Radio Shack

TRS-80 Microcomputer NEWS

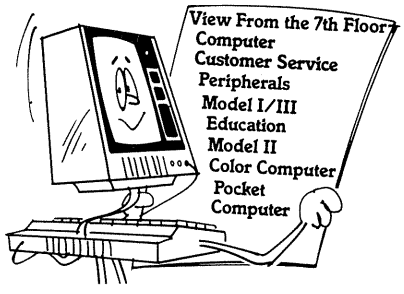
November, 1981

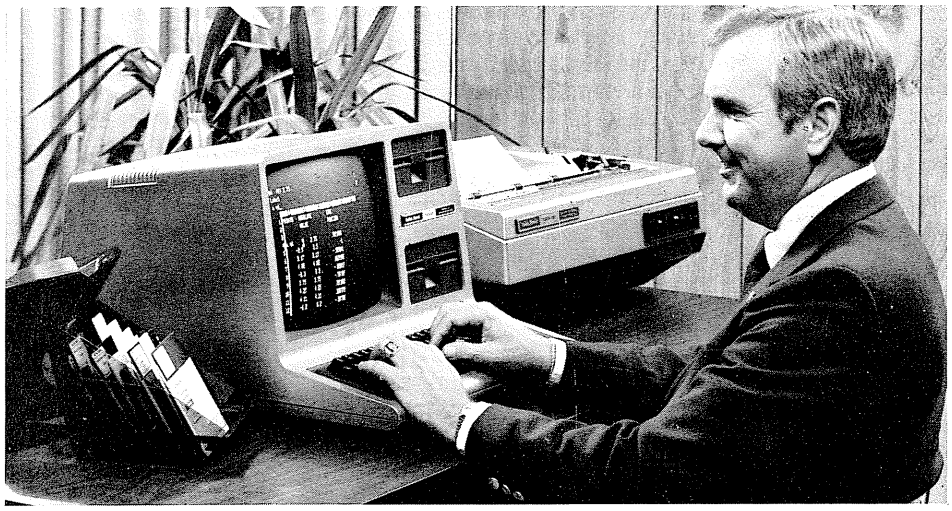
P.O. Box 2910, Fort Worth, Texas 76113-2910

THE MICROCOMPUTER NEWSLETTER PUBLISHED FOR TRS-80 OWNERS

Volume 3, Issue 11

NEWSLETTER INDEX IN THIS ISSUE





TRS-80 Applications—City of Davis

What kind of political future does a TRS-80 computer have in city government? Recently the city of Davis, California tried out a Model III in their day to day operations where it proved itself to be an extremely useful and versatile tool.

The Model III as a Word Processor

A spokesperson for the city relates: "Using the TRS-80 has been an exciting opportunity and challenge. The Scripsit program has been an invaluable aid to our office. We write several policy memos every month and the ability to change and rewrite sections while leaving the rest of the document intact has saved hundreds of hours of secretarial time. Two of the major projects have been an energy element of Davis' General Plan and a Solar Access Ordinance. As you know, the precise wording of documents like these is critical and the word processor's ability to change and rearrange is fantastic—I'm surprised anyone in the legal profession can do without one.

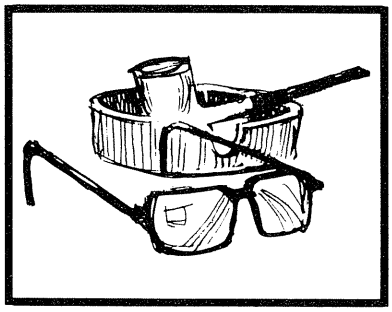
"Another subtle advantage is that of being able to sit down and go from concept to finished copy which avoids breaking the continuity of thought. Not having to go through all the steps of getting work to a secretary, correcting typos, additions, and retyping has improved document quality. In addition, I feel Scripsit has increased our productivity. Scripsit's ability to make corrections neatly on the screen in front of you has also made rewrites much easier.

"The Scripsit program itself has performed admirably. I have used other word processing systems before, including one which cost over $12,000, and Scripsit compares well to any of them. Several people in our office who had never used a word processing system listened to your tape sessions and became fairly proficient by the end of the lessons. I congratulate you on a fine software program."

VisiCalc

"As with Scripsit, I have also found Visicalc to be an extremely useful program. Its chief assets seem to be flexibility and power. It has been portrayed as an "accounting oriented" program, but it is really much more versatile than that. It is an excellent tool for making managerial decisions by summarizing data and making projections.

(Continued on Page 39)



View From the 7th Floor

by Jon Shirley, Vice President Computer Merchandising

For the last two months I have been on the rubber chicken circuit. It included talking to every Radio Shack store, computer center and computer department manager plus some of our dealers. In addition I occasionally talk before various industry groups and I also participated in our introduction of ARCNET in New York in September (see story about ARCNET elsewhere in this issue). So when I had to write this month's column, I was pretty well out of things to say, and was searching desperately for inspiration, especially since I was way behind the deadline. I even toyed with a blank page saying I was on vacation in Africa or that I had acquired some strange new malady. But, as I started to catch up with the three foot high stack of unread magazines that accumulated these last two months, I found my inspiration.

Don't believe everything you read in the press—revisited.

That's right gang, its rumor time again. The September Byte provided the initial force in a column called BYTELINES titled "News and Speculation About Personal Computing." Now before I start, please do not get me wrong. I have no gripes about Byte except perhaps that it has gotten about as big as a small city telephone directory and won't fit in my briefcase. I have been on a few panels with Chris Morgan, BYTE editor in chief, and he's a great guy. I have never met Sol Libes who writes BYTELINES so I can't comment on the man but I suggest he retitle his column to leave out the word news.

Anyway, this September issue BYTELINES starts off "Peering into Radio Shack's Crystal Ball." Item 1, the Model II will be offered with an optional 10 megabyte hard disk with a 15k bpi tape backup for less than $5000. Not bad, there is no tape backup, but we did indeed announce our hard disk in August but it has not been a secret since last year. Item 2, we are working on a multiprocessing and -programming system that will allow up to 16 Model IIs to be chained together. I don't even know what that means, but perhaps it is an as yet unreleased item called the Network 3 that will let a Model II or III host up to 16 Model IIIs via RS-232 for use in the classroom. Again not too bad.

Item 3, a new system, Model IV?, will be announced between the Model III and II, with 8 inch drives, not CP/M compatible and base price $1500-$1800. Sorry Sol, that is a 100% miss, not any part of it is true. Item 4, software, we will introduce vertical market software, and a package to read and write IBM format disks . . . hardly crystal ball as we introduced the stuff at the NCC show. Also on software, we will have upgraded versions of Model I, II, III TRSDOS—which is as safe a prediction as saying it will snow in Fargo, North Dakota this winter.

Now to be fair, Sol then said that most of our new Model III software will run on the Model I and praised us for doing that, and I forgive anything else he ever says about us just for that compliment. It is costing us a ton of time and money to produce and check those compatible versions but it is our policy not to forget the Model I owners and if you read our new RSC-6 computer catalog I think you will find only 3 Model III only disk packages and those all required the larger disk capacity. Unfortunately the crystal ball must have rebooted when it came up with item 5 or perhaps it's RAMs were hit with cosmic rays. We are absolutely not going to rename the Computer Centers to Office Product Centers, although we will sell copiers and perhaps some other office products. Now remember, I did start this with don't believe everything you read in the press, not don't believe anything.

Now on to the September 14 edition of Computer Business News where the "Page Two Report" says we will introduce the TRS-88 which is an Intel 8088 (as used in the IBM personal computer) before Christmas. Again a 100% miss or our R & D guys are smoking pot. The neatest "news" item I hit in the stack, and I am sorry but I just can't remember where I read it, was that IBM planned the introduction of their personal computer to be the day before we held a major meeting to pump up our troops. Well they did introduce it the day before the second of our 5 divisional meetings but come on, do you know how big IBM is? To put their size in perspective they spent on R and D just $200 million less than our total $1.7 billion sales last year. Would they really time a product introduction based on one of our sales meetings???? Hard to believe, but if true I guess we should be proud.

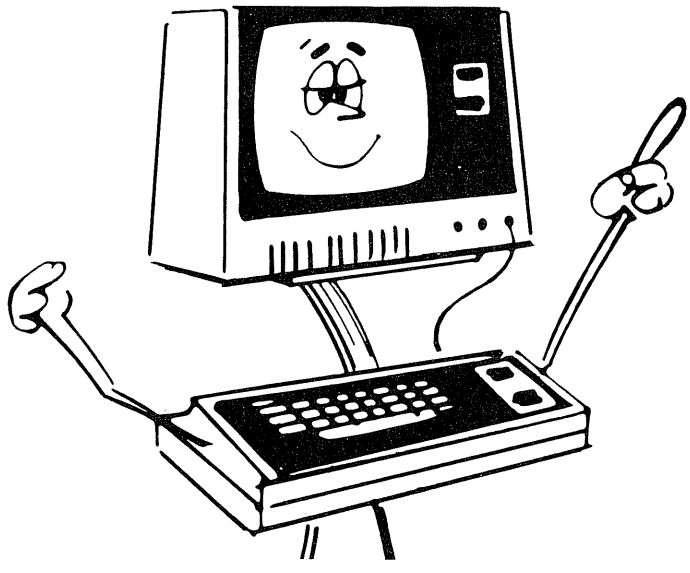
In the last few months both IBM and Xerox have entered the personal computer business which means, I guess, that the business is now legitimate. Should you run out and trade in your TRS-80? Well that is up to you, dear owner, but take a hard look before you do. The Xerox 820 has single density drives and virtually no Xerox supported software, which is very disappointing as they had just introduced the Star system which has fantastic software. An IBM Personal Computer comparable to a Model III is 50% more expensive and allows only two disk drives max. It also has a very limited software offering, none of it done by IBM.

Perhaps I am old fashioned, but I believe that good software and good software support are the keys to this business and that almost anyone can make the iron, even Maytag, if they wanted to.

And speaking of good software, you Color Computer owners really should look in that new RSC-6 Computer Catalog because we've got a lot of really neat new products. One I would like to pitch is the Color Computer Learning Lab (26-3153), a program to teach you how to write programs. If you are tired of the games, and want to learn BASIC, I think this is a great way to do it.

Would you like a look in my crystal ball? It's cloudy once in a while also, but when it's clear I see more software, more peripherals, more computers, more accessories, more service locations and more of all the good things that have made us the "biggest name in little computers ™ "

Until next month.



COMPUTER Customer Service

Computer Hints, Help and Tips

Radio Shack Computer Customer Service answers between 500 and 1000 letters monthly. In the past we have not been able to provide the timely turnaround that we would like. We feel that if we give you some guidelines in writing to us, we will be able to respond more rapidly.

WHAT TYPE OF INFORMATION CAN COMPUTER CUSTOMER SERVICE PROVIDE?

Questions relating to applications software, that we market, are our strongest suit. If you are having problems with our software, we want to know it, so that we can correct any "bugs" that might exist, or update and clarify our documentation. Radio Shack wants to provide you with the best possible software it can, and your input is both welcome and respected.

We can answer questions about our BASIC interpreters and our various compilers, especially questions about how particular functions work. It will help if you provide us with an example of the specific problem(s) that you have. If a lengthy example is needed (more than four or five short lines of code), please provide a copy of the program on tape or disk. We simply do not have time to key every customer's example into a computer to investigate a problem.

Questions about hardware should be limited to how a particular system should function. For example, "If I use a "," when inputting information, I get an "extra ignored" error. Is this normal?" (Editor's note: "Yes") We can try to help you hook up your XYZ serial printer to a Model II, but our response may be limited to telling you what inputs the Model II's serial port must have to make the print function work. Obviously, we cannot go out and buy everything on the market to see how it works with a TRS-80.

Pin outs and schematics are in our technical reference manuals and service manuals. All service manuals for our computer products are available through your local Radio Shack store.

WHAT TYPE OF INFORMATION DOES COMPUTER CUSTOMER SERVICE NOT PROVIDE?

Please do not ask us questions about undocumented addresses in ROM, TRSDOS, etc. Questions about addresses not documented in the manuals will not be answered. Radio Shack has chosen not to publish certain addresses, especially ones that are subject to change in future releases. This is done to minimize the adverse effects on any software using hard coded addressing. (We deliberately stay away from using undocumented addresses in our own software for the same reason.)

Please do not ask us to help you modify software (ours, yours, or someone else's). Radio Shack software is written to a particular design, and we test the software based upon that design. Some of our software is difficult to modify and the results can often be disastrous if the modifications are not made and tested with the utmost care.

Please do not ask us questions about Radio Shack merchandising decisions such as "Why did you add 16K more memory on the Model III's and sell them for the same price?" We may be able to confirm or deny a particular piece of information for you, but that is about the limit of our knowledge. In the case of the increased RAM in the Model III, this came about because the price of RAM keeps coming down.

Requests for owner and service manuals, books, software updates, and replacements for defective software should not be directed to computer customer services. Your local Computer Center and Radio Shack store or dealer has a procedure to follow to get these items for you. A supply of these items is not maintained by Customer Service.

Yes, occasionally, our distribution center may get behind on shipping some of these items. To help this, Radio Shack recently installed a new inventory system in our distribution center. During the installation we did get behind on the shipment of the 700 series software updates. The new system seems to be working quite well now and should help our stores to serve you better.

Problems with your computer equipment are best handled by your local store manager, service shop manager, or regional service coordinators. If you call or write us about a problem we simply contact one of these people to help solve the problem. Your local store manager can contact his service shop manager or regional service coordinator for you.

Radio Shack Computer Customer Service does not authorize refunds on equipment. Your local dealer is responsible for handling these requests. If you do not get satisfaction from the store manager, the next step is the local district manager. We are not saying that we do not want to hear from our unsatisfied customers, your information is valuable to us—we are saying that we have given our store managers the authority and responsibility to handle these problems.

