help map out the permutations involved in arraying various numbers of segments across various numbers of disk drives. There's no problem distributing segments when the number of segments and the number of available drives are evenly divisible by one another. It gets tricky, however, when they are not. Especially tricky is arraying three data segments across two drives.

The problem is that when one segment fills its drive, then the entire system is full as far as the computer is concerned. Segments are not divisible—you can't put the first half of segment 1 on one drive, and the second half on a second drive. You also want to be able to incorporate another drive efficiently when the budget allows.

If you consider your entire file equal to 1, and establish relative lengths of 1/2, 1/4 and 1/4 for three data segments while using two drives (grouping the two smaller segments and reserving drive 0 for programs and formats), both drives will fill up at the same rate. However, when you add a fourth drive, you cannot increase the system's capacity by moving one of the two smaller segments onto the new drive—the drive containing the larger segment will fill its disk at the same rate as before. The answer is to make four segments with ratios of 1/3, 1/3, 1/6 and 1/6. While using two drives, bunch them as follows: drive 1—1/3 and 1/6, drive 2—1/3 and 1/6. After adding a third drive, move the segments this way: drive 1—1/3, drive 2—1/3, drive 3—1/6 and 1/6.

## EXTRA HINTS

A few nuts and bolts suggestions to make life easier: 1) When you format your data diskettes, name them in such a way as to identify the database and drive. For example, SALES 1.2 identifies the disk as your SALES program, region 1 territory, drive 2 diskette. Print the same name on the label of the diskette.

2) Set up your entire system on one disk, expanding your files by only five or ten records to test the formats. That is, keep your Creation diskette in drive 0 and your Runtime disk in drive 1, and always answer "1" when prompted for "Drive number to hold this segment." Then, when everything is working to your satisfaction, transfer the various segments and formats to the drives you've determined they should inhabit and expand your files to working levels. This will save extras steps and disk-switching during file creation procedures.

*PROFILE Editor's Note: This is Mr. Sygoda's second article in a series of 'how-to' PROFILE III+ articles. Other articles in the series will be published over the next few issues in this column. We hope that you enjoy this new feature, and we look forward to your comments and questions on PROFILE III+.*

*Pentacle is a New York City-based non-profit service organization specializing in administrative services for performing art groups.*

| Number of Segments In Your Data Base | Number of Drives Available For Data Assuming Drive Ø Is Reserved For Runtime Programs & Formats | | |
|---|---|---|---|
| | 1 Drive | 2 Drives | 3 Drives |
| 1 segment | filename/KEY /KX1 /KX2 /KX3 will fill additional disk drives as needed (true only of single-segment data bases) | | |
| 2 segments | no problem | keep segments equal | not possible—segments cannot be divided |
| 3 segments | no problem | use proportions of thirds & sixths | keep segments equal |
| 4 segments | no problem | keep segments equal | use proportions of thirds & sixths |

**Figure 1**

## ACCESSING PROFILE DATA FROM AN INDEX, PART 2

In the last issue of this newsletter, we published a program that lets you access PROFILE data sequentially from a PROFILE index. This issue, we've written a program that lets you access PROFILE data randomly (in other words, in a way other than in record number order), again using an index.

The computer does "binary" searches of the index to find a particular piece of data. A binary search is one in which the computer cuts the indexed list in half and in half again as many times as is necessary to pinpoint the desired piece of information. The charm of this demonstration program is that you can watch the process on the screen.

You can also use the program as a module, building a BASIC program around it. Please note that the program shown is a demonstration program only—you should adapt it for your own purposes.

A few notes are in order. This demonstration program accesses data only from the key segment, which is the most difficult to access because it must be unblocked. If you want to access non-key data, include statements between lines 800 and 900 reading as follows:

OPEN "D", 3, F$ + "/DAT," nnn

where "nnn" is the length of the segment and "/DAT" is the second data segment; DA2 and DA3 are the other data segments.

FIELD 3, n AS XX$, m AS YY$ . . .

where "n", "m", etc. are the lengths of your PROFILE fields, and "XX$", "YY$" etc. are field names.

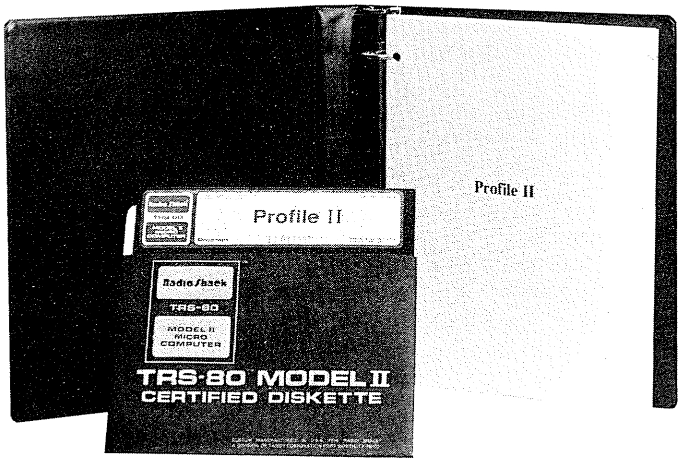
If you want an entire record, you can then access it by typing (between lines 3600 and 3700; make sure you have a third file):

GET 3, LR

where "LR" is the logical record.

To open: BASIC FIND1/BAS-F:2 (two files are needed, one for the key and one for the index)

```
10 '    FIND1/BAS
20 '    The small Computer Company, Inc.
30 '
40 '    Index demo program
50 '    Random access of PROFILE+ records from BASIC using a PROFILE+ index
60 '
70 '
100 CLS : CLEAR 5000 : DEFINT A-Z
200 INPUT " ENTER FILE NAME"; F$
300 F$ = LEFT$(F$+"00000000",8)      ' Pad with zeros to make 8-byte file name
400 INPUT "ENTER KEY LENGTH"; KY
500 OPEN "D", 1, F$ + "/IX1", KY+2   ' Open index file
600 FIELD 1, (KY) AS KY$, 2 AS LR$   ' Key field and two-byte pointer
700 OPEN "D", 2, F$ + "/KEY"         ' Open /KEY segment
800 '
900 '
1000 PRINT : INPUT "       ENTER KEY"; K$
1050 PRINT
1100 '
1200 PT = 0                          ' PT will point to current position in index
1300 HI = LOF(1)+1                   ' Set HI to last index entry plus one
1400 LO = 4                          ' Set LO to first index entry
1500 '
1600 OP = PT                         ' Save old pointer
1700 PT = INT((LO+HI)/2)             ' Set pointer to middle of unexamined portion
1800 IF OP = PT THEN 4400            ' Search completed; no match has been found
1900 GET 1,PT                        ' Read index record
1950 PRINTUSING "####: "; PT;        ' Display index record number being examined,
2000 PRINT "   "; KY$; "  ";         '     key value, and
2050 PRINTUSING"####"; CVI(LR$)      '     PROFILE+ record number
2100 GOSUB 4800                      ' Set index key length to length of search key
2200 IF K$ = RK$ THEN 2800           ' Match found; display PROFILE+ record
2300 IF K$ > RK$ THEN LO = PT        ' Index pointer was still too low
2400 IF K$ < RK$ THEN HI = PT        ' Index pointer was still too high
2500 GOTO 1600                       ' Loop and try again
2600 '
2700 '
2800 GET 1, PT-1 : GOSUB 4800        ' Now look backwards through index
2900 IF K$ <> RK$ THEN 3100          ' Did we overshoot the first match?
3000 PT = PT-1 : GOTO 2800           '     If not, keep looking backwards
3100 GET 1, PT                       ' This is the first match
3200 LR = CVI(LR$)                   ' PROFILE+ record number from index pointer
3300 PR = INT((LR-1)/3)              ' Deblock the KEY segment record:
3400 SR = LR - PR*3 - 1              '     SR = sub-record number (0, 1 or 2)
3500 PR = PR+1                       '     PR = physical record number
3600 GET 2, PR                       ' Get the /KEY record
3700 FIELD 2, (85*SR) AS DD$, 85 AS A$  ' A$ = /KEY segment data fields
3800 PRINT : PRINT LR; "  "; A$      ' Print record number and data
3900 PT = PT+1                       ' Advance pointer to next index record
3920 IF PT > LOF(1) THEN 1000        ' Past end of file?
3950 GET 1, PT : GOSUB 4800          ' Get it and adjust key field for compare
4000 IF K$ = RK$ THEN 3200           ' Loop back if it's also a match
4100 GOTO 1000                       ' Otherwise, ask for next key
4200 '
4300 '
4400 PRINT "NO MATCH FOUND"
4500 GOTO 1000
4600 '
4700 '
4800 RK$ = LEFT$(KY$,LEN(K$))        ' Make index key equal length to K$
4900 RETURN
```



# Introducing SATN, The Journal for VisiCalc Users

Reprinted by permission of Software Arts, Inc.

These articles originally appeared in SATN, Vol. 1, No. 1, Copyright ©, on September/October, 1981, Software Arts, Inc., P.O. Box 494, Cambridge, MA 02139.

*Editor's Note: From time to time we will be publishing articles from the SATN journal in hopes that they may aid our readers and users of VisiCalc® to better understand and utilize the many features of this powerful software package.*

The user of any program deserves strong technical and professional support. When a program is as widely applicable as the VisiCalc® program is, its users benefit from the support already provided by a variety of sources: the manual, the reference card, the distributor, computer dealers, and users' clubs. The SATN pronounced "satin"—journal is designed to supplement, not supplant those sources of support.

The VisiCalc program, a visible calculator program created by Software Arts, and published and distributed exclusively by VisiCorp under their registered trademark VisiCalc, is used by many people in many ways. Some users are novices, others experts; some use it casually, others intensively.

The purpose of SATN, a publication from Software Arts, is to show all VisiCalc users how to use the program more effectively and extensively. The articles in SATN will focus as much as possible on information that is not specific to any particular computer in order to be helpful to all users of the VisiCalc program. SATN's editorial aim is to provide VisiCalc users with informative, in-depth articles about the program drawn from experts both within Software Arts and from outside the company.

This and subsequent issues will include practical applications of the program in sufficient detail to allow you to solve similar problems immediately. To expand your scope of applications, we will also include tutorials and techniques useful in solving a wide variety of problems. We will pass on some pointers about the VisiCalc program to clarify misunderstandings, and describe brief accounts of why VisiCalc features were designed as they were, to provide insight into better ways to use the program.

### A CLOSE LOOK AT @LOOKUP

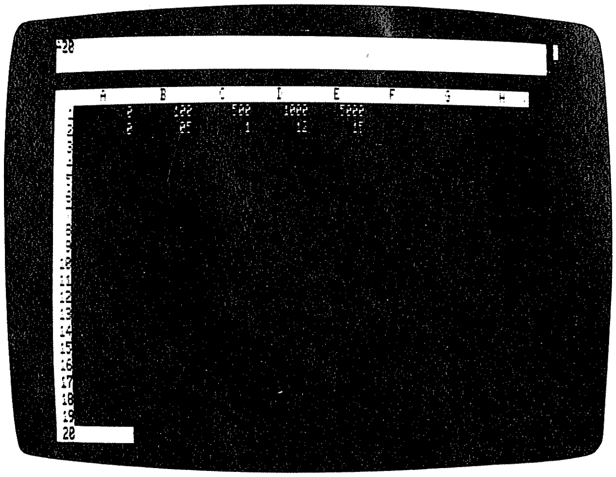
The @LOOKUP function of the VisiCalc program can be an invaluable tool in a number of applications. It is particularly helpful in tax calculations, and in fact was designed with tax tables in mind. But you can use @LOOKUP with tables that you create yourself for your own particular applications.

As we will see @LOOKUP can be used to search an inventory table, or a table of discount percentages; to help calculate the costs of items in which a number of variables come into play; or to perform some payroll calculations involving taxes. Among the other uses of @LOOKUP that come to mind are searches of insurance premium and interest rate tables.

As entered on the sheet, an @LOOKUP table consists of a range of values to be searched and their corresponding values. The corresponding values must be entered on the sheet in the row directly below or the column directly to the right of the range of values.

In operation, the @LOOKUP function takes a number that you have entered, the input value, and sequentially searches the range of the table until it reaches the first number in the range that is greater than the input value. It then returns to the preceding number and selects the corresponding value. The @LOOKUP function assumes that the range is in ascending order, though the corresponding values need not be.

Our first example shows how @LOOKUP is used with a small table of discount percentage. Screen 1 contains the sheet for such a table.



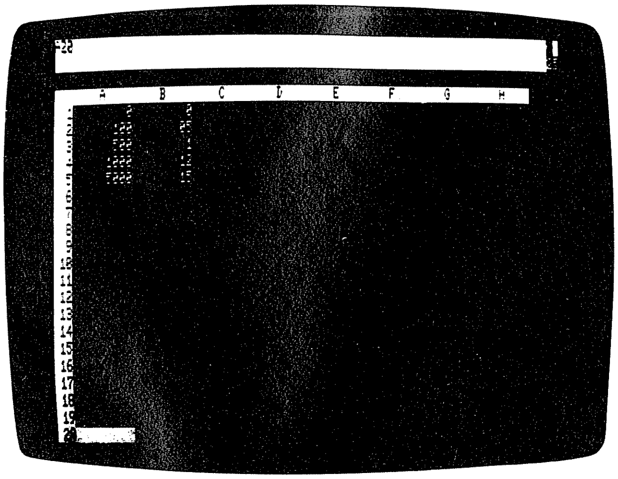
```
        A      B      C      D      E      F      G      H
 1      0    100    500   1000   5000
 2      0    .05     .1    .12    .15
```

Now suppose you need, at C5, the discount applicable to a purchase of 1000 items. To get the appropriate percentage in that location on the sheet you would enter:

```
>C5:@LOOKUP(1ØØØ,A1..E1)
```

@LOOKUP will find the value in the specified range A1 . . . E1, and will return .12, the corresponding value in the second row.

If the sheet is set up in columns, as in Screen 2, the @LOOKUP search is the same. The range, however, is A1 . . . A5 and the corresponding value is found in the column immediately to the right of the input value.



If you wish to find the discount applied to 800 items, @LOOKUP will search the table, matching 800 to each value in the range consecutively. When it reaches 1000, it will stop, back up to the preceding value, and return .10, the correct discount to apply to a purchase of 800 items.

If you use @LOOKUP to find a value that is lower than the first one in the range, the result will be NA (Not Available). If you try to find a discount, for example, for −50 items in the tables in Screens 1 and 2, NA would be returned. If you use @LOOKUP to find a value that is greater than the last value at the end of the table, it goes back to the last value it found. Thus 6000 would return to .15.

Once you have entered an @LOOKUP table on your VisiCalc sheet, you can save it and use it repeatedly, either to retrieve a value that is an end in itself or one that will be used for further calculations. You may of course, enter that table at any location on the sheet and retrieve values by giving the coordinates for the range or by pointing at them with the cursor.

# Business Checkwriter

Business Checkwriter (Cat. No. 26-1585) has been introduced by Radio Shack to provide another approach to the previously released Checkwriter 80. Those customers who felt that Checkwriter 80 did not fully meet their needs, may find Business Checkwriter more to their satisfaction. Additionally, Radio Shack will provide Business Checkwriter to any present owners of Checkwriter 80 at a minimal fee.

The differences between the two packages should be fully understood before a purchase or a switch is made. For instance, a current bank balance, distribution of check to multiple expense codes, detail check stubs, and payee summaries are all available on the new Business Checkwriter, where they were non-existent on the Checkwriter 80 package.

Checkwriter 80 has year-to-date summaries which are available on check register and end of year report. Business Checkwriter does not offer this particular facility.

On current expense summaries, the Business Checkwriter user determines the period for which he desires a summary. Current detail is cleared when an expense summary is printed. Capacity limitations suggest that one month is the maximum length of time. Checkwriter 80 users see current period total printed on the check register.

Business Checkwriter maintains only one account directly, but the user has the option of setting up multiple systems. Checkwriter 80 handles up to a maximum of nine accounts.

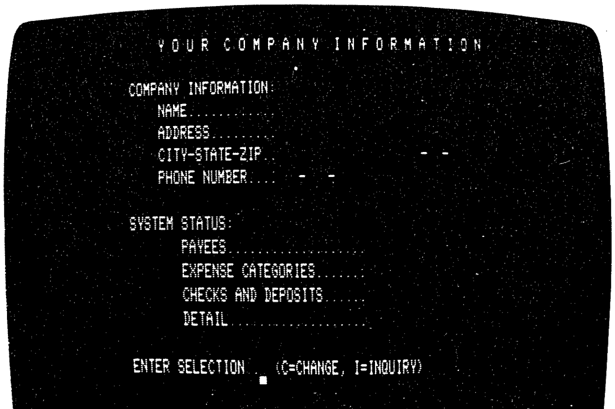
Business Checkwriter sustains up to nine daily deposits. Checkwriter 80 is limited to one deposit per day.

The entry of manual checks is recorded and detail printed with Business Checkwriter. It is simply recorded with Checkwriter 80.

With both packages bank reconciliations are available with report on demand.

The two sets of files are not compatible. Users cannot move their Checkwriter 80 files to Business Checkwriter files and vice versa. Furthermore, Business Checkwriter uses different check forms than Checkwriter 80.

A note to the buyer/user: Please be sure to provide verification, such as a sales receipt, for Checkwriter 80 in order to acquire Business Checkwriter at the minimal fee.



```
            YOUR COMPANY INFORMATION

COMPANY INFORMATION:
   NAME...........
   ADDRESS........
   CITY-STATE-ZIP..                    - -
   PHONE NUMBER....  -  -

SYSTEM STATUS:
     PAYEES..................
     EXPENSE CATEGORIES......
     CHECKS AND DEPOSITS.....
     DETAIL..................

ENTER SELECTION   (C=CHANGE, I=INQUIRY)
```

# Have We Got Printers for You!

While writing about the new Radio Shack printers, it was hard to keep the superlatives that kept flowing from my keyboard to a minimum, but a concerted effort was made to simply state the facts so that you could see for yourselves the abundance of features to be found in the new printer line.

## DMP, DWP, CGP?

In reading about the new printer line, you may notice right off that when naming our latest and greatest in printers we have gone in for acronyms in a big way. DMP stands for Dot Matrix Printer, and DWP? . . . Right, Daisy Wheel Printer. You may even have noticed in October that CGP stood for Color Graphics Printers. We have a lot of clever things like that up our sleeves.

So much for chit chat, on to the facts.

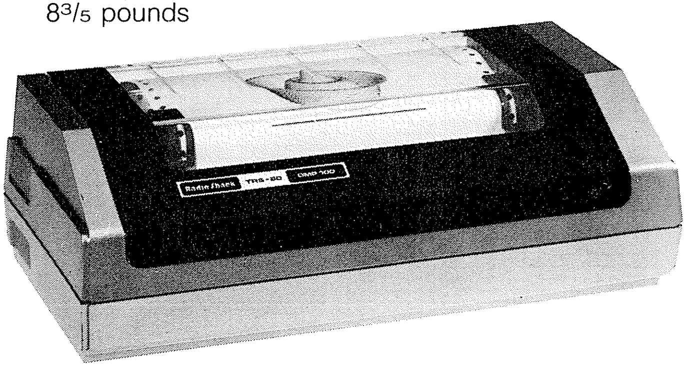
## DMP-100

The DMP-100 has all the capabilities of the Line Printer VII along with the three important differences below.

- It's faster. For instance, the DMP-100 prints 50 characters per second on a ten character per inch (cpi) line at 27 lines per minute.
- It has underlining capability.
- It has a 480 byte full-line dot buffer for much faster graphic printing. Compare this to the 90 byte buffer of the Line Printer VII.

Other features of the DMP-100 are:

- A bit image mode with 480 addressable 7 dot columns for printing high-density graphics—up to 3780 dots per square inch 80 upper/lower case 5 x 7 matrix characters (a total of 96 ASCII characters) on an 8″ line that can be printed at either 5 or 10 cpi
- Both parallel and serial interfaces to run with the Color Computer, Model I, II, III, or 16
- A ribbon cassette and an adjustable 4½ x 9½″ tractor which can use up to 9½″ fanfold paper
- A compact 5⁵/₁₆ x 16 x 8¹/₄″ printer that weighs only 8³/₅ pounds



The many features of the DMP-100 (catalog no. 26-1253, $399.00) make it a very useful printer at a very attractive price.

## THREE MORE DOT MATRIX PRINTERS

The DMP-200, DMP-400, and DMP-500 share many outstanding features. They all have built-in graphics capability, bi-directional logic-seeking 9-wire print heads, and the ability to print lower-case descenders and true underlines.

More features of these three printers are:

- A Word Processing mode to emulate the Daisy Wheel printers which includes:
- Proportionally spaced and mono-spaced (Standard at 10 CPI, Elite at 12 CPI, and Correspondence at 10 CPI) character sets as well as 16.7 CPI Condensed.
- Backspacing, boldface, underlined characters, subscripts, and superscripts
- A Bit Image dot-addressable mode for graphics and special repeat and column addressing codes to simplify graphic programming
- A Bold mode to allow a wide range of character styles and a character elongation mode for printing wider characters which like the bold mode can be used in all fonts
- A Data Processing Mode for faster throughput

The DMP-200, DMP-400, and DMP-500 also include switches to allow manual selction of modes, pitches and interface options as well as the ability to select these options through software.

Because all three printers have a friction platen, they all can use 8″ single sheet paper (The 15″ printers can use wider paper.) in addition to standard fanfold paper on the adjustable tractor. The general specifications for these three printers are:

Character Set
96 ASCII, 32 special and 30 block graphics characters
Line Feed
Forward, reverse; 1/6, 1/12, 1/72, 1/216
Print Density
10, 12, or 16.7 characters per inch in either a 9 x 9 or 15 x 9 dot matrix, n-23 matrix in a proportionally spaced font—all fonts can be elongated
Resolution
60 dots per inch at 10 cpi
72 dots per inch at 12 cpi
100 dots per inch at 16.7 cpi
Buffer
2K RAM

For a more detailed comparison of these three printers plus the DMP-100 and the predecessors of these three printers, refer to the Radio Shack Dot Matrix Printer Comparison Chart included in this article.

Now for a look at the features that make the DMP-200, DMP-400, and DMP-500 unique.

## DMP-200

The DMP-200 is about the size of the Line Printer VIII but there are some noticeable differences. It has a paper guide which functions similarly to a typewriter paper guide. When in the up position, the paper guide holds the paper up as it comes through the printer so the page can be seen as it is printed. When the paper guide is in the down position, it works as a paper separator for multipart, fanfold paper. Other new features of the DMP-200 include:

A removable, adjustable tractor instead of the non-adjustable pin feed of the Line Printer VIII

Bi-directional carriage to give printing speed of 55 full lines per minute and 120 characters per second at 10 characters per inch

Selectable parallel and serial interfaces (600/1200 baud)

The DMP-200 includes line feed increments of 1/6, 1/8, 1/72, and 1/216 and on/off line, restart, and paper feed controls. The 8″ line lets you print 80 columns of characters at 10 pitch or 143 characters at 16.7 pitch. The DMP-200 (catalog #26-1254, $799.00) is a compact—5×6 1/2×13 1/2″, 16 lbs.—but very versatile printer.

### DMP-400

This terrific 15″ wide printer offers the same size efficient styling found in the Line Printer VI. But just look at some of the additional features of the DMP-400.

Cover design which aids in noise reduction

Line feeds in 1/6, 1/8, 1/72, 1/216 increments

Selectable parallel and serial interfaces

Paper movement switch which when in the up position advances the paper 1/72″. If the switch is held up continuously, the paper continues to advance. When the switch is pressed down, the paper does a reverse feed (moves backwards) of 1/72″.

For operator convenience there is a LineFeed/Form Feed switch on the front panel. When the switch is set to Line Feed, the paper advances one line. If held in the up position, the paper will feed continuously. When the switch is set to Form Feed, the paper will advance to the logical top of the next form.

Software controllable Form Feed for graphics.

Full 132 column printing capacity on a 13.2″ line

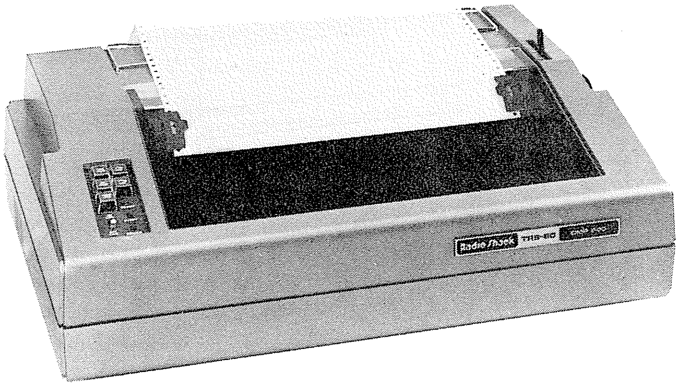
The printing speeds of the DMP-400 vary according to typeface and the number of columns per line. For example, with a standard type face and a 20 column line the DMP-400 prints 181 lines per minute (lpm) while when using the same type face to print 132 columns across the page, it prints at a rate of 51 lpm. Using the correspondence type face at 10 cpi, it prints at a rate of 705 words per minute (wpm).

The DMP-400 (catalog no. 26-1251, $1195.00) includes an adjustable, removable tractor and don't forget the list of features that apply to this dot matrix printer as well as the DMP-200 and DMP-500.

### DMP-500

The DMP-500 is a heavy duty, high performance, dot matrix printer which lends itself favorably to a business environment requiring large volumes of through-put. One of the most outstanding features of the DMP-500 is its speed. This high speed printer prints 220 characters per second and seventy-six 132-column lines per minute.

Since it can line feed in 1/6, 1/8, 1/72, 1/216 increments, the paper can advance in fine-line increments for aligning preprinted forms.



The DMP-500 offers:

A standard parallel interface so it can be used with the Models I, II, III, or 16

Paper feed and form feed controls plus an on-line indicator

The DMP-500 (catalog no. 26-1252, $1795.00) is a great value in a powerful printer.

### DOT MATRIX PRINTER COMPARISON CHART

The printer comparison chart gives you an indication of the family history of the new dot matrix printers. The old printers are those that have been replaced in the Radio Shack printer line by the corresponding new printers, e.g. the Line Printer VII has been replaced by the DMP-100. Many similarities between the old and the new exist in appearance, capability, and price, but there are, as we have seen, noteworthy differences.