If you have information about false advertising, we would like to (and want to) hear about it. Radio Shack does not intentionally advertise products it does not have nor state incorrect product specifications. From time to time a product design must be changed. For example, our RSC-5 catalog stated the direct connect modem would be a 300/600 baud unit. At the last minute the product was changed to 300 baud only, and we notified our stores and customers before the first one was shipped. The long lead times for advertising and catalogs often require that we write about a product before it is actually produced. We will always give you the best information available at a particular time, and tell you as soon as we know that a change is being made.

Radio Shack Computer Customer Service will handle your letters as promptly as possible. Of course, if you need a quicker solution or answer to a question, call us on the phone. That is what we are here for, to support you—our computer customer.

Model II Communications

We have had several inquiries about using the Model II as either a HOST computer, or a terminal. Here is some information which we think will help.

First, some definitions: a HOST computer is a computer whose memory, disk drives, and peripherals are being used by some other computer or terminal. Once the HOST computer system is turned on and the appropriate diskettes are loaded, it is not necessary that a person be at the HOST computer. All the functions of the computer can be controlled remotely.

A TERMINAL is a "device" which is able to use the memory, disk drives, and peripherals of a HOST computer to accomplish some task. Terminals generally fall into one of two classes, "dumb" which means no memory or processing ability of its own, and "smart" which means that the terminal has its own memory and microprocessor and can process information and data independent of the HOST computer.

Model II TRSDOS 2.0a comes with both a HOST program and a "smart" TERMINAL program. The following is an example of using the Model II for communication both as the HOST COMPUTER and as the TERMINAL COMPUTER.

(Continued on Page 4)

COMPUTER SERVICES (From Page 3)

The equipment we will use for the HOST computer includes the Model II 64K with R/S Direct Connect Modem I (26-1172) connected to the Channel A serial port and a terminator plug connected to the Channel B serial port. The Direct Connect Modem I will be set as originate and full duplex.

The equipment being used for the TERMINAL includes the Model II 64K with R/S Telephone Interface II (26-1171) connected to the Channel A serial port and a terminator plug connected to the Channel B serial port. The Telephone Interface II will be set as answer and full duplex.

The software will be TRSDOS 2.0a which has both a HOST program and a TERMINAL program as part of the operating system.

Please note that our choice of which modem to use and where is arbitrary. You could use the 1171 or the 1172 on either or both machines. Likewise, either machine could have been set to the Originate mode, with the other machine being set to Answer mode. The choice of full or half duplex is also arbitrary, although we generally prefer full duplex when it is available.

Once you have the hardware set up, and all of the cables plugged in correctly, we need to load the proper software.

Let's start with the HOST system.

Load TRSDOS as usual. When TRSDOS Ready is displayed, check the status of the serial ports:

```
SETCOM [ENTER]
```

The response should be:

PORT A IS OFF
PORT B IS OFF

We want to use Port A for our communications, so turn A on and initialize it to 300 Baud, 7 bits, Even parity, and 1 stop bit. Does that sound like a lot to remember? Well, since these are the default values, the command is simply:

```
SETCOM A=() [ENTER]
```

If you want to use some other settings, you can type:

```
HELP SETCOM [ENTER]
```

from TRSDOS Ready to see what the proper syntax is for the SETCOM command.

Now that we have initialized (turned on) the serial port, we are ready to turn on the HOST program:

```
HOST ON [ENTER]
```

At this point, the Host system is on, the printer should be on and selected if there is a printer. The Direct Connect Modem I is connected to PORT A, on, with Originate mode selected. We have initialized the Serial port and turned HOST on.

What is left to do? The only thing we still need to do is make the telephone connection to the TERMINAL Model II. What else can you do? Your HOST Model II is still a computer and you can still use its keyboard to enter commands, run programs, etc. But you can also go away and leave the computer alone. With few exceptions (like SCRIPSIT), once you establish the phone link to the terminal, the terminal operator will be able to access and use any of the programs (TRSDOS, BASIC, machine language, etc.) that are available on the disks which are in the Host system.

If you will be using your Model II as either a HOST or terminal very often, you may want to BUILD a DO file (we called ours TALK) then have the system automatically execute this program by AUTO DO TALK. This could initialize the serial port, turn the HOST program on and even run a BASIC program whenever you boot that particular disk.

The set-up procedure for the terminal Model II is very similar to the one for the HOST Model II. The two differences are:

1. The modem must be set to Answer if the Host uses Originate, or Originate if the Host uses Answer.

2. Instead of turning the HOST feature on, we will run the TERMINAL program.

Note that we use

```
SETCOM A=() [ENTER]
```

again. To execute the terminal program use:

```
TERMINAL [ENTER]
```

You should now have the TERMINAL MENU from which to choose what you want to do. The T option puts you directly into the interactive terminal mode where you would see the prompts displayed on your screen that the basic program from the host system is seeking. The P option turns the printer on and off so the incoming text from the host system will go to the printer and the video display. The R option deals with the RAM BUFFER which allows you to save the incoming text in memory. The C option allows you to copy the ram buffer to the disk. The D option displays whatever is currently in RAM (For more details, see your Model II Owners Manual).

You now have both the HOST and TERMINAL systems on line and you are ready to perform. One of the operators will need to place the phone call to connect the two systems. The operator of the HOST system then switches his Direct Connect Modem I to the Originate position. On the TERMINAL, switch the Telephone Interface II to Answer. As soon as you do this, you will hear a high pitched tone. Place the telephone receiver into the cradle (make sure the cord is facing the proper way). Your system should now be complete.

Anything the operator of the Terminal machine enters on his/her keyboard will be sent to the HOST machine to be executed. All the resources of the HOST system can be utilized. You could enter and store a BASIC program on the HOST systems disk, print using the HOST printer, etc.

Most frequently asked questions—Model I/III

Question 1: How can I tell if I've got the latest version of In-Memory Information?

Answer: The newest version has a new catalog number—26-1508. The previous version was catalog number 26-1502. Also, the tapes in the new version have a Model III side and a Model I side (sorry, the new version will not work with Level I). The old version tapes had a level I side and a level II side.

Question 2: Can I make partial payments on one invoice with Model I/III Accounts Payable (26-1554)?

Answer: No, this program does not allow partial payments. When paid, an invoice must be paid in full.

Question 3: When using VisiCalc, if I already have some information on the screen and load in another file, the screen is not erased—the previous information is still there. What's wrong with the program?

Answer: There's nothing wrong with the program. This is a feature designed to allow you to combine files on the screen. If you wish to clear the screen before loading a new file, use the following sequence: / C Y

Question 4: I bought the Model I/III Business Mailing List program (26-1558) with the understanding that it would handle the new nine digit Zip Codes, but the Post Office has included a dash, so now I need ten digits. What do I do?

Answer: It will be at least a year before the nine (or ten) digit zip codes will be fully implemented. By that time, we expect to release a new version of the program that will print the dash—although it will not be stored (and take up disk space).

Question 5: Why will my Accounts Receivable program (26-1555) not age the accounts properly? It always prints out statements showing nothing due, and it shows that all accounts are current!

Answer: You probably haven't given the account a minimum payment percentage. If you don't set up a minimum, the computer will default to zero, causing the account to have no minimum payment. When your customers make that payment (zero), they will be current. (Continued on Page 5)

COMPUTER SERVICES (From Page 4)

COMPUTER CUSTOMER SERVICES ADDRESS AND PHONE NUMBERS

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

Model I/III Business Software
Outside Texas 1-800-433-5641
In Texas 1-800-772-5973
Model II Business Software
Outside Texas 1-800-433-5640
In Texas 1-800-772-5972
Education Software
Outside Texas 1-800-433-1679
In Texas 1-800-772-5914
All Other Calls Related to Computers
Outside Texas 1-800-433-1679
In Texas 1-800-772-5914
Switchboard - 1-817-390-3583

What I Do

Priscilla A. Ridgway Hot Springs, Arkansas

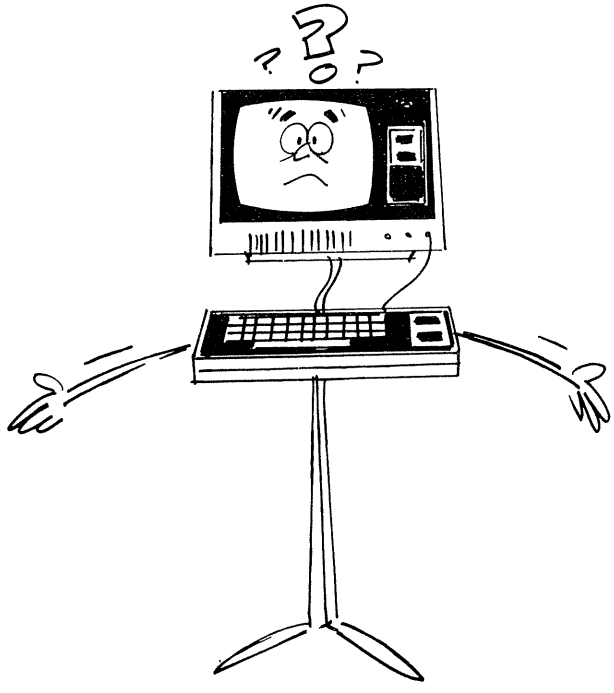
I am fourteen years old and own a Model III. The enclosed program is designed to answer the question, "What Does It Do?" I appropriately gave it the filename "WID," standing for "What I Do".

Lines 80-160 are a revised version of Les Childers' Guess Number (see June Microcomputer News, page 16).

I request that you put this program in Microcomputer News so others may use it too. I hold no rights to this program. Anyone is free to use it.

I enjoy your magazine and hope you keep up the good work.

```
5 REM PRISCILLA RIDGWAY, 1981
1Ø CLS
     : PRINT "HI! I'M A TRS-8Ø MICROCOMPUTER AND I CAN DO
     ANYTHING YOU CAN THINK UP AND PROGRAM ME TO DO!"
2Ø INPUT "WHAT IS YOUR NAME"; A$
     : PRINT "NICE TO MEET YOU "; A$
     : INPUT "WOULD YOU LIKE TO SEE SOME OF WHAT I CAN DO";
     B$
     : IF B$="YES" THEN 4Ø
3Ø PRINT "WELL, "; A$; " YOU ARE ANYWAY!"
     : GOTO 5Ø
4Ø PRINT "I'M GLAD YOU WANT TO "; A$
5Ø PRINT "FIRST I'LL SHOW YOU MY MATHEMATICAL TALENTS"
     : INPUT "GIVE ME A NUMBER"; A
6Ø PRINT "EASY NUMBER "; A$
7Ø PRINT A "+" A "=" A+A "..." A "-12=" A-12 "..." A "*" A
     "=" A*A "...327/" A "=" 327/A "...THE SQUARE ROOT OF
     "; A; "IS " SQR(A) "...NOT TOO BAD HUH "; A$;
     "?...DON'T ANSWER THAT!"
8Ø PRINT "NOW FOR A ROUSING GAME OF GUESS THE NUMBER.
     I'LL TAKE THE NUMBER YOU GAVE ME BEFORE ("A") AND MIX
     IT UP TO MAKE A NEW NUMBER..."
9Ø A=INT(A*A/12+1Ø-3Ø)
1ØØ INPUT "YOUR GUESS IS "; F
11Ø IF F=A THEN 15Ø
12Ø IF F>A THEN 14Ø
13Ø PRINT "YOUR GUESS IS LOW"
     : GOTO 1ØØ
14Ø PRINT "YOUR GUESS IS HIGH"
     : GOTO 1ØØ
15Ø CLS
     : PRINT CHR$(23) "CORRECT!!"
     : FOR I=1TO 1ØØØ
     : NEXT I
     : CLS
     : INPUT "DO YOU WANT TO PLAY AGAIN"; C$
     : IF C$="YES" THEN 16Ø ELSE 17Ø
16Ø INPUT "GIVE ME ANOTHER NUMBER, WILL YOU PLEASE"; A
     : GOTO 9Ø
17Ø CLS
     : PRINT "NOW WE SHALL ENGAGE IN A LITTLE TRUE OR FALSE
     QUIZ. PLEASE USE T OR F"
18Ø INPUT "LINCOLN WAS THE 4ØTH PRESIDENT...T/F"; D$
19Ø IF D$="T" THEN GOSUB 31Ø
2ØØ IF D$="F" THEN GOSUB 3ØØ
21Ø CLS
     : INPUT "WASHINGTON D.C. IS THE U.S. CAPITOL"; D$
22Ø IF D$="F" THEN GOSUB 31Ø
23Ø IF D$="T" THEN GOSUB 3ØØ
24Ø CLS
     : INPUT "COMPUTERS ARE AS SMART AS THEIR OWNER"; D$
25Ø IF D$="F" THEN GOSUB 31Ø
26Ø IF D$="T" THEN GOSUB 3ØØ
27Ø CLS
     : PRINT G; "OUT OF 3 AIN'T BAD"
28Ø PRINT "THE PRECEDING WAS TO GET YOU ACQUAINTED WITH
     THE TRS-8Ø MICROCOMPUTERS. OF COURSE YOU HAVEN'T
     SEEN ALL OF MY CAPACITY YET. I CAN FIGURE TAXES, DO
     BOOKKEEPING, HOOK INTO OTHER UNITS, PLUS A LOT MORE
     WITH THE PROPER PROGRAMING AND CORDS."
29Ø FOR I=1 TO 5ØØØ
     : NEXT I
     : CLS
     : PRINT CHR$(23)
     : PRINT @ 176, "BYE"
     : FOR I=1 TO 5ØØ
     : NEXT I
     : CLS
     : END
3ØØ CLS
     : PRINT CHR$(23)
     : PRINT @17Ø, "CORRECT!"
     : G=G+1
     : FOR W=1 TO 1ØØØ
     : NEXT
     : RETURN
31Ø CLS
     : PRINT CHR$(23)
     : PRINT @164, "YOU MISSED!"
     : FOR W=1 TO 1ØØØ
     : NEXT
     : RETURN
```



CompuServe

CompuServe Information Service

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.