# Radio Shack Dot Matrix Printer Comparison Chart

| | Old<br>LP VII<br>26-1167 | New<br>DMP-100<br>26-1253 | Old<br>LP VIII<br>26-1168 | New<br>DMP-200<br>26-1254 |
|---|---|---|---|---|
| Matrix Format | | | | |
| Size in dots (H×W) | 7×5 | 7×5 | 9×〈=23 | 9×〈=23 |
| Descenders | No | No | Yes | Yes |
| Underline Mode | No | Yes | Yes | Yes |
| Bold Mode | No | No | No | Yes |
| Max. Line Length | 8″ | 8″ | 8″ | 8″ |
| Horizontal Pitches (cpi) | | | | |
| Fixed | 5,10 | 5,10 | 5,10,16.6 | 10,12,16.7 |
| Proportional | No | No | Yes | Yes |
| Elongate any pitch | NA | Yes | NA | Yes |
| Mixed pitch per line | Yes | Yes | Yes | Yes |
| Line Pitches (LF per inch) | 6 | 6(9) | 6,8,12 | 6,8,12,72,216 |
| Speed (Throughput) | | | | |
| CPS @ 10 cpi | 30 | 50 | 80 | 120 |
| LPM @ 10 cpi | 19/80 col<br>53/20 col | 26/80 col<br>47/40 col<br>78/20 col | 23/80 col<br>55/20 col | 55/80 col<br>80/40 col<br>104/20 col |
| Logic Seeking (1) | Short Line | Short Line | Yes | Yes |
| Bi-directional | No | No | No | Yes |
| Character Set | | | | |
| Standard ASCII (7 bit) | Yes | Yes | Yes | Yes |
| International (8 bit) | No | No | Yes | Yes |
| Graphics | | | | |
| Block Characters | No | No | Yes | Yes |
| Bit Image: Dots per line | 480 | 480 | 960 (3) | 480-800 |
| Paper Handling | | | | |
| Maximum Size | 9½″ | 9½″ | 9½″ | 9½″ |
| Friction Feed | No | No | Yes 8.5″ (s,r) | Yes 8.5″ (s,r) |
| s—Single Sheet | | | | |
| r—Roll Paper | | | | |
| Tractor (Fan fold) | Adj (4.5″-9.5″) | Adj (4.5″-9.5″) | Fixed 9.5″ | Adj. 10″ |
| Form Feed | NA | No | NA | Yes |
| Copies | Orig. +2 (2) | Orig. | Orig. +2 | Orig. +2 |
| Word Processing Features | | | | |
| Correspondence Fonts | | | | |
| 10 cpi | NA | No | NA | Yes |
| PS | No | No | Yes | Yes |
| Incremental Spaces | | | | |
| Backspace | No | No | Yes | Yes |
| For/Rev Linefeed | No | No | Yes | Yes |
| For/Rev ½ Linefeed | No | No | Yes | Yes |
| Controls | Power | Power | Power<br>On/Off Line<br>Reset<br>Restart | Power<br>On-line/Off-line<br>Reset<br>Paper feed<br>form feed |
| Indicators | 1 | 1 | 2 | 2 |
| User Self Test | No | No | Yes | Yes |
| Reliability | | | | |
| Head Life (Chars)(2) | 30 Million | 30 Million | 〉30 Million | 30 Million |
| MTBF@100% | 300 Hrs (50% Duty) | 〉300 Hrs (50% Duty) | 300 Hrs | 〉300 Hrs |
| Ribbon | | | | Refillable |
| Type | Re-inking cass | Re-inking cass | Cartridge | Cartridge |
| Life | 1 Mil chr. | 1 Mil chr | 1 Mil chr | 4 Mil Char |
| Catalog Number | 26-1424 | 26-1424 | 26-1418 | 26-1483 cart<br>26-1489 refill |
| Interface | | | | |
| Buffer | 90 bytes | 480 bytes | 160 bytes | 2K bytes |
| Parallel Data Bits | 8 | 8 | 8 | 8 |
| Status: Busy | Yes | Yes | Yes | Yes |
| Bits: Fault | No | No | Yes | Yes |
| : Paperout | No | No | Yes | Yes |
| : Busy | | No | | Yes |
| : Select | No | NA | Yes | NA |
| Serial (3) | | | | |
| Data Bits | 7,8 | 7,8 | 8 | 8 |
| Baud Rate | 600 | 600/1200 | 600/1200 | 600/1200 |
| Size | 5.32×16×8.25″ | 5.32×16×8.25″ | 4.7×15.4×11″ | 5×16.5×13.5″ |
| Weight | 8.6 lbs. | 8.6 lbs. | 16.5 lbs. | 16.5 lbs. |
| Power 120 VAC±15% | 15W | 15W | 85W | 85W |
| Price (less cable) | NA | $399.00 | NA | $799.00 |

NOTE: Specifications are based on the most accurate information available at the time of publication, subject to change.

| | Old | New | Old | New |
|---|---|---|---|---|
| | LP VI | DMP-400 | LP V | DMP-500 |
| | 26-1166 | 26-1251 | 26-1165 | 26-1252 |
| Matrix Format | | | | |
| Size in dots (H × W) | 7 × 9 | 9 × ⟨ = 23 | 9 × 9 | 9 × ⟨ = 23 |
| Descenders | No | Yes | Yes | Yes |
| Underline Mode | No | Yes | Yes | Yes |
| Bold Mode | No | Yes | Yes | Yes |
| Max. Line Length | 13.2" | 13.2" | 13.2" | 13.2" |
| Horizontal Pitches (cpi) | | | | |
| Fixed | 5,7.5,10,15 | 10,12,16.7 | 5,7.5,10,15 | 10,12,16.7 |
| Proportional | No | Yes | No | Yes |
| Elongate any pitch | NA | Yes | NA | Yes |
| Mixed pitch per line | Limited | Yes | Limited | Yes |
| Line Pitches (LF per ") | 6,8,12 | 6,8,12,72,216 | 6,8,12 | 6,8,12,72,216 |
| Speed (Throughput) | | | | |
| CPS@10 cpi | 100 | 140 | 160 | 220 |
| LMP@10 cpi | 33/132 col | 51/132 col | 60/132 col | 76/132 col |
| | 42/80 col | 70/80 col | 100/66 col | 119/80 col |
| | 120/20 col | 126/40 col | 198/22 col | 200/40 col |
| | | 181/20 col | | 293/20 col |
| Logic Seeking (1) | Yes | Yes | Yes | Yes |
| Bi-directional | Yes | Yes | Yes | Yes |
| Character Set | | | | |
| Standard ASCII (7 bit) | Yes | Yes | Yes | Yes |
| International (8 bit) | Yes | Yes | Yes | Yes |
| Graphics | | | | |
| Block Characters | Yes | Yes | Yes | Yes |
| Bit Image: Dots per line | No | 792-1320 | No | 792-1320 |
| Paper Handling | | | | |
| Maximum Size | 15" | 15" | 15" | 15" |
| Friction Feed | Yes (s) | Yes (s) | No | Yes (s) |
| s—Single Sheet | | | | |
| r—Roll Paper | | | | |
| Tractor (Fan Fold) | Adj. (4"-15") | Adj. (4"-15") | Adj. (4"-15") | Adj. (4"-15") |
| Form Feed | NA | Yes | NA | Yes |
| Copies | Orig. + 3 | Orig. + 3 | Orig. + 4 | Orig. + 4 |
| Word Processing Features | | | | |
| Correspondence Fonts | | | | |
| 10 cpi | NA | Yes | NA | Yes |
| PS | No | Yes | No | Yes |
| Incremental Spaces | | | | |
| Backspace | No | Yes | No | Yes |
| For/Rev Linefeed | No | Yes | No | Yes |
| For/Rev 1/2 Linefeed | No | Yes | No | Yes |
| Controls | Power | Power | Power | Power |
| | On-line/Off-line | On-line/Off-line | | On-line/Off-line |
| | Reset | Paperfeed/Form Feed | Reset | Form Feed |
| | | Restart | Restart | Restart |
| | Line feed | 1/12"paper feed up, down | Line feed | Line feed |
| | 1/12 line feed | | For 1/8 line feed | 1/12" paper feed up,down |
| | | | Rev 1/8 line feed | |
| Indicators | 3 | 3 | 3 | 3 |
| User Self Test | Yes | Yes | Yes | Yes |
| Reliability | | | | |
| Head Life (Chars)(2) | 100 Million | 100 Million | 100 Million | 100 Million |
| MTBF@100% | 300 Hrs. | ⟩300 Hrs. | 300 Hrs. | ⟩300 Hrs. |
| Ribbon | | | | |
| Type | Cartridge | Cartridge | Cartridge | Cartridge |
| Life | 1 Mil Char | 1 Mil Char | 2 Mil Char | 4 Mil Char |
| Catalog No. | 26-1418 | 26-1418 | 26-1414 | 26-1482 |
| Interface | | | | |
| Buffer | 1 Line | 2K bytes | 1 Line | 2K bytes |
| Parallel Data Bits | 8 | 8 | 8 | 8 |
| Status: Busy | Yes | Yes | Yes | Yes |
| Bits: Fault | Yes | Yes | Yes | Yes |
| : Paperout | Yes | Yes | Yes | Yes |
| : Busy | NA | Yes | NA | Yes |
| : Select | Yes | NA | Yes | NA |
| Serial (3) | | | | |
| Data Bits | NA | 8 | NA | NA |
| Baud Rate | NA | 600/1200 | NA | NA |
| Size | 6.3 × 24.2 × 13.3" | 6.3 × 24.2 × 13.3" | 7.4 × 24.4 × 15.9" | 7.4 × 24.4 × 15.9" |
| Weight | 28 lbs. | 28 lbs. | 42 lbs. | 42 lbs. |
| Power | 85W | 85W | 85W | 85W |
| Price (less cable) | NA | $1195.00 | NA | $1795.00 |

Notes: (1) Eliminates leading and trailing spaces in print line
(2) Heavy use of graphics can accelerate head wear
(3) 4 Pin DIN Connector (RS232 DATA, STATUS, GROUND)

### BEYOND DOT MATRIX

All the news in Radio Shack printers is not just dot matrix printers. There's the CGP-155 Color Graphics Printer which we told you about in October and a new Daisy Wheel printer, the DWP-410.

### DWP-410

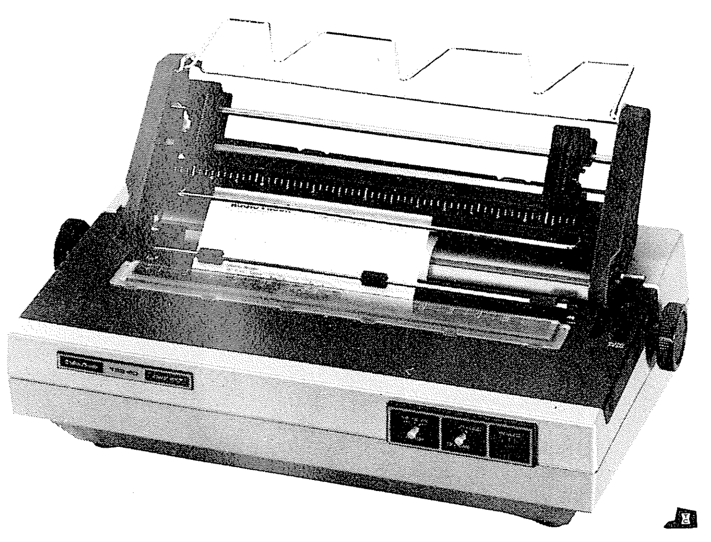
The DWP-410 is ideal for giving a professional look in word processing applications using Scripsit. Some if its features are:

- Switch selectable print densities (10 or 12 characters per inch or proportional pitch) on a 13.6" line
- Forward and reverse paper feed and 1/2-line feed
- Underline, boldface and programmable backspace
- 1/120" space and 1/48" line feed
- Automatic paper set to ease paper insertion. Simply place the sheet of paper at the rear of the 15" platen and pull the Autoset lever to insert the paper.
- It accepts paper up to 16" wide and it will print an original plus 5 copies.
- An external program mode to let you use print wheels with different pitch or special characters
- Interchangeable 124 character print wheels (Courier at 10 pitch, Prestige Elite at 12 pitch, Madeline PS, Cubic PS, Tile Italic at 12 pitch, OCR B, Letter Gothic at 12 pitch, Cubic 15, and PS Bold)
- A nominal print speed of 25 characters per second and 11.7 lines per minute at 10 cpi on a 132 column printout which is over 300 wpm
- Character set: 96 ASCII characters plus special and international for a total of 124 characters plus 21 control codes
- Ribbon—multi-strike carbon or nylon cartridge
- Standard 8 bit parallel interface
- Unidirectional, optimal motion seeking carriage return

Also available for the DWP-410 is an optional bi-directional tractor feed (26-1459 at $239.95). With the addition of the easily installed tractor feed, the DWP-410 is ideal for printing labels, multi-part forms or long DP runs. The DWP-410 (catalog no. 26-1250, $1495.00) is an outstanding Daisy Wheel printer.

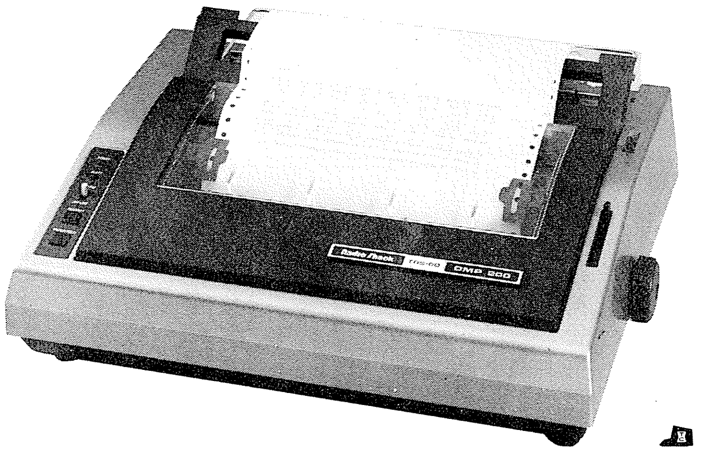
### THE FINAL ANALYSIS IS . . .

Each printer whether Dot Matrix or Daisy Wheel is impressive in its own right, and most printing applications from graphics to word processing can be met by these printers.



# Note to SuperSCRIPSIT Users with the DMP-200

If you are using SuperSCRIPSIT (26-1590) with the DMP-200 (26-1254), you will need to use the printer driver for the DMP-400 which is on the SuperSCRIPSIT program disk. To load the driver, type DMP400 for the printer type when opening a document.



# Notes on Previous Newsletters

## JANUARY 1982

### Cassette Telephone File

**Keith Theriot**
**P.O. Box 46**
**Luling, LA 70070**

When attempting to create the cassette telephone file, I experienced trouble in obtaining the particular record I desired.

I remedied this by rewriting program line 540:

From:

```
54Ø  PRINT
     : PRINT A$, B$
```

To:

```
54Ø  IF A$ = Z$ THEN PRINT
     : PRINT A$, B$
```

After this correction the proper file appeared every time program execution occurred.

# FEBRUARY 1982

## Random Files

**John K. Segars, Ph.D.**
**P.O. 829**
**142-B South Dargan Street**
**Florence, SC 29503**

I would like to note an error in the feature article. On page 5, you stated that if the record number is omitted from the PUT statement, the record written will be number one. This is not true on my Model II (TRSDOS 2.0) or Model III (TRSDOS 1.3). The first record used will be number two. Record one is skipped and contains hexadecimal "e". This can cause problems if you later read record one by using a record number with the GET statement.

## Typewriter

**John E. Blank, M.D.**
**3352 E. State St.**
**Sharon, PA 16146**

The February issue included a very neat little program that was written by David Salisbury for use on the Model II. The program runs perfectly well on the Model III with one tiny little flaw— no cursor. You can get by without it as long as you keep your mind on your work. Or a simple fix in Line 500 eliminates the problem.

```
5ØØ  CLS
     : PRINT CHR$(14)
     : F=Ø
```

## Mailing List Files

**Neal H. Krape**
**986 Northbriar Drive**
**York, PA 17404**

Mr. William G. S. Smith suggests that the solution to the problem of alphabetizing a file of people's names for mailing lists and other purposes by using two fields, one for the last name and the other for the first name, middle name, etc. "is not an ideal solution" because it "can lead to some pretty strange looking mailing labels, with huge gaps between first name and last name because of unused spaces". He suggests typing the last name twice, once in a special field for alphabetizing, and again in the field used for printing.

He might wish to go back to the solution which he said is not an ideal one, if he includes one more line in his program, probably just before the "PRINT" in which the name is printed. Assuming that the field of last names is "NL" and the field of first names, etc. is "NF" (placing "N" first makes one-variable definition as string possible: DEFSTR N), then insert the following line just before the PRINT instruction:

```
nnnn NP = NF +" "+ NL
```

Then print NP instead of NF and NL separately

Concatenating the strings to obtain a single string assures that the huge gaps will not appear between first and last name, and avoids the necessity of typing the last name twice.

# MARCH 1982

## WILLIAM BARDEN JR.'S SERIES

**Ralph Bloch**
**202 Rosemary Lane**
**Ancaster, Ont.**
**Canada**
**L9G 2K7**

With the scarcity of articles for the TRS-80 Color Computer, William Barden Jr.'s series fills a great need. His lighthearted style makes for easy reading and alleviates the neophyte's fear of machine language programming.

However, I would like to suggest a change in the use of subroutine calls. In the first statement of the SET routine in the Graphics program, he makes a Long Branch to subroutine INTCNV, an absolute address in ROM. The assembler will automatically generate the correct offset, that is no problem. If a user would like to relocate the machine language program, it will bomb, unless reassembled at the new address, since with the LBSR instruction, only the relative offset is carried in the machine code. The same holds true for line 00310.

To his four rules, I would therefore like to add three more (at least, if you value relocatability):

5. Distinguish between subroutine calls within your own program, and subroutine calls to utilities residing in ROM.
6. Access utilities with absolute calls, e.g. JSR $B3ED.
7. Access your own subroutines relatively, e.g. BSR ADDMSK, (LBSR, if your program is longer).

I hope that letters like this one will gradually lead to a correspondence of interested assembly language programmers, to discuss the pros and cons of various programming strategies.

## Blinking Prompt and INKEY$

**Marc Poulin**
**6750 El Colegio #103**
**Goleta, CA 93117**

I am writing in regard to Mr. Sal Vescera's letter which appears on page 17. Mr. Vescera describes a simple technique he uses to make a prompt message flash on the screen and I would like to expand on his comments.

Mr. Vescera uses the following four lines of code to achieve the blinking effect:

```
1Ø PRINT @ 96Ø,"< >ES OR < >O";
2Ø FOR I=1 TO 2Ø
     : NEXT I
3Ø PRINT @ 96Ø,"<Y>ES OR <N>O";
4Ø GOTO 1Ø
```

This results in a rapid flickering of the "Y" and "N" rather than a steady blinking.

The University of Michigan's Highway Safety Research Institute has determined that a light flashing at about four times per second produces the strongest response in automobile drivers (see "Headlight Modulators: Their Function and Use," Rider:July 1982, p.51). With this in mind, I would like to suggest modifying Mr. Vescera's algorithm slightly:

```
1Ø PRINT @ 96Ø,"<Y>ES OR <N>O";
2Ø FOR I=1 TO 5Ø
     : NEXT I
3Ø PRINT @ 96Ø,"< >ES OR < >O"
4Ø FOR I=1 TO 3Ø
     : NEXT I
5Ø GOTO 1Ø
```

This flashes the "Y" and "N" at about four times per second and leaves the letters "on" for about 60% of the flashing cycle.

## MAY 1982

### SuperSCRIPSIT—An Overview

There is a correction to the article. The SuperSCRIPSIT catalog number should be 26-1590.

## JUNE 1982

### Disk Head Cleaner

**Paul G. Kriz**
**25 Stonehedge Drive**
**West Nyack, NY 10994**

The program on page 17 is a useful addition to everyone's disk system program library. It needs, however, one change to make it work properly on the Model III.

Statement line 100 issues an 'OUT 241,TR' to set track number. While it is true that PORT 241 addresses the floppy disk controller's track register, this register should contain the track you are currently at, not the track that you want.

The track that you desire to seek should be sent to the floppy disk controller's data register which is addressed by PORT 243. Since the "RESTORE OPERATION" on LINE 65 also resets the track register to zero, no further track register commands are needed.

To make the program work properly, change the 'OUT 241,TR' on Line 100 to 'OUT 243,TR'.

# Software Driver for Model I Voice Synthesizer

**Stephen D. Griffis**
**RD #1, Truver Road**
**Valencia, PA 16059**

The following routine is a software driver for the TRS-80 Voice Synthesizer. It is an alternate method to that given in the operator's manual. Instead of entering strings of 32 characters or less, you can enter strings of up the 254 characters. This will save considerably on subroutine calls.

Lines 20-110 demonstrate the use of the subroutine. The driver is in lines 130-220. Line 140 adds the space to the end if the string in the last character is other than a space. This space is necessary. Without it, you will get an "Illegal function error". The main part of the program explains how the spaces are used. Line 150 looks for the first space. VC then points to the end of the first words plus one. Line 160 then stores the first word minus the space into variable VX$. VY$ is then shortened by removing the first word and its space. Lines 170-190 are directly out of the manual. They load the individual characters of VX$ into the Synthesizer buffer. Line 200 is a delay so that the separate words are not loaded into the buffer too fast. If this delay is too short, the words will sound clipped. Line 210 checks to see if done. If VY$ is only one character long that character has to be the final space. Line 220 is another delay, though this one is for effect. Without it, the Synthesizer will sound like its talking in one complete sentence.

The timing delay in line 200 is about the best compromise I could come up with between speed (or should I say slowness) of speech and intelligibility. If you keep each word short, you can shorten the delay to about 130.

```
10 CLS
     : CLEAR 500
20 PRINT "THIS PROGRAM IS A DEMONSTRATION OF AN
     ALTERNATE METHOD OF DRIVING THE TRS-80 VOICE
     SYNTHESIZER."
30 VY$="<=ISS PØR[UGR95M IISS 66
     DØ435M678NSTR)@>4N 768VV 99NN 12LTRN435T
     M435=78DØ 768VV DR;#&V!E+ <86 TE. ;8R 43#SS
     @*&D& VO85&SS SIN=78SA;5#ZR/"
     : GOSUB 140
40 PRINT "THE SUBROUTINE BELOW ACCEPTS COMPLETE
     SENTENCES AND PHRASES EVEN THOUGH THEY MAY
     BE LONGER THAN 32 CHARACTERS."
50 VY$= "<86 S678BR(UUTE.N BEEL[OU 6KS435PTS
     K7MPOLE.T S43NT43NS43S 995NDD FR))*Z435
     EV435NN =[OU <5))*& M@** BEE& L12NGR/ <99N
     ==/RD& T((UU K34/45KTR/Z"
     : GOSUB 140
60 PRINT "THE SUBROUTINE BREAKS THE SENTENCE INTO
     ITS INDIVIDUAL WORDS AND THEN PROCESSES THEM
     THROUGH THE VOICE SYNTHESIZER."
70 VY$="<86 S678BR(UUTE.N BR@*&KS <86 S43NT43NS
     IINT(UU IITTØS IINDØDIIVIIDJ(U2LW/RDZ 995NND
     <43NN PR;AS43S43S <43M =R((U <86 VOR5&SS
     SIN=78SA;5#ZR/"
     : GOSUB 140
80 PRINT "THE FIRST LINE CHECKS TO SEE IF THE
     LAST CHARACTER IN THE ENTRY IS A SPACE. IF
     NOT, IT ADDS ONE TO THE END."
90 VY$="<86 F/RST L;5#&N TC35KS T((UU SE. IIF <86
     L995ST K34/45KTR/ IINN <86 55NTØR& IISS 66
     SP)@*&S"
     : GOSUB 140
     : VY$="IIF NA;8T IITT 995DZ W877NN T((UU <86
     35NDD"
     : GOSUB 140
100 PRINT "THIS SPACE IS NECESSARY SINCE THE
     SPACE IS USED TO DELINEATE THE INDIVIDUAL
     WORDS."
110 VY$="<=ISS SP)@*&S IIZZ N435S435S)@R& SINS
     <86 SP)@*&S IIZZ &(UUZDØ T((UU DE.LINE))*T
     <86 IINDØDIIVIIDØ(U2L W/RDZ"
     : GOSUB 140
120 END
130 REM DRIVER SUBROUTINE
140 IF RIGHT$(VY$,1)<>CHR$(32) THEN
     VY$=VY$+CHR$(32)
150 FOR VC=1 TO LEN(VY$)
     : IF ASC(MID$(VY$,VC,1))<>32 NEXT
160 VX$=LEFT$(VY$,VC-1)
     : VY$=RIGHT$(VY$,LEN(VY$)- VC)
170 POKE 16383,63
     : POKE 16383,32
180 FOR VX=1 TO LEN(VX$)
     : POKE 16383,ASC(MID$(VX$,VX,1))
     : NEXT
190 POKE 16383,32
     : POKE 16383,63
     : POKE 16383,32
200 FOR I=1 TO 200
     : NEXT
210 IF LEN(VY$)>1 THEN 150
220 FOR I=1 TO 500
     : NEXT
     : RETURN
```

# LDOS—The New Operating System for Your TRS-80

by Logical Systems Inc.

Back in 1980 several alternative operating systems offering added features for TRS-80 users with special needs appeared for the TRS-80 Model I and soon after for the Model III. One of the alternative operating systems to appear at that time was LDOS from Logical Systems, Inc.

LDOS is the operating system that Radio Shack has selected for the new Hard Disk, and it will also be available on floppy for those who have not yet purchased Hard Disk.

### LDOS AND THE HARD DISK

The power and speed of your Model I or III can now be increased many-fold with the new Hard Drive and its LDOS operating system. Five million characters of storage in one small, quiet and highly reliable 14 by 15 inch package, with access speeds many times faster than floppy drives. Best of all, this new storage device is very cost effective at less than .05 (1/20th) cents per byte versus double density floppy storage at more than .25 (1/4th) cents per byte. This means that Radio Shack Hard Disk storage is about FIVE times more cost effective than floppy drive's for on-line storage.

Getting up and running on the new Radio Shack Hard Drive system is simple and straight-forward as LDOS allows many existing applications to run without any modifications. The Hard Drive system features simplicity, compatibility, speed, reliability and cost effectiveness, too . . . Think about it. As your storage demands increase (and there never does seem to be enough disk space) consider a Radio Shack Hard Drive. It is probably the answer.

### LDOS ON FLOPPIES

LDOS is available in floppy disk versions as catalog numbers 26-2213 (Mod I) and 26-2214 (Mod III). The LDOS package is similar in both the floppy and the hard disk versions. Therefore, the following discussion of features will pertain to all versions.

### LDOS—AN OVERVIEW

LDOS offers the TRS-80 user many features and enhancements that are not included in TRSDOS. The disk BASIC (LBASIC) that is provided with LDOS is, for the most part, upward compatible with the disk BASIC on TRSDOS Model III. Most BASIC programs for either the Model I or III should run on LDOS without change. LBASIC has several additional features that TRSDOS BASIC does not have. One of the most important features being the high speed LOAD and SAVE of programs and the default extension of /BAS on LBASIC programs. LBASIC also has the ability to go from one program to another and have the values of all variables carried from one program to the next. This makes it possible to break a program into small pieces and yet have it run as one program. Combined with the high speed load this makes a whole new world available to BASIC programmers because memory constraints become less of a problem. Renumbering and variable cross-referencing are available as well as several new file modes and single character abbreviations for commonly used command line directives like LIST, AUTO, EDIT and DELETE. There are many other features in LBASIC which I hope to address in future articles, along with ideas on how to use these features.

### COMMUNICATIONS

LDOS comes with a very versatile communications utility called LCOMM. LCOMM allows direct upload and download from your RS-232 to/from disk. It also provides many high level "terminal" type functions like printer support (with spooling) and the ability to generate and send all 128 ASCII characters from the keyboard, plus many other features. If communications is a major consideration for you then LDOS is probably worth having for LCOMM alone.

### DEVICE INDEPENDENCE

Now I'll touch on the system itself. There is a major design difference between TRSDOS and LDOS that is "Device Independence." Sounds impressive, but what is it? Well, it's really quite simple. Each input or output device (Keyboard, video, printer, RS-232, Diskfile, etc.) is treated very similarly by LDOS. For the most part, these "devices" can be interchanged, interconnected, and redirected. For example, you could tell LDOS to send all characters that would have gone to the video screen to the printer instead. This is called a ROUTE. You could even ROUTE your printer output to a disk file. Then all characters that would have gone to the printer will actually go to the disk file instead. The file can then be viewed or printed out later. You can also have all output to a given device go to another device simultaneously. This is called a LINK.

You can even create new devices or alter the utility of existing devices by giving them a name and turning on a compatible "driver." This is called a SET. For the most part this independence is the same for input and output devices. The standard LDOS devices are *KI for the keyboard, *DO for the video display, and *PR for the printer. Additional devices can be created for special purposes. For instance, we could create a device called *CL to act as a communications line. To use *CL, we might SET *CL TO RS232T. This command sequence would SET the our newly created communications line (*CL) to the Model III RS-232 driver (RS232T), using default settings for the RS-232.

### LDOS FILTERS

This Device Independent concept also brings about another powerful feature available in LDOS, the FILTER command. Let us imagine that your computer sends information around inside it, to and from its various devices, one character at a time. This then is the I/O stream between devices and files. Now let us assume that we know the start and end of a certain data path in the machine. We know that a certain character will from time to time pass down this data path on its way to its destination. When this character does appear we wish to take some predetermined action. We may wish it changed to another character, count the number of times this character comes by, or maybe simply throw this particular character away. A good example would be FILTERing your printer to keep certain control characters, (which it is unable to handle), from being presented to it. Some filters are provided with LDOS and are invoked by simply telling LDOS to filter a certain device, with a certain filter. LDOS will then attach this filter to the I/O stream for the specified device, and the data associated with that device will pass through the filter.

### PR/FLT

One filter that is provided is the LDOS Printer Filter, PR/FLT. This filter allows you to provide complete forms control to your printer. With this filter the user can set lines per page, length of page, characters per line, indent on wrap around, left margin, tab expansion, and translate a character from/to and more.

### MINIDOS

LDOS also provides a filter called MINIDOS. This filter allows several common DOS commands to be invoked directly from the keyboard, EVEN IN THE MIDDLE OF A RUNNING PROGRAM. MINIDOS will also allow the last DOS command to be repeated with a single key stroke as well as several other features, like sending a Top of Form command to the printer or even killing a file or viewing a disk's directory. All of this is available with single keystroke commands and even from within most running programs.

### KSM

Another filter that is provided is the Key Stroke Multiply filter (KSM). This one allows the user to predefine 26 strings or phrases which can then be invoked with a single keystroke. Words, strings and phrases that are commonly used in your day-to-day operations become very easy to enter. This avoids errors and will increase the speed at which you can provide keyboard input to your machine.

### LIBRARY COMMANDS

Almost all the library commands in the LDOS system have been endowed with the same functions as those in TRSDOS. These functions in most cases form but a subset of the full utility of the comparable LDOS command. To clarify this a bit, let's just say that a user may execute most commands in the identical manner and syntax and expect a similar result from LDOS. However, in most cases the command has had several additional extensions and/or enhancements added that allow additional functions to be performed. For instance, all LDOS users are familiar with the DIRectory command. In LDOS you can do a DIR :1 just like you can in TRSDOS, but the LDOS directory will be sorted in alphabetical order when it is displayed on the screen. Wherever possible, every effort has been made to maintain upward compatibility to TRSDOS command syntax in order to make transition by existing TRSDOS users painless, if not pleasant.

Here is a complete list of the LDOS library commands. Most of these commands have many parameters associated with them to modify and control their actions. It should also be noted that there are TWO libraries ⟨A⟩ and ⟨B⟩. This is so the user may remove (KILL) one of these library modules from a system type disk to gain back the space when needed and still have access to a selected library. Further discussions on the unique features of some of the LDOS library commands will be provided in future articles.

| | | | |
|---|---|---|---|
| Library ⟨A⟩ (SYS6) | | | |
| Append | Copy | Device | Dir |
| Do | Filter | Kill | Lib |
| Link | List | Load | Memory |
| Rename | Reset | Route | Run |
| Set | Spool | | |
| Library ⟨B⟩ (SYS7) | | | |
| Attrib | Auto | Boot | Build |
| Clock | Create | Date | Debug |
| Dump | Free | Purge | System |
| Time | Trace | Verify | |

### JOB CONTROL LANGUAGE

LDOS contains a full Job Control Language (JCL) very similar to those EXEC languages used on Minis and Mainframes. JCL allows the user to write a computer controlling job stream. JCL is much more than the simple "DO" type command systems that are available in other microcomputer operating systems. LDOS JCL allows the construction of very sophisticated job streams with conditionals (if, then, else), tokens (variables), macros (keywords), user inputs, flashing prompts, sound (if you are set up for sound) and even special start points in the JCL file. JCL can be used in a very simple manner such as going into a language and then running a specified program automatically. As you become familiar with JCL, it could do things for you like cause all the needed commands to be invoked to assemble, link together and create an optimized master LDOS disk, all without user intervention. (Note that when we at LSI assemble LDOS and its utilities in this manner it takes several hours but it would take much longer and be subject to much human error were it to be done without JCL.)

### JOBLOG

Most large computers also have a function they call the JOBLOG. With this feature invoked, a record of all commands issued to the DOS and any error reports are recorded to either a disk file or a device, most likely your printer. Each entry to this JOBLOG contains a "time stamp" to show when the action occurred. Of course, the reason for mentioning this high level feature of large systems is that it is fully implemented (as described above) in the LDOS operating system.

### PRINTER SPOOLER

A printer SPOOLER that makes use of both RAM and DISK is included with LDOS. The spooler is very simple to invoke and use, and in most cases will greatly increase the through-put of programs that produce large amounts of printed output. In effect, what a printer spooler does is allow the program to output to the printer even when the printer is still busy printing some previous line. Because the printer is one of the slowest devices (relatively speaking) in a computer system, it is not unusual for the rest of your system, and your program, to be idle while waiting for the printer. The LDOS print spooler helps correct this time wasting situation.

### SPEED

In almost all cases, the overall speed of LDOS is slightly greater than that of TRSDOS and in some cases LDOS is much faster than TRSDOS. This can make certain disk oriented programs function faster, and therefore, more pleasant to use.

### FILE MAINTENANCE

The file maintenance done by LDOS is a bit different than that of TRSDOS. All files have information in their directory entry to indicate whether or not the file has been backed up since it was last written to, and the date that the file was last written to. This information can be used in conjunction with BACKUP and PURGE, and is viewable with the DIR command.

### BACKUP

Speaking of the backup command, I should mention that this command is very much enhanced. The LDOS backup will do a mirror image backup like TRSDOS, but it will also do much more. You can "backup by class" with LDOS. This could be done by specifying to the BACKUP command a partial file specification (partspec). This could be as simple as BACKUP /CMD:0 TO :1, this command will move all the files with the extension "CMD" from drive :0 to drive :1. Backup also allows you to use wildcard characters in the partspec. You can also do controlled backups based on the MOD flag. This would backup only the files that had been written to since they were last backed up. Controlled backups are also available by date or range of dates, by file type, or by their presence or absence on the destination disk. Maintenance of your backups and archive diskettes is simple and very efficient in the LDOS system.

### DOCUMENTATION

The documentation provided with LDOS consists of a large three ring binder of about 350 pages, plus an easy to use "User's Guide". It is all logically organized and written in plain English with examples for every function. There is even a glossary and a handy cross reference index. A complete section is devoted to providing detailed technical information on the structure and the entry points of LDOS, as well as RAM storage assignments and detailed specifications regarding the layout of an LDOS type diskette.

### COMPATIBILITY

One of the main design features of the LDOS environment is COMPATIBILITY!!! Any LDOS Data disk can be read from and written to by either a Model I or Model III version of LDOS. If you have a Model I and a Model III and use the appropriate LDOS on each machine, the data diskettes are completely compatible. Yes, LDOS will read and write single density 35 track disks on the MODEL III and it will also read and write 40 track double density disks on the MODEL I (with the Radio Shack Double Density modification and a 40 track drive). With LDOS you can have data such as SCRIPSIT files, BASIC program files, PROFILE data file, VISICALC data files . . . etc. and use them between models without any type of conversions. LDOS will automatically determine the density and track count of the disk it is accessing and respond accordingly (provided the disk is an LDOS type disk). This powerful feature makes combined Model I and III installations both practical and pleasant.

The only problem with cross compatibility is with some of the programs themselves. It must be noted that programs written to be machine dependent will not be cross compatible between models. Should you have any question about the direct cross compatibility of a program or about its compatibility with LDOS, contact the author or the publisher of the program involved. They should be able to provide you with answers.

Please note that while most Radio Shack programs will run on LDOS without change or with only slight modification, they are only supported by Radio Shack on the operating system that Radio Shack released them on (i.e., TRSDOS 2.3 for Model I and TRSDOS 1.3 for Model III.) Many Radio Shack packages are being checked for compatibility with LDOS and the information will be included with the product. Please note that we will NOT be attempting to correct, modify, patch, or in any other way make non-Radio Shack programs or hardware work with LDOS or the Hard Drive system. This would be an impossible undertaking. For non-Radio Shack software, you will have to contact the author or publisher of the product.

One more important point should be made before you run down to your local Radio Shack and order yourself a copy of LDOS. You should be aware that the authors of LDOS and Radio Shack strongly recommend that LDOS not be purchased by single drive owners. LDOS will function with a single drive, but some LDOS functions will be very cumbersome, and certain high level functions will not function at all with a single drive.

### DO YOU NEED LDOS?

All this may sound great, but one major question must be addressed. Do you need to have LDOS? The answer is, in many cases, a simple "No, you don't." If the major use of your computer is with functional applications that you are comfortable and pleased with, then chances are LDOS would be of little value to you.

I don't want you to think that I'm saying most users should not buy LDOS. LDOS is a fine operating system that can do many things that TRSDOS cannot. But you must justify some use for these additional features, or LDOS will be of little value other than the increase in the speed of disk activity. The price of LDOS is $129 and this cost should be weighed against the additional features that you may use. Also, if you are just beginning your use of the TRS-80 you will find TRSDOS to be sufficient in most cases. In all instances it is suggested that you become thoroughly familar with the use of TRSDOS before "moving up" to the additional capabilities of LDOS.

If you are totally familiar with TRSDOS and/or you actively write software for your own use or for commercial purposes, you will find many new and intriguing possibilities with the LDOS system and should, therefore, seriously consider obtaining LDOS.

In the months to come this column will contain hints and tips on the use of LDOS and its special features, which are aimed at the general user and the BASIC programmer, as well as technical information on the inner workings of LDOS and interfacing machine language programs to LDOS in the proper manner.

# Printing Titles

**Bill Christ**
**3514 Balfour Street**
**Saskaktoon, Saskatchewan**
**CANADA S7H 3Z6**

This is a little routine which will center a title of your choice and underline it.

```
10 INPUT "ENTER YOUR TITLE HERE"; A$
20 L=LEN(A$)
    : T=35- INT(.5*L)
30 LL=6*L*2
35 LPRINT CHR$(31)
40 LPRINT TAB(T/2); A$;
    : LPRINT CHR$(30); CHR$(26)
45 FOR X=1 TO T:LPRINT " ";
    : NEXT
50 LPRINT CHR$(18);
60 LPRINT CHR$(28); CHR$(LL); CHR$(129)
65 LPRINT CHR$(30)
70 END
```

# A Pascal Division Program

**James W. Wood**
**424 N. Missouri**
**Atwood, IL 61913**

Do you ever wonder what numbers don't show on your calculator or computer when dividing? My calculator display reads .07692308 when I divide one by thirteen. I know from Algebra that if the answer to this problem has a repeating block of numbers that it is no longer than twelve digits long (one less than the divisor).

I decided to write a computer program to calculate the answer "exactly." Since I want to increase my knowledge of Pascal, I decided this was a good time to become bilingual. The Pascal program works fine unless you try to divide numbers which exceed Tiny Pascal's integer range. On a repeating decimal press ⟨ENTER⟩ once to pause, twice to terminate program.

```
VAR A,B,C,D,E,F:INTEGER;
BEGIN
  WRITE(28,31,'ENTER YOUR DIVIDEND');
  READ(A#);
  WRITE(13,13,'ENTER YOUR DIVISOR');
  READ(B#);
  C:=A DIV B;
  WRITE(13,13,C#,'.');
  D:=(A MOD B)*10;
  REPEAT
    E:=D DIV B;
    WRITE(E#);
    F:=D-(E*B);
    D:=F*10;
  UNTIL D=0;
END.
```

# Model I/III Bugs, Errors, Fixes

## NOTE TO USERS:

The following changes and corrections are provided for your information. If you have an applications program which is working correctly, you should probably NOT make any changes to it. If you feel that changes should be made, but you do not feel qualified to make the change yourself, please contact your local Radio Shack Computer Center or Expanded Computer Department for assistance. If you do not have access to one of these stores, then you may want to call Computer Customer Service in Fort Worth for assistance.

## MEDICAL OFFICE SYSTEM (26-1568)

The program does not print zero balance accounts when option 2 of statement printing is selected. Make the following changes to the program BLLPRT.

```
1100 PRINT@728,"SELECT OPTION: ";:FL=-1:GOSUB210
     :Q=VAL(IN$):PRINT:IFQ<1ORQ>3THEN1100ELSE
     PF=Q=1:X0=Q<>2:IFQ=1THENPOKE16424,43ELSEPOKE
     16424,67
1130 IFA=AHTHEN1140ELSEIFX0AND(BA=0ORI$="N")THEN
     1210ELSEGOSUB1215:IFPFTHENLPRINTLF$:LPRINT
     LF$:LPRINTLF$:IFBA<0THELPRINTTAB(40)"CREDIT
     BALANCE"ELSELPRINTTAB(42)"PLEASE REMIT"
1140 IFX0AND(BA=0ORI$="N")THENGOSUB1250:GOTO1210
     ELSEGOSUB1215
```

## MODEL III DISK COURSE (26-2014)

In Lesson 1 Part 2 when running the section of the course that says:

A BASIC program is saved with the filename PAYROLL/BAS.RT33N:0

Which of the following commands will load and run this program?

1. RUN"PAYROLL/BAS.RT33N"
2. RUN"PAYROLL.RT33N:0"
3. RUN"PAYROLL/BAS"
4. LOAD"PAYROLL/BASD.RT33N:0"

(1, 2, 3, OR 4) 4
The following error message will appear:
Syntax Error in 450
READY
450

The first colon (:) in line 450 should be deleted so that the line reads:

```
450 PRINT"No, LOAD"Z$"PAYROLL/BAS.RT33N:0"Z$" is
      incorrect because ":PRINT"'LOAD' will not
      execute the program after it is loaded
      into:PRINT" memory.":GOSUB 7
```

# More TAB and USING in a Total Inventory Program

**Robert E. Perkins**
**2914 Ashley River Road**
**Charleston, SC 29407**

Recent discussion on how to use both the "TAB" and "USING" functions to line up decimal points prompted the following program. It works on a Model I with a LP VI. First look at what an example run of the program might look like.

```
SUMMARY OF OTHER BRANDS
     ARMSTRONG        $     123.55
     SEMPERIT         $  1,256.25
     OTHER            $  1,254.25
     VALVES           $     12.25
          TOTAL OTHER BRANDS                 $   2,646.3Ø
          CARLISLE                           $  12,365.25
          MICHELIN                           $       Ø.ØØ
          GILLETTE                           $  12,354.25
          GOODRICH                           $  32,145.25
          INDUSTRIAL                         $      13.22
          TUBES                              $   1,225.25

          TOTAL INVENTORY                    $ 6Ø,749.52
```

## The Total Inventory Program

```
2 DEFBL A, O, S, G, C, H, Z, Y, I
4 K$="$###,###.##"
5 LPRINT CHR$(31)
1Ø LPRINT TAB(5) "SUMMARY OF INVENTORY LISTING"
15 INPUT "ENTER THE DATE-MM/DD/YY"; A$
2Ø LPRINT TAB(5); A$
3Ø FOR X=1TO6
    : LPRINT CHR$(138)
    : NEXT X
4Ø LPRINT TAB(1Ø) "SUMMARY OF OTHER BRANDS"
5Ø INPUT "ARMSTRONG"; A
    : INPUT "OTHER BRANDS"; O
    : INPUT "SEMPERIT"; S
6Ø INPUT "GILLETTE"; G
    : INPUT "CARLISLE"; C
65 INPUT "GOODRICH"; H
7Ø INPUT "TUBES"; T
75 INPUT "VALUES"; V
76 INPUT "INDUSTRIAL"; I
8Ø INPUT "MICHELIN"; M
9Ø LPRINT TAB(15) "ARMSTRONG"; TAB(3Ø) USING K$; A
1ØØ LPRINT TAB(15) "SEMPERIT"; TAB(3Ø) USING K$; S
11Ø LPRINT TAB(15) "OTHER"; TAB(3Ø) USING K$; O
115 LPRINT TAB(15) "VALVES"; TAB(3Ø) USING K$; V
12Ø Y=A+S+O+V
13Ø LPRINT TAB(2Ø) "TOTAL OTHER BRANDS"; TAB(5Ø)
     USING K$; Y
14Ø LPRINT TAB(2Ø) "CARLISLE"; TAB(5Ø) USING K$; C
15Ø LPRINT TAB(2Ø) "MICHELIN"; TAB(5Ø) USING K$; M
16Ø LPRINT TAB(2Ø) "GILLETTE"; TAB(5Ø) USING K$; G
165 LPRINT TAB(2Ø) "GOODRICH"; TAB(5Ø) USING K$; H
166 LPRINT TAB(2Ø) "INDUSTRIAL"; TAB(5Ø) USING K$; I
17Ø LPRINT TAB(2Ø) "TUBES"; TAB(5Ø) USING K$; I
18Ø LPRINT CHR$(138)
19Ø Z=Y+C+M+G+T+H+I
2ØØ LPRINT TAB(2Ø) "TOTAL INVENTORY"; TAB(5Ø) USING
     K$; Z
```

# Line Printer Set Up

**William Allen**
**2012 Metairie Court, #24**
**Metairie, LA 70002**

```
1Ø ' LINE PRINTER SET UP PROGRAM (TITLE - LPSU)
2Ø CLS
3Ø PRINT "LINE PRINTER SET UP COMMANDS"
4Ø PRINT "FOR QUICK PRINTER II       2"
5Ø PRINT "FOR LINE PRINTER IV        4"
6Ø PRINT "FOR LINE PRINTER VII       7"
7Ø INPUT "<ENTER> WHICH PRINTER ";W
   : CLS
8Ø IF W=2 THEN 2ØØ
9Ø IF W=4 THEN 3ØØ
1ØØ IF W=7 THEN 5ØØ
2ØØ PRINT "FOR QUICK PRINTER II"
21Ø PRINT "CARRIAGE RETURN   1"
22Ø PRINT "DOUBLE WIDTH      2"
23Ø INPUT "<ENTER> CHOICE ";C
24Ø IF C=1 THEN 118Ø
25Ø IF C=2 THEN 114Ø
3ØØ PRINT "FOR LINE PRINTER IV"
31Ø PRINT "DOUBLE WIDTH       1"
32Ø PRINT "PROPORTIONAL       2"
33Ø PRINT "STANDARD PRINT     3"
34Ø PRINT "CONDENSED PRINT    4"
35Ø PRINT "START UNDERLINE    5"
36Ø PRINT "STOP UNDERLINE     6"
37Ø INPUT "<ENTER> CHOICE ";C
38Ø IF C=1 THEN 1ØØØ
39Ø IF C=2 THEN 1Ø2Ø
4ØØ IF C=3 THEN 1Ø4Ø
41Ø IF C=4 THEN 1Ø6Ø
42Ø IF C=5 THEN 114Ø
43Ø IF C=6 THEN 116Ø
5ØØ PRINT "FOR LINE PRINTER VII"
51Ø PRINT "CARRIAGE RETURN/LINE FEED              1"
52Ø PRINT "GRAPHIC MODE                           2"
53Ø PRINT "STANDARD PRINT                         3"
54Ø PRINT "DOUBLE WIDTH                           4"
55Ø INPUT "<ENTER> CHOICE ";C
56Ø IF C=1 THEN 116Ø
57Ø IF C=2 THEN 112Ø
58Ø IF C=3 THEN 11ØØ
59Ø IF C=4 THEN 1Ø8Ø
1ØØØ LPRINT CHR$(27); CHR$(14)
1Ø1Ø NEW
1Ø2Ø LPRINT CHR$(27); CHR$(17)
1Ø3Ø NEW
1Ø4Ø LPRINT CHR$(27); CHR$(19)
1Ø5Ø NEW
1Ø6Ø LPRINT CHR$(27); CHR$(2Ø)
1Ø7Ø NEW
1Ø8Ø LPRINT CHR$(31)
1Ø9Ø NEW
11ØØ LPRINT CHR$(3Ø)
111Ø NEW
112Ø LPRINT CHR$(18)
113Ø NEW
114Ø LPRINT CHR$(15)
115Ø NEW
116Ø LPRINT CHR$(14)
117Ø NEW
118Ø LPRINT CHR$(13)
119Ø NEW
```

# Changing Print Size on the LP V and VI

**Robert P. Graham**
**2709 N. Sibley Street**
**Metairie, LA 70003**

Here are four short machine language programs that can be run on the Model II to change the print size of the Line Printer V and VI. They reside above user RAM, which means they can be run from TRSDOS READY or "called" from BASIC (using the SYSTEM " " command) or from VisiCalc (using /SE). We find them particularly useful in VisiCalc where it is often necessary to print more than 132 columns on a line. With SET15, you can print 198 columns, with SET75 or SET5, you can print anything in large size type, and with SET10, you can return the printer to normal operation.

You can execute these while the printer is anywhere on the page. The only restriction is that they must be executed while the printer is at the beginning of a line (no problem in VisiCalc). This is because the printer initialization function sets the print head to the beginning of the current line, and anything already printed on that line will be printed over. There is nothing copyrighted here, as it is just a series of Model II supervisor calls.

Program lines 110-130 get the current printer status. PUSH HL saves the current line count. Lines 150-190 initialize the printer, setting a different number of columns for each program. The next five lines send a series of codes to the printer, setting the print size. The current line count is then POP-ed off the stack. Finally, a supervisor call restores the line count to TRSDOS. This is necessary because the printer initialization sets it to zero.

The assembly language and the hex code is included, so if you do not have an assembler, you can enter the hex code in DEBUG beginning at address F380. To store the programs, return to TRSDOS after entering the code, and use the DUMP command (DUMP SET15 [START=F380, END=F3A7]). Use a different file name for each program. The beginning and ending addresses shown above can be used for all four.

I hope these can be as useful to you as they have been to us.

**SET5** (For printing in large size type)

```
F38Ø           ØØ1ØØ             ORG     ØF38ØH
F38Ø  3E5F     ØØ11Ø  START      LD      A,95
F382  Ø6ØØ     ØØ12Ø             LD      B,Ø
F384  CF       ØØ13Ø             RST     8
F385  E5       ØØ14Ø             PUSH    HL
F386  3E11     ØØ15Ø             LD      A,17
F388  Ø642     ØØ16Ø             LD      B,66
F38A  ØE3C     ØØ17Ø             LD      C,6Ø
F38C  1642     ØØ18Ø             LD      D,66
F38E  CF       ØØ19Ø             RST     8
F38F  3E13     ØØ2ØØ             LD      A,19
F391  21A1F3   ØØ21Ø             LD      HL,MSG1
F394  Ø6Ø4     ØØ22Ø             LD      B,MSGL1
F396  ØEØØ     ØØ23Ø             LD      C,ØØ
F398  CF       ØØ24Ø             RST     8
F399  E1       ØØ25Ø             POP     HL
F39A  3E5F     ØØ26Ø             LD      A,95
F39C  4D       ØØ27Ø             LD      C,L
F39D  Ø6Ø3     ØØ28Ø             LD      B,3
F39F  CF       ØØ29Ø             RST     8
F3AØ  C9       ØØ3ØØ             RET
F3A1  1B       ØØ31Ø  MSG1       DEFB    27
F3A2  ØF       ØØ32Ø             DEFB    15
F3A3  1E       ØØ33Ø             DEFB    3Ø
F3A4  1F       ØØ34Ø             DEFB    31
F3A5  ØØ       ØØ35Ø  MSG2       DEFB    ØØ
ØØØ4           ØØ36Ø  MSGL1      EQU     MSG2-MSG1
F38Ø           ØØ37Ø             END     START
ØØØØØ  Total Errors

MSG2    F3A5
MSGL1   ØØØ4
MSG1    F3A1
START   F38Ø
```

**SET75** (To print with large size type)

```
F38Ø           ØØ1ØØ             ORG     ØF38ØH
F38Ø  3E5F     ØØ11Ø  START      LD      A,95
F382  Ø6ØØ     ØØ12Ø             LD      B,Ø
F384  CF       ØØ13Ø             RST     8
F385  E5       ØØ14Ø             PUSH    HL
F386  3E11     ØØ15Ø             LD      A,17
F388  Ø642     ØØ16Ø             LD      B,66
F38A  ØE3C     ØØ17Ø             LD      C,6Ø
F38C  1662     ØØ18Ø             LD      D,98
F38E  CF       ØØ19Ø             RST     8
F38F  3E13     ØØ2ØØ             LD      A,19
F391  21A1F3   ØØ21Ø             LD      HL,MSG1
F394  Ø6Ø6     ØØ22Ø             LD      B,MSGL1
F396  ØEØØ     ØØ23Ø             LD      C,ØØ
F398  CF       ØØ24Ø             RST     8
F399  E1       ØØ25Ø             POP     HL
F39A  3E5F     ØØ26Ø             LD      A,95
F39C  4D       ØØ27Ø             LD      C,L
F39D  Ø6Ø3     ØØ28Ø             LD      B,3
F39F  CF       ØØ29Ø             RST     8
F3AØ  C9       ØØ3ØØ             RET
F3A1  1B       ØØ31Ø  MSG1       DEFB    27
F3A2  ØF       ØØ32Ø             DEFB    15
F3A3  1E       ØØ33Ø             DEFB    3Ø
F3A4  1F       ØØ34Ø             DEFB    31
F3A5  1B       ØØ35Ø             DEFB    27
F3A6  ØE       ØØ36Ø             DEFB    14
F3A7  ØØ       ØØ37Ø  MSG2       DEFB    ØØ
ØØØ6           ØØ38Ø  MSGL1      EQU     MSG2-MSG1
F38Ø           ØØ39Ø             END     START
ØØØØØ Total Errors

MSG2    F3A7
MSGL1   ØØØ6
MSG1    F3A1
START   F38Ø
```

**SET10** (Returns printer to normal operation)

```
F38Ø           ØØ1ØØ             ORG     ØF38ØH
F38Ø  3E5F     ØØ11Ø  START      LD      A,95
F382  Ø6ØØ     ØØ12Ø             LD      B,Ø
F384  CF       ØØ13Ø             RST     8
F385  E5       ØØ14Ø             PUSH    HL
F386  3E11     ØØ15Ø             LD      A,17
F388  Ø642     ØØ16Ø             LD      B,66
F38A  ØE3C     ØØ17Ø             LD      C,6Ø
F38C  1684     ØØ18Ø             LD      D,132
F38E  CF       ØØ19Ø             RST     8
F38F  3E13     ØØ2ØØ             LD      A,19
F391  21A1F3   ØØ21Ø             LD      HL,MSG1
F394  Ø6Ø3     ØØ22Ø             LD      B,MSGL1
F396  ØEØØ     ØØ23Ø             LD      C,ØØ
F398  CF       ØØ24Ø             RST     8
F399  E1       ØØ25Ø             POP     HL
F39A  3E5F     ØØ26Ø             LD      A,95
F39C  4D       ØØ27Ø             LD      C,L
F39D  Ø6Ø3     ØØ28Ø             LD      B,3
F39F  CF       ØØ29Ø             RST     8
F3AØ  C9       ØØ3ØØ             RET
F3A1  1B       ØØ31Ø  MSG1       DEFB    27
```

```
F3A2  ØF          ØØ32Ø          DEFB    15
F3A3  1E          ØØ33Ø          DEFB    3Ø
F3A4  ØØ          ØØ34Ø  MSG2    DEFB    ØØ
ØØØ3              ØØ35Ø  MSGL1   EQU     MSG2-MSG1
F38Ø              ØØ36Ø          END     START
ØØØØØ Total Errors

MSG2    F3A4
MSGL1   ØØØ3
MSG1    F3A1
START   F38Ø
```