To Insulate or Not to Insulate: The CompuServe Energy Management System

No doubt one of the most disheartening thoughts about the winter season is that homeowners will probably spend a lot more money for energy this year than they did last year. It's a puzzling problem, and often consumers have nowhere to turn for answers to the enigma of energy usage versus misuse.

But now, thanks to the CompuServe Information Service and the talents of information providers Charles Kolb and Gene McManus, CIS subscribers won't be left out in the cold. Instead, by typing R ENERGY in the Personal Computing section (main menu item 9), users can access the CompuServe Energy Management System, which was developed by Kolb and McManus to help consumers understand the home/energy conservation puzzle. The energy management system can also be used for small commercial buildings.

After keying in R ENERGY, the heading CompuServe Energy Management System will appear on the screen, followed by a list of programs in numerical order and a question mark if additional help is needed. Keying the question mark will result in a brief explanation of the function of the programs.

The system will also ask if a hardcopy terminal is being used. If the answer is yes, it will give the message to set the paper at the top of the form, so the system can print out the first program, the introduction.

The introduction explains briefly what the system will do and how it works. It will alsc ask that the user literally go out and take stock of his home. Broken down, component by component, this information will be used by the system to help analyze the energy needs of the structure.

The program will also ask that information regarding area fuel costs be entered. This data may be obtained from local utility companies. The last type of information that the user is prompted to enter comes from contractors or equipment dealers regarding the building's Energy Efficiency Ratings.

The introduction goes on to tell what the other functions in the system do. For instance, it explains that the Summary Report will give the user a statement showing estimated annual heating losses for all components of the structure. Should the user request a Summary Report, a message appears informing the user that there is a charge for production of the report. However, before a Summary Report may be generated, program two, Data Collection, must be completed.

The program Data Collection asks that a project ID name be assigned to the new set of information about to be entered. Once this is completed, questions about the structure of the house appear on the screen, beginning with those concerning the exterior walls. Should the user, at any time, not understand the question fully, or some of the terminology used within the question, he need only enter a question mark to receive an explanation.

The program will ask the makeup of each wall, component by component, and will provide a list of possible components, such as aluminum siding, fiberglass batts, drywall, etc. It will then give an approximate R-value to the wall, which will help to analyze the energy efficiency of the home later in the program.

The program repeats these kinds of questions for windows, doors, ceilings, roofs—just about all variable components of an individual home. For those users who don't know dimensions in square feet, feet and inch measurements can be used by keying ENTER at the questions prompt; the program will ask individual dimensions and then calculate the area in square feet automatically.

The program also takes into account air infiltration, which it asks for in terms of number of air changes per hour, the type of fuel used to heat the structure, the fuel cost per hour (which may be obtained by calling the local utility), the type of heating system, the color of the roof, and the air changes per hour in the attic. Again, should the user not understand these questions, he needs only to enter a question mark to receive an explanation.

After all the data is collected, the user keys program three, Data Analysis, and the program goes to work to analyze his home. It will ask for the average daytime and nighttime winter thermostat settings, the number of hours for the nighttime setting, the estimated annual rate of inflation and the estimated annual increase in fuel cost.

In a matter of just a few seconds, the user receives an explanation, by component, of heat loss (expressed in BTUs per hour), the cost per year of each component, the dollars that would be necessary to make improvements and the payoff, in terms of years, of those improvements. In other words, the program allows the user to "suppose" he installed storm windows, or extra fiberglass batts, and give the approximate amount of time, in years, it will take to recover that cost.

For a more detailed analysis of an individual component, the user enters the number of the component, which will instruct the system to display information in that manner.

So there you have the basic facts of the Energy Management System. It's easy to see then, that the system really does have the capability to determine when a user reaches the crossover point; where dollars spent to reduce energy usage result in even greater savings in energy costs. He will be able to look at the investments that were impractical at a time of lower energy costs that have today become money-savers due to ever-climbing prices.

Also within the program is the provision to "edit" or modify the structure to find out what the approximate cost and effectiveness of the modification might be without actually hiring a contractor. Perhaps the user might call these two versions of his home HOUSE.OLD and HOUSE.NEW. He may then compare the relative costs of the two.

One interesting feature of the program is its ability to priortize energy improvements for the user. In other words, it gives information which will help the homeowner decide which areas of his home need improvements most.

And, much to the relief of the average homeowner, this program explains some of the energy industry buzzwords that may be confusing and often lead consumers down the expensive path of unnecessary home improvements. When a user doesn't understand what a BTU is, for example, he only needs to key in the question mark, and the resulting explanation will give him the answer.

(Continued on Page 7)

CompuServe (From Page 6)

The Micro Advisor Has the Answers

Are there questions about personal computing you'd like to ask but just don't know where to find the answer?

Then CompuServe's Micro Advisor may be for you. A new offering on CompuServe, the Micro Advisor is a clearinghouse for information about microcomputers and personal computing.

For both the experienced and novice user, the Micro Advisor provides details on equipment use and selection, peripheral devices and software, to name just a few areas.

The Micro Advisor also lists current information on computer clubs throughout the various regions of the country, presents industry magazines and other publications for user groups and gathers the latest news on developments in the microcomputing industry.

The Micro Advisor will answer specific questions of yours through an interactive feedback service.

Read the Micro Advisor by accessing main menu item 6, Special Services, on CompuServe.

CompuServe in Canada

The CompuServe Information Service is now available to owners of personal computers and computer terminals through Radio Shack stores in Canada.

Using a personal computer or terminal, a telephone and a modem, subscribers in Canada can now access CompuServe through the Datapac telecommunications network.

Canadian customers can take advantage of the full range of services on CompuServe including the latest news from major newspapers and a national U.S. wire service, corporate stock and commodities information, electronic mail and real-time communications, computer games, family information and computing power for programming activities.

Questions and comments about the CompuServe Information Service can be sent to Richard A. Baker, Editorial Director, CompuServe Information Service, 5000 Arlington Centre Blvd., Columbus, Ohio 43220 or through Feedback, main menu item 5, CompuServe User Information.

Binary Search

Dan Belemecich Jamestown, California
R. Lawrence Clark Atlanta, Georgia
Frank Rowe Oak Ridge, Tennessee

From Mr. Dan Belemecich, author of the original Binary Search article which we published in the July Newsletter:

I appreciate the privilege of having a useful algorithm published in TRS-80 Microcomputer NEWS.

As published in the July '81 issue, my algorithm does not work. Originally, the program had records with five elements each and the first lines of the listings were:

```
DIM S(255Ø),N=254Ø, N1=125Ø
```

Something happened to the first digits. (Editor: the something was our rewriting the routine for single element records, and then not throughly testing the result.)

To make listing 1 of the published version work, change:

```
2Ø N=5ØØ
     : N1=25Ø
     : N2=Ø
8Ø 'THE SEARCH IS SUCCESSFUL. S(N+1)
```

The purpose of listing 2 was not a repeat of listing 1 as published, but a test of every record in the array; mainly to check the validity of N and N1. Make the following changes in the published listing 2:

```
3Ø FOR W=1 TO 5ØØ
4Ø N=5ØØ
     : N1=25Ø
     : N2=Ø
1ØØ PRINT S(N1+1)
     : NEXT W
     : END
```

Mr. R. Lawrence Clark, of Perimeter Data Systems in Atlanta, Georgia suggested the following changes to make the routines work properly:

In listing 1, change the following lines to read:

```
2Ø N=5ØØ
     : N1=25Ø
     : N2=Ø
3Ø FOR I=Ø TO 9
     : IF W>S(N1) THEN 4Ø ELSE 5Ø
5Ø IF W=S(N1) THEN 8Ø
```

In listing 2, change the following lines to read:

```
2Ø S(I)=I
     : NEXT
4Ø N=5ØØ
     : N1=25Ø
     : N2=Ø
5Ø FOR I=Ø TO 9
     : IF W>S(N1) THEN 6Ø ELSE 7Ø
7Ø IF W=S(N1) THEN 1ØØ
1ØØ PRINT S(N1)
     : GOTO 3Ø
```

With these changes, the above routines will work. They are very restricted however. If you wanted to use the binary search for arrays with other than 500 elements, you would need to recalculate the constants in the FOR-NEXT loop, change several other constants, and generally have to rewrite the entire routine.

As an alternative, Mr. Clark provided the following program, which he indicates is derived directly from the algorithm in the third volume of Donald Knuth's series, The Art of Computer Programming (Addison-Wesley, 1973):

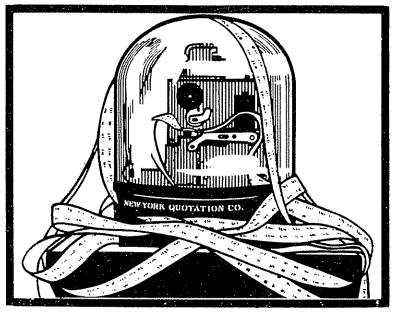
```
1Ø N=5ØØ
     : DIM S(N) 'Change N to adjust array size
2Ø FOR I=1 TO N
     : S(I)=I
     : NEXT 'Initialize array with test data
3Ø INPUT "VALUE TO SEARCH FOR"; W  'Get search key
35 REM The actual search function begins below....
4Ø L=1
     : H=N  'Initialize lower and upper search bounds
5Ø IF H<L THEN 1ØØ  'Range is null; search fails
6Ø T=INT((H+L)/2)  'Get midpoint
7Ø IF W<S(T) THEN H=T-1
     : GOTO 5Ø  'Argument is below midpoint
8Ø IF W>S(T) THEN L=T+1
     : GOTO 5Ø  'Argument is above midpoint
9Ø PRINT S(T)
     : GOTO 3Ø  'Search was successful
1ØØ PRINT "NOT FOUND"
     : STOP
     : GOTO 3Ø  'Search failed
```

Except for storage limitations, this routine will work for any size array by simply changing the value of N in line 10. This gives you a practical and very useful binary search routine. Our thanks to Mr. Clark for this routine and for his several comments.

From Frank Rowe in Oak Ridge, Tennessee we received the following information:

I noted with interest Dan Belemecich's "simple" binary search which appeared in the July, 1981 issue. In my relatively limited experience in programming, I have encountered several situations in which I needed a binary search, and through trial and error have developed my own binary search algorithm, which I feel is somewhat superior to Mr. Belemecich's.

(Continued on Page 23)



Dow Jones

Dow Jones Information Services

SOMETHING OLD, SOMETHING NEW . . .

This month's article from Dow Jones Information Services will address Dow Jones News, one of the most popular and respected data bases. It will highlight one of the newest additions to the service, Wall Street Journal Highlights Online, and will also answer many questions you may have on where to find specific information within the News data base.

DOW JONES NEWS

The News data base of Dow Jones Information Services draws on the worldwide resources of The Wall Street Journal, Barron's Business and Financial Weekly, and the Dow Jones News Service.

The Wall Street Journal is the recognized leader in business journalism. It is the largest circulation daily newspaper in the country—and the nation's only national daily business newspaper. With a readership of more than five million, it provides an accurate and objective account of daily events of interest to men and women in business, government, education, the professions and all other segments of the economy.

The Journal's content reflects the broad interest range of its readership. International affairs, labor and management, agriculture and accounting, marketing and finance, energy, taxes, politics, education, social concerns, cultural affairs and countless other areas of coverage are familiar fare to Journal readers.

Wall Street Journal reporters in bureaus around the world are responsible for gathering and reporting the most significant news. Their first task is to supply news to the ticker (the Dow Jones News Service). Then they set to work writing the story for the Journal. At the same time, they are almost always busy working on more in-depth feature stories for publication in the Journal.

Barron's Business and Financial Weekly magazine is an outstanding source of investment information for the nation's investing community. With a readership of over one million, it fills the more specialized needs of the U.S. financial and investment community. Barron's publishes in-depth articles on industries, analyses of companies and stock market trends, interviews with important investment executives, and thoughtful editorials—all of which are considered essential reading for the financial and investment community.

Dow Jones News Service—Known familiarly as the "broad tape" or news "ticker," the Dow Jones News Service speeds timely financial and business news to all its customers—banks, corporations, brokerage firms and newspapers—simultaneously. It does so by means of an electronic newsprinter and display units, capable of highspeed transmission.

News flows into the wire room in New York via a proliferation of electronic equipment from nearby and from remote parts of the globe. This wire room links the News Service with all of Dow Jones in the U.S., Canada, Europe and the Far East, as well as with the Associated Press, United Press International and various specialized business wires. Each story is quickly and skillfully edited and then is transmitted to thousands of subscribers in business and finance centers throughout the U.S.

Dow Jones News—the Dow Jones data base makes available to our subscribers select information and stories from these unique resources. Because of the timely nature of the news wire, many stories are available online within 90 seconds after being filed. They are held in our computer data banks for 90 days. Many of these articles are cross-referenced by both company code and industry for your convenience. We offer this library of information on over 6000 companies, over 80 news categories and over 50 industries. And you can retrieve the information you want in seconds by asking for headlines and stories on the specific company or industry you are interested in. This direct access feature saves you time that would otherwise be spent looking through paper after paper for the information you want. Direct access also saves you money online for it allows you to quickly retrieve the specific information you have requested. Stock symbols, industry and category symbols, as well as access codes to the data bases can be found in the Dow Jones Operating Guide and Symbol Directory, which all subscribers receive. Dow Jones News is available twenty-one hours a day; usage charges are 20c per minute during non-prime time and $1.00 per minute during prime time.

FREE TEXT SEARCH

But, you may ask, what if I want to retrieve a story that goes back longer than 90 days? Dow Jones Information Services offers a Free Text Search data base which uses a key word search system of logical operators, through which a user can pinpoint articles of interest that have appeared in the News Service dating from the present back to June, 1979. Free Text Search offers an excellent research tool and a real time saver for locating those old but not forgotten articles. Another benefit is the ability to track private companies that are not on the stock exchanges, and therefore are not directly accessible on our retrieval system through a company code. With Free Text Search, by properly keying in the name of the company or the subject matter of interest, you can retrieve all related stories we have carried back to June, 1979.