### **SET15** (To print 198 columns)

```
F38Ø              ØØ1ØØ          ORG     ØF38ØH
F38Ø  3E5F        ØØ11Ø  START   LD      A,95
F382  Ø6ØØ        ØØ12Ø          LD      B,Ø
F384  CF          ØØ13Ø          RST     8
F385  E5          ØØ14Ø          PUSH    HL
F386  3E11        ØØ15Ø          LD      A,17
F388  Ø642        ØØ16Ø          LD      B,66
F38A  ØE3C        ØØ17Ø          LD      C,6Ø
F38C  16C6        ØØ18Ø          LD      D,198
F38E  CF          ØØ19Ø          RST     8
F38F  3E13        ØØ2ØØ          LD      A,19
F391  21A1F3      ØØ21Ø          LD      HL,MSG1
F394  Ø6Ø5        ØØ22Ø          LD      B,MSGL1
F396  ØEØØ        ØØ23Ø          LD      C,ØØ
F398  CF          ØØ24Ø          RST     8
F399  E1          ØØ25Ø          POP     HL
F39A  3E5F        ØØ26Ø          LD      A,95
F39C  4D          ØØ27Ø          LD      C,L
F39D  Ø6Ø3        ØØ28Ø          LD      B,3
F39F  CF          ØØ29Ø          RST     8
F3AØ  C9          ØØ3ØØ          RET
F3A1  1B          ØØ31Ø  MSG1    DEFB    27
F3A2  ØF          ØØ32Ø          DEFB    15
F3A3  1E          ØØ33Ø          DEFB    3Ø
F3A4  1B          ØØ34Ø          DEFB    27
F3A5  ØE          ØØ35Ø          DEFB    14
F3A6  ØØ          ØØ36Ø  MSG2    DEFB    ØØ
ØØØ5              ØØ37Ø  MSGL1   EQU     MSG2-MSG1
F38Ø              ØØ38Ø          END     START
ØØØØØ Total Errors

MSG2    F3A6
MSGL1   ØØØ5
MSG1    F3A1
START   F38Ø
```

# Descenders on the LP VII

**Joe Diel**
**337 Rhode Island NE**
**Albuquerque, NM 87108**

This program uses the graphics capability of the Line Printer VII to print descenders or lower case g, j, p, q, and y. All other symbols are standard line printer symbols. The program was written for Model III BASIC, 16K RAM. Input is in response to a prompt; to exit use the ⟨BREAK⟩ key.

```
1Ø REM TRS-8Ø MODEL III LINE PRINTER VII
2Ø REM PROGRAM TO PRINT LOWER CASE WITH DESCENDERS
3Ø REM J.DIEL, ALBUQUERQUE, NM, DECEMBER 1981
35 CLEAR 1ØØØ
4Ø DEFINT I-N
    : DIM L(13,4),IP(2Ø),ID(2Ø),B$(4)
45 REM SET UP PATTERNS FOR LOWER CASE G, J, P, Q, Y
5Ø DATA 18, 184, 196, 196, 168, 252, 3Ø
6Ø DATA 18, 128, 132, 132, 132, 131, 3Ø
7Ø DATA 18, 128, 128, 128, 128, 253, 3Ø
8Ø DATA 18, 128, 132, 132, 132, 131, 3Ø
9Ø DATA 18, 252, 168, 196, 196, 184, 3Ø
1ØØ DATA 18, 135, 128, 128, 128, 128, 3Ø
11Ø DATA 18, 184, 196, 196, 168, 252, 3Ø
12Ø DATA 18, 128, 128, 128, 128, 135, 3Ø
13Ø DATA 18, 188, 192, 192, 16Ø, 252, 3Ø
14Ø DATA 18, 128, 132, 132, 132, 131, 3Ø
15Ø FOR I=Ø TO 4
16Ø FOR J=Ø TO 13
17Ø READ L(J,I)
18Ø NEXT J
19Ø NEXT I
2ØØ REM SET UP ARRAY B$ FOR TESTING FOR LOWER CASE
     G,J,P,Q, OR Y
21Ø DATA "g", "j", "p", "q", "y"
22Ø FOR I = Ø TO 4
     : READ B$(I)
     : NEXT I
3ØØ REM INPUT LINE TO BE PRINTED
31Ø REM FOR MORE COMPLEX LPRINT PROGRAMS MAIN PROGRAM
32Ø REM SHOULD BE BETWEEN 3ØØ AND 5999
33Ø PRINT "INPUT LINE(S) TO BE PRINTED"
34Ø INPUT A$
     : GOSUB 6ØØØ
     : GOTO 34Ø
6ØØØ REM SUBROUTINE TO FIND LOWER CASE G,J,P,Q, OR Y
     IN A$
6Ø1Ø REM AND PRINT A$ TO LINE PRINTER VII
6Ø2Ø NP=Ø
6Ø3Ø FOR I=1 TO LEN(A$)
6Ø4Ø FOR J=Ø TO 4
6Ø5Ø IF MID$(A$,I,1)=B$(J) THEN 6Ø8Ø
6Ø6Ø NEXT J
6Ø7Ø GOTO 6Ø9Ø
6Ø8Ø NP=NP+1
     : IP(NP)=I
     : ID(NP)=J
6Ø9Ø NEXT I
61ØØ IF NP=Ø THEN 629Ø
611Ø IF IP(NP(1)) <> 1 THEN LPRINT
     MID$(A$,1,IP(1)-1);
612Ø FOR J=Ø TO 6
     : LPRINT CHR$(L(J,ID(1)));
     : NEXT J
613Ø LPRINT CHR$(18); CHR$(128); CHR$(3Ø);
614Ø IF NP=1 THEN 6185
615Ø FOR I=2 TO NP
616Ø LPRINT MID$(A$,IP(I-1)+1,IP(I)-IP(I-1)-1);
617Ø FOR J=Ø TO 6
     : LPRINT CHR$(L(J,ID(I)));
     : NEXT J
618Ø NEXT I
6185 IF LEN(A$)>IP(NP) THEN LPRINT MID$(A$,IP(NP)+1);
619Ø LPRINT CHR$(18); CHR$(1Ø); CHR$(3Ø);
62ØØ IF IP(NP(1)) <> 1 THEN LPRINT STRING$(IP(1)-1,
     "  ");
621Ø FOR J=7 TO 13
     : LPRINT CHR$(L(J,ID(1)));
     : NEXT J
6215 LPRINT CHR$(18); CHR$(128); CHR$(3Ø);
622Ø IF NP=1 THEN 627Ø
623Ø FOR I=2 TO NP
624Ø LPRINT STRING$(IP(I)-IP(I-1)-1," ");
625Ø FOR J=7 TO 13
     : LPRINT CHR$(L(J,ID(I)));
     : NEXT J
626Ø NEXT I
627Ø LPRINT CHR$(18); CHR$(1Ø); CHR$(3Ø);
628Ø RETURN
629Ø LPRINT A$;
     : FOR I=1 TO 2
     : LPRINT CHR$(18); CHR$(1Ø); CHR$(3Ø);
     : NEXT I
     : RETURN
```

# Model II Scripsit 2.0, LP VIII and the New Printers

Bruce Elliott

As stated in the Scripsit manual, Scripsit 2.0 was designed to work with a number of different printers. In this article we will suggest several techniques which will help optimize the use of Scripsit 2.0 and Radio Shack's LP VIII, the new DMP series printers and the new DWP-410. This information should also provide you with a guide for using other dot matrix printers. Check your printer manual for specific information on how your printer works.

## START-UP

Page 74 of the Scripsit 2.0 manual tells how to initialize a printer automatically during startup. This is especially valuable for a printer like the Line Printer VIII. The Line Printer VIII powers up in data processing mode, and you will want it to be in the word processing mode for use with Scripsit 2.0. Page 24 of the LP VIII manual tells you what control codes are available in the LP VIII.

The Scripsit manual goes on to show a sample startup procedure for an LP VIII:

## STARTUP 141B0E1B14

This code sequence can be deciphered by checking page 24 of the LP VIII manual:

14—Select Word Processing Mode
1B0E—Start Character Elongation
1B14—Select the Condensed Character Set

This startup method works very well, but you should remember one thing: Don't RESET the printer! Resetting the printer will reset the standard character set and data processing modes, not at all what you want.

Radio Shack's new DMP (Dot Matrix Printer) series allows you to "custom-set" the power-up mode and pitch, so you should not need the special start-up capabilities of Scripsit 2.0.

With any printer which offers changeable pitch and font, you may find yourself wanting to use these features in the documents you create.

You can do this by using Scripsit's printer codes. I use "I" (for Initialize) to allow resetting my LP VIII if needed. Here is how I set up the "I" key: Using the above sequence, I see that there are 5 pairs of hexadecimal (hex) values—14 1B 0E 1B and 14. In the Scripsit Printer Control Code Utility, I select "I", then enter a 5 for the number of ASCII characters, then 0 for the Justification Value.(Since these are printer control codes, not printable characters, I don't want them counted when Scripsit tries to justify a line.) Next, I enter the 5 pairs of values. Finally I press ⟨ENTER⟩ and enter a short description "Init 8.3 CPI".

Now, anytime I need to initialize my LP VIII to 8.3 cpi (Characters Per Inch) I simply place a ⟨CTRL⟩⟨X⟩⟨I⟩ at the beginning of the text to be printed.

## UNDERLINING AND BOLDFACE

Two features of Scripsit 2.0 which enhance the look of a document are underlining and boldface. If you are using one of the printers we are discussing, and you type in a line of text containing Scripsit's underline code, you may not be overly pleased by the result. It will be obvious that something is not quite right.

If you watch the printer closely, you will notice that while underlining (or trying to), the printer prints a character, backs up (the wrong distance), prints an underscore, prints the next letter, and continues until the text to be underlined has been printed.

The problem is that Scripsit comes set-up for the DW II printer. The DW II accepts a single hex character (08) as the command for backspacing one full character. Printers like the LP VIII, DMP series and the DWP-410 use a two character hex sequence to indicate the distance to move the printhead back.

Page 74 of the Scripsit manual gives us information about changing the "distance for backspace." Scripsit is designed to be patched for printers which use the two hex digit backspace.

When you patch Scripsit 2.0's backspace distance, we recommend that you put in the value for your most commonly used character size. The reason for this is that you can only insert a single value, and you can not change that value during the course of printing a document.

In Scripsit 2.0 there are two basic functions which use the patched backspace distance: underlining and bold face.

The DMP series of printers and the DWP-410 have built-in underline and boldface capabilities (LP VIII has built-in underline, but not boldface.) This means that you can avoid having to patch Scripsit, and use the capabilities of your printer to the fullest.

For the LP VIII, a quick glance at the manual shows that the printer uses hex 0F to turn underline on, and hex 0E to turn underline off. I have set these codes up as "U" to turn underline on, and "O" to turn underline off.

We can now underline any phrase, in any of the LP VIII type sizes, by using the codes ⟨CTRL⟩⟨X⟩⟨U⟩ and ⟨CTRL⟩⟨X⟩⟨O⟩.

To use boldface on normal LP VIII characters, we make the following patch to Scripsit 2.0:

PATCH SCRIPSIT A=DDFC F=00 C=0C

The "0C" in the C= string is hex for 12, which is the number of dot columns to backspace on the LP VIII. I also recommend reducing the number of strikes for bold characters from the preset five to three:

PATCH SCRIPSIT A=DDFD F=05 C=03

This change produces a slightly thinner bold character, which I prefer. You may wish to experiment with several values to see what you like best.

For the DMP and DWP-410 printers, the printer manual contains a chart which gives you the proper codes to turn underline or boldface on or off. The manuals also include information of the dot column widths of characters for patching Scripsit.

### CHANGING CHARACTER SIZE

One of the reasons for having a dot matrix printer like the ones we are discussing is that the printer is capable of printing more than one size character. Page 34 of the LP VIII manual shows six different combinations of character font and size.

LP VIII is capable of printing three basic character fonts: Proportional, Condensed and Ordinary. Since Model II Scripsit does not provide justification with proportional characters, I do not use this font. The two remaining fonts, when used normal size or elongated, provide me with four character sizes that I can use in my documents.

To be able to use all of these character sizes, I need some additional printer control codes. Again checking the LP VIII manual, I find the following codes that look useful:

1B 13—Select Normal Character Set
1B 14—Select Condensed Character Set
1B 0E—Start Elongation
1B 0F—Stop Elongation

I have set these up as printer codes "N", "C", "E" and "P" respectively. You need to remember that elongation will remain on until you turn it off. Use care when you go from one character size to another, and turn elongation off when you don't want it.

Here is how Scripsit 2.0 is set up for my LP VIII, and my printer control codes. As you can see from the chart, I have defined print codes for several special characters as well as the control codes we have discussed.

### SCRIPSIT PRINTER CONTROL CODES

```
xA  1 1 AB          Copyright       xN  2 0 1B13      Normal Char
xB  1 1 AB          Dagger          xO  1 0 0E        Undrline Off
xC  2 0 1B14        Condensed Ch    xP  2 0 1B0F      Elongate Off
xD  1 1 A6          Degree Symb     xQ  1 1 AC        1/4
xE  2 0 1B0E        Start Elongt    xR  3 3 285229    (R)
xF  1 1 BF          script f        xS  1 1 B4        cents
xG  0 0                             xT  4 4 28544D29  (TM)
xH  1 1 AE          1/2             xU  1 0 0F        Undrline On
xI  5 0 141B0E1B14  Init 8 3 cpi    xV  0 0
xJ  0 0                             xW  0 0
xK  0 0                             xX  0 0
xL  0 0                             xY  0 0
xM  0 0                             xZ  0 0
```

Which control code would you like to change?

Enter A—Z, ESC to store, BREAK to cancel, or HOLD to restore previous:

### SUBSCRIPTS AND SUPERSCRIPTS

Once initialized to the word processing mode, (regardless of character size), the printers we have been discussing will handle subscripts and superscripts correctly. No special codes needed other than the normal Scripsit codes.

One more comment about Scripsit and Dot Matrix Printers. If you use character justification, speed of the printer is reduced dramatically. In order to place the proper number of micro-spaces between characters, the printer must stop its printhead, back up, regain its forward speed, locate the proper dot column, skip it, and then print the next character. This process takes a lot of time, and the speed of the printer suffers.

# Navigation with a CC

**Leo McNeil**
**6665 Koralee**
**Tucson, AZ 85710**

Earlier this year I purchased my Extended BASIC Color Computer with the intent of programming Navigation problems, thus eliminating the tedium of cross-referencing one list from one table with another.

Every thing worked out fine until the bottom line: The answer, by formula, was given as an argument of COS(M). No built in ARCCOS!

I resolved the problem through the laws of Sines, Cosines and Tangents. The formula for Great Circle Sailing:

$\text{Cos}(c) = \text{Cos}(a)\text{Cos}(b) + \text{Sin}(a)\text{Sin}(b)\text{Cos}C$

was resolved thus:

$\text{Sin}^2(c) + \text{Cos}^2(c) = 1$

$\text{Sin}^2(c) = 1 - \text{Cos}^2(c)$

$\text{Sin}(c) = \text{SQR } 1 - \text{Cos}^2(c)$

$\text{Tan}(c) = \text{Sin}(c)/\text{Cos}(c)$

Where M in Line 300 = Cos(c), CC in Line 310 converts M to Tan(c).

```
10 CLS
15 CLEAR 300
20 'A PROGRAM TO COMPUTE - WITH EXTENDED COLOR
     COMPUTER - GREAT CIRCLE DISTANCES
21 'CASE I :LAT/LONG SAME NAMES
22 'CASE II:LATS SAME NAMES;LONGS DIFFERENT NAMES
23 'CASE III:LAT DIFFERENT NAME;LONGS SAME NAMES
24 'CASE IV:LATS/LONGS DIFFERENT NAMES
25 'K IS THE RAD/DEG/RAD CONVERSION FACTOR
30 K=.0174532933
35 'KM IS THE MILES/MS/MILES CONVERSION FACTOR
40 KM=.6213699
45 D$="#####.## %    %"
100 'PD=POINT(LAT/LONG) OF DEPARTURE
105 'PA=POINT(LAT/LONG) OF ARRIVAL
150 'INPUT LARGEST LATITUDE FIRST
155 INPUT "LAT:DG, MN, N OR S"; DA, MA, A$
160 INPUT "LONG:DG, MN, E OR W"; AD, AM, AA$
165 INPUT "LAT:DG, MN, N OR S"; DB, MB, B$
170 INPUT "LONG:DG, MN, E OR W"; BD, BM, BB$
175 INPUT "PORT OF DEPARTURE"; DP$
176 INPUT "PORT OF DESTINATION"; DS$
200 A=DA+MA/60
205 AL=AD+AM/60
210 B=DB+MB/60
215 BL=BD+BM/60
220 A1=(90-1)*K
225 B1=(90-B)*K
230 IF AL<BL THEN 240 ELSE 235
235 C=(BL-AL)*K
     : GOTO 300
240 C=(AL-BL)*K
     : GOTO 300
300 M=((COS(A1)*COS(B1))+(SIN(A1)*SIN(B1)*COS(C)))
305 'CC IS COS(M) EXPRESS AS ARCTAN(M)
310 CC=ATN((SQR(A-(M^2)))/(M))/K
315 'DM IS THE DISTANCES IN MILES:DK IN KMS
320 DM=CC*60
325 DK=DM*KM
400 PRINT#-2, "DISTANCE FROM "; DP$;
401 PRINT#-2, " TO  "; DS$"  IS "
     : PRINT#-2
405 PRINT#-2, USING D$; DM, "MILES"
     : PRINT#-2
410 PRINT#-2, USING D$, DK, "KMS"
     : PRINT#-2
     : PRINT#-2
415 PRINT#-2, "L. MCNEIL, CDR USN (RET)"
500 END
```

# Word Processing on the Pocket Computer

**Don Hergert**
**2300 Fremontia Dr. #106**
**San Bernardino, CA 92404**

*Editor's Note: We are very pleased to announce that an excellent unpublished manuscript containing programs and text for the pocket computer (PC-1) was offered to us by Don Hergert. And of course, we accepted. In the coming months we will be publishing several segments from the manuscript.*

## INTRODUCTION

The Pocket Computer represents perhaps one of the greatest challenges available to the experienced BASIC programmer: to produce valuable working programs with practicality, for an extremely limited computer. As much as I like the Pocket Computer, the facts of life in the personal computer market say that any computer with under 64K bytes of RAM, or without the capacity to upgrade to Disk Operating Systems, or without the other features available in many of today's computers, is limited. If you've never done any programming on a larger computer, you probably don't understand the comments made by those who have. If they, in turn, haven't done programming on the Pocket Computer, they probably cannot fathom working on something so small.

Undoubtedly the biggest hurdle to overcome in working with the Pocket Computer is not tied to the limitations of the hardware. The biggest hurdle is a limited imagination. We all have it. For you it might be impossible to imagine this "alphanumeric calculator" being able to play a game. For me, imagining the Pocket Computer doing an inventory is difficult. It's all possible, though. We may have to sacrifice in some applications, but we'll reach our major goals if our imagination will let us.

The Pocket Computer has only 1.9K bytes of Random Access Memory. That means that a program usually cannot be over forty lines long. However, the BASIC which the Pocket Computer uses in programming incorporates a "chain" command which is more simple and easy to use than the "chain" commands found in most larger systems. By "chain"ing we can write programs in segments that would, when put together, make a 64K computer blush. Imagine that!

The Pocket Computer's BASIC is missing a few common functions. There is no provision for "DATA" or "READ" instructions. There is no "INSTRING" capability for looking into and taking apart the contents of a String Variable. There is no provision for "PEEK"ing or "POKE"ing into the Machine Language of the Pocket Computer, or using Assembly Language with it. There are other limitations, too. They all make the challenge more exciting!

My first programming experience was on the Pocket Computer. Later I got a computer with more capabilities. I like my larger system a great deal, but I always come back to the Pocket Computer. It's like recreation. I guess I like to move mountains.

There are two interfaces available for the Pocket Computer. One is the Cassette Interface, which allows programs and data to be saved on a Cassette Recorder for storage and later loading back into the Pocket Computer. The other is the Printer/Cassette Interface, which is a Cassette Interface and Printer for producing hard copy of program runs and listings. Most of the programs in this series utilize and are formatted for use either with or without the Printer/Cassette Interface and a Cassette Recorder for program and data loading and storing. If you have these peripherals, or can get them, they very definitely will broaden the scope of applications for the Pocket Computer. They, and the programs in this series, make up a few of the unlimited number of Pocket Operating Systems.

Now, imagine with me a program . . .

## THE WORD PROCESSING CHALLENGE

Perhaps the most useful and popular type of program used in real world systems today is the word processor. They are used for many needs, from producing many letter-quality form letters, to developing manuscripts into books. Usually word processors come as a package, which completely take over the computer functions, turning it into a page oriented editing device for producing final copy suitable for whatever application the user has in mind. Available in many packages are spelling editors, grammar editors, and even editors that will go through the entire manuscript and find cliches. Most word processors are expected to at least be able to justify lines to the left and right margins without hyphenating any words, thus making very presentable finished copy.

By definition, a word processor is a computer-based operating system that can take raw manuscript and transform it into finished copy. That implies, but does not require, that editing functions may exist within the package. Also not mentioned in the definition is the disk system which normally would be involved, allowing speedy access of different programs and data as needed. There are many other functions that are normally associated with word processors, and in the future, there will undoubtedly be many more. It is in the nature of the computer industry itself to continue advancing the capabilities of operating systems, in order to make them more competitive on the market.

Obviously, with all that can be involved with word processors, it makes it difficult to talk about them and the Pocket Computer in the same sentence. However, if we stick to our definition, there are some programs that can allow the Pocket Computer to do some word processing functions. Whether you want to call them word processors or not is up to you.

The two programs we will be working with in this article both require that the user have some knowledge of what the Pocket Computer can do with string variables. The user must, when asked for input by the computer, enter the copy seven (or less) characters at a time. This means that an example sentence like "Mary had a little lamb." could be entered like this:

```
(USER) MARY HA _ ⟨ENTER⟩
       D A LIT _ ⟨ENTER⟩
       TLE LAM _ ⟨ENTER⟩
       B. _ ⟨ENTER⟩
```

The " _ " is what the computer does.

There's another complication, though. If you are using the Printer Interface, as normally would be run with these programs, the printed copy is limited to a 16 character per line format. Since our maximum string length is 7 characters, we can either limit our output (and input) to two strings (14 characters), or to three strings per line, and limit the third string to only two characters. With the second method, the capacity of each line is greater, that being 16 characters. Because of the greater capacity for line length, this is the method that is used in both Pocket Typewriter, and Pocket Processor. Here is an example of how to enter copy into these systems:

When the "?" appears—

```
(USER) MARY HA _ ⟨ENTER⟩ OUTPUT:
         D A LIT _ ⟨ENTER⟩
             - _   ⟨ENTER⟩ MARY HAD A LIT-
       TLE LAM _ ⟨ENTER⟩
       B WHOSE _ ⟨ENTER⟩
               _   ⟨ENTER⟩ TLE LAMB WHOSE
        FLEECE _ ⟨ENTER⟩
        WAS AS _ ⟨ENTER⟩
               _   ⟨ENTER⟩ FLEECE WAS AS
       WHITE A _ ⟨ENTER⟩
       S SNOW. _ ⟨ENTER⟩
               _   ⟨ENTER⟩ WHITE AS SNOW.
```

It takes a little practice to get used to, but the two extra characters will be worth it. Just remember that if you enter any more than two characters into the third string of any given line, the printed copy of that line will scroll off into the space that the next line should be in. Lines like that can be corrected, but they take some time and effort.

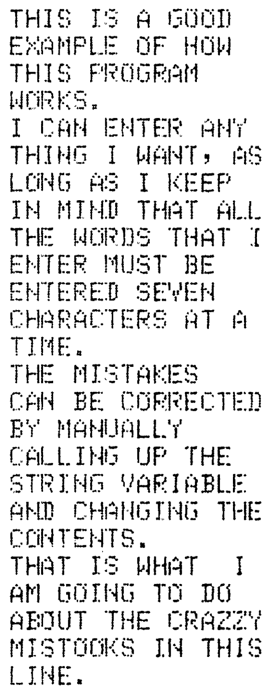
The main differences between Pocket Typewriter and Pocket Processor are in the way they deal with data storage. Pocket Typewriter fills the computer memory first if necessary, then upon the user's command, stores the data on tape for later use. Since the data remains within the computer, individual strings can be called up (manually) for correction and editing. Pocket Processor, on the other hand, enters each line into tape storage upon completion, avoiding the need to watch for overloading the computer. Editing on this system can be accomplished either before the lines are stored on tape, or after.

You've probably noticed that automatic editing has not been mentioned yet. Nor has justification of the margins, or a dictionary function. That is because our little computer cannot handle the string manipulation necessary to do these things. But, for what it can do, there's nothing to be ashamed of. The main purposes of these programs are to take in the user's manuscript, provide editing opportunities, make a hard copy, and store the data in an accessible form for later use. The computer can handle these functions very well. The user, then, must handle the editing.

These programs are both fun to use for notes, or letters, etc. I had even considered using Pocket Processor in the development of this series, as it could provide the finished copy (in 16 character per line format) in a suitable format for printing. As you will see, though, this would take a great deal of time. But it was sort of fun just imagining it . . .

## POCKET TYPEWRITER

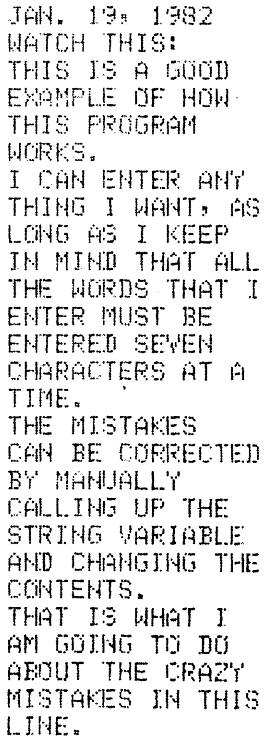
Sample Run:
(PAUSE) POCKET TYPEWRITER
C/1981 BY DON HERGERT
TO ENTER SHFT A
TO PRINT SHFT B
TO LOAD DATA SHFT C
TO SAVE DATA SHFT D
TO CLEAR SHFT F
(USER) SHFT A
(THE PROGRAM NOW ASKS FOR DATA, STORES IT INTO THE COMPUTER'S MEMORY, AND PRINTS IT ON THE PRINTER.)



```
THIS IS A GOOD
EXAMPLE OF HOW
THIS PROGRAM
WORKS.
I CAN ENTER ANY
THING I WANT, AS
LONG AS I KEEP
IN MIND THAT ALL
THE WORDS THAT I
ENTER MUST BE
ENTERED SEVEN
CHARACTERS AT A
TIME.
THE MISTAKES
CAN BE CORRECTED
BY MANUALLY
CALLING UP THE
STRING VARIABLE
AND CHANGING THE
CONTENTS.
THAT IS WHAT  I
AM GOING TO DO
ABOUT THE CRAZZY
MISTOOKS IN THIS
LINE.
```

(USER) * ⟨ENTER⟩ * ⟨ENTER⟩ * ⟨ENTER⟩ A ⟨ENTER⟩
(PRINT TO DISPLAY) 103
(USER) A$(98) ⟨ENTER⟩
(PRINT) MISTOOK
(USER) A$(98) = "MISTAKE" ⟨ENTER⟩
(PRINT) MISTAKE
(USER CAN CONTINUE CORRECTIONS IN THIS MANNER)
(USER) SHFT D (AFTER PREPARING CASSETTE)
(COMPUTER NOW RECORDS DATA ON TAPE)
(USER) SHFT F
(COMPUTER NOW CLEARS ALL VARIABLES)
(USER) SHFT C (AFTER PREPARING CASSETTE)

(COMPUTER NOW LOADS DATA FROM TAPE)
(USER) SHFT B
(INPUT) DATE _
(USER) JAN. 19 ⟨**ENTER**⟩, 1982 ⟨**ENTER**⟩
(INPUT) NAME
(USER) WATCH T ⟨**ENTER**⟩ HIS: ⟨**ENTER**⟩
(PRINT)



```
JAN. 19, 1982
WATCH THIS:
THIS IS A GOOD
EXAMPLE OF HOW
THIS PROGRAM
WORKS.
I CAN ENTER ANY
THING I WANT, AS
LONG AS I KEEP
IN MIND THAT ALL
THE WORDS THAT I
ENTER MUST BE
ENTERED SEVEN
CHARACTERS AT A
TIME.
THE MISTAKES
CAN BE CORRECTED
BY MANUALLY
CALLING UP THE
STRING VARIABLE
AND CHANGING THE
CONTENTS.
THAT IS WHAT I
AM GOING TO DO
ABOUT THE CRAZY
MISTAKES IN THIS
LINE.
```

Probably even if we could handle justification of the margins, the lines would still look a little strange, because we can only get so many words into a 16 character line. For this system, it might be more trouble than it's worth.