This service does require reading of the Free Text Search Guide before using it. Guides are available free of charge by calling the Dow Jones Customer Service Hotline at 1-800-257-5114 or in New Jersey, 609-452-1511. However, please note that Free Text Search is an 80-column wide data base. Model II users will have no problem in reading the information on their screen. Model I and III subscribers will find that the line of type will "wrap around" after your 64 characters are used up, thus making reading difficult but still legible. Those of you with a 32 x 16 character screen (color computer and Videotex terminal) might find that the screen will be very difficult to read and choose not to utilize this particular aspect of the service. Free Text Search is available Monday 8:30 a.m. to 8:00 p.m. and Tuesday - Friday 8:00 a.m. to 8:00 p.m. Eastern Time. It is not available on weekends, evenings or holidays. The cost for this service is $1.33 per minute at any time.

NEW DATA BASE

"EXTRA, EXTRA, READ ALL ABOUT IT!"

THE WALL STREET JOURNAL HIGHLIGHTS ONLINE—delivered electronically each morning.

This new, exclusive data base from Dow Jones Information Services brings you edited versions of many of your favorite Wall Street Journal features, available each business day at 7:00 a.m. Eastern Time. Over breakfast you can scan the headlines and summaries of the Journal's in-depth front page stories; the full text of the Business and Finance column plus a brief summary of the Worldwide column on page one. In addition there will be concise summaries of the front page of Section 2, descriptions of major

(Continued on Page 9)

Dow Jones (From Page 8)

editorials, columns and commentary, edited versions of "Heard on the Street" and "Abreast of the Market" columns, and a summary of the back page feature.

You can get a broad overview of the day's Journal in a matter of minutes through the convenience of your computer or Videotex terminal. And, we will keep the five most recent issues of the Journal online, so you can catch up on any issues during the week you may have missed. The cost of the Wall Street Journal Highlights Online is the same as our News data base: 20c per minute non-prime time and $1.00 per minute prime time. If you have any questions, please call our customer service hotline (number listed at the end of this article).

BORROWING FROM OUR SUBSCRIBERS . . .

We have accumulated some of the most frequently asked questions from our subscribers as to where to find certain items of information within the News data base. You may find you have often had the same question. So here are the answers.

Q. How can I get commodities quotes?

A. At this time, complete commodities quotations are not available through Dow Jones Information Services. However, the news category I/CMD contains many current day prices and the Dow Jones Commodity Futures Index.

Q. Where can I find up and down volume?

A. Up and down volume can be found in the news category I/STK under New York Stock Exchange transactions.

Q. Do you carry information about odd lot trading and block trades?

A. Block trades over 250,000 shares can be found in I/BLK. Odd lots can be found under I/STK.

Q. Does Dow Jones carry advances and declines?

A. Yes. Net change percent leaders can be found in the news category I/ACT.

Q. How do I get Dow Jones Averages?

A. Dow Jones averages can be found in the news category I/DJA, updated every half hour.

Q. How do I get updated stock symbols?

A. To get updated stock symbols just call the Dow Jones Customer Service Hotline. As the Symbols Guide is revised, new copies are sent to subscribers.

Q. What is the best method to begin a search?

A. Many people search only for the most recent story. However, by beginning with headlines, the search will usually yield better results faster.

Q. Where can I find a list of the most active daily stocks?

A. The news category I/ACT contains a list of the 10 most active stocks for the New York and American exchanges.

HINTS FOR SHORTER ON-LINE SESSIONS

Before logging on the Dow Jones Information Services, know what information you want. Then, even while the response to your first query is appearing on your screen, you can be keying in your next request. Thus you will shorten the response time and have a continuous feed of data onto your screen. However, this will allow you little time to actually review the material on the screen. If you have a color computer or Videotex Terminal with at least 16K, however, you can get up to 27 pages stored and review the information off-line. If you have a Model I, II, or III, you can use Videotex with a line printer to save the information. Also, remember to type DISC when you wish to go off-line. Tymnet will then give you your log-on, log-off times, and you can keep track of your usage.

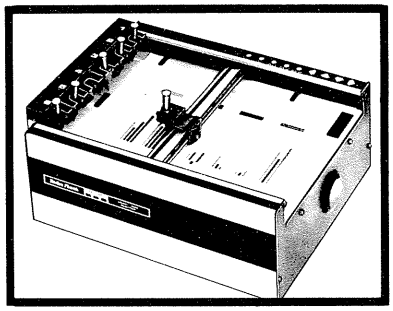
If you have any questions about the service, or wish to request additional information, please call our Customer Service Hotline at 1-800-257-5114, or in New Jersey 609-452-1511. Someone will be there to assist you 8:00 a.m. to 11:00 p.m. Eastern time Monday through Friday, and 9:00 a.m. to 5:00 p.m. on Saturdays.

Calories Revision

Richard Eidmann Philadelphia, Pennsylvania

With thanks to Michael Mayer-Kielmann for his program on calories used while running. I could not resist so I made some changes so that the computer saves all the previous runs, times, weights and distance. Further I added an average on the distance, calories and weight. I hope you like it.

```
3Ø CLEAR 1ØØ
    : DIM DA$(365), W(365), D(365), T(365), C(365)
4Ø CLS
43 ON ERROR GOTO 1ØØØ
45 OPEN "I", 2, "CAL2/TXT"
    : INPUT#2, N
    : CLOSE
5Ø OPEN "I", 1, "CAL1/TXT:1"
55 FOR J=1 TO N
6Ø INPUT#1, DA$(J), W(J), D(J), T(J), C(J)
7Ø NEXT
8Ø CLOSE
1ØØ REM * CALORIES USED RUNNING
1Ø5 N=N+1
    : PRINT"THIS IS SESSION # "; N
12Ø INPUT "ENTER DATE"; DA$
13Ø CLS
14Ø INPUT "CURRENT WEIGHT"; W
15Ø PRINT
    : INPUT "DISTANCE RUN"; D
16Ø PRINT
    :PRINT "ENTER MINUTES AND SECONDS USING A PERIOD "
17Ø PRINT "INSTEAD OF A COLON"
175 PRINT "(E.G.- FOR 16:3Ø ENTER 16.3Ø)
18Ø PRINT
    : INPUT "TIME"; T
19Ø T1=INT(T)+(T-INT(T))/.6
2ØØ C=(W*(.735993-(T1/D*.Ø1325))+3.6)*D
215 DA$(N)=DA$
    : W(N)=W
    : D(N)=D
    : T(N)=T
    : C(N)=INT(C+.5)
22Ø PRINT "DATE       WEIGHT   DISTANCE   TIME
    CALORIES"
225 PRINT STRING$(63,45)
23Ø J=Ø
    : FOR J=1 TO N
24Ø PRINT DA$(J); TAB(1Ø); W(J) TAB(19); D(J);
    TAB(31); T(J); TAB(39); C(J)
25Ø NEXT
255 FOR J=1 TO N
    : W1=W1+W(J)
    : T1=T1+T(J)
    : D1=D1+D(J)
    : C1=C1+C(J)
    : NEXT
    : PRINT STRING$(63,45)
    : PRINT "TOTALS"; TAB(19); D1; TAB(39); C1
256 PRINT "AVERAGE DISTANCE TRAVELED "; D1/N; " AVERAGE
    CALORIES PER RUN "; C1/N
257 PRINT "  AVERAGE WEIGHT  "; W1/N
26Ø OPEN "O", 1, "CAL1/TXT:1"
27Ø J=Ø
    : FOR J=1 TO N
28Ø PRINT#1, DA$(J); ","; W(J), D(J), T(J), C(J)
29Ø NEXT
3ØØ CLOSE
31Ø OPEN "O", 2, "CAL2/TXT:1"
    : PRINT#2, N
    : CLOSE
1ØØØ IF ERR/2+1=54 RESUME 1ØØ
1Ø2Ø ON ERROR GOTO Ø
```



Peripherals

Product Line Manager's News

Those of you who have seen the new computer catalog, RSC-6, which is available now in all RADIO SHACK stores and computer centers should have discovered by now that we have our best ever line-up of printers. I sincerely hope that we can come up with a more inventive number system before "Line Printer IX" comes along. Roman numerals are not my strong point.

The printers are varied; there's a wide range of price and performance represented in the stable. In fact, if you are like me, you have trouble remembering which printer does what or how fast. So . . . this month I will make good another one of my promises made in earlier columns.

Accompanying this month's article is a full page chart outlining most details of the performance specifications of our current dot matrix printers. The chart allows easy reference to key features and invites comparison of one machine to another (ours or theirs). "Spec for spec," feature for feature these units represent unmatched price-performance levels in the industry. All the features described in the chart are standard; none are extra cost. Some printers available in the marketplace need extra cost options to reach the performance levels of our standard products; and you know our products are backed by Radio Shack standards of product quality and service absolutely unmatched in the industry.

Here are a few random comments which come to mind as I scan the chart which might prove to be helpful. It is important to understand the concept of throughput (the total work accomplished, measured in completed lines) when comparing speeds of printers. Note that the chart quotes figures for a variety of line lengths; you can see the results.

I have always had difficulty in finding a set of numbers which will adequately express the reliability factors in our mechanisms. "MTBF" stands for "meantime between failure." This number alone is almost meaningless with out an indication of the "duty cycle" i.e. the duration of non-stop operations.

All Radio Shack products are tested extensively at several stages of development. The most rigorous of these tests are given to a sampling of actual production models (just like you will receive from the store). Before final release of the product, five of these samples are put into a life test. These machines are run for 300 hours straight. We expect no problems to occur during this time. At the end of the run we retest the machines to see that important specs are still being met.

Preventive maintenance is also important to the life of your printer. Read your owner's manual. Follow its recommendations and your printer will give you excellent service.

I hope the chart will help you understand our printers. If you are contemplating an investment, this information should allow you to make the best choice for your needs.

NOTES THAT FLOAT TO THE TOP OF MY DESK

I have a few odds and ends this month to get out of the way. The Plug'n Power controller comes supplied with a full compliment of software. Color computer owners, however, were a little shortchanged: We did not include a subroutine module to allow control of the device from another BASIC program. The following program provides such a capability. When installed, it allows any BASIC program to send commands to the controller modules.

```
7Ø G=Ø
75 ' SET G=Ø FOR 4K SYSTEM     ;
76 ' SET G=3 FOR 16K SYSTEM    ;
77 ' SET G=7 FOR 32K SYSTEM    ;
9Ø G=4Ø96*G
1ØØ DIM A(22)
     : FOR I=1 TO 22
     : READ A(I)
     : NEXT
11Ø FOR X=3712+G TO 3823+G
     : READ Y
     : POKE X, Y
     : NEXT
12Ø CLS
     : INPUT "INPUT HOUSE CODE A-P"; Z$
     : Z=ASC(Z$)-64
     : POKE 3825+G, A(Z)
13Ø PRINT " "
     : INPUT "INPUT UNIT CODE 1-16"; D
14Ø PRINT " "
     : PRINT "INPUT COMMAND CODE"
145 INPUT "ON=1 OFF=2 CLR=3 ALL=4 BR=5 DIM=6"; E
15Ø E=E+16
     : C=3
     : IF E<21 GOTO 17Ø
16Ø PRINT " "
     : INPUT "INPUT NUMBER OF STEPS 1-1Ø"; C
     : C=C*2+1
17Ø POKE 3826+G, A(D)
     : POKE 3824+G, 3
     : GOSUB 2ØØ
18Ø POKE 3826+G, A(E)
     : POKE 3824+G, C
     : GOSUB 2ØØ
19Ø GOTO 12Ø
2ØØ DEFUSRØ=3712+G
     : X=USRØ(Ø)
     : RETURN
2Ø2 ';************************************;
2Ø3 ';  FOR 4K COLOR BASIC CHANGE 2ØØ TO   ;
2Ø4 ';  2ØØ POKE 275, 14: POKE 276, 128:   ;
2Ø5 ';       X=USR(Ø): RETURN              ;
2Ø6 ';                                     ;
2Ø7 ';  REM FOR 16K USE POKE 275, 62       ;
2Ø8 ';  REM FOR 32K USE POKE 275, 126      ;
2Ø9 ';************************************;
21Ø DATA 96, 224, 32, 16Ø, 16, 144, 8Ø, 2Ø8, 112, 24Ø
22Ø DATA 48, 176, Ø, 128, 64, 192, 4Ø, 56, 8, 24, 88, 72
23Ø DATA 48, 141, Ø, 1Ø8, 23Ø, 132, 52, 4, 141, 47
24Ø DATA 141, 45, 141, 43, 141, 33, 166, 1, 198, 4
25Ø DATA 141, 12, 198, 5, 166, 2, 141, 6, 53, 4, 9Ø
26Ø DATA 38, 229, 57, 72, 36, 2, 141, 18, 141, 8, 37, 2
27Ø DATA 141, 12, 9Ø, 38, 242, 57, 52, 6, 52, 1, 198, Ø
28Ø DATA 32, 6, 52, 6, 52, 1, 198, 255, 182, 255, 32, 68
29Ø DATA 36, 25Ø, 134, 52, 74, 38, 253, 134, 3
3ØØ DATA 247, 255, 32, 52, 2, 134, 178, 74, 38, 253
31Ø DATA 183, 255, 32, 53, 2, 74, 38, 5, 53, 1, 53, 6
32Ø DATA 57, 52, 2, 134, 248, 138, Ø, 74, 38, 251
33Ø DATA 53, 2, 32, 22Ø
```

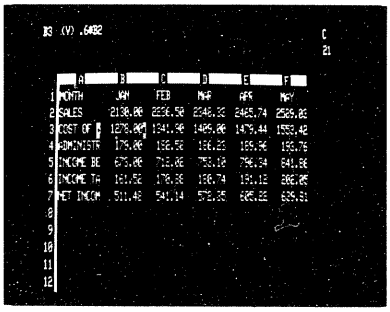
There has been a recent change at CompuServe which can affect users of the Direct Connect Modem I (26-1172) or Radio Shack's Videotex terminals (which contain a direct connect modem in them). If you are using a CompuServe node to access CompuServe, the system now looks at data received from your system when you begin the sign-on process to determine the baud rate in use. If the sequence described below is not followed, room noise can be transmitted to the CompuServe host system, causing an incorrect computation of the baud rate. The result may hang everything up. Follow this procedure and you will avoid tears and

(Continued on page 32)

Line Printer Comparison Chart

	LP 5	LP 6	LP 7	LP 8
	26-1165	26-1166	26-1167	26-1168
Matrix Format				
Size in dots (H×W)	9 × 9	7 × 9	7 × 5	9 × <=23
Descenders	Yes	No	No	Yes
Underline Mode	Yes	No	No	Yes
Bold Mode 10 cpi	Yes	No	No	No
Max. Line Length	13.2"	13.2"(4)	8"	8"
Horizontal Pitches (cpi)				
Fixed	5,7.5,10,15	5,7.5,10,15	5,10	5,10,16.6
Proportional	No	No	No	PS, PS Elong, PS Cond
Mixed Pitch per line	Limited	Limited	Yes	Yes
Line Pitches (1pi)	6,8,12	6,8,12	6	6,8,12
Speed (Throughput)				
CPS @ 10 cpi	160	100	30	80
LPM @ 10 cpi	60/132 col	33/132 col	19/80 col	23/80 col
	100/66 col	42/80 col	53/20 col	55/20 col
	198/22 col	120/20 col		
Logic Seeking (1)	Yes	Yes	Short Line	Yes
Bi-directional	Yes	Yes	No	No
Character Set				
Standard ASCII (7 bit)	Yes	Yes	Yes	Yes
International (8 bit)	Yes	Yes	No	Yes
Graphics				
Block Characters	Yes	Yes	No	Yes
Bit Image: Dots per line	No	No	480	960 (3)
Paper Handling				
Maximum Size	15"	15"	9½"	9½"
Friction Feed	No	Yes (s)	No	Yes 8.5"(s,r)
s - Single Sheet				
r - Roll Paper				
Tractor (Fan-fold)	Adj.(4"-15")	Adj.(4"-15")	Adj.(4.5"-9.5")	Fixed 9.5"
Copies	Orig + 4	Orig + 3	Orig + 2 (2)	Orig + 2
Word Processing Features				
Backspace	No	No	No	Yes
Near Correspondence				
Hi-Density PS	No	No	No	Yes
Incremental Spaces				
and Backspace	No	No	No	Yes
For/Rev Linefeed	No	No	No	Yes
For/Rev ½ Linefeed	No	No	No	Yes
Controls	Power	Power	Power	Power
		On/Off Line		On/Off Line
	Reset	Reset		Reset
	Restart			Restart
	Line Feed	Line Feed		
	For 1/8 LF	1/12 LF		
	Rev 1/8 LF			
Indicators	3	3	1	2
User Self Test	Yes	Yes	No	Yes
Reliability				
Head Life (Chars)	100 Million	100 Million	30 Million	>30 Million
MTBF @100% (6)	300 Hrs.	300 Hrs.	300 Hrs.(50% Duty)	300 Hrs.
Ribbon				
Type	Cart. ($13.95)	Cart. ($11.95)	Cass. (2/$8.95)	Cart. ($11.95)
Life	2 Mil Char	1 Mil Char	1 Mil Char	1 Mil Char
Interface				
Buffer	1 Line	1 Line	90 bytes	160 bytes
Parallel Data Bits	8	8	8	8
Status : Busy	Yes	Yes	Yes	Yes
Bits : Fault	Yes	Yes	No	Yes
: Paperout	Yes	Yes	No	Yes
: Select	Yes	Yes	No	Yes
Serial (5)				
Data Bits	N/A	N/A	7,8	7,8
Baud Rate	N/A	N/A	600	600/1200
Size	7.4×24.4×15.9	6.3×24.2×13.3	5.32×16×8.25	4.7×15.4×11
Weight	42 lbs	28 lbs	8.6 lbs	16.5 lbs
Power 120 VAC +/- 15%	85W	85W	15W	85W
Price (less cable)	$1860.00	$1160.00	$399.00	$799.00

Notes:
(1) Eliminates leading and trailing spaces in print line
(2) Carbonless forms only
(3) 480 Addressable double dot columns.
(4) Max Line Length 132 Columns
(5) 4 Pin DIN Connector (RS232 DATA, STATUS, GROUND)
(6) See Text.



VISICALC™

The Visible Calculator

Add Some Structure to Your VisiCalc® Models

VisiCalc software provides tremendous power to your TRS-80 for handling financial and other numerical models. What's a model? Simply a set of mathematical relationships between variable values.

For example:

A + B = C

is a model; C always equals any value in A added to any value in B. When either A or B have their values changed, C will change. If the plus sign is changed to a minus, C will naturally be changed too.

What the VisiCalc program does is allow you to build that kind of dependent relationship among hundreds of locations on an electronic work sheet, so that when one number (or formula) is changed, ALL the numbers dependent on it will change too. And those numbers are displayed on the screen of your TRS-80 continuously. Planning for your "bottom line" becomes quick and easy.

This concept of an "interactive" model is fairly easy to understand, and your Radio Shack dealer will be happy to demonstrate how powerful the VisiCalc program is for analyzing business decision-making situations. But where do YOU begin when you sit down to plan out YOUR model for use with the VisiCalc program?

After reading the detailed tutorial manual and becoming familiar with the way the VisiCalc commands work, you need to know two other basic items:

- What do you want to find out from the model?
- An idea of what factors affect your business.

That sounds simplistic; but with the VisiCalc program handling the calculations, you can go much farther into calculating the factors which affect the operation of your business than you ever were able to with pencil and paper. The editing ability of the program (much like a word processor) allows you to take account of new factors when they come to your attention.

Too, no one knows your business like you do; it would be hard for some distant programmer to take every factor important to you into consideration. That's why the flexibility of the VisiCalc program is appreciated by a wide range of business people. It's your good business sense and judgement which makes the calculation abilities of the program so valuable.

Starting With Structure

What's structure? It's a way of laying out your VisiCalc model so that it is easy to change without disruption or confusion and easy to understand by someone else working with it. With the VisiCalc program, this translates into grouping your working variables in one logical place, and labeling them well so that other people using the model can understand it easily.

For example, if you're doing a sales forecast, your first month's sales figure may be a value you'll want to play with. Or, perhaps the RATE at which your sales increase or decrease over the course of the year (seasonal fluctuations, for example) may be brought out in a table. That way, only the table need be changed when you need a change reflected in the entire model.

Having your variables all in one portion of the worksheet (or in two or three specific areas for a larger model) comes in handy if you don't expect to use a given model very often. After all, you may forget precisely why you included a variable, or how it was used in your model.

A bonus which comes from fully-labeling your variables is that another person can understand the way your model is put together. That means changes and improvements can be made without disrupting the model.

USER TIP: placing your variable area in the upper left hand corner of your worksheet is advantageous because VisiCalc software calculates either across rows from left column to right, top row to bottom, or down the columns, left column to right. In the case where you wish to have more than one variable area, they should be positioned above your model.

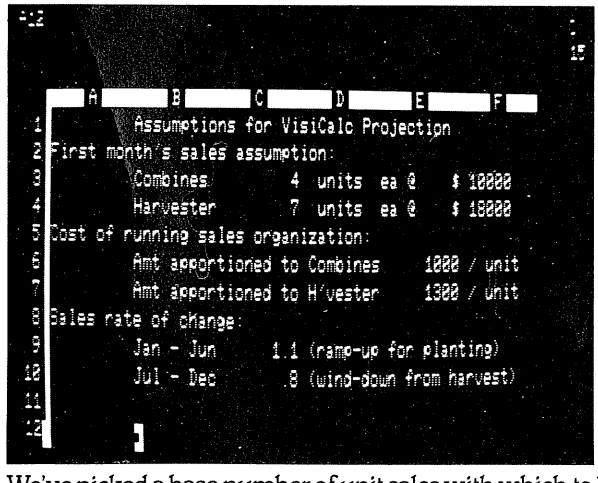
Another Way Structure Can Help

Beyond making your VisiCalc model simpler and more understandable, structuring your model has other advantages. It provides you with an area of the screen which can act as a "fill in the blanks" page—for real estate estimate, insurance quotation, construction bid or any other type of analysis which "plugs" information into a process.

For example, a real estate agent might construct a VisiCalc model to calculate the first five years' amortization on a mortgage. He'd need to know asking price, down payment, tax and interest rate. From that the model would immediately calculate monthly principal and interest payment, monthly tax assessments and show the growth of equity for five years. A few additions to the model would allow the inclusion of income, show percentage of income bound into the mortgage payment, adjust for condo or income property.

A Structured Projection

Screen One shows all the assumptions we're making for a sales projection for Smith Combine & Harvester Sales—a non-existent company.



We've picked a base number of unit sales with which to begin our projection planning. January will start with four and seven units—combines and harvesters are VERY expensive. We might also have divided our sales among territories as percentages of the total sales, rounded to integers because it's difficult to sell half a harvester.

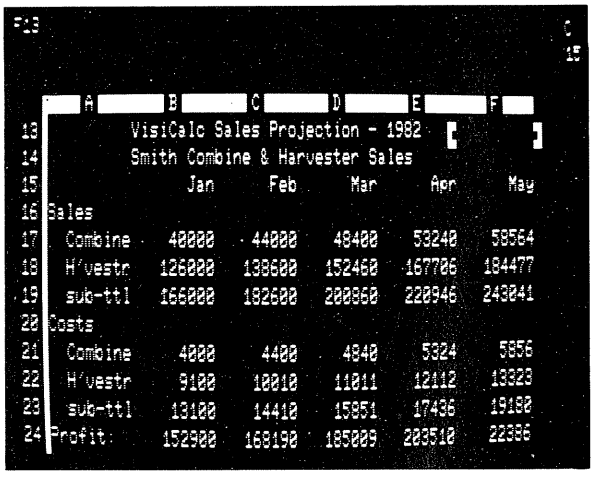
The costs of running our efficient sales organization have been apportioned on a per unit basis. We figure it will cost us

(Continued on Page 13)

VisiCalc (From Page 12)

$1000 per unit to sell a combine, $1300 to sell a harvester. At first glance, selling a combine might be more attractive; we'd have to look at costs as a percentage of sales for each product to be sure.

The rate of change in sales, both up and down, varies with the time of year. We chose combines and harvesters for this reason; we might also have picked toys, auto accessories, lawn furniture or other salable products which fluctuate with the seasons or other time dependent considerations—such as U.S. tax time or a fiscal year.

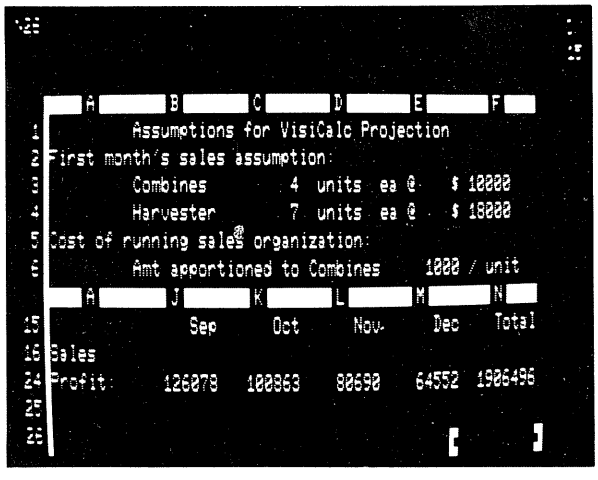


On Screen Two is our model proper. All the numbers in it derive directly from the assumptions made on our assumption screen. For example, the January Combine sales are the 4 units times the price per unit($10,000). Thus, even the first entry in our model is a formula which may be influenced by a change in our assumptions. After all, maybe the manufacturer will have a price increase, and we'd have to adjust our model.

How is this fancy structure an advantage? The MODEL need not be touched. There is no danger of introducing typographical errors into the model or accidentally changing the "look" when presenting multiple versions of identical reports.

Using the VisiCalc window command, we can split the screen horizontally (or vertically) and examine the effect on our bottom line of altering the assumptions. Screen Three shows the model split by a window to allow this kind of immediate analysis with the TRS-80 Model III personal computer.

A little forethought about the results you're looking for, your knowledge of the factors which affect your business, and the flexibility of the VisiCalc program let you accomplish more with your TRS-80 in a shorter time than you ever thought possible.



How do you use the VisiCalc program?

We're looking for interesting ways you're using the VisiCalc program, and would like to share them with our other readers. Please let us know. Drop a line to:

The VisiCalc Page
Microcomputer News
P.O. Box 2910
Fort Worth, Texas 76113-2910

VisiCalc is a registered trademark of Personal Software Inc.

CASSCOM and Screen Printer

Peter Mazur Oak Ridge, Tennessee

The User's Guide to RADIO SHACK'S CASSETTE COM (Cat. No. 26-1139) states that one limitation to this otherwise useful terminal program is that it will not output to a printer. Actually, in a non-disk Model I, with a short patch the program can be used very effectively with the RADIO SHACK Screen Printer (Cat. No. 26-1151 Note: The 1151 screen printer is no longer available from Radio Shack).