As mentioned earlier, the BASIC that resides within the Pocket Computer has limited string handling capabilities. You may want to further explore or experiment in order to get an idea of exactly what those limitations are. It is possible to do some word searching, and even editing, but you will find that the more you force into this program, the less practical it will become.

In the sample run, the user first keys in ⟨**SHFT**⟩, and ⟨**SPC**⟩. The computer then presents the introduction and menu. In order to enter new copy, the user keys in ⟨**SHFT**⟩ ⟨**A**⟩, and upon the appearance of the cursor, enters the data for processing 7 characters at a time for the first two inputs, and a maximum of 2 characters at a time for the third input of each line. At the end of the third input, the computer prints the line, and asks for the first input of the next line. When the user has completed the final line, an asterisk is entered for the next three inputs for what would be the next line. The computer reads these as an escape command and presents the menu again.

After the escape from the data entering subroutine, the user in this example keys in an ⟨**A**⟩ and ⟨**ENTER**⟩. The variable A in this program is holding the number of the last dimensioned string variable that was used. In the sample that number was 103. The user then stepped back through the contents of strings A$(103) to A$(98), where the first correction was made by simply keying in manually the string name and what it should contain, and then entering it. The remainder of the sample run is self-explanatory.

Although it is limited by the amount of memory available for strings (enough for about 40 lines), the convenience of not needing to be connected to a cassette recorder in order to print is welcomed. For a note or letter, the copy can be printed as often as the user wants, using as many headings or greetings as wanted. If it is a manuscript that the user wants to re-use later, the data can be saved on tape, and loaded at will. If the user wants, manuscript can be longer than normal memory space would allow. This is accomplished by setting up multiple files, and simply calling one file after the other as needed, thus making it possible to have pre-recorded letters or notes that could have interchangeable paragraphs for use in different situations.

Program Listing:

```
1 " "PAUSE "POCKET TYPEWRITER"
     : PAUSE "C/1981 BY DON HERGERT"
3 PAUSE "TO ENTER SHFT A"
     : PAUSE "TO PRINT SHFT B"
     : PAUSE "TO LOAD DATA SHFT C"
4 PAUSE "TO SAVE DATA SHFT D"
     : PAUSE "TO CLEAR SHFT F"
     : END
5 "F"
     : CLEAR
     : END
1Ø "A"
     : A=26
2Ø B=A+1
     : C=B+1
     : D=C+1
3Ø INPUT A$(B),A$(C)
     : BEEP 1
     : INPUT A$(D)
35 IF A$(B)="*" IF A$(C)="*" IF A$(D)="*" LET
     A$(B)=""
     : A$(C)=""
     : A$(D)=""
     : GOTO 3
4Ø PRINT A$(B);A$(C);A$(D)
5Ø A=D
     : GOTO 2Ø
6Ø "B"
     : INPUT "DATE",E$,F$,"NAME",G$,H$
     : PRINT E$;F$
     : PRINT G$;H$
     : FOR A=27TO D STEP 3
     : B=A+1
     : C=B+1
7Ø PRINT A$(A);A$(B);A$(C)
8Ø NEXT A
9Ø GOTO 3
1ØØ "C"
     : INPUT #"DATA"
     : GOTO 3
15Ø "D"
     : PRINT #"DATA"
     · GOTO 3
```

Now let's move on to the second program.

## POCKET PROCESSOR

Sample Run:
(PAUSE) POCKET PROCESSOR
C/1981 BY DON HERGERT
TO ENTER PRESS SHFT A
FOR COPIES PRESS SHFT B
(USER) SHFT A
(INPUT) TITLE: _
(USER) SAMPLE (ENTER)

```
(PAUSE) COPY:
(INPUT) ?
(USER)                    THIS ISA GOOD
(INPUT) ANY CORRECTIONS? (Y/N) _
(USER) Y (ENTER)
(PAUSE) COPY:
(INPUT) ?
(USER)                    THIS IS A GOOD
(HERE THE                 EXAMPLE OF WHAT
PROGRAM                   THIS PROGRAM CAN
MAKES THE                 DO.
COMPUTER                  IT IS NOT LIMIT-
ASK THE                   ED TO THE AMOUNT
USER FOR                  MEMORY AVAILABLE
COPY,                     IN THE COMPUTER.
PRINT THAT                THE ONLY LIMIT-
COPY, ASK                 ATION IS IN HOW
IF THERE                  LONG YOUR TAPE
ARE ANY                   IS.
CORRECTIONS, AND IF NOT, RECORD THAT LINE ON
TAPE. IF THERE ARE CORRECTIONS, THE PROGRAM
WILL ASK FOR THE CORRECT LINE, AND AFTER GET-
TING IT, PRINT, VERIFY, AND RECORD IT. THE USER MUST
AFTER THE LAST LINE TO BE RECORDED ENTER A
SPACE FOR EACH OF THE NEXT THREE ?S, OR THE
COMPUTER WILL, WHEN SEARCHING THE TAPE FOR
DATA, NEVER KNOW WHEN TO STOP.)
(USER) SHFT B
(INPUT) TITLE TO FIND: _
(USER) SAMPLE (ENTER)
(PRINT)                   THIS IS A GOOD
                          EXAMPLE OF WHAT
                          THIS PROGRAM CAN
                          DO.
                          IT IS NOT LIMIT-
                          ED TO THE AMOUNT
                          MEMORY AVAILABLE
                          IN THE COMPUTER.
                          THE ONLY LIMIT-
                          ATION IS IN HOW
                          LONG YOUR TAPE
(END)                     IS.
```

This sample run is fairly self-explanatory. The program has only two modes, those being a mode for data entering, and a mode for calling and printing data. Since this program immediately records the edited data onto tape, there is no need to be concerned about remaining memory space when in the entry mode. As mentioned in the sample run, the user must at the completion of the final line, key in ⟨**SPC**⟩ and ⟨**ENTER**⟩ for the next three inputs. This not only serves as a key to a conditional escape routine, but it also provides a pointer to the program when in the calling and printing mode the computer reaches the end of the current data file on tape.

This program does an interesting thing as it loads data from tape for eventual printing. It searches from file to file for the lines that carry the title that the user has instructed it to find, and as it does this, it will skip over those lines that do not have the correct title. This means, among other things, that data blocks for different manuscripts can be mixed without mixing up the hard copy.

This program can provide some good examples of the nature of data storage on tape. One of the exciting possibilities of this system is editing already existing tape files. Provided that care is taken to not crash surrounding files, editing can be done by simply recording over the line (file) that is to be changed (or deleted), using the data entry mode of the program. If you are careful, you can even add a line (or two if you are tricky), by recording the new line(s) on the leading 6-second non-signal beep that normally precedes each data block on tape. This takes practice, so don't try it on an important file until you know you can do it without crashing the surrounding lines. The ability to do this with this system makes editing possible on two levels, and generally much easier and simpler than with Pocket Typewriter, thus making Pocket Processor the choice for heavy duty usage.

Program Listing:

```
10 " "PAUSE "POCKET PROCESSOR"
     : PAUSE " C/1981 BY DON HERGERT
20 CLEAR
     : BEEP 5
     : PAUSE "TO ENTER PRESS SHFT A"
     : PAUSE "FOR COPIES PRESS SHFT B"
     : END
100 "A"INPUT "TITLE:  ";A$(142)
110 PAUSE "COPY:  "
     : INPUT A$(143),A$(144)
     : BEEP 1
     : INPUT A$(145)
120 PRINT A$(143);A$(144);A$(145)
130 INPUT "ANY CORRECTIONS? (Y/N) ";C$
     : IF C$="Y" THEN 110
140 PRINT #"TITLE";A$(142)
150 IF A$(143)=" "IF A$(144)=" "IF A$(145)=" "THEN 20
160 GOTO 110
200 "B"INPUT "TITLE TO FIND:  ";T$
210 INPUT #"TITLE";A$(142)
     : IF T$=A$(142)THEN 230
220 GOTO 210
230 IF A$(143)=" "IF A$(144)=" "IF A$(145)=" "THEN 20
240 PRINT A$(143);A$(144);A$(145)
     : GOTO 210
```

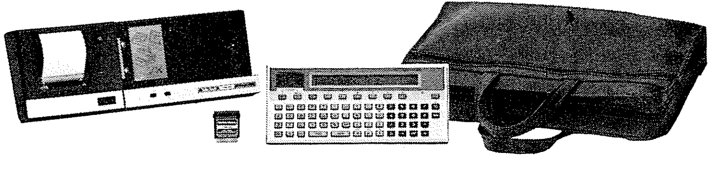
## SUMMARY

The key to any successful program is convenience. One of the main reasons that a program or system might fail on the market is the lack of this convenience, or "User Friendliness".

Convenience doesn't just include the functions that a program might do. Speed is an important factor, as are ease of operation and clarity of the output. Any of these attributes require memory space, though, and in our little computer, that is what is missing. Consequently function by function we have to go through a perspective program and decide what is important enough to keep, and then file the remaining ideas for use in another time and place.

This is what was done with these two programs, and in fact most of the programs in this series. It's the challenge that we've been discussing and working with. It is such a thrill to see a Pocket Computer at work on some of these projects that no matter how many conveniences we need to give up, the outcome is worth it.

Pocket Typewriter and Pocket Processor are practical programs. It is difficult in some ways to equate these programs with their real world counterparts. Large word processing systems definitely eclipse these little systems, but for what we have to work with, these programs are fairly impressive. Let's go on imagining from here . . .



# Circle Subroutine For PC-2

Here is a subroutine that you might want to use in some of your graphics programs. It will draw a circle using a set of parameters that you can specify. The parameters are as follows:

X,Y coordinate—the coordinate of the centerpoint of the circle

Radius—the radius of the circle (1-108)

Color—the color of the circle (0-3)

Height/Width Ratio—This parameter gives you the ability to make ellipses. A value of 1 will make the circle a "perfect" circle. A value from 2 to 99 will make the circle range from a near "perfect" circle (99) to a vertical line (2). A value from .01 to .99 will make the circle range from a near "perfect" circle (.99) to a horizontal line (.01).

Start and End Points—These parameters will allow you to make arcs. Both values must be between 1 and 360 (inclusive). Also, you should take note that the circle will always be drawn counterclockwise.

Density—The density is the number of lines that will be used to draw the circle. A value of 3 would actually give you a triangle; a value of 4, a square; a value of 5, a pentagon, etc. The value can range from 2 to ???. However, use 20 to 100 for a circle.

Additional Note: You must be in GRAPH mode and already have your origin set before calling this subroutine.

```
100:GRAPH
110:LPRINT "        ";
120:SORGN
130:INPUT "X COORDINATE: ";X
135:IF X<-107 OR X>107 GOSUB 300
     :GOTO 130
140:INPUT "Y COORDINATE: ";Y
145:IF Y<-107 OR Y>107 GOSUB 300
     :GOTO 140
150:INPUT "RADIUS: ";R
160:INPUT "COLOR: ";C
164:C=INT C
165:IF C<0 OR C>3 GOSUB 300
     :GOTO 160
170:INPUT "HEIGHT/WIDTH RATIO: ";H
171:H=ABS H
172:IF H<1 LET H=INT (H*100)/100
     :GOTO 180
175:H=INT H
180:INPUT "START POINT: ";S
184:S=INT S
185:IF S<0 OR S>360 GOSUB 300
     :GOTO 180
190:INPUT "END POINT: ";E
194:E=INT E
195:IF E<0 OR E>360 GOSUB 300
     :GOTO 190
200:INPUT "DENSITY: ";D
210:IF D<2 GOSUB 300
     GOTO 200
220:GOSUB 20000
240:GOTO 130
300:PAUSE "INPUT ERROR! - TRY AGAIN"
     :RETURN
500:REM ..........
600:REM ..........
10000:REM  CIRCLE SUBROUTINE
10010:REM  X = X COORDINATE
10020:REM  Y = Y COORDINATE
10030:REM  R = RADIUS
10040:REM  C = COLOR
10050:REM  H = HEIGHT/WIDTH RATIO
10060:REM  S = START POINT
10070:REM  E = END POINT
10080:REM  D = DENSITY
20000:DEGREE
20005:IF S>=ELET E=E+360
20030:IF H=1LET XM=1
     :YM=1
20040:IF H>1LET XM=H/100
     :YM=1
20050:IF H<1LET XM=1
     :YM=H
20052:OX=X+COS (S)*R*XM
     :LX=OX
20057:OY=Y+SIN (S)*R*YM
     :LY=OY
20060:Z=360/D
20070:FOR I=STO E+1STEP Z
20075:J=I
     :IF I>360 LET J=I-360
20080:XC=X+COS J*R*XM
20090:YC=Y+SIN J*R*YM
20100:LINE (OX,OY)-(XC,YC),0,C
20110:OX=XC
     :OY=YC
20120:NEXT I
20140:RETURN
```

# Quick Printer Graph Routine

**Greg Patek**
**Route 1 Box 309**
**Katy, TX 77449**

I own a TRS-80 Model I 16K Level II with the Quick Printer (26-1153). The program that I wrote will let the printer do "Character Graphics" on the printer line by line (a total of 16 lines). The program can be used as a subroutine in your program, or the program can store the drawing in memory and print it at a later time. It's up to you. I use this program with a program that draws a graph on the screen and then the printer types it on the line printer.

```
1000 REM  * PRINTING ROUTINE *
1010 FOR S=15360 TO 16383
     : P=PEEK(S)
     : IF P=32 OR P=128 THEN LPRINT " ";
     : POKE S, 32
     : ELSE LPRINT "#";
     : POKE S, 35
1020 T=T+1
     : IF T>63 THEN LPRINT CHR$(13)
     : T=0
1030 NEXT S
1040 FOR H=1 TO 4
     : LPRINT " "
     : NEXT H
1050 RETURN
```

I use this routine a lot with all of my graphic drawings and other graphic designs.

# Christmas Tree Letterhead

**Art Voisard**
**1310 Cornwall**
**Ocean Springs, MS 39564**

I continue to marvel at the 'classic' programs some of the readers send, including their Christmas efforts.

I thought your followers might enjoy (and wish to modify) a program I wrote to put a letterhead on the notes we include with Christmas cards (using my Line Printer VII, of course). It does not contain mathematical wizardry, but I am proud of the result it achieves.

How about devoting more space to the Color Computer . . . after all, that is the playful one, isn't it?

```
2 '==========> XMAS TREE LETTERHEAD
4 '==========> ART VOISARD
6 '==========> 131Ø CORNWALL
8 '==========> OCEAN SPRINGS, MS 39564
1Ø '=========> WRITTEN 1981
15 PCLEAR 4
2Ø PMODE 4,1
25 PCLS Ø
3Ø SCREEN 1,Ø
32 PSET (128,4)
35 PSET (128,8)
    : PSET (126,1Ø)
    : PSET (13Ø,1Ø)
    : PSET (128,6)
4Ø FOR X = 124 TO 132 STEP 4
    : PSET (X,12)
    : NEXT X
45 FOR X = 126 TO 13Ø STEP 4
    : PSET (X,14)
    : NEXT X
5Ø FOR X = 124 TO 132 STEP 4
    : PSET (X,16)
    : NEXT X
55 FOR X = 122 TO 134 STEP 4
    : PSET (X,18)
    : NEXT X
6Ø FOR X = 12Ø TO 136 STEP 4
    : PSET (X,2Ø)
    : NEXT X
65 FOR X = 122 TO 134 STEP 4
    : PSET (X,22)
    : NEXT X
7Ø FOR X = 118 TO 138 STEP 4
    : PSET (X,24)
    : NEXT X
75 FOR X = 122 TO 138 STEP 4
    : PSET (X,26)
    : NEXT X
8Ø FOR X = 12Ø TO 136 STEP 4
    : PSET (X,28)
    : NEXT X
85 FOR X = 118 TO 138 STEP 4
    : PSET (X,3Ø)
    : NEXT X
9Ø FOR X = 114 TO 142 STEP 4
    : PSET (X,32)
    : NEXT X
95 FOR X = 118 TO 142 STEP 4
    : PSET (X,34)
    : NEXT X
1ØØ FOR X = 116 TO 138 STEP 4
    : PSET (X,36)
    : NEXT X
1Ø5 FOR X = 112 TO 14Ø STEP 4
    : PSET (X,38)
    : NEXT X
11Ø FOR X = 114 TO 144 STEP 4
    : PSET (X,4Ø)
    : NEXT X
115 FOR X = 116 TO 142 STEP 4
    : PSET (X,42)
    : NEXT X
12Ø FOR X = 114 TO 144 STEP 4
    : PSET (X,44)
    : NEXT X
125 FOR X = 11Ø TO 146 STEP 4
    : PSET (X,46)
    : NEXT X
13Ø FOR X = 112 TO 148 STEP 4
    : PSET (X,48)
    : NEXT X
135 FOR X = 114 TO 144 STEP 4
    : PSET (X,5Ø)
    : NEXT X
14Ø FOR X = 112 TO 146 STEP 4
    : PSET (X,52)
    : NEXT X
145 FOR X = 11Ø TO 148 STEP 4
    : PSET (X,54)
    : NEXT X
15Ø FOR X = 1Ø8 TO 15Ø STEP 4
    : PSET (X,56)
    : NEXT X
155 FOR X = 112 TO 148 STEP 4
    : PSET (X,58)
    : NEXT X
16Ø FOR X = 11Ø TO 15Ø STEP 4
    : PSET (X,6Ø)
    : NEXT X
165 FOR X = 1Ø8 TO 152 STEP 4
    : PSET (X,62)
    : NEXT X
17Ø FOR X = 1Ø6 TO 15Ø STEP 4
    : PSET (X,64)
    : NEXT X
175 FOR X = 1Ø4 TO 152 STEP 4
    : PSET (X,66)
    : NEXT X
18Ø FOR X = 1Ø2 TO 154 STEP 4
    : PSET (X,68)
    : NEXT X
185 FOR X = 1Ø8 TO 158 STEP 4
    : PSET (X,7Ø)
    : NEXT X
19Ø FOR X = 1Ø6 TO 158 STEP 4
    : PSET (X,72)
    : NEXT X
195 FOR X = 1Ø4 TO 154 STEP 4
    : PSET (X,74)
    : NEXT X
2ØØ FOR X = 1Ø2 TO 156 STEP 4
```

```
    : PSET (X,76)
    : NEXT X
2Ø5 FOR X = 98 TO 158 STEP 4
    : PSET (X,78)
    : NEXT X
21Ø FOR X = 1Ø2 TO 156 STEP 4
    : PSET (X,8Ø)
    : NEXT X
215 FOR X = 1ØØ TO 16Ø STEP 4
    : PSET (X,82)
    : NEXT X
22Ø FOR X = 96 TO 158 STEP 4
    : PSET (X,84)
    : NEXT X
225 FOR X = 98 TO 16Ø STEP 4
    : PSET (X,86)
    : NEXT X
23Ø FOR X = 94 TO 162 STEP 4
    : PSET (X,88)
    : NEXT X
235 FOR X = 9Ø TO 16Ø STEP 4
    : PSET (X,9Ø)
    : NEXT X
24Ø FOR X = 92 TO 158 STEP 4
    : PSET (X,92)
    : NEXT X
245 FOR X = 94 TO 16Ø STEP 4
    : PSET (X,94)
    : NEXT X
25Ø FOR X = 96 TO 164 STEP 4
    : PSET (X,96)
    : NEXT X
255 FOR X = 94 TO 162 STEP 4
    : PSET (X,98)
    : NEXT X
26Ø FOR X = 88 TO 164 STEP 4
    : PSET (X,1ØØ)
    : NEXT X
265 FOR X = 92 TO 166 STEP 4
    : PSET (X,1Ø2)
    : NEXT X
27Ø FOR X = 9Ø TO 162 STEP 4
    : PSET (X,1Ø4)
    : NEXT X
275 FOR X = 88 TO 164 STEP 4
    : PSET (X,1Ø6)
    : NEXT X
28Ø FOR X = 86 TO 68 STEP 4
    : PSET (X,1Ø8)
    : NEXT X
285 PSET (9Ø,11Ø)
    : PSET (138,11Ø)
    : PSET (96,112)
29Ø PSET (94,11Ø)
    : PSET (98,11Ø)
295 FOR X = 1Ø2 TO 11Ø STEP 4
    : PSET (X,11Ø)
    : NEXT X
3ØØ FOR X = 114 TO 134 STEP 4
    : PSET (X,11Ø)
    : NEXT X
3Ø5 FOR X = 142 TO 17Ø STEP 4
    : PSET (X,11Ø)
    : NEXT X
31Ø PSET (1Ø4,112)
    : PSET (1Ø8,112)
315 PSET (116,112)
    : PSET (12Ø,112)
32Ø PSET (124,112)
    : PSET (128,112)
    : PSET (136,112)
325 PSET (142,112)
    : PSET (146,112)
33Ø PSET (15Ø,112)
    : PSET (152,112)
    : PSET (156,112)
```

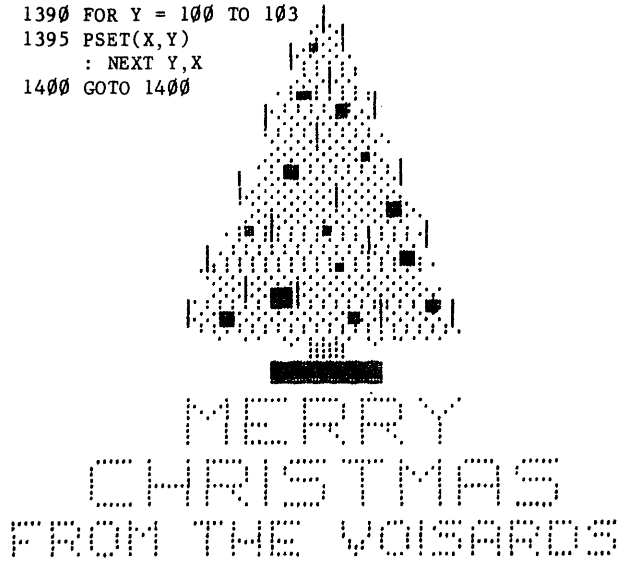
```
335 PSET (16Ø,112)
    : PSET (164,112)
    : PSET (118,114)
34Ø PSET (144,114)
    : PSET (132,112)
345 FOR X = 124 TO 134 STEP 2
35Ø FOR Y = 114 TO 118 STEP 2
355 PSET (X,Y)
    : NEXT Y,X
36Ø FOR X = 112 TO 146
365 FOR X = 12Ø TO 126
37Ø PSET (X,Y)
    : NEXT Y,X
4ØØ '=======> MERRY CHRISTMAS
4Ø5 '==> M
41Ø FOR Y = 132 TO 146 STEP 2
    : PSET (86,Y)
    : NEXT Y
415 PSET (88,134)
    : PSET (9Ø,136)
    : PSET (92,138)
42Ø PSET (94,136)
    : PSET (96,134)
425 FOR Y = 132 TO 146 STEP 2
    : PSET (98,Y)
    : NEXT Y
43Ø '==> E
435 FOR Y = 132 TO 146 STEP 2
    : PSET (1Ø4,Y)
    : NEXT Y
44Ø FOR X = 1Ø6 TO 116 STEP 2
    : PSET (X,132)
    : NEXT X
445 PSET (1Ø6,14Ø)
    : PSET (11Ø,14Ø)
    : PSET (1Ø8,14Ø)
45Ø FOR X = 1Ø6 TO 116 STEP 2
    : PSET (X,146)
    : NEXT X
455 '==> R
46Ø FOR X = 122 TO 132 STEP 2
    : PSET (X,132)
    : NEXT X
465 FOR Y = 134 TO 146 STEP 2
    : PSET (122,Y)
    : NEXT Y
47Ø FOR X = 124 TO 132 STEP 2
    : PSET (X,14Ø)
    : NEXT X
475 FOR Y = 134 TO 138 STEP 2
    : PSET (134,Y)
    : NEXT Y
48Ø PSET (13Ø,142)
    : PSET (132,144)
    : PSET (134,146)
485 '==> R
49Ø FOR X = 14Ø TO 15Ø STEP 2
    : PSET(X,132)
    : NEXT X
495 FOR Y = 134 TO 146 STEP 2
    : PSET(14Ø,Y)
    : NEXTY
5ØØ FOR X = 142 TO 15Ø STEP 2
    : PSET(X,14Ø)
    : NEXT X
5Ø5 FOR Y = 134 TO 138 STEP 2
    : PSET(152,Y)
    : NEXT Y
51Ø PSET(148,142)
    : PSET(15Ø,144)
    : PSET(152,146)
515 '==> Y
52Ø PSET(158,132)
    : PSET(17Ø,132)
    : PSET(16Ø,134)
    : PSET(168,134)
```

```
525 PSET(162,136)
    : PSET(166,136)
53Ø FOR Y = 138 TO 146 STEP 2
    : PSET(164,Y)
    : NEXT Y
535 '==> C
54Ø FOR Y = 154 TO 164 STEP 2
    : PSET(56,Y)
    : NEXT Y
545 FOR X = 58 TO 68 STEP 2
    : PSET(X,152)
    : NEXT X
55Ø FOR X = 58 TO 68 STEP 2
    : PSET(X,166)
    : NEXT X
555 '==> H
56Ø FOR Y = 152 TO 166 STEP 2
    : PSET(74,Y)
    : NEXT Y
565 FOR X = 76 TO 84 STEP 2
    : PSET(X,16Ø)
    : NEXT X
57Ø FOR Y = 152 TO 166 STEP 2
    : PSET(86,Y)
    : NEXT Y
575 '==> R
58Ø FOR Y = 152 TO 166 STEP 2
    : PSET(92,Y)
    : NEXT Y
585 FOR X = 94 TO 1Ø2 STEP 2
    : PSET(X,152)
    : NEXT X
59Ø FOR X = 94 TO 1Ø2 STEP 2
    : PSET(X,16Ø)
    : NEXT X
595 FOR Y = 154 TO 158 STEP 2
    : PSET(1Ø4,Y)
    : NEXT Y
6ØØ PSET(1ØØ,162)
    : PSET(1Ø2,164)
    : PSET(1Ø4,166)
6Ø5 '==> I
61Ø FOR Y = 152 TO 166 STEP 2
    : PSET(11Ø,Y)
    : NEXT Y
615 '==> S
62Ø FOR X = 118 TO 128 STEP 2
    : PSET(X,152)
    : NEXT X
625 PSET(116,154)
    : PSET(116,156)
    : PSET(128,16Ø)
63Ø FOR X = 118 TO 126 STEP 2
    : PSET(X,158)
    : NEXT X
635 PSET(128,16Ø)
    : PSET(128,162)
    : PSET(128,164)
64Ø FOR X = 116 TO 126 STEP 2
    : PSET(X,166)
    : NEXT X
645 '==> T
65Ø FOR X = 134 TO 146 STEP 2
    : PSET(X,152)
    : NEXT X
655 FOR Y = 154 TO 166 STEP 2
    : PSET(14Ø,Y)
    : NEXT Y
66Ø '==> M
665 FOR Y = 152 TO 166 STEP 2
    : PSET(152,Y)
    : NEXT Y
67Ø FOR Y = 152 TO 166 STEP 2
    : PSET(164,Y)
    : NEXT Y
675 PSET(154,154)
```

```
     : PSET(156,156)
     : PSET(158,158)
68Ø PSET(16Ø,156)
     : PSET(162,154)
685 '==> A
69Ø FOR X = 172 TO 18Ø STEP 2
     : PSET(X,152)
     : NEXT X
695 FOR Y = 154 TO 166 STEP 2
     : PSET(17Ø,Y)
     : NEXT Y
7ØØ FOR Y = 154 TO 166 STEP 2
     : PSET(182,Y)
     : NEXT Y
7Ø5 FOR X = 172 TO 18Ø STEP 2
     : PSET(X,16Ø)
     : NEXT X
71Ø '==> S
715 FOR X = 19Ø TO 2ØØ STEP 2
     : PSET(X,152)
     : NEXT X
72Ø FOR X = 19Ø TO 198 STEP 2
     : PSET(X,158)
     : NEXT X
725 FOR X = 188 TO 198 STEP 2
     : PSET(X,166)
     : NEXT X
73Ø PSET(188,154)
     : PSET(188,156)
735 FOR Y = 16Ø TO 164 STEP 2
     : PSET(2ØØ,Y)
     : NEXT Y
74Ø '==> F
745 FOR X = 32 TO 4Ø STEP 2
     : PSET(X,172)
     : NEXT X
75Ø FOR Y = 174 TO 182 STEP 2
     : PSET(32,Y)
     : NEXT Y
755 PSET(34,178)
     : PSET(36,178)
76Ø '==> R
765 FOR X = 44 TO 5Ø STEP 2
     : PSET(X,172)
     : NEXT X
77Ø FOR X = 46 TO 5Ø STEP 2
     : PSET(X,178)
     : NEXT X
775 FOR Y = 174 TO 182 STEP 2
     : PSET(44,Y)
     : NEXT Y
78Ø PSET(52,174)
     : PSET(52,176)
     : PSET(5Ø,18Ø)
785 PSET(52,182)
79Ø '==> O
795 FOR X = 58 TO 62 STEP 2
     : PSET(X,172)
     : NEXT X
8ØØ FOR X = 58 TO 62 STEP 2
     : PSET(X,182)
     : NEXT X
8Ø5 FOR Y = 174 TO 18Ø STEP 2
     : PSET(56,Y)
     : NEXT Y
81Ø FOR Y = 174 TO 18Ø STEP 2
     : PSET(64,Y)
     : NEXT Y
815 '==> M
82Ø FOR Y = 172 TO 182 STEP 2
     : PSET(68,Y)
     : NEXT Y
825 FOR Y = 172 TO 182 STEP 2
     : PSET(76,Y)
     : NEXT Y
83Ø PSET(7Ø,174)
```

```
     : PSET(72,176)
     : PSET(74,174)
835 '==> T
84Ø FOR X = 88 TO 96 STEP 2
     : PSET(X,172)
     : NEXT X
845 FOR Y = 174 TO 182 STEP 2
     : PSET(92,Y)
     : NEXT Y
85Ø '==> H
855 FOR Y = 172 TO 182 STEP 2
     : PSET(1ØØ,Y)
     : NEXT Y
86Ø FOR Y = 172 TO 182 STEP 2
     : PSET(1Ø8,Y)
     : NEXT Y
865 FOR X = 1Ø2 TO 1Ø6 STEP 2
     : PSET(X,178)
     : NEXT X
87Ø '==> E
875 FOR X = 112 TO 12Ø STEP 2
     : PSET(X,172)
     : NEXT X
88Ø FOR X = 112 TO 12Ø STEP 2
     : PSET(X,182)
     : NEXT X
885 FOR X = 114 TO 118 STEP 2
     : PSET(X,178)
     : NEXT X
89Ø FOR Y = 174 TO 182 STEP 2
     : PSET(112,Y)
     : NEXT Y
895 '==> V
9ØØ FOR Y = 172 TO 176 STEP 2
     : PSET(134,Y)
     : NEXT Y
9Ø5 FOR Y = 172 TO 176 STEP 2
     : PSET(142,Y)
     : NEXT Y
91Ø PSET(136,178)
     : PSET(136,18Ø)
     : PSET(14Ø,178)
     : PSET(14Ø,18Ø)
     : PSET(138,182)
     : PSET(138,18Ø)
915 '==> O
92Ø FOR X = 148 TO 152 STEP 2
     : PSET(X,172)
     : NEXT X
925 FOR X = 148 TO 152 STEP 2
     : PSET(X,182)
     : NEXT X
93Ø FOR Y = 174 TO 18Ø STEP 2
     : PSET(146,Y)
     : NEXT Y
935 FOR Y = 174 TO 18Ø STEP 2
     : PSET(154,Y)
     : NEXT Y
94Ø '==> I
945 FOR Y = 172 TO 182 STEP 2
     : PSET(158,Y)
     : NEXT Y
95Ø '==> S
955 FOR X = 164 TO 17Ø STEP 2
     : PSET(X,172)
     : NEXT X
96Ø FOR X = 164 TO 168 STEP 2
     : PSET(X,176)
     : NEXT X
965 FOR X = 162 TO 168 STEP 2
     : PSET(X,182)
     : NEXT X
97Ø PSET(162,174)
     : PSET(17Ø,178)
     : PSET(17Ø,18Ø)
975 '==> A
```

```
98Ø FOR Y = 174 TO 182 STEP 2
     : PSET(174,Y)
     : NEXT Y
985 FOR Y = 174 TO 182 STEP 2
     : PSET(182,X)
     : NEXT Y
99Ø FOR X = 176 TO 18Ø STEP 2
     : PSET(X,172)
     : NEXT X
995 FOR X = 176 TO 18Ø STEP 2
     : PSET(X,178)
     : NEXT X
1ØØØ '==> R
1ØØ5 FOR X = 186 TO 192 STEP 2
     : PSET(X,172)
     : NEXT X
1Ø1Ø FOR X = 188 TO 192 STEP 2
     : PSET(X,178)
     : NEXT X
1Ø15 FOR Y = 174 TO 182 STEP 2
     : PSET(186,Y)
     : NEXT Y
1Ø2Ø PSET(194,174)
     : PSET(194,176)
     : PSET(192,18Ø)
     : PSET(194,182)
1Ø25 '==> D
1Ø3Ø FOR X = 2ØØ TO 2Ø4 STEP 2
     : PSET(X,172)
     : NEXT X
1Ø35 FOR X = 2ØØ TO 2Ø4 STEP 2
     : PSET(X,182)
     : NEXT X
1Ø4Ø FOR Y = 172 TO 182 STEP 2
     : PSET(198,Y)
     : NEXT Y
1Ø45 FOR Y = 174 TO 18Ø STEP 2
     : PSET(2Ø6,Y)
     : NEXT Y
1Ø6Ø '==> S
1Ø65 FOR X = 212 TO 218 STEP 2
     : PSET(X,172)
     : NEXT X
1Ø7Ø FOR X = 212 TO 216 STEP 2
     : PSET(X,176)
     : NEXT X
1Ø75 FOR X = 21Ø TO 216 STEP 2
     : PSET(X,182)
     : NEXT X
1Ø8Ø PSET(21Ø,174)
     : PSET(218,178)
     : PSET(218,18Ø)
1Ø85 FOR X = 128 TO 13Ø
1Ø9Ø FOR Y = 76 TO 78
1Ø95 PSET(X,Y)
     : NEXT Y,X
11ØØ FOR Y = 42 TO 5Ø
     : PSET(126,Y)
     : NEXT Y
11Ø5 FOR Y = 4 TO 12
     : PSET(128,Y)
     : NEXT Y
111Ø FOR Y = 16 TO 24
     : PSET(118,Y)
     : NEXT Y
1115 FOR Y = 1Ø TO 18
     : PSET(134,Y)
     : NEXT Y
112Ø FOR Y = 24 TO 32
     : PSET(13Ø,Y)
     : NEXT Y
1125 FOR Y = 36 TO 44
     : PSET(11Ø,Y)
     : NEXT Y
113Ø FOR Y = 32 TO 4Ø
     : PSET(144,Y)
     : NEXT Y
1135 FOR Y = 42 TO 50
     : PSET(126,Y)
     : NEXT Y
1140 FOR Y = 58 TO 66
     : PSET(102,Y)
     : NEXT Y
1145 FOR Y = 54 TO 62
     : PSET(152,Y)
     : NEXT Y
1150 FOR Y = 62 TO 70
     : PSET(132,Y)
     : NEXT Y
1155 FOR Y = 72 TO 80
     : PSET(112,Y)
     : NEXT Y
1160 FOR Y = 82 TO 90
     : PSET(92,Y)
     : NEXT Y
1165 FOR Y = 74 TO 82
     : PSET(160,Y)
     : NEXT Y
1170 FOR Y = 76 TO 84
     : PSET(142,Y)
     : NEXT Y
1175 FOR Y = 92 TO 100
     : PSET(104,Y)
     : NEXT Y
1180 FOR Y = 100 TO 108
     : PSET(86,Y)
     : NEXT Y
1190 FOR Y = 94 TO 102
     : PSET(120,Y)
     : NEXT Y
1195 FOR Y = 100 TO 108
     : PSET(146,Y)
     : NEXT Y
1200 FOR Y = 100 TO 108
     : PSET(168,Y)
     : NEXT
1205 FOR X = 124 TO 126
1210 FOR Y = 16 TO 18
1215 PSET(X,Y)
     : NEXT Y,X
1220 FOR X = 120 TO 124
1225 FOR Y = 32 TO 34
1230 PSET(X,Y)
     : NEXT Y,X
1235 FOR X = 132 TO 135
1240 FOR Y = 36 TO 40
1245 PSET(X,Y)
     : NEXT Y,X
1250 FOR X = 116 TO 120
1255 FOR Y = 56 TO 60
1260 PSET(X,Y)
     : NEXT Y,X
1265 FOR X = 140 TO 142
1270 FOR Y = 52 TO 54
1275 PSET(X,Y)
     : NEXT Y,X
1280 FOR X = 148 TO 152
1285 FOR Y = 68 TO 72
1290 PSET(X,Y)
     : NEXT Y,X
1295 FOR X = 104 TO 106
1300 FOR Y = 76 TO 78
1305 PSET(X,Y)
     : NEXT Y,X
1310 FOR X = 112 TO 118
1315 FOR Y = 96 TO 102
1320 PSET(X,Y)
     : NEXT Y,X
1325 FOR X = 96 TO 100
1330 FOR Y = 104 TO 108
1335 PSET(X,Y)
     : NEXT Y,X
1340 FOR X = 132 TO 134
1345 FOR Y = 88 TO 90
1350 PSET(X,Y)
     : NEXT Y,X
1355 FOR X = 136 TO 139
1360 FOR Y = 104 TO 107
1365 PSET(X,Y)
     : NEXT Y,X
1370 FOR X = 152 TO 156
1375 FOR X = 84 TO 88
1380 PSET(X,Y)
     : NEXT Y,X
1385 FOR X = 160 TO 163
1390 FOR Y = 100 TO 103
1395 PSET(X,Y)
     : NEXT Y,X
1400 GOTO 1400
```

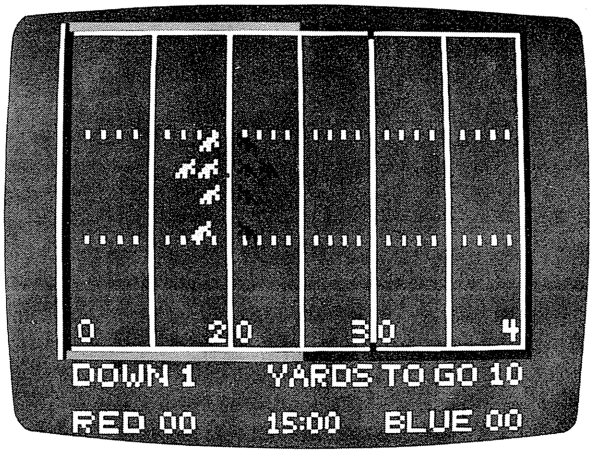


# Football Program Pak Tip

**Matthew Staller**
**200 Winston Drive**
**Cliffside Park, NJ 07010**

For all you Football PROGRAM PAK owners, you can return a punt. But you have to place the "receiver" right along the sideline. Usually you can "grab" the ball with your head or feet on the upper and lower sidelines respectively. Sometimes the punt will stop next to the sideline and be treated as a "Quick Kick."

Note: I have 32K Extended Color BASIC.



# Using Your Printer To Make Math Worksheets For Grades K-8

The Radio Shack K-8 Math Worksheet Generator program lets you use any Radio Shack line printer to print out math worksheets and answer sheets for grades K through 8. In this article, we'll look at the features of this time-saving program; then we'll look at some helpful hints from program user Lou Weiner, a fourth-grade teacher in Saugus, California.

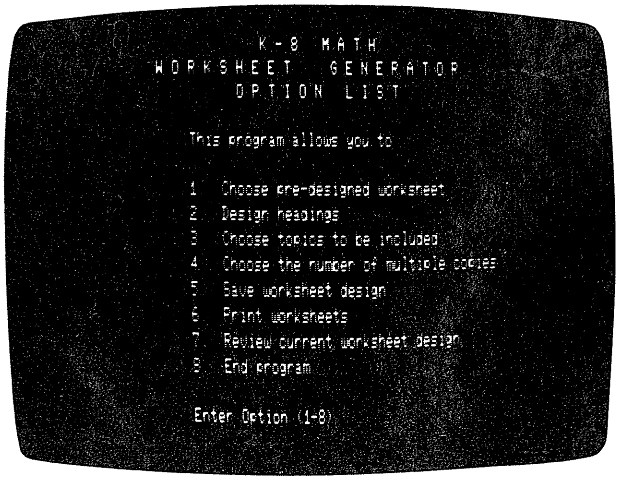
Teachers know that making math worksheets can be time consuming if you use traditional methods. The teacher has to make up the exercises, write or type them onto duplicating masters, and often make corrections on the masters before mimeographing can be done. To give a class two or more worksheets on the same level of difficulty, or to vary the worksheets for different classes or semesters, means that the teacher has to repeat the process from scratch.

The Radio Shack K-8 Math Worksheet Generator program (Cat. No. 26-2162) helps teachers produce worksheets and answer sheets in a fraction of the time it usually takes. And once the teacher has designed a worksheet, the computer can print out any number of different worksheets on the same topic and level of difficulty.

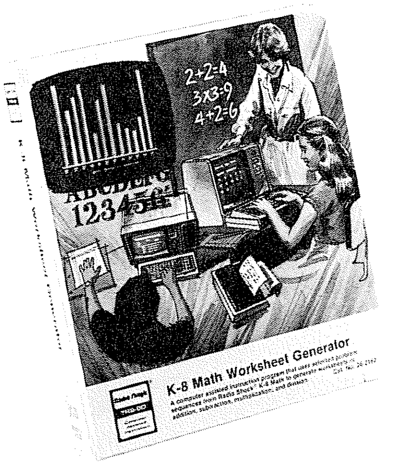
Worksheets can be printed on lesson levels selected from the addition, subtraction, multiplication, and division sequences of the Radio Shack K-8 Math Program, Volume One (Cat. No. 26-1715; or K-8 Math With Student Management, Cat. No. 26-1725). You select the topic(s) for the worksheet, the lesson number(s), and the number of problems from each lesson. The computer then uses these guidelines to randomly generate exercises. For each worksheet generated, an answer sheet is also generated and printed. Special features make the program very versatile. You can choose the total number of worksheets to be printed and can have these worksheets be all alike, or you can have the computer print different worksheets on the same skill level. You can also design headings for the worksheet, and can save the worksheet design on diskette for later use.

A Curriculum Summary in the Worksheet Generator teacher's manual helps you match lesson numbers with the skills you are teaching, even if you aren't using Radio Shack's K-8 Math Program, Volume One. The Worksheet Generator Cross-Reference, also in the teacher's manual, correlates K-8 lesson numbers with specific pages from six of the major K-8 basal mathematics series.

Let's take a closer look at the program's features. Once Worksheet Generator is loaded and running, you'll see a list of options:



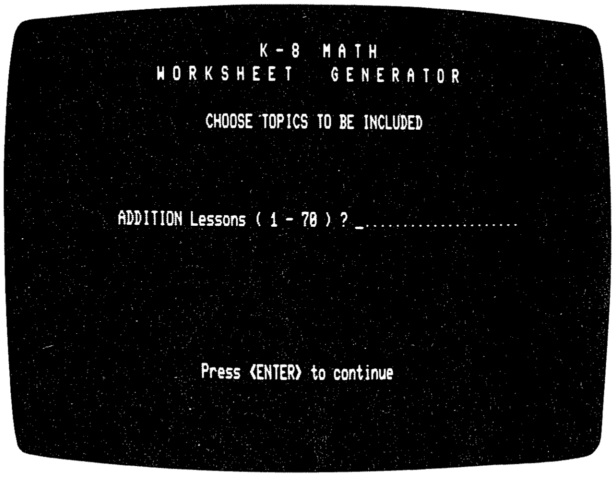
```
            K - 8   M A T H
   W O R K S H E E T   G E N E R A T O R
            O P T I O N   L I S T

      This program allows you to

      1   Choose pre-designed worksheet
      2   Design headings
      3   Choose topics to be included
      4   Choose the number of multiple copies
      5   Save worksheet design
      6   Print worksheets
      7   Review current worksheet design
      8   End program

      Enter Option (1-8)
```



To design a worksheet, you'll need to use Options 2 ("Design headings"), 3 ("Choose topics to be included"), and 4 ("Choose the number of multiple copies"). Option 7 lets you take a look at the overall design of your worksheet. Option 5 then lets you save your worksheet design, and Option 6 lets you print out worksheets. Option 1 lets you use any worksheet design that has been saved on disk (using Option 5). Option 8 ends the program.

Through Option 2, you can enter as many as four headings (for example: NAME, DATE, CLASS, and GRADE). These headings will appear at the top of the worksheet, followed by blanks for the teacher or student to fill in. Entering headings is very easy. When you choose Option 2, some brief directions and a numbered dotted line appear on the screen. To enter a heading, you type the heading on the line and press ⟨ENTER⟩. When you have entered all the headings you wish, simply press ⟨ENTER⟩ at the dotted line to return to the option list.

When you choose Option 3 ("Choose topics to be included"), you'll see this screen:

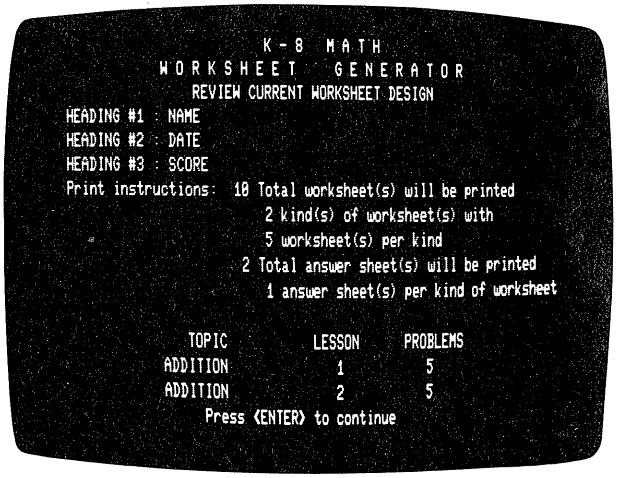


```
        K - 8  M A T H
WORKSHEET   GENERATOR

   CHOOSE TOPICS TO BE INCLUDED


ADDITION Lessons ( 1 - 70 ) ? _.....................




      Press <ENTER> to continue
```

The screen displays the name of each topic (addition, subtraction, multiplication, and division) in turn, and asks which lessons from that topic you want the worksheet to cover. For example, the screen above is asking you to select lessons from the topic "addition." To include some addition problems, you'd type the numbers of the desired lessons, separated by commas, and press ⟨ENTER⟩. For each lesson number you chose, you'd be asked for the number of problems to include from that lesson. To "skip" addition, you'd simply press ⟨ENTER⟩, without selecting any lessons. Before you return to the option list, the program gives you the chance to redo your selections.

Option 4 lets you choose the total number of worksheets to be printed, the number of kinds of worksheets (different worksheets on the same level of difficulty), and the number of answer sheets you want for each kind of worksheet.

Option 7 lets you review the current worksheet design. Included in the review are the names of the headings entered in Option 2, the print instructions entered in Option 4, and the topics, lesson numbers, and number of problems entered in Option 3. For example:



```
                  K - 8  M A T H
          W O R K S H E E T   G E N E R A T O R
             REVIEW CURRENT WORKSHEET DESIGN
HEADING #1 : NAME
HEADING #2 : DATE
HEADING #3 : SCORE
Print instructions:  10 Total worksheet(s) will be printed
                       2 kind(s) of worksheet(s) with
                       5 worksheet(s) per kind
                     2 Total answer sheet(s) will be printed
                       1 answer sheet(s) per kind of worksheet

            TOPIC          LESSON    PROBLEMS
          ADDITION           1          5
          ADDITION           2          5
             Press <ENTER> to continue
```

Once you have completed Options 2, 3, and 4, your worksheet is ready to be saved on diskette. When you select Option 5 to "Save worksheet design," you'll be asked to enter a worksheet name. You are also given the option to enter a password, which protects your worksheet design from unauthorized users. Finally, you enter a drive number, and the computer saves the worksheet design.

Option 6 lets you print out the worksheets. First, the computer pauses to generate exercises. Then you simply ready the printer and start the printing. The computer does the rest.

Option 1 lets you recall a worksheet design that you saved on the diskette earlier, in order to print out more copies. Each time you recall and print a specific worksheet, the problems which appear on the worksheet will be different. This is because the computer randomly generates new problems, using the guidelines of your worksheet design, each time that worksheet is printed. This feature is particularly convenient for a teacher who will be using a worksheet for more than one class or for more than one semester. Not only through Option 4, but also with every printing, the teacher can get different worksheets on the same topic and level of difficulty without having to rewrite the worksheet. Since an answer key is generated for each new worksheet, the time spent in grading homework and exercises is no greater than it would be if all the worksheets were identical.

## HELPFUL HINTS FOR USING THE K-8 MATH WORKSHEET GENERATOR

Fourth-grade teacher Lou Weiner, of Saugus, California, has been using TRS-80 microcomputers for about three and one-half years, and has been using the Worksheet Generator program for almost a year. Before using Worksheet Generator, Weiner programmed some of his own worksheet-producing software. But Weiner is impressed with Worksheet Generator because, he says, it has "a lot more parameters." For example, his multiplication programs didn't force carrying, while many of the lesson levels in Worksheet Generator's multiplication topic do. He also uses Radio Shack's K-8 Math With Student Management Program, Volume One, in his classroom.

Weiner has come up with some good ideas for using Worksheet Generator in the classroom. When a worksheet contains a whole page of long-division problems, for example, it takes time for the computer to print out enough sheets for an entire class. So instead of starting up the printer and leaving it, Weiner puts a spirit master in the printer. The computer prints the worksheet directly onto the master, which Weiner then uses to make as many copies as desired.

Weiner says that the answer sheet that is also printed can be a single answer key for the teacher's use, or it can be used to make many copies so that students can check their work in class. (If you don't plan to use spirit masters, remember that the program allows you to choose how many copies of the answer sheet you want to print.) Weiner finds it useful for students to check their work in class because, he says, the answer sheet "shows each line of the problem as it should be worked." Students can look at the answer sheet and see, for example, whether they are multiplying incorrectly or just making mistakes in addition.

Weiner has also found a diagnostic use for the Worksheet Generator program. He uses the program not only for worksheets, but also says, "all my tests are done with it." When he designs a worksheet, he prints out the screen display from Option 7 ("Review current worksheet design") by pressing ⟨SHIFT⟩ ⟨↓⟩ ⟨*⟩ at the Model III keyboard. This gives him a quick-reference list of the lesson numbers covered in the worksheet or test. For students who do not do well on the worksheet, he uses the list as an aid in designing individualized supplementary worksheets that target the concepts missed.

### HARDWARE REQUIREMENTS, PRICE, AND AVAILABILITY

The K-8 Math Worksheet Generator program (Cat. No. 26-2162) can be used with a TRS-80 Model III 32K or 48K disk system and any Radio Shack line printer. The suggested retail price of the program is $99.95, though prices may vary at individual stores and dealers.

Worksheet Generator is available through any Radio Shack store or Computer Center. Also, Radio Shack has Regional Educational Sales Coordinators throughout the country to help schools determine and meet their educational computing needs. For details, call your local Radio Shack store or Computer Center.

# Single Disk FORTRAN

**Tim Daneliuk**
**4927 North Rockwell Street**
**Chicago, IL 60625**

Here is a step-by-step method for running the Radio Shack FORTRAN Compiler with a single disk drive on a TRS-80 Model I (with special thanks to Spencer Lepley of Tallahassee Florida who figured out how to link on a single drive without L80 being on the program diskette):

1) Load the disk with EDIT on it and load this program by typing in EDIT. Now remove the disk.
2) Place the disk that is to contain your FORTRAN Source file into the drive and EDIT the file as desired. Save the source on this disk. Now remove this disk.
3) Insert the disk with F80 on it and load the program by typing F80. Now remove this disk.
4) Insert the disk with your source program on it. Compile the program as usual (e.g. TEMP, TEMP=TEMP). This writes the /REL and /LST files to the program disk. Now remove this disk.
5) Insert the disk with L80 on it and load the linker by typing in L80. Remove this disk.
6) Insert the disk with the /MAC, /REL, and /LST files on it (really all you need is the /REL file, so if have a copy of the source elsewhere and need the disk space, kill the /MAC and /LST files: they are real space hogs!). Now key in the file name you want to link (e.g. TEMP). This loads the relocatable code and displays the undefined globals (like the ones from FORLIB). Remove this disk.
7) Insert the disk with FORLIB on it. Note, at this point you are still in L80! To resolve the undefined globals type in FILENAME-S, where FILENAME is the file you want to search to find the global definitions. Usually, this will be FORLIB/REL-S (S is for search) unless you have your own library files. Remove this disk.
8) Insert the disk which is to have the final executable /CMD file on it. Key in FILENAME-N to output object code, then -E to write the file and then exit L80. The process is complete.

You can save space on a disk by killing the FORMAT/CMD, BASIC/CMD, BASICR/CMD, and BACKUP/CMD files (they are invisible protected files whose password is their own name. For example BASIC and BASICR have the password BASIC. FORMAT has the password of FORMAT and so on.

# Color Logo Now Available for the TRS-80 Color Computer

Color LOGO is Radio Shack's new LOGO program for the 32K Disk-based or 16K ROM pack TRS-80 Color Computer. "LOGO" is an educational computer language that can be used to draw pictures on the computer's video display. When simple instructions are entered at the computer keyboard, a shape on the screen (called a "turtle") moves around and leaves a trail of color.

Color LOGO can be a lot of fun for a computer enthusiast or novice of any age, but its main educational use has been with young children.

Color LOGO isn't a language for teaching. You wouldn't use the language to present traditional lesson material. But educators who have been involved in the development and use of LOGO-type programs have been enthusiastic about the language's power to let children learn by exploring. To make the turtle move, children have to plan an action and then describe it to the turtle through commands like "forward," "back" ,"right," and "left". This involves them in thinking logically about direction, distance, and angle. They can define a problem such as; "How can I draw a square?" and then break the problem into its parts (in this case, four equal moves forward, with a 90 degree turn between each move). The language gives children vast opportunities to be creative.

### HOW TO USE COLOR LOGO

Let's look at how Radio Shack's Color LOGO program can be used and at a few of its many special features.

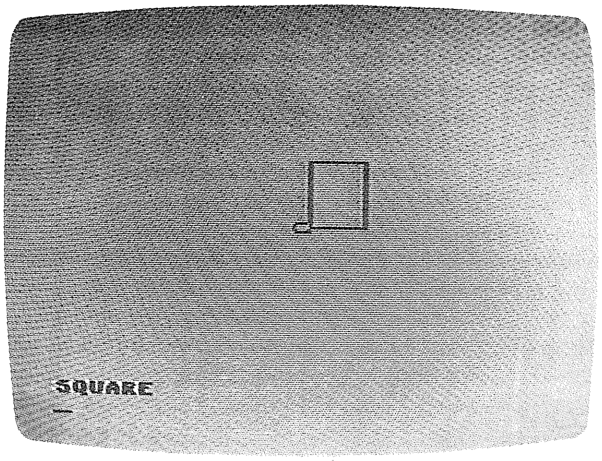
Color LOGO can be loaded into the TRS-80 Color Computer via diskette or ROM pack. User programs written in Color LOGO can be saved onto diskette or cassette.