The patch is a slightly modified version of that given in the Screen Printer manual. This printer produces "garbage-free" copy only when the TRS-80 is executing routines in ROM as is the case with a BASIC program. In order for it to produce clean copy with a machine language program in RAM like CASSCOM, it is necessary to provide a routine which keeps execution in ROM during the actual screen print. Control R was used to jump to this routine since CASSCOM can not otherwise use it in non-disk systems. The patch is as follows:

```
CASSCOM and Screen Printer
727F   C3 AE 73          JP   73AE      ;JUMP TO PRINT ROUTINE
73AE   F5                PUSH AF
73AF   C5                PUSH BC
73BØ   Ø6 Ø8             LD   B,ØBH
73B2   ØE ØØ             LD   C,Ø
73B4   D3 FE             OUT  (ØFEH),A ;EXECUTE SCREEN PRINT
73B6   CD 2B ØØ PAUSE    CALL ØØ2BH    ;REMAIN IN ROM
73B9   ØB                DEC  BC
73BA   78                LD   A,B
73BB   B1                OR   C
73BC   2Ø F8             JR   NZ,PAUSE
73BE   C1                POP  BC
73BF   F1                POP  AF
73CØ   C3 15 7Ø          JP   7Ø15  ;RETURN TO KB SCAN IN
;MAIN PROGRAM
```

The Screen Printer Manual gives the instruction at 73B0 as:

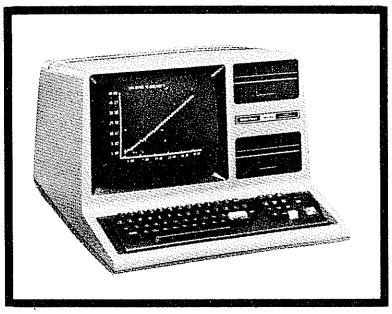
```
LD  B, Ø8
```

but I found the 0BH to give many fewer garbage @'s.

The indicated modifications can be made with TBUG's M command, and the modified program then saved as a SYSTEM program on cassette tape with TBUG's Punch command as:

```
P 7ØØØ 73C2 7ØØ1 CASPRT
```

A screen print is executed when desired by depressing the SHIFT, DOWN ARROW, and R keys to produce CONTROL R.



Model I/III

Product Line Manager's News

Roll those pennies you have stashed in the jar, return your pop bottles that have accumulated in the garage, recycle the cans, and continue along with the rest of us to look for ways to handle the costs of living. If you are cost conscious, like most people today, then a bargain is noteworthy and putting off purchases can become an everyday occurrence.

Whether it's your departmental budget that is shrinking or your personal funds, it is now easier than ever to get started with the Model III Disk System. We have just introduced our 1 disk 48K Model III (26-1065 $1995). The 1 disk system is similar to our Desktop Model III Business System, but we took out the RS-232C communications interface and one of the disk drives.

This system is for those of you who want disk, but don't need all the capacity of two double density drives. The new single disk Model III has all of the disk space and memory you need to run such popular programs as: VisiCalc®, SCRIPSIT ™ , Profile, and Profile III Pulus ™ . Of course, the RS-232C and/or second drive can be added later.

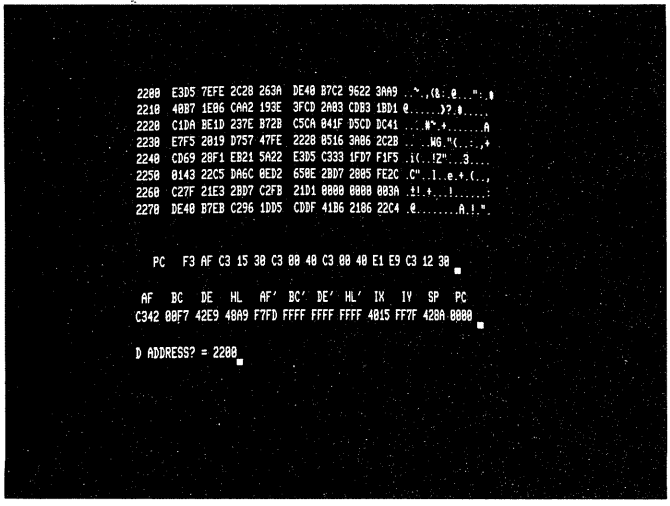
If you are planning to move up to disk anyway, this system will save you the installation charges for the two 16K RAM upgrade Kits and the Single Drive Kit. If you had those three kits installed on a 16K Model III, you total cost would be about $2100.

There is one caution for those of you considering the one drive system: the XFERSYS and CONVERT TRSDOS Utilities both require two drives, so you will not be able to use these functions.

The one drive Model III without RS-232C is available now, so contact your local store to order one.

DEBUG for Cassette based Model I/III systems

Model I/III DEBUG (26-2000, suggested retail price $19.95) is ready! It is a cassette based "monitor" for entering, testing and debugging machine language programs. If you plan to use our Series I Editor Assembler (26-2011), then you will probably also want to purchase DEBUG. Few of us can avoid errors during software development and DEBUG is designed for finding and correcting errors as you develop assembly language programs. It works by allowing you to examine your TRS-80 memory and its registers to watch what is happening as your program executes. If you don't like what is happening, DEBUG allows you to modify memory locations and the values associated with the Z-80 registers to debug your program. A typical use might be to assemble a source code program with our Series I Editor Assembler and then Jump with Debug to the object code program generated from your assembly. This would begin execution of your program. If you didn't get what you expected then use DEBUG to single step thru the program and examine DEBUG's display to see what is really happening at every step and to verify that your program is doing what you intended.



Advantages of this program over our Model I T-BUG (26-2001) are:

- In the normal mode, all registers and machine status is displayed.
- Register contents can be modified directly with the R command.
- Two single step commands, I (absolute single step) and C (execute CALLs in full, otherwise single step) are available which allow you to step through your programs.
- The U (Update) command causes the display to be updated continuously.
- The M (Modify Memory) command will allow you to modify from low memory addresses to higher addresses, or to modify from high addresses to lower addresses.

Two Radio Shack books which should be quite helpful for DEBUG users are How to Program the Z-80 (62-2066) and TRS-80 Assembly-Language Programming (62-2006). The first of these books is about the Z-80 as a microprocessor, NOT the TRS-80, and is especially useful for information about particular Z-80 commands. The second book was written especially for the Model I. In general most of the information in this second book will apply directly to the Model III.

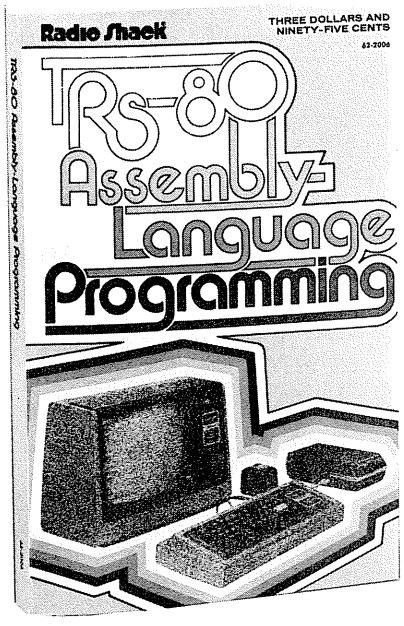
Here is some additional information which you should find helpful:

1. DEBUG can be considered to reside and use memory from 4200 Hex to 49FF Hex (not 39FF as in the manual). The actual DEBUG program resides from 4332H to 493FH as shown in the memory map on page 10 of the manual. Your programs can reside from 4A00H to the end of memory.
2. Model I users with Expansion Interfaces may not be able to use DEBUG unless the Expansion Interface is turned OFF. This will restrict those Model I users to 16K. Note: this appears to be a problem only with Expansion Interfaces which require a Buffered Interface Cable.
3. On page 4 of the DEBUG manual, the last example of the M (Modify memory) command states that it will modify 65ABH and the FOLLOWING two bytes. This should be the PRECEDING two bytes.
4. On Page 5 of the manual, under the U command, the address listed for the real time clock is for the Model I only. For Model IIII use 4217H.
5. Also under the U command, when you want to terminate the Update function, you need to press any key and HOLD the key until the function terminates.
6. If you are writing a data tape using the W (Write system tape) command in a Level II or Model III BASIC system, the Transfer (T) address should be 4909H to return to DEBUG, 01C9H to return to LEVEL I BASIC READY, and 1A19H to return to Level II or Model III BASIC READY.

(Continued on Page 15)

Model I/III (from page 14)

I like the new DEBUG program, and find it much nicer to work with than the older T-Bug. If you are curious about what is really happening with your TRS-80 programs, or if you are developing assembly language/machine language programs, this program should be a great help to you.



Model I/III Bugs, Errors, and Fixes

26-2111 Model III Disk Owner's Manual

Page 85 of the Model III Owner's Manual contains an error.

$RAMDIR

If C contains the decimal value 255 when your machine language routine executes a call to $RAMDIR, you will not get the ASCII-coded message "*** nnnnn Free Granules***", as the manual indicates.

What occurs is that, on return from the CALL to $RAMDIR, the register pair HL points to four bytes of information.

The first two bytes are the HEX number of granules that have been used (Decimal 163 in the example.) The second two bytes contain the HEX number of free granules (Decimal 77 in the example.) You as the assembly language programmer will have to make any needed Hex to Decimal conversions, and you must provide any messages that you may feel are appropriate.

26-2203 RSCOBOL for Model I/III

We have had reports of erroneous I/O Error 30s occurring during line feeds to a printer. The following patches will prevent the erroneous I/O Error 30s:

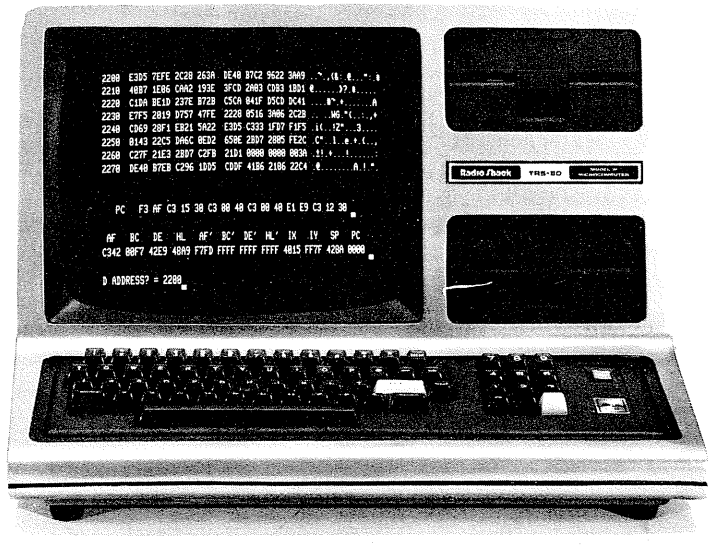
```
PATCH RUNCOBOL/CMD (ADD=A5F4, FIND=28, CHG=18)
PATCH RUNCOBOL/CMD (ADD=A6Ø2, FIND=28, CHG=18)
PATCH RUNCOBOL/CMD (ADD=A612, FIND=C2CEA6, CHG=ØØØØØØ)
PATCH RUNCOBOL/CMD (ADD=A642, FIND=CA, CHG=C3)
PATCH RUNCOBOL/CMD (ADD=A653, FIND=C2CEA6, CHG=ØØØØØØ)
PATCH RUNCOBOL/CMD (ADD=A665, FIND=CA, CHG=C3)
```

These patches should be applied to both Model I and Model III versions of RUNCOBOL. Note: the Model I TRSDOS which is supplied with RSCOBOL includes the PATCH utility.

The following information is provided to help you compute the RAM requirements for your user COBOL programs:

System memory requirements:	64K Model II	48K Mod I/III
ROM	0K	16K
TRSDOS	10K	4.5K
End of Usable Memory	60K	64K*
Total RAM available for RUNTIME plus User Program (bytes)	51,200	44,544
RSCOBOL Runtime Size (bytes)	24,060	24,435
Total RAM available for User Programs, Files, and Data (bytes)	27,140	20,109
User Memory Space Requirements Calculation:		
User Program Size (TOTAL BYTES ON COMPILE) plus overhead	+ 115	+ 96
plus Number of Files OPEN at the same time multiplied by buffer size:		
Model II 626* _____ =	+	
Model I/III 348* _____ =		+
plus space for ISAM files if used:		
1. 46+(4*(# ALT KEYS)) [PER FILE]	+	+
2. ((LENG LARGEST KEY)*2)+4 [PER FILE]	+	+
3. LARGEST ISAM RECORD LENGTH [ALL FILES]	+	+
Space used by User Program, TOTAL		
If COBOL DEBUG is to be used, add (bytes)	+ 1127	+ 1127
TOTAL MEMORY SPACE REQUIRED		
If Total Space Required is greater than	27,140	20,109
you need to shorten or segment the program		

*Model I/III RAM is contiguous from 4000 to FFFF Hex (16,384 to 65,536)



Model III Directory Program

James Ditucci

Sharing experiences with fellow computerists is one of the more pleasurable aspects of my job. As manager of the Radio Shack Computer Center in Rochester NY, I have many opportunities to discuss business and personal computing with customers. Through my contacts I have been exposed to many points of view, and one single aspect that keeps surfacing time and time again is the need for more "user-friendly" access to Model III disk based computer programs.

Common complaints expressed are; "My wife won't even look at the computer because she thinks it is too complicated to operate . . . ," or . . . "My kids can't tell the difference between a machine code program executed under TRSDOS and a BASIC program . . . ," or . . . "My secretary doesn't use the system very often because it takes too much time to get to the point where it can do something."

If any of those complaints sound familiar to you, then this program is just what you need to create a little more interest in your TRS-80 Model III Disk system. You might even find yourself using it on all your disks.

This program will allow the user to review the directory of programs on a Model III diskette and easily select the one desired. Program selection is made by positioning a flashing cursor over the name of the program, using the four arrow keys. Then press **[ENTER]** and the computer decides whether to execute the program under BASIC or TRSDOS and does so automatically.

After typing in the program as listed, save it out to disk using the filename M3DIR/BAS (or your own file name). Create a "DO FILE" program as follows:

```
BUILD DIR <ENTER>
CLS <ENTER>
BASIC <ENTER>
<ENTER>
<ENTER>
RUN "M3DIR/BAS" <ENTER>
<BREAK>
```

Now type:

```
AUTO DO DIR <ENTER>
```

From that point on, every time you load in that diskette (or press RESET) the directory program takes control and takes the guesswork out of program initialization. Naturally, both the "DO FILE" and the M3DIR/BAS program file can be copied for use on another diskette.