To move the turtle on the screen, you get into the program's "Run" mode. There you can enter commands such as "FORWARD" (which can be abbreviated as "FD"), "LEFT" (or "LT"), and so on. Along with a command like "FD", you type a number that tells the turtle how far forward to move. With a command like "LEFT", the number indicates how many degrees the turtle should turn in place before receiving the next command to move forward or back. As you enter each command, the turtle makes one move or one turn. By entering commands one at a time, you finally create a picture on the screen.

Once you've mastered the basics of turtle movement, you can write a series of turtle movements and save the series. Such a set of turtle movements is called a "procedure." Color LOGO provides line-oriented editing for procedures, through the program's "Edit" mode. That is, Color LOGO lets you type each command ("FD", "LT", etc.) into the computer and save it as part of the procedure, without having the turtle follow each command as you type it.

For example, the directions below form a procedure for drawing a square. The prefix "TO" marks the name of the program. "END" signals the end of the program. The information between these two lines consists of ordinary turtle commands. (For example, "FD 30" means "move forward 30 units"; "RT 90" means "turn right 90 degrees."):

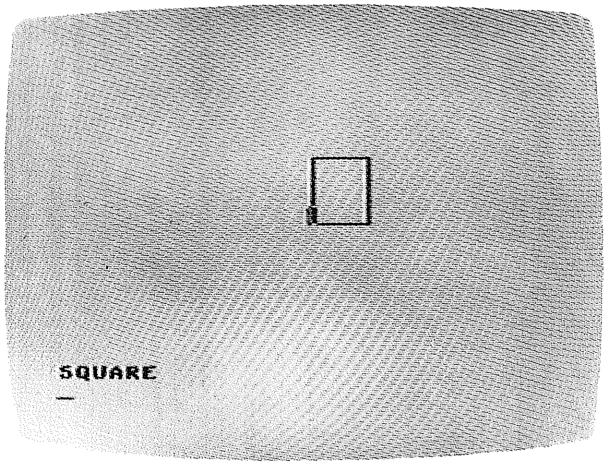
```
TO SQUARE
FD 3Ø  RT 9Ø  FD 3Ø  RT 9Ø  FD 3Ø
RT 9Ø  FD 3Ø
END
```



```
SQUARE
_
```

You can use Color LOGO's "REPEAT" command to save yourself some typing while you write the same procedure. The directions below tell the turtle to repeat the action in parentheses four times. (The only difference in this version of "SQUARE" is that the turtle makes an extra right turn after completing the square.)

```
TO SQUARE
REPEAT 4 (FD 3Ø  RT 9Ø)
END
```



```
SQUARE
_
```

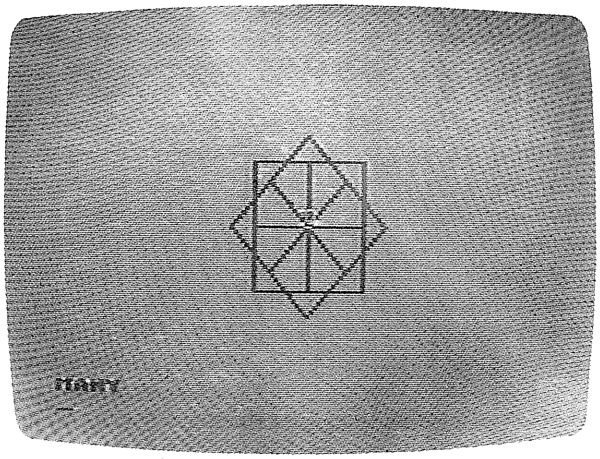
Once you've entered a procedure in memory, you can get into the "Run" mode and use the procedure just like any turtle command. That is, entering "SQUARE" as a command (instead of entering a simple turtle command like "FD 30"), would make the turtle follow all the steps outlined in the procedure "SQUARE".

## PROCEDURES WITHIN PROCEDURES

Since a whole procedure can be used just like a simple turtle command, this means that you can include procedures within procedures, just as you include simple turtle commands within procedures. For example, the procedure below tells the turtle to repeat "SQUARE" ten times, making a 45 degree turn to the left before each repetition:

```
TO MANY
REPEAT 1Ø (SQUARE  LT 45)
END
```

Once the procedure "MANY" was in memory, entering the command "MANY" in the Run Mode would produce a more complicated design made with ten squares:



```
MANY
_
```

## VARIABLES IN PROCEDURES

Now suppose that you want to draw a lot of squares of different sizes. Instead of writing a different square procedure for each different size, you can use variables. Using one or more variables in a procedure allows you to change the procedure every time you run it.

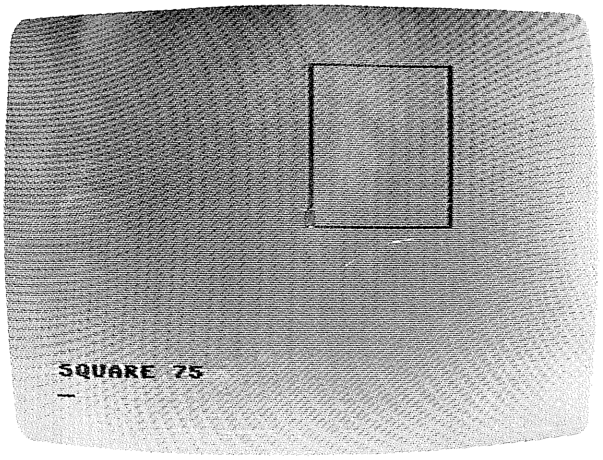
To change the size of a square, you need to change the length of its sides. In the procedure "SQUARE", the sides are drawn by the turtle command "FD 30". To change the length of the sides, we just need to change the number after "FD". With a variable, we can leave the length of the sides open for change each time we run the procedure. For example:

```
TO SQUARE :LENGTH
REPEAT 4 (FD :LENGTH  RT 9Ø)
END
```

This way, entering "SQUARE 10" in the "Run" mode will produce a square with sides 10 units long. Entering "SQUARE 50" will make the sides 50 units long, and so on.

## SPECIAL FEATURES FOR VERY YOUNG CHILDREN

Color LOGO's "Doodle Mode" is a special feature for children who can't read yet. In Doodle Mode, keys on the top row of the TRS-80 Color Computer correspond to turtle commands like "RT 45", "FD 10", and so on. A plastic overlay for these keys contains graphic representations of each key's meaning. This feature allows children who can't yet read or



```
SQUARE 75
_
```

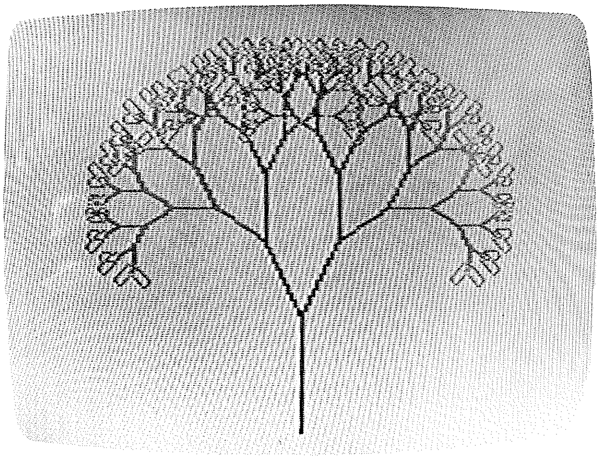
type reliably to use the language and to benefit from the practice in structured thinking that the language offers.

Or, parents or teachers can set up "One-Key Doodling" in the Run mode. The adult writes a procedure that is named by one key. For example, a procedure like "SQUARE" might be written and given the name "S" instead of "SQUARE". When the child works with the program, pressing the "S" key and pressing "ENTER" will make the turtle draw a square. For children who don't recognize letters, parents can cut a small square out of colored tape and place it over the "S" key.

### MULTIPLE TURTLES

Unlike most versions of LOGO, Color LOGO allows more than one turtle to be at work on the screen at once. Several parts of a complex drawing can be drawn at once, with the effect that the drawing seems to evolve rather than being drawn piecemeal. For example, you can write a procedure that draws a many-branched tree, with a turtle ending up at the end of each branch.

Multiple turtles allow you to program simple games in Color LOGO. Each player can be assigned one turtle that maintains its position until that player's next turn.



Color LOGO also allows you to create new shapes for turtles. You can use different-shaped turtles as different pieces in a game, or can program simple animation and use turtles as animated figures.

### OTHER COLOR LOGO FEATURES

A "SLOW" command lets you vary the speed at which procedures are carried out. Speeds range from "0" (full speed) to "127" (slowest speed).

A "PENUP" command lets you move the turtle without leaving a trail on the screen.

A "HIDETURTLE" command lets you make the turtle invisible.

Arithmetic expressions can be used with variables in Color LOGO procedures.

As mentioned above, limited animation is possible in Color LOGO.

Screen colors can be changed, including the overall colorset, the background color, and the color of the line drawn by the turtle.

The Color LOGO user's manual contains numerous sample procedures, including some at the end of the manual that demonstrate special effects. For example, one of these procedures draws a clock face with moving hands!

### HOW YOU CAN USE COLOR LOGO

The Radio Shack Color LOGO program is ideal for classroom or home. The program is available through Radio Shack stores and Computer Centers nationwide. Color LOGO for 32K disk-based TRS-80 Color Computer (catalog number 26-2721) is listed at $99.00. The suggested retail price of the ROM pack version (Cat. No. 26-2722) for 16K ROM-based TRS-80 Color Computer is $49.95. Prices may vary at individual stores and dealers.

Much has been written about the educational theory behind LOGO-type programs. For more information, you may wish to consult Seymour Papert's book Mindstorms: Children, Computers, and Powerful Ideas (Basic Books, 1980) and the book Turtle Geometry: Computation as a Medium for Exploring Mathematics (M.I.T. Press) by H. Abelson and A. DiSessa. Chapter One of Mindstorms was condensed in the March 1981 issue of Creative Computing, under the title "Computers and Computer Cultures."

### MODES IN COLOR LOGO

The Color LOGO system can be in one of 4 "modes" depending upon what the user is doing at the time.

BREAK MODE is entered automatically upon starting Color LOGO, and can be entered from any other mode by pressing the BREAK key at any time. It is signified by the "LOGO:" prompt on the screen. The following single-letter commands may be used:

R enters RUN MODE

E enters EDIT MODE

P prints contents of internal program on the printer connected to the serial port.

Q prints same as P common except the Q sends a line feed after a return character.

L prompts for module designation with a "LOAD:", then reads from the specified source into the internal program area.

S prompts for the module name with a "SAVE:", writes the internal program area to the specified destination.

EDIT MODE is entered from BREAK MODE by pressing E. In EDIT MODE one can edit the currently loaded modules. To start with a blank program area, press SHIFT CLEAR in BREAK MODE before pressing E.

RUN MODE is entered form the BREAK MODE by entering "R." When RUN MODE is entered, the screen is cleared and the turtle appears at the home position. A TEXT WINDOW of three lines exists at the bottom of the screen. The user enters turtle graphics commands or calls Color LOGO procedures that have been entered earlier via the EDIT or DOODLE MODE. The user can enter any of the following commands directly in RUN MODE. However, in RUN MODE no more than one command may be entered on a line. Once the ENTER key is pressed, the command is executed.

CLEAR
HOME
FORWARD
BACK
RIGHT
LEFT
PENUP
PENDOWN
PENCOLOR
SHOWTURTLE
HIDETURTLE
SETX
SETY
SETHEADING
SLOW
COLORSET
BACKGROUND
WRAP
NOWRAP
SEND
PRINT
HATCH
VANISH

DOODLE MODE is entered from RUN MODE by pressing the "@" key. DOODLE MODE allows the creation of a turtle graphics procedure that will draw a shape without requiring that the user even know how to read. In DOODLE mode the screen displays an "=" sign. The user enters a word (nonsense or otherwise) of at least one letter or number and presses **⟨ENTER⟩**. The word is the name of the procedure to be created as a picture is drawn.

The numeric keys (marked by the special keyboard overlay) can be used to enter turtle graphic commands. Each time a key is pressed, the specified command is executed by the turtle. At the same time, a procedure is created in the program area. This procedure can be viewed by entering EDIT MODE.

The DOODLE MODE commands are:
(1) CLEAR
(2) HOME
(3) PU
(4) PD
(5) RT 45
(6) LT 45
(7) FD 1
(8) FD 10
(9) RT 15
(0) LT 15

## COLOR LOGO STATEMENTS AND COMMANDS

Control statements include:

IF expr The expression is evaluated. If the value is true (non-zero) the list of statements in parenthesis is executed. If it is false (0) then the list of statements is skipped.

ELSE This statement must follow an IF statement. If the expression value on the IF statement is false, then the list of statements after ELSE is executed. Otherwise it is skipped.

REPEAT expr The expression is evaluated; if it has a value less than or equal to zero, then the list of statements is skipped. Otherwise the list of statements is executed the specified number of times.

WHILE expr The expression is evaluated; if it is false (0) then the list of statements is skipped. If it is true (non-zero) then the list of statements is executed. After the list is executed, control returns to the WHILE and the process is repeated.

STOP This terminates the execution of a procedure. Control is returned to the calling procedure if there is one. If the procedure was called from RUN MODE, then control returns to RUN MODE.

TO procname parmlist This statement defines the start of a Color LOGO procedure. It must start in column 1 of a line and must be the only statement on the line. The "proname" may be any name of one or more letters.

END This is the last statement in a procedure. Execution of an END is equivalent to that of the STOP statement.

VANISH VANISH takes the current turtle out of existence.

MAKE :var expr The value of the expression is assigned to the variable.

PRINT "TEXT"

PRINT expr The literal text of the expression value is displayed at the turtle location. The turtle is not moved.

NOWRAP Execution of the NOWRAP statement takes the screen out of its normal wrap mode. If a turtle then runs off the screen the program will terminate with an "OUT OF BOUNDS" error message.

WRAP Puts the screen back in wrap mode.

HATCH expr procname arglist Creates a new turtle. The turtle will start at the same (X,Y) position as its parent (the turtle which HATCHed it) and will be pointed in the same direction.

SEND expr expr A message is sent to the specified turtle. The first expression value denotes the identification of the turtle to which the message is sent. The value of the second expression is the value sent to the other turtle.

procname arglist This is referred to as a CALL statement, even though it does not contain the word CALL. To CALL any procedure, just code its name followed by any arguments to be passed.

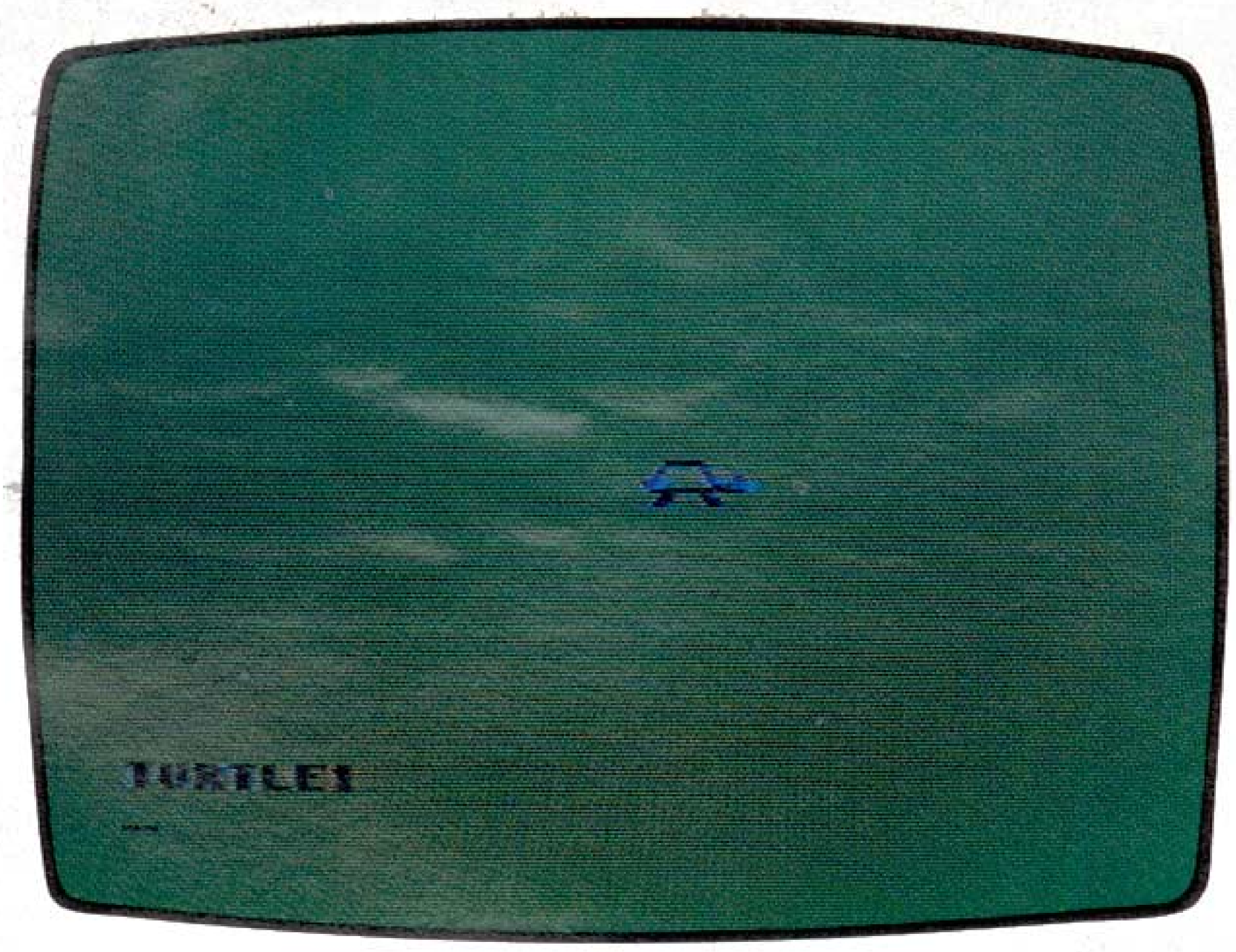
SLOW expr The SLOW statement causes execution to slow down so that it can be watched more closely. The value of the expression denotes how slow to go.

**Turtle Graphics Commands include:**

BACK expr
BACKGROUND expr
COLORSET expr
CLEAR
FORWARD expr
HIDETURTLE
HOME
LEFT expr
PENCOLOR
PENDOWN
PENUP
RIGHT expr
SETHEADING expr
SETX expr
SETY expr
SHAPE shape list
SHOWTURTLE

## TURTLE SHAPE LIST

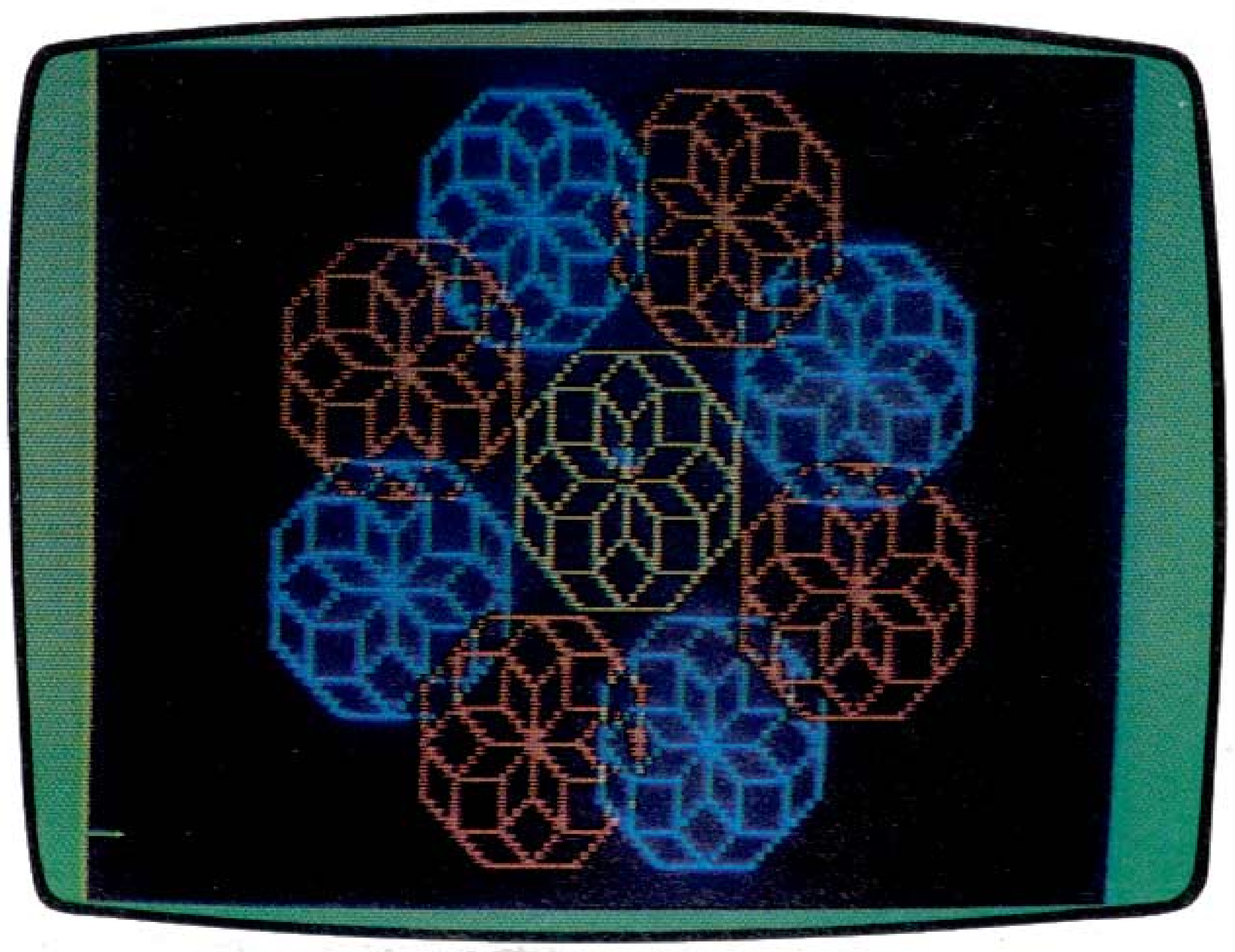
The SHAPE statement is used to assign a new shape to the current turtle. The shape of a turtle is made up of a pattern of dots on a grid. The shape list tells Color LOGO how to draw the turtle pattern. The turtle shape is automatically rotated to face in the direction the turtle is headed. Drawing the turtle shape is similar to using normal turtle graphics commands to draw any shape. The difference is that the commands which make up the shape list are a restricted and simplified form of the normal turtle graphics commands. The commands allow a step of one pixel (one square on a piece of graph paper) in any of the 8 possible directions. The 8 directions are: up, down, right, left, and the four diagonal directions.



### THE CLOCK PROGRAM FROM COLOR LOGO

**Copyright © MicroPi**

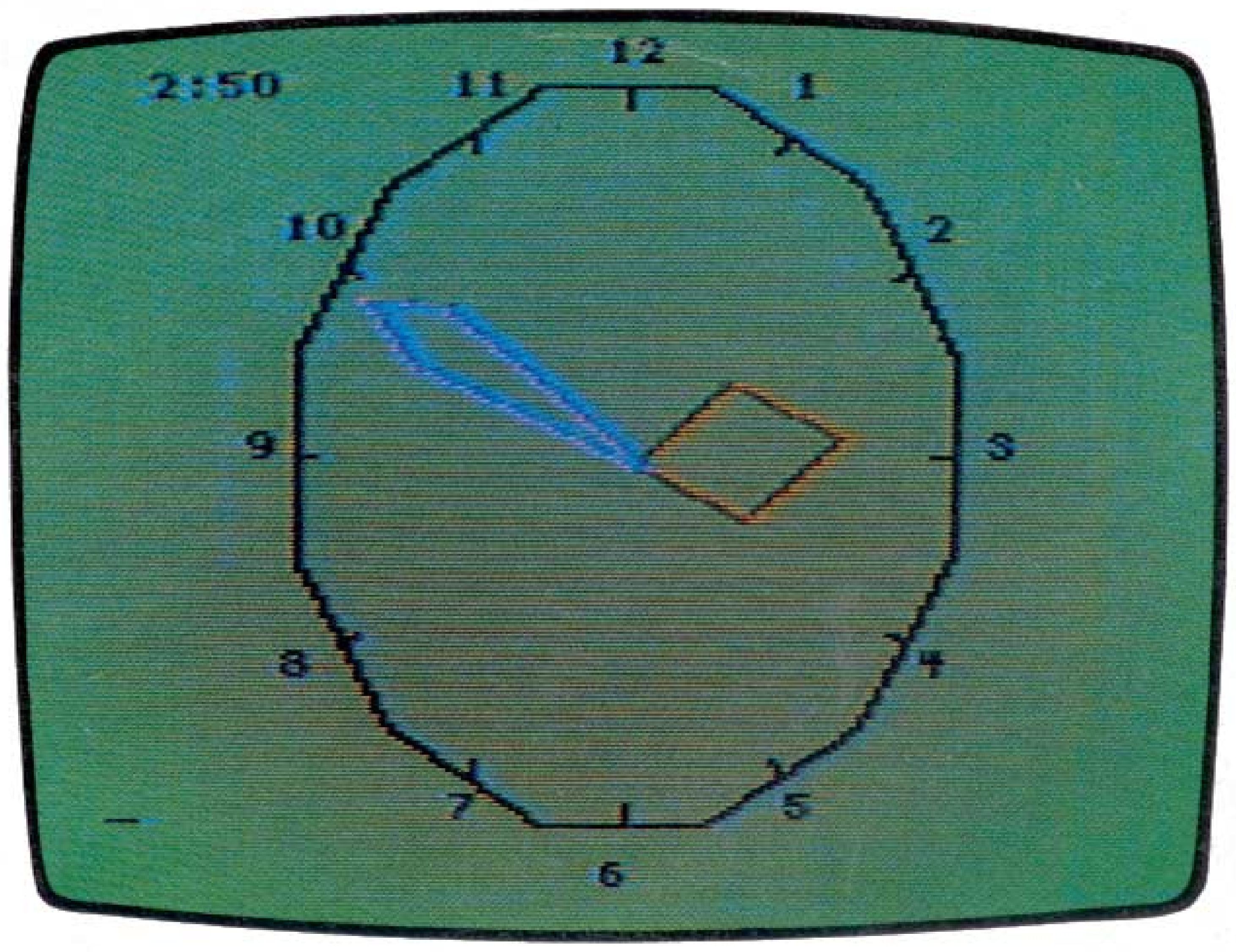
```
TO CLOCK :DELAY :INT
  CLEAR
  CLOCKFACE
  TIME :DELAY :INT
END

TO CLOCKFACE
  MAKE :NUMBER 12
  SY 18Ø  SX 1Ø4  SH 9Ø
  REPEAT 12
    (FD 22  RT 9Ø  FD 5  BK 5
     PU  BK 1Ø  PRINT :NUMBER
     FD 1Ø  PD  LT 9Ø  FD 22
     RT 3Ø
     MAKE :NUMBER  :NUMBER+1
     IF :NUMBER > 12
       (MAKE :NUMBER 1))
END
```



```
TO TIME :DELAY :INTERVAL
  HT
  REPEAT 24
    (MAKE :HR Ø
     WHILE :HR<12
      (DIGITAL :HR :MIN
       PC 1 LITTLEHAND :HR :MIN
       PC 2 BIGHAND :MIN
       REPEAT :DELAY ()
       PC 3  LITTLEHAND :HR :MIN
       BIGHAND :MIN
       MAKE :MIN
            :MIN + :INTERVAL)
     MAKE :HR :HR + 1))
END

TO BIGHAND :MINUTE
  SX 128  SY 96  SH 6*:MINUTE
  LT 8  FD 6Ø  RT 3Ø  FD 18
  RT 13Ø  FD 18  RT 32  FD 6Ø
END

TO LITTLEHAND :HOUR :MINUTE
  SX 128  SY 96
  SH 3Ø*:HOUR + :MINUTE/2
  LT 32  FD 3Ø  RT 6Ø  FD 3Ø
  RT 12Ø  FD 3Ø  RT 6Ø  FD 3Ø
END

TO DIGITAL :HOUR :MINUTES
  SX Ø  SY 18Ø  PRINT "       "
  SX 8*(:HOUR<=9 & :HOUR<>Ø)
  IF :HOUR (PRINT :HOUR)
  ELSE (PRINT 12)
  SX 16  PRINT ":"  SX 24
  IF :MINUTES<1Ø
    (PRINT "Ø"  SX 32)
  PRINT :MINUTES
END
```

Notice that you can set the interval to any number of clock minutes and that you can set the speed with :DELAY. Try running

```
CLOCK 3ØØ 5
```



# Call or Visit the Radio Shack Computer Center Near You for More Information

**ALABAMA**

**BIRMINGHAM** 2428 Green Springs Hwy., (205) 945-0792
**HUNTSVILLE** 1400 N. Memorial Pkwy., (205) 536-1581
**MOBILE** 405 Bel-Air Blvd., (205) 471-1617
**MONTGOMERY** #24 Union Square S/C, (205) 271-1500

**ARIZONA**

**PHOENIX** 10233 Metro Pkwy. E., (602) 861-1124, 4301 N. 7th Ave., (602) 277-3031
**SCOTTSDALE** 2525 N. Scottsdale Rd., (602) 990-2241
**TEMPE** 83 E. Broadway, (602) 894-2065
**TUCSON** 5622 E. Broadway, (602) 748-0101; Campbell Plaza, 2830 N. Campbell Ave., (602) 881-8003

**ARKANSAS**

**LITTLE ROCK** Town & Country S/C, University & Asher, (501) 568-5694

**CALIFORNIA**

**ANAHEIM** 509 Katella, (714) 776-9540
**BAKERSFIELD** 2018 Chester Ave., (805) 323-8074
**BERKELEY** 1922 Grove St., (415) 848-9170
**BEVERLY HILLS** 8500 Wilshire Blvd., (213) 659-8870
**CANOGA PARK** 8371 Topanga Canyon, (213) 347-9800
**CARMICHAEL** 6305 Fairoaks Blvd., (916) 484-6815
**CHICO** 1834 Mangrove Ave., (916) 894-0101
**CHULA VISTA** 1201 3rd Ave., (714) 420-3810
**CITRUS HEIGHTS** 7405 Green Back Ln. at San Juan, (916) 723-5183, 969-8752
**DOWNEY** 8031 Florence Ave., (213) 927-7882
**ESCONDIDO** 347 W. Mission Ave., (714) 741-6032
**FREMONT** Fremont Hub, 39114 Fremont, (408) 790-1112
**FRESNO** Princeton S/C, 2721 N. Blackstone Ave., (209) 225-5551
**GARDEN GROVE** 12821 Knott St., (714) 891-1160
**GLENDALE** 236 N. Brand Blvd., (213) 246-9310
**HAYWARD** 24784 Hesperian Blvd., (415) 887-6420
**HOLLYWOOD** 6922 Hollywood Blvd., (213) 466-8681
**LAKEWOOD** 5830 Lakewood Blvd., (213) 920-9671
**LA MESA** 5346 Jackson Dr., (714) 460-3610
**LONG BEACH** 2119 Bellflower Blvd., (213) 597-3377
**LOS ANGELES** 740 S. Olive St., (213) 629-2455
**MONTCLAIR** 5237 Arrow Hwy., (714) 625-1541
**MONTEREY** 484 Washington St., (408) 375-8430
**MOUNTAIN VIEW** 1933 El Camino Real W., (415) 961-0542
**OAKLAND** 1733 Broadway, (415) 763-3183
**PASADENA** 575 S. Lake Ave., (213) 449-5424
**RIVERSIDE** 3844 La Sierra Ave., (714) 689-0340
**SACRAMENTO** 4749 J. St., (916) 454-3287
**SAN BERNARDINO** 764 Inland Center Dr., (714) 884-6871
**SAN DIEGO** 3062 Clairemont Dr., (714) 276-6050; 3902 El Cajon Blvd., (714) 280-0227
**SAN FRANCISCO** One Market Place, (415) 777-9810; 2920 Geary Blvd., (415) 221-8716
**SAN JOSE** 1228 S. Bascom Ave., (408) 297-2603
**SAN MATEO** 3180 Campus Dr., (415) 573-8607
**SANTA ANA** 2320 S. Fairview St. (at Warner), (714) 966-2787
**SANTA BARBARA** 4141 State St. A-1, (805) 967-4538
**SANTA FE SPRINGS** 14138 East Firestone Blvd., (213) 921-0702
**SANTA MONICA** 511 Wilshire Blvd., (213) 458-1802
**SHERMAN OAKS** 14936 Ventura Blvd., (213) 788-2639
**STOCKTON** College Sq. S/C, 963 West March Lane, (209) 957-3676
**TARZANA** 18545 Ventura Blvd., (213) 343-1696
**TORRANCE** 3840 Sepulveda at Hawthorne, (213) 373-0306
**VENTURA** 4005 E. Main St., (805) 654-0196
**WEST COVINA** 2516 E. Workman St., (213) 915-5791

**COLORADO**

**BOULDER** Arapahoe Plaza, 3550 Arapahoe, (303) 443-7142
**COLORADO SPRINGS** 4341 N. Academy, (303) 593-7500
**DENVER** 8000 E. Quincy, (303) 770-1362; Green Briar Plaza, 7075 Pecos, (303) 428-6787
**LAKEWOOD** 2099 Wadsworth Blvd., (303) 232-6277

**CONNECTICUT**

**EAST HAVEN** 51 Frontage Rd., (203) 467-8864
**FAIRFIELD** 1196 Kings Hwy. & Rt. 1, (203) 255-6099
**HARTFORD** The Richardson Bldg., 942 Main St., (203) 549-5717
**MANCHESTER** 228 Spencer St., (203) 649-8210
**NORWALK** Rt. 7-345 Main Ave., (203) 846-3418
**ORANGE** Woolco S/C, 538 Boston Post Rd., (203) 795-1291
**WATERBURY** 105 Bank St., (203) 573-8800
**WATERFORD** 122 Boston Port Rd., (203) 443-0716
**WEST HARTFORD** 39 S. Main St., (203) 523-4283

**DELAWARE**

**DOVER** Edgehill Shop. Ctr., Rt. #13, (302) 734-5223
**WILMINGTON** 3847 Kirkwood Hwy., (302) 999-0193

**DISTRICT OF COLUMBIA**

**WASHINGTON** 1800 M St. NW., (202) 822-3933

**FLORIDA**

**ALTAMONTE SPRINGS** 766 B. East Altamonte Dr., (305) 339-1313
**CLEARWATER** 2460-D US 19 North, (813) 797-3223
**DAYTONA BEACH** Volusia Plaza, (904) 255-5909
**FT. LAUDERDALE** 4368 N. Federal Hwy., (305) 772-6797
**FORT MYERS** Edison Mall, (813) 939-3454
**GAINESVILLE** 3315 Archer Rd., (904) 375-7063
**HOLLYWOOD** 429 S. State Rd. #7, (305) 966-4382
**JACKSONVILLE** 8252 Arlington Exprwy., (904) 725-2594
**LAUDERDALE LAKES** 4317-25 N. State Rd. 7, (305) 486-2240
**MIAMI** 9459 S. Dixie Hwy., (305) 667-2316; 1601 Biscayne Blvd., (305) 374-6433; 15 SE. 2nd Ave., (305) 374-7310; 20761 S. Dixie Hwy., (305) 238-2518
**N. MIAMI BEACH** The Promenade, 1777 N.E. 163rd St., (305) 940-6887
**ORLANDO** 1238 E. Colonial Dr., (305) 894-0570
**ST. PETERSBURG** 3451 66th St. N., (813) 381-2366
**SARASOTA** 5251 S. Tamiami Tr. (Hwy.41), (813) 923-4721
**TALLAHASSEE** 2529 S. Adams, (904) 222-4440
**TAMPA** 4555 W. Kennedy, (813) 879-7470; 1825 E. Fowler Ave., (813) 971-1130
**W. PALM BEACH** 2271-A Palm Beach Lakes Blvd., (305) 683-3100

**GEORGIA**

**AUGUSTA** 3435 Wrightsboro Rd., (404) 738-5998
**ATLANTA** 2108 Henderson Mill, (404) 939-9888; 49 W. Paces Ferry, (404) 231-9604; Akers Mill S/C, 2937 Cobb Parkway NW., (404) 955-5235; 113 Peachtree St., (404) 223-5904
**COLLEGE PARK** 5309 Old National Hwy., (404) 761-3055
**DORAVILLE** 5697 Buford Hwy., (404) 458-2691
**SAVANNAH** Chatham Plaza, 7805 Abercorn St., (912) 355-6074

**IDAHO**

**BOISE** 691 S. Capitol Blvd., (208) 344-5450

**ILLINOIS**

**AURORA** 890 North Lake St., (312) 844-2224
**CHICAGO** 4355 S. Archer Ave., (312) 376-7617; CNA Plaza, 309 S. Wabash, (312) 922-0536; 72 West Adams, (312) 236-7262
**ELMWOOD PARK** 7212 W. Grand Ave., (312) 452-7500
**FAIRVIEW HEIGHTS** #4 Market Place, (618) 398-6410
**HOMEWOOD/GLENWOOD** 329 Glenwood Lansing, (312) 758-0440
**LaGRANGE** One S. LaGrange Rd., (312) 482-3484
**LIBERTYVILLE** 1350 S. Milwaukee Ave., (312) 367-8230
**LOMBARD** 4 Yorktown Center, (312) 629-5350
**NILES** 8349 Golf Rd., (312) 470-0670
**OAK LAWN** 4815 W. 95th St., (312) 425-9130
**PEORIA** 4125 N. Sheridan Rd., (309) 685-7056
**ROCKFORD** North Town S/C, 3600 N. Main St., (815) 282-1001
**SCHAUMBURG** 651 Mall Dr., (312) 884-8600
**SPRINGFIELD** Sherwood Plaza, 2478 Wabash, (217) 787-3066

**INDIANA**

**EVANSVILLE** 431 Diamond Ave., (812) 426-1715
**FT. WAYNE** 747 Northcrest S/C, (219) 482-9547
**GRIFFITH** 208 W" Ridge Rd., (219) 838-3000
**INDIANAPOLIS** 6242 E. 82nd St., Castleton Plz., (317) 849-6896; Speedway Plaza, 6129 B Crawfordsville, (317) 244-2221; 10013 E. Washington St., (317) 898-4887
**TERRE HAUTE** 3460 U.S. Hwy. 41 S., (812) 234-3212

**IOWA**

**CEDAR RAPIDS** 111 First Ave., S.E. (Downtown), (319) 362-7399
**DAVENPORT** 616 E. Kimberly Rd., (319) 386-3457
**DES MOINES** 7660 Hickman Rd., Sherwood Forest S/M, (515) 270-0193

**KANSAS**

**OVERLAND PARK** 8619 W. 95th, (913) 642-1301
**TOPEKA** White Lakes Plaza, West Tower, 3715 Plaza Dr., (913) 267-6420
**WICHITA** 2732 Blvd. Plaza S/C, (316) 681-1212

**KENTUCKY**

**FLORENCE** 7727 Mall Rd., (606) 371-2811
**LEXINGTON** 2909 Richmond Rd., (606) 269-7321
**LOUISVILLE** 2900 Taylorsville Rd., (502) 459-9901; Louisville Galleria, (502) 589-1013

**LOUISIANA**

**BATON ROUGE** 7007 Florida Blvd., (504) 928-5260
**HOUMA** 2348 W. Park Ave. (Hwy. 24), (504) 873-7713
**LAFAYETTE** University Square at Congress Blvd., (318) 235-6177
**METAIRIE** 3750 Veterans Hwy., (504) 454-3681
**NEW ORLEANS** 327 St. Charles Ave., (504) 523-6408
**SHREVEPORT** 1545 Line Ave., (318) 221-5125

**MAINE**

**BANGOR** Maine Square, (207) 945-6491

**MARYLAND**

**BALTIMORE** 7942 Belair Rd., Putty Hill Plaza, (301) 882-9583; 115 N. Charles St. at Lexington, (301) 539-7251
**BETHESDA** 7900 Wisconsin Ave., (301) 656-0892
**CATONSVILLE** One Mile West S/C, 6600 B Balt. Nat'l. Pike, (301) 788-3277
**FREDERICK** Shoppers World, Rt. 40W, (301) 695-8440
**NEW CARROLLTON-LANHAM** 7949 Annapolis Rd., (301) 459-8030
**PASADENA** 8120 Ritchie Hwy., (301) 544-2352
**ROCKVILLE** Congressional Plaza, 1673 Rockville Pike, (301) 984-0424
**SALISBURY** Shoppers World S/C, Rt. 50, (301) 546-9223
**TEMPLE HILLS** 4520 St. Barnabas Rd., (301) 899-0740
**TOWSON-LUTHERVILLE** Yorktown S/C York Rd. at Ridgley Rd., (301) 561-2001

**MASSACHUSETTS**

**BOSTON** 730 Commonwealth Ave., (617) 739-1704; 111 Summer St., (617) 542-0361
**BRAINTREE** South Shore Plaza, 250 Granite St., (617) 848-9290
**BROCKTON** 675 Belmont, (617) 583-2270
**BURLINGTON** Crossroads Plaza, Rt. 3 S., (617) 229-2850
**CAMBRIDGE** Harvard Square, 28 Boylston St., (617) 354-7694
**CHESTNUT HILL** 200 Boylston St., (617) 969-2031
**NATICK** 1400 Worcester Rd., (617) 875-8721
**SAUGUS** 343 Broadway, (617) 233-4985
**SPRINGFIELD** 1985 Main St., Northgate Plz., (413) 732-4745
**WORCESTER** Lincoln Plaza, (617) 852-8844

**MICHIGAN**

**ANN ARBOR** 2515 Jackson Rd., (313) 761-6266
**BIRMINGHAM** 3620 W. Maple Rd., (313) 647-2151
**DETROIT DWNTN** 1559 Woodward Ave., (313) 961-6855
**FLINT** G3298 Miller Rd., Yorkshire Plaza, (313) 732-2530
**GRAND RAPIDS** 3142 28th St. SE., (616) 957-2040
**KALAMAZOO** 25 Kalamazoo Center, (616) 343-0780
**LANSING** 2519 S. Cedar St., (517) 372-1120
**PLAINFIELD** North Kent Mall, (616) 364-7558
**LIVONIA** 33470 W. 7 Mile Rd., (313) 476-6800
**ROSEVILLE** 31873 Gratiot Ave., (313) 296-6210
**SOUTHFIELD** 17651 West 12 Mile Rd., (313) 569-1027
**TROY** Oakland Plaza, 322 John R. Rd., (313) 585-3900

**MINNESOTA**

**BLOOMINGTON** 10566 France Ave. S., (612) 884-1641
**GOLDEN VALLEY** Golden Valley S/C, 8016 Olson Memorial Hwy., (612) 542-8471
**ST. PAUL** 6th & Wabasha, (612) 291-7230

**MISSISSIPPI**

**JACKSON** 979 Ellis Ave., (601) 352-5001

**MISSOURI**

**DES-PERES** 11960 Manchester Rd., (314) 965-5911
**FLORISSANT** 47 Florissant Oaks S/C, (314) 921-7722
**INDEPENDENCE** 1325 S. Noland Rd., (816) 254-3701
**KANSAS CITY** 4025 N. Oak Trafficway, (816) 455-3381
**ST. ANN** 10472 St. Charles Rock Rd., (314) 428-1400
**SPRINGFIELD** 2684 S. Glenstone, (417) 883-4320

**NEBRASKA**

**OMAHA** 3006 Dodge St., (402) 346-4003

**NEVADA**

**LAS VEGAS** Commercial Center, 953 E. Sahara #31-B, (702) 731-3956
**RENO** 3328 Kietzke Lane, (702) 826-6327

**NEW HAMPSHIRE**

**MANCHESTER** Hampshire Plaza, 1000 Elm St., (603) 625-4040
**NASHUA** 429 Amherst St., Rt. 101A, (603) 881-8588

**Radio Shack**

TRS-80 Microcomputer News
P.O. Box 2910
Fort Worth, Texas 76113-2910

FIRST CLASS
U.S. POSTAGE
PAID
DALLAS, TX
PERMIT 2180

**ADDRESS CHANGE**

□ **Remove from list** □ **Change as shown**

Please detach address label and mail to address shown above

**NEW JERSEY**

**BRIDGEWATER** 1472 U.S. Highway 22 East, (201) 469-3232
**E. BRUNSWICK** 595 A Rt. 18, (201) 238-7142
**E. HANOVER** Rt. 10, Hanover Plaza, (201) 884-1200
**LAWRENCEVILLE** Rt. 1 & Texas Ave., (609) 771-8113
**NEWARK** 595 Broad, (201) 622-1339
**NORTHFIELD** 322-24 Tilton Rd., (609) 645-7676
**PARAMUS** 175 Rt. 17 S., (201) 262-1920
**SPRINGFIELD** Rt. #22 Center Isle, (201) 467-9827
**VOORHEES** 35 Eagle Plaza, (609) 346-0600

**NEW MEXICO**

**ALBUQUERQUE** 2108 San Mateo NE., (505) 265-9587

**NEW YORK**

**ALBANY** Shoppers Pk., Wolf Rd., (518) 459-5527
**BAYSHORE** 1751 Sunrise Hwy., (516) 666-1800
**BETHPAGE** 422 N. Wantagh Ave., (516) 822-6403
**BROOKLYN** 516 86th St., (212) 238-3576
**BUFFALO** 839 Niagara Falls Blvd., (716) 837-2590
**FRESH MEADOWS** 187-12 Horace Harding Exp., (212) 454-1075
**JOHNSON CITY** Giant Shopping Center, Harry L. Drive, (607) 729-6312
**KINGSTON** Kings Mall, Rt. 9W, (914) 336-6262
**MELVILLE** TSS Mall, Rt. 110, (516) 673-4646
**NEWBURGH** Zayre Plaza, Rt. #17K, (914) 561-2960
**NEW ROCHELLE** 242 North Ave., (914) 636-0700
**NEW YORK** 385 Fifth Ave., (212) 889-1345; 139 E. 42nd St., (212) 953-6053; 19 W. 23rd St., (212) 691-1861; 347 Madison Ave., (212) 867-8650
**NIAGARA FALLS** Pine Plaza, 8351 Niagara Falls Blvd., (716) 283-2041
**REGO PARK** 97-77 Queens Blvd., (212) 897-5200
**ROCHESTER** 3000 Winton Rd., (716) 244-6400
**SCARSDALE** 365 Central Park Ave., (914) 472-2520
**SPRING VALLEY** White House Center, 88 W. Rt. 59, (914) 425-2828
**STATEN ISLAND** 2409 Richmond Ave., (212) 698-3100
**SYRACUSE** 2544 Erie Blvd., (315) 446-3017; Hotel Syracuse, 510 S. Warren St., (315) 471-6663
**UTICA** Riverside Mall, (315) 735-1933

**NORTH CAROLINA**

**CHARLOTTE** 3732 Independence Blvd., (704) 535-6320; Tyvola Mall, 5401 South Blvd., (704) 527-3373
**FAYETTEVILLE** Eutaw Shopping Center, 815 Elm St., (919) 483-2418
**GREENSBORO** 3718 High Point Rd., (919) 294-5529
**RALEIGH** Townridge Sq., Hwy. 70 W., (919) 781-9380
**WINSTON-SALEM** 629 Peters Creek Pkwy., (919) 722-0030

**OHIO**

**AKRON** Fairlawn Plaza, 2727 W. Market St., (216) 836-9303
**BEDFORD HEIGHTS** 5217 Northfield Rd., (216) 662-2477
**CANTON** 5248 Dressler Rd. NW., (216) 494-7230; Mellet Plaza, 3826 W. Tuscarawas, (216) 478-1878
**CENTERVILLE** 2026 Miamisburg-Centerville Rd., (513) 435-5167
**CINCINNATI** 9725 Montgomery, (513) 793-8688; 16-18 Convention Way (on Skywalk), (513) 381-4664
**CLEVELAND** 419 Euclid (Dwntwn), (216) 575-0800; 27561 Euclid Ave., (216) 289-6823
**COLUMBUS** 862 S. Hamilton, Great Eastern S/C, (614) 864-2806; The Patio Shop. Ctr., 4661 Karl Rd., (614) 436-4666; 400 N. High St., (614) 464-2781
**DAYTON** Northwest Plaza, 3279 West Siebenthaler, (513) 277-6500
**FAIRFIELD** 7255 Dixie Hwy. (1/4 Mi. North of I-275), (513) 874-5994
**NORTH OLMSTED** Great Northern S/C, (216) 734-2255
**PARMA** 7551 W. Ridgewood Dr., (216) 842-4030
**TOLEDO** 5844 W. Central Ave., (419) 531-5797
**YOUNGSTOWN** Union Square Plaza, 2543 Belmont Ave., (216) 744-4541

**OKLAHOMA**

**OKLAHOMA CITY** 4732 SE 29th St., (405) 670-4561; Springdale S/C, 4469 NW 50th, (405) 943-8712, 1101 SW 59th St., (405) 634-2406
**TULSA** 7218 & 7220 E. 41st St., (918) 663-2190

**OREGON**

**EUGENE** 390 Coburg Rd., (503) 687-0082
**PORTLAND** 7463 SW Barbur Blvd., (503) 246-1157; 9131 SE Powell, (503) 777-2223
**SALEM** Salem Plaza, 403 Center, (503) 588-7095

**PENNSYLVANIA**

**ALLENTOWN** Crest Plaza S/C, Cedar Crest Blvd. US 22, (215) 395-7155
**BALA CYNWYD** 67 E. City Line Ave., (215) 668-9950
**ERIE** 5755 Peach St., (814) 868-5541
**HARRISBURG** Union Deposit Mall, Union Deposit Rd. #17, (717) 564-6753
**LANCASTER** Park City Plaza, US 30, (717) 393-5817
**MONROEVILLE** 3828 Wm. Penn. Hwy., (412) 823-3400
**MONTGOMERYVILLE** Airport Sq., Rt. 309, (215) 362-1200
**PHILADELPHIA** 7542 Castor Ave., (215) 342-2217; 1002 Chestnut St., (215) 923-3080; 1801 Market St., 10 Penn Center, (215) 568-0901
**PITTSBURGH** 5775 Baptist Rd., Hills Plaza, (412) 831-9694, 303 Smithfield St., (412) 391-3150
**SCRANTON** 206 Meadow Ave., (717) 348-1801
**WYOMISSING** Berkshire Mall West, 1101 Woodland Rd., (215) 372-8610

**PUERTO RICO**

**HATO REY** 243 Franklin D. Roosevelt Ave., (809) 759-8248

**RHODE ISLAND**

**E. PROVIDENCE** 850 Waterman Ave., (401) 438-2860
**PROVIDENCE** 177 Union St., (401) 831-0320

**SOUTH CAROLINA**

**COLUMBIA** Old Sears Bldg., 1001 Harden St., (803) 799-2065
**GREENVILLE** N. Hills S/C, (803) 292-1835
**N. CHARLESTON** 5900 Rivers Ave., (803) 747-5580

**TENNESSEE**

**CHATTANOOGA** 636 Northgate Mall, (615) 870-1366
**JOHNSON CITY** Peerless Center, (615) 282-6829
**KNOXVILLE** Cedar Bluff S/C, 9123 Executive Park Dr., (615) 690-0520
**MEMPHIS** 4665 American Way, (901) 795-4963, 1997 Union Ave., (901) 278-7935
**NASHVILLE** 2115 Franklin Pike, (615) 298-5484; Rivergate Plaza, (615) 859-3414

**TEXAS**

**AMARILLO** Wellington Sq. S/C, 1619 S. Kentucky, (806) 358-4567
**ARLINGTON** 2500 E. Randol Mill, Suite 113, (817) 274-3127
**AUSTIN** 8764 E. Research Blvd., (512) 459-4238, Southwood Mall, 1501 Ben White Blvd., (512) 447-0371
**BROWNSVILLE** 1639 Price Rd. (Hwy. 77), (512) 544-6800
**BEAUMONT** 5330 Eastex Frwy., (713) 898-7000
**CORPUS CHRISTI** 1711 S. Staple St., (512) 887-8901
**DALLAS** 15340 Dallas Pkwy., Suite 1100, (214) 934-0275, 2930 W. Northwest Hwy., (214) 350-4144; 1517 Main St., (214) 760-8601; 2588 Royal Ln., (214) 484-9947
**EL PASO** 9515 Gateway West, (915) 594-8211
**FT. WORTH** 15 One Tandy Center, (817) 335-7198; 2801 Alta Mere, (817) 738-0251
**HARLINGEN** 1514 S. Hwy 77, Sunshine Strip, (512) 425-8890
**HOUSTON** 211C-FM 1960, (713) 444-7006; 10543 Gulf Fwy., (713) 943-9310; 5900 North Fwy., (713) 699-1932; 6813 SW Fwy., (713) 777-7907; 809 Dallas St., (713) 651-3002; Holland Square Center, 10920 East Freeway, (713) 453-0600
**HURST** Northeast Mall, (817) 284-1518
**LAREDO** 102 East Calton Rd., (512) 727-4768
**LUBBOCK** 3625 34th St., (806) 793-1467
**ODESSA** 1613 ''A'' East 8th Street, (915) 334-8355
**RICHARDSON** Fleetwood Sq. S/C, 202 W. Campbell Rd., (214) 669-1494
**SAN ANTONÍO** 6018 West Ave., (512) 344-8792; 4249 Centergate, (512) 657-3958

**UTAH**

**OGDEN** K-Mart Shopping Center, 3672 Wall Ave., (801) 394-1666
**MURRAY** 6051 S. State Ave., (801) 268-8978
**SALT LAKE CITY** 415 5th Ave., (801) 322-4893

**VIRGINIA**

**ALEXANDRIA** 3425 King St. at Quaker Ln., (703) 820-2600
**ARLINGTON** Crystal City, 2301 So. Jefferson Davis Hwy., (703) 521-7479
**FAIRFAX** Westfair Center, 11027 Lee Hwy., (703) 273-6500
**LYNCHBURG** Hill's Plaza, Ward's Rd., (804) 237-1664
**NEWPORT NEWS** Newmarket South Shop. Ctr., (804) 245-8203
**NORFOLK** 5731 Poplar Hall Dr., (804) 461-0798
**RICHMOND** Willow Lawn S/C, 1617 Willow Lawn Dr., (804) 282-3453; 7728 Midlothian Turnpike, (804) 272-8803
**ROANOKE** Franklin Bldg., 3561 Franklin Rd. S.W., (703) 342-6335
**ROSSLYN** 1911 N. Ft. Myer Dr. at Rt. 29, (703) 243-5955

**WASHINGTON**

**BELLEVUE** Crossroads Mall, North East 8th & 156 St., (206) 644-1804
**BELLINGHAM** 1111 Cornwall Ave., Suite B & C, (206) 676-0811
**FEDERAL WAY** 33505 Pacific Hwy. South, (206) 838-6830
**SEATTLE** 18405 Aurora Ave. N., (206) 542-6184; 1521 3rd Ave., (206) 447-1959; 5030 Roosevelt Way NE, (206) 527-1019
**SPOKANE** 7702 N. Division, (509) 484-7000; E. 12412 Sprague, (509) 922-2800
**TACOMA** 7030 S. Sprague, (206) 473-7333
**TUKWILA** 15425 53rd Ave. S., (206) 248-3710
**YAKIMA** 1111 N. First St., (509) 248-9667

**WEST VIRGINIA**

**HUNTINGTON** 2701½ 5th Ave., (304) 523-3527

**WISCONSIN**

**MADISON** 57 West Towne Mall, (608) 833-6130
**MILWAUKEE** 6450 N. 76th St., (414) 353-6790
**WEST ALLIS** 2717 South 108th St., (414) 327-4240