The secret to a clean looking directory display is to assign an invisible attribute to all program and data files you do not want executed from the program. Examples are:

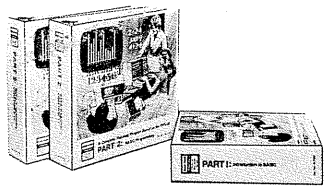
```
ATTRIB M3DIR/BAS (I) <ENTER>
```

and

```
ATTRIB DIR/BLD (I) <ENTER>.
```

How effective is the program? Since I wrote this program six months ago, my four year old daughter has become the master of the Model III at our house!

```
3 REM *** 19 JAN 81   NO RIGHTS RESERVED ***
5 REM *** MOD III DIRECTORY/MENU PROGRAM ***
6 REM ***  WRITTEN BY: JAMES C. DITUCCI  ***
7 REM ***  RADIO SHACK COMPUTER CENTER   ***
8 REM ***  ROCHESTER, NEW YORK    14623  ***
9 REM          *** INITIALIZE ***
10 CLEAR 150
     : ON ERROR GOTO 10
     : CMD"B","OFF"
     : CMD"D:0"
     : POKE 15367, 49
20 PRINT CHR$(15)
     : PRINT@960, "* USE ARROWS TO POSITION CURSOR-PRESS
     ENTER TO SELECT PROGRAM *";
30 V=15360
     : C=143
     : K=14400
     : I=0
     : P$=""
40 REM          **** SCAN KEYBOARD ****
50 IF I=1 GOTO 160
60 IF I=64 AND PN<>32 THEN V=N
70 IF I=16 AND PD<>32 THEN V=D
80 IF I=32 AND PP<>32 THEN V=P
90 IF I=8  AND PU<>32 THEN V=U
100 L=PEEK(V)
     : I=PEEK(K)
     : N=(V+15)
     : P=(V-15)
     : D=(V+64)
     : U=(V-64)
110 IF I=64 AND PN=32 THEN N=(V+19)
120 PN=PEEK(N)
     : PP=PEEK(P)
     : PU=PEEK(U)
     : PD=PEEK(D)
130 IF V<15345 THEN V=(V+64)
     : GOTO 100 ELSE IF V<15360 THEN 30
140 IF V>15360 AND PEEK(V-1)<>32 V=V-1
     : GOTO 100
150 POKE V, C
     : FOR X=1 TO 15
     : NEXT
     : POKE V, L
     : FOR X=1 TO 10
     : NEXT
     : GOTO 40
160 REM *** EXECUTE PROGRAM SELECTED ***
170 FOR X=0 TO 11
     : C=PEEK(V+X)
     : IF C<32 THEN C=(C+64)
180 IF C=32 NEXT ELSE P$=P$+CHR$(C)
     : NEXT
182 LP$=LEFT$(P$,7)
     : IF LP$="DiskDri" OR LP$="Ready" THEN RUN
183 IF LP$="DRIVE:1" THEN CMD"D:1"
     : POKE 15367, 48
     : GOTO 20
185 IF LP$="DRIVE:0" THEN RUN ELSE CMD"B","ON"
200 L=LEN(P$)
     : IF RIGHT$(P$,4)="/CMD" P$=LEFT$(P$,L-4)
     : GOTO 220
205 IF RIGHT$(P$,4)="/BLD" THEN P$="DO "+P$
     : GOTO 220
210 GOSUB 230
     : RUN P$
220 GOSUB 230
     : CMD"I",P$
230 ZB$="NOW   LOADING   "+CHR$(244)+CHR$(245)+CHR$(246)+
     " "+P$
250 ZN$=STRING$(63,32)
     : Z=960+(64-LEN(ZB$))/2
260 PRINT CHR$(21);
     : FOR X=1 TO 5
     : PRINT@960, ZN$;
     : FOR T=1 TO 30
     : NEXT T
270 PRINT@Z, ZB$;
     : FOR T=1 TO 90
     : NEXT T
     : NEXT X
     : PRINT CHR$(21);
     : RETURN
```



Education

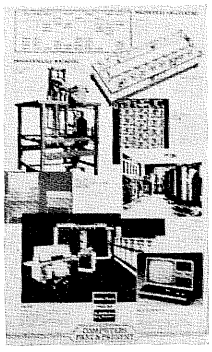
Educational Products News

BASIC COMPUTER LITERACY: COMPUTERS PAST AND PRESENT

Radio Shack's new Basic Computer Literacy Program (26-2755) is a classroom package designed to help students develop an understanding of computers: their history, their functions, and their applications in modern society. The package contains a teacher's manual, a large four-color wall chart for classroom display, and six spirit masters.

The BCL Program's teacher's guide provides a background survey of computers and computer history plus a bibliography with additional resources.

The background material on computers will help the teacher present each unit effectively, no matter how much or how little the teacher knows about computers. The BCL Program assumes that neither the teacher nor the students have access to a computer. All of the unit's activities can be completed effectively with or without the use of computer equipment.



The wall chart (24 inches by 36 inches in full color) serves as a visual reference for the student and as a colorful motivational device, and can be used as a permanent classroom display. Its large, attractive graphics illustrate the development of calculators and programmable machines through history. The chart includes the abacus (still widely used in Far Eastern countries), the La Pascaline calculator developed by Blaise Pascal, Joseph Jacquard's programmable loom, the Analytical Engine designed by Charles Babbage, Herman Hollerith's census tabulating machine, the monstrous ENIAC, and finally the first true computer—UNIVAC.

Each preprinted spirit-duplicating activity master will reproduce up to 200 copies. Simply attach the master to any standard spirit-duplicating machine and run the copies needed. Masters can be stored for later use. Lesson objectives, answers to the various activities, and suggestions for using each class activity sheet are printed on the face of each master. This information will not reproduce when making student copies.

The selected bibliography provides twenty-two references for computing literature suitable for grades four through twelve. Each reference supports the use of the activity sheets in answering these basic questions: What is a computer? What can a computer do? How does a computer work?

A basic understanding of computers is a critical need of students in the 1980's, but actual access to computers remains limited for thousands of children. The TRS-80® Basic Computer Literacy Program (26-2755) brings computer awareness into classrooms with or without computers. This is part of the Radio Shack Education Division's continuing effort to make quality educational computer resource materials available to teachers.

Introducing Talk/Tutor™ and "The History of Technology"

Late this fall a new educational software series called the "History of Technology: Part 1" (26-2624) will be available for the 16K and 32K Color Computers. "The History of of Technology: Part 1" is the first of many courseware packages for the Color Computer to be developed with the new software development system called Talk/Tutor.

Talk/Tutor is licensed to Radio Shack for the development of instructional software (or "courseware") for the TRS-80® Color Computer. The educational programs developed with the Talk/Tutor system will have upper and lower case text, high-resolution color graphics created with a special television camera, and high-quality audio recordings including spoken narration and music.

A 16K Color Computer (26-3001,26-3015) and an ordinary audio cassette recorder like the CTR-80A (26-1206) are the minimum equipment requirements for using any of the Talk/Tutor programs. Students are permitted to interact with the lessons using the optional joysticks (26-3008) as well as the keyboard. The use of the joystick option provides the flexibility of more convenient response input when the learner may have difficulty using a standard keyboard.

Talk/Tutor makes low cost computer assisted instruction possible in the subject areas of reading, foreign language, history, and social studies. These subjects require extensive use of upper and lower case text, realistic pictures, and clear spoken instruction. Talk/Tutor is one of the first programming systems to address these needs in computer assisted instruction.

In recognition of this instructional potential, selected educational publishers will be sublicensed to develop and sell instructional programs for the Color Computer using the Talk/Tutor system, according to William T. Gattis, director of Radio Shack's Educational Division. Owners of the Color Computer can look forward to a growing number of programs from both Radio Shack and outside publishers.

They won't have to wait long. The "History of Technology: Part 1" package (26-2624) will contain the following four lesson tapes: "The Electric Car," "The Age of Flight," "The History of Computers," and "Space Exploration." Later in the year, the "History of Technology: Part 2" (26-2625) will contain four additional titles on the historical development of technology in the Western world: "Edison's Electric Inventions," "Bell and the Telephone," "The Story of the Railroad," and "The Age of Television."

Each fifteen minute lesson contains lesson-related questions. The student selects the correct answer from those presented on the screen either by typing the number of the answer selection on the keyboard, typing a word as the answer, or moving a marker over the correct answer with the optional joystick and then pressing the joystick button. The student receives immediate feedback on the response. If the student selects a correct answer, the narrator's voice may tell the student that the answer selected is correct. If a wrong selection is made, a tone may sound, and a hint or explanation may appear on the screen. Feedback response is varied within the lesson, presenting a mix of audio-visual and verbal reinforcement immediately to the student.

The "History of Technology" series lends itself to many classroom activities. The TRS-80® Color Computer can be used with any standard television set or large screen projector, permitting a

(Continued on Page 18)

Education (From Page 17)

single student or a whole class to learn about the dramatic history of modern technology. Talk/Tutor tapes can be used as an incentive for individual study, in the classroom or in the home. Talk/Tutor tapes offer extra instruction for a remedial student or extra challenge to highly motivated students.

Let's imagine a few possibilities.

A class can be divided into groups, and each group works with a different "History of Technology" program. Later, each group could be responsible for reporting to the class about the particular subject they studied.

The whole class could view the same "History of Technology" program once, and the program could then be placed in the library so that students can refer to it later. Individual students or groups could be assigned different kinds of reports or projects related to the subject matter.

When viewed by a group of students, the Talk/Tutor material can be used as an interactive documentary; when the program presents questions, students in small study groups can take turns answering, or answers can develop out of group discussion. The computer becomes a moderator that challenges teams of children in a scholastic bowl game.

Science projects related to "The Electric Car," for example, might involve learning about the kinds of energy used by trains, ships, and other forms of transportation, collecting information about new forms of energy like gasohol, or finding out how electric power is produced at a local power plant.

A history project for "Edison's Electric Inventions" might involve reading about the life of Thomas Edison, or students interested in history or current events could report on the events that occured during Edison's life time. Students can explore the social impact of Edison's inventions on their own daily lives.

The "History of Technology: Part 1" (26-2624) and "History of Technology: Part 2" (26-2625) are only the latest additions to Radio Shack's growing selection of validated, tested courseware for use in the classroom.

Sorting 'Multi-Dimensioned' String Arrays

Richard W. Saxen Port Orange, Florida

To keep the ball rolling on Allan Emert's machine language sort (July 1980), which lit a fuse under me and others as well, I have put together bits and pieces of ideas gleaned from many sources to come up with a method to perform what amounts to sorting multi-dimensional string arrays using a single dimension sort.

The idea of concatenating the various elements of a record into one string has been advanced with the purported advantage of speeding up cassette input/output. Reverend Harry Jansing's adaptation demonstrates the use of field dividers, or keys ("/" and "#"), in his program—July 1981. This has the advantage of conserving space, since it permits each element (field) of the final string (record) to vary in length from record to record. However, depending upon where the particular divider is located in the record, the search for it on a character by character comparison basis may be faster or slower.

However, if one first designs the record (be it a name and address list, inventory list, etc.) by setting fixed lengths to each field in the record, then, if each record contains N fields, it is possible to sort all the records in N ways.

Assuming each record is constructed by concatenating five fields such as a + b + c + d + e, and all records are stored in Array A, it is a simple matter using the machine language sort to quickly sort Array A based on field 'a' as a key. Different routines can be designed to output each record in any desired order of fields. For example, using the MID$ function, a record can be output in the order: a + b + c + d + e. This is only possible, however, if the length of each field on a record by record basis is the same.

With this as a starting point, it is possible then to construct an entirely new record by using the MID$ function to isolate field 'c' for example, and concatenate it with the original record thusly: c + (a + b + c + d + e). Note that the original field 'c' remains, and the new record is now LEN('c') longer than the original record. This new record can now be stored back into Array A which can then be sorted using field 'c' as the sorting key. The last step before output is to strip the extra field 'c' from each record in Array A, and you now have Array A sorted in field 'c' order, but with each record in its original input format.

The following program for 16K Models I or III (cassette only) demonstrates this. It is an elementary mailing list and, as constructed, sorts on the Last name which is the first field in the original input format. However, it will also sort on the Zip Code field, which in this case happens to be the last input field (I leave it to the readers to see how it could also be programmed to sort on any of the other fields, e.g. City, State, etc.). Each record is formatted as follows: Last Name (15 characters); First Name (12 characters); Street Address (20 characters); City (15 characters); State (2 characters); Zip Code (5 characters).

As each field is input, it is measured for maximum length, and, in the case of the Last Name, shortened to proper length if necessary. The other fields are rejected if they are too long, and in the case of the Last Name, First Name, Street Address and City, the fields are padded out with spaces (" ") to the proper length if they are too short. Each field after being input is concatenated to the preceding fields, and upon completion of the entire record, the completed string is stored in Array A.

The print routine makes use of the MID$ function to output each record in the order:

First Name Last Name
Street Address
City, State Zip Code

It also includes part of a search routine which only works on the Last Name; however, there is no reason why it could not be expanded to search on any field and be used as part of an Add/Change/Delete routine.

Sorting by Zip Code is accomplished by isolating the Zip Code field with the MID$ function (in this case I could have used the RIGHT$ function) and creating a new temporary record in the form: Zip; Last name; First name; Address; City; State; Zip. The new record is temporarily stored right back into Array A. After all the records have been so modified, Array A is now sorted by Mr.

(Continued on Page 19)

Sorting (From Page 18)

Emert's trusty machine language routine. Upon completion of the sort, the leading Zip Code fields are stripped off each record, and Array A is now sorted by Zip Code, with each record in its original input format ready for output.

When designing your own record format, one caveat is in order. Although string length is purported to be 255 characters (bytes), in reality you are limited to about 249 when concatenating and storing the strings into arrays. The extra bytes are apparently used by the TRS-80 in its housekeeping chores. Therefore the maximum length of any record you design can be no longer than 249 characters minus Len (longest field you intend to use as a sorting key). This is because, as noted, the length of the temporary working string is its original length plus the length of sorting key field.

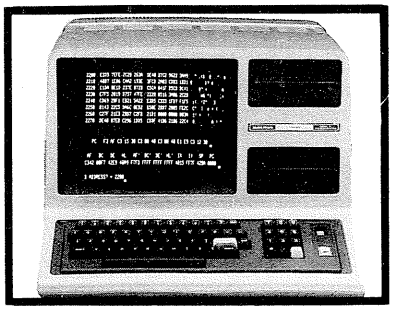
Here is the program. It incorporates a cassette input/output routine to enable the user to continually add new names and addresses to the stored list.

```
Ø CLS
     : PRINT@512, "SUBMITTED BY:   RICHARD W. SAXEN"
     : PRINT@591, "235 PEPPERMINT WAY"
     : PRINT@655, "PORT ORANGE, FL  32Ø19"
     : PRINT@719, "7/2Ø/81"
     : FOR I=1 TO 2ØØØ
     : NEXT
1 CLS
     : PRINT "SHORT WAIT FOR DATA TO BE READ IN"
     : REM FOR MODEL I/III CASSETTE SYSTEMS ONLY -- MEM SIZE
     SHOULD BE SET TO 32512-MACHINE LANGUAGE SORT FROM
     TRS-8Ø NEWS   7/8Ø - THANK YOU ALLAN EMERT!!!
1Ø CLEAR 8ØØØ
     : POKE 16526,Ø
     : POKE 16527,127
2Ø DEFSTR A-H
     :  DEFINT I-Z
     :  DIM A(15Ø), X(2)
     : Z=Ø
     : J=Ø
     : NF=Ø
25 REM 'J IS ARRAY COUNTER; X(2) AND Z ARE FOR SORT; NF IS
     FLAG USED IN SEARCH ROUTINE
3Ø GOSUB 3ØØØØ
2ØØ REM-MENU ROUTINE
21Ø CLS
     : PRINT "TYPE IN THE NUMBER OF YOUR SELECTION."
22Ø PRINT "  1 - eNTER NEW RECORD       2 - STORE RECORDS"
23Ø PRINT "  3 - READ IN TAPE           4 - SORT DATA"
24Ø PRINT "  5 - PRINT DATA             6 - END PROGRAM"
245 PRINT "          7 - RECORD SEARCH"
25Ø INPUT ZZ
     :  REM ZZ IS TEMPORARY INTEGER VARIABLE
26Ø IF ZZ<1 OR ZZ>7 THEN 25Ø
27Ø ON ZZ GOSUB 1ØØØ, 2ØØØ, 3ØØØ, 4ØØØ, 5ØØØ, 6ØØØ, 7ØØØ
28Ø ZZ=Ø
     : GOTO 21Ø
     : REM RETURN TO MENU
1ØØØ REM INPUT ROUTINE
1Ø1Ø F=""
     : FL=""
     : REM INITIALIZE WORKING STRINGS
1Ø4Ø CLS
     : PRINT
     : INPUT "LAST NAME   "; F
     : REM 15 CHARACTER MAXIMUM
1Ø45 IF F="" PRINT "NOTHING ENTERED"
     : FOR ZZ=Ø TO 5ØØ
     : NEXT
     : GOTO 1Ø1Ø
1Ø5Ø IF LEN(F)=>15 THEN F=LEFT$(F,15)
     : REM IF NAME EXCEEDS 15 CHARACTERS, IT WILL BE
     SHORTENED TO EQUAL 15.
1Ø6Ø KL=15-LEN(F)
     : REM COMPUTES BLANKS IN NAME STRING IF NEEDED
1Ø7Ø F=F+STRING$(KL," ")
     : REM F NOW HAS 15 CHRS
1Ø8Ø FL=FL+F
     : F=""
     : REM ADD F TO FL THEN REINITIALIZE F
1Ø9Ø PRINT
     : INPUT "FIRST NAME (MIDDLE INITIAL OPTIONAL)"; F
1Ø95 IF F="" PRINT "NOTHING ENTERED"
     : FOR ZZ=Ø TO 5ØØ
     : NEXT
     : GOTO 1Ø9Ø
11ØØ IF LEN(F)>12 PRINT "TOO LONG - LIMIT TO 12 CHARACTERS"
     : GOTO 1Ø9Ø
111Ø KL=12-LEN(F)
     : F=F+STRING$(KL," ")
     : REM NOW HAS 12 CHRS
112Ø FL=FL+F
     : F=""
     : REM (SEE 1Ø8Ø)
113Ø PRINT
     : INPUT "STREET ADDRESS   "; F
     : REM 2Ø CHARACTER MAX.
1135 IF F="" PRINT "NOTHING ENTERED"
     : FOR ZZ=Ø TO 5ØØ
     : NEXT
     : GOTO 113Ø
114Ø IF LEN(F)>2Ø PRINT "UP TO 2Ø CHARACTERS-REDO"
     : GOTO 113Ø
115Ø KL=2Ø-LEN(F)
     : REM COMPUTE REMAINING BLANKS
116Ø F=F+STRING$(KL," ")
     :  REM NOW HAS 2Ø CHARACTERS
117Ø FL=FL+F
     : F=""
     : REM (SEE 1Ø8Ø)
118Ø PRINT
     : INPUT "CITY/TOWN/ETC   "; F
1185 IF F="" PRINT "NOTHING ENTERED"
     : FOR ZZ=Ø TO 5ØØ
     : NEXT
     : GOTO 118Ø
119Ø IF LEN(F)>15 THEN PRINT "UP TO 15 CHARACTERS-REDO"
     : GOTO 118Ø
12ØØ KL=15-LEN(F)
     :  REM COMPUTE BLANKS
121Ø F=F+STRING$(KL," ")
     :  REM NOW HAS 15 CHARACTERS
122Ø FL=FL+F
     : F=""
     : REM (SEE 1Ø8Ø)
123Ø PRINT
     : INPUT "STATE (2 LETTER ABBREV)  "; F
124Ø IF LEN(F)<>2 THEN PRINT"ERROR-REDO"
     : GOTO 123Ø
125Ø FL=FL+F
     : F=""
     : REM (SEE 1Ø8Ø)
126Ø PRINT
     : INPUT "ZIP CODE    "; F
127Ø IF LEN(F)<>5 THEN PRINT "ERROR-REDO"
     : GOTO 126Ø
128Ø FL=FL+F
     : F=""
     : REM (SEE 1Ø8Ø)
147Ø PRINT
     : PRINT "IF THIS IS LAST ENTRY, ENTER  'YES'"
1475 INPUT "  OTHERWISE ENTER 'NO'"; F
148Ø A(J)=FL
     : J=J+1
     : IF F="NO" GOTO 1Ø1Ø
149Ø RETURN
     :  REM END OF INPUT-RETURNS TO MENU
1999 END
```

(Continued on Page 20)

Sorting (From Page 19)

```
2000 REM * ROUTINE TO STORE RECORDS ON TAPE
2010 PRINT "READY CASSETTE FOR STORAGE"
     : INPUT "PRESS ENTER TO BEGIN"; ZZ
2020 PRINT #-1, J-1
     : REM  J-1 = TOTAL RECORDS
2030 FOR X=0 TO J-1
2040 PRINT #-1, A(X)
2050 NEXT X
2060 RETURN
     :  REM * TO MENU
2999 END
3000 REM * ROUTINE TO READ FILE FROM TAPE
3010 PRINT "READY CASSETTE TO READ IN DATA FROM TAPE"
3020 INPUT "PRESS ENTER WHEN READY"; ZZ
3030 INPUT#-1, J
     : REM J=RECORD COUNT (STORED AS 1ST DATA ITEM FOR USE
     IN SETTING UP ARRAY A(0) TO A(J) )
3040 FOR X=0 TO J
3050   INPUT #-1, A(X)
3060 NEXT X
3070 RETURN
     :  REM * TO MENU
3999 END
     :  REM SORT ROUTINE FOLLOWS
4000 CLS
     : PRINT "TYPE IN YOUR CHOICE"
4010 PRINT
     : INPUT "1. SORT BY LAST NAME   2. SORT BY ZIP CODE";
     SS
4020 IF SS<1 OR SS>2 THEN 4000
4030 IF SS=1 GOTO 4070 ELSE GOSUB 8000
4040 RETURN
     :  REM TO MENU
4060 REM SORT ROUTINE FROM TRS-80 NEWSLETTER 7/80
4070 X(0)=J
4080 X(1)=VARPTR(A(0))
4090 Z=USR(VARPTR(X(0)))
4100 RETURN
     : REM TO MENU
4999 END
     :  REM PRINT ROUTINE BEGINS HERE
5000 CLS
     : FOR JJ=0 TO J-1
5010 C=A(JJ)
     :  REM C IS TEMPORARY STRING USED TO SIMPLIFY
5020 PRINT MID$(C,16,12); LEFT$(C,15)
     :  REM FIRST NAME    LAST NAME
5030 PRINT MID$(C,28,20)
     : REM STREET ADDRESS
5040 PRINT MID$(C,48,15); " "; MID$(C,63,2); " ";
     MID$(C,65,5)
     : REM CITY/STATE/ZIP
5130 PRINT
     : INPUT "PRESS <ENTER> TO CONTINUE"; ZZ
     : CLS
     : REM THIS DISPLAYS ONE RECORD AT A TIME
5135 IF NF=1 THEN NF=0
     : RETURN
     : REM BREAK OUT OF LOOP RE SEARCH-NF IS SEARCH ROUTINE
     FLAG AND IS RESET ON EXIT
5140 NEXT JJ
5150 RETURN
5999 END
6000 PRINT "PROGRAM ENDED - TO RECOVER TYPE IN 'GOTO 210'"
     : END
6999 END
7000 REM SBRT TO SEARCH BY NAME
7005 NF=1
     : REM SET FLAG TO BREAK OUT OF PRINT LOOP(5135)
7010 CLS
     :  INPUT "TYPE IN LAST NAME"; F
7015 IF F="" THEN PRINT "NOTHING ENTERED"
     : GOTO 7080
7020 IF LEN(F)>15 THEN GOTO 7070
7025 KL=15-LEN(F)
7030 F=F+STRING$(KL," ")
     :  REM PAD OUT TO 12 SPACES
7040 FOR JJ=0 TO J
7050 IF F=LEFT$(A(JJ),15)
     : CLS
     : GOSUB 5010
     : RETURN
7060 NEXT JJ
7070 PRINT
     : PRINT "NO SUCH LISTING"
7080 NF=0
     : FOR ZZ=0 TO 500
     : NEXT ZZ
     : RETURN
7999 END
8000 REM SBRT TO MAKE ZIP CODE THE SORT POINTER
8010 CLS
     :  PRINT "PROCESSING"
8020 FOR JJ=0 TO J-1
     : C=A(JJ)
8030 A(JJ)=MID$(C,65,5)+C
     :  REM ZIP CODE PLACED IN FRONT HERE TO BE USED AS KEY.
     ORIGINAL ZIP CODE IS STILL IN PLACE ALSO!
8040 NEXT JJ
8050 GOSUB 4070
     : REM SORT BY ZIP AND RETURN TO 8060 TO RESTORE
8060 FOR JJ=0 TO J-1
     : C=A(JJ)
     : REM RESTORE ORIGINAL ORDER
8070 A(JJ)=MID$(C,6)
     : REM STRIP OFF ZIP CODE FROM FRONT
8080 NEXT JJ
     : RETURN
     : REM TO MENU
29999 END
     : REM ALLEN EMERT'S MACHINE LANGUAGE SORT FOLLOWS
30000 REM LIST FOR 16K (ENTRY = &H7F00)
30010 REM MEM SIZE = 32512
30020 DATA 205, 127, 10, 94, 35, 86, 237, 83, 19, 127, 35,
      94, 35, 86, 237, 83
30030 DATA 213, 127, 33, 0, 0, 34, 211, 127, 237, 91, 211,
      127, 203, 59, 175
30040 DATA 203, 58, 48, 2, 203, 251, 237, 83, 211, 127, 122,
      179, 200, 42, 19
30050 DATA 127, 237, 82, 34, 207, 127, 33, 0, 0, 34, 205,
      127, 42, 205, 127, 34
30060 DATA 203, 127, 42, 203, 127, 237, 91, 211, 127, 25,
      34, 209, 127, 235, 33
30070 DATA 0, 0, 25, 25, 25, 229, 237, 91, 203, 127, 33, 0,
      0, 25, 25, 25, 237
30080 DATA 75, 213, 127, 9, 235, 225, 9, 229, 213, 14, 0,
      126, 71, 26, 184, 48
30090 DATA 3, 14, 1, 71, 175, 176, 40, 25, 197, 19, 35, 78,
      35, 70, 197, 225
30100 DATA 235, 78, 35, 70, 197, 225, 193, 26, 150, 56, 10,
      32, 39, 19, 35, 16
30110 DATA 246, 203, 65, 32, 31, 209, 225, 6, 3, 78, 235,
      126, 113, 235, 119
30120 DATA 35, 19, 16, 246, 42, 211, 127, 235, 42, 203, 127,
      175, 237, 82, 34
30130 DATA 203, 127, 48, 144, 24, 2, 209, 225, 42, 205, 127,
      17, 1, 0, 175, 25
30140 DATA 34, 205, 127, 237, 91, 207, 127, 237, 82, 218,
      58, 127, 195, 24, 127
30150 N=0
30160 FOR I=1 TO 203
30170   READ Y
30180   N=N+Y
30190   POKE I+32511, Y
30200 NEXT
30210 IF N<>22393 THEN PRINT "DATA ERROR"
     : END
31000 RETURN
```



Level I

Model I and III

Good Old Days

Warren Whitcomb Port Lavaca, Texas

I thought I would respond to your request for input from Level I users.

I was a happy Level One user who moved to Level II but was lost at Level I.

In the computer world it is impossible to grasp the idea that there are people and certified teachers who do not know how to use the cassette tape recorder. The children in Kindergarten through fifth grade know how to use the recorders, but most teachers never had an opportunity to learn.

In Level I when data was stored one had no idea if it recorded or not. I also had Level I questions which were not in the book. Pepper at the Victoria, Texas store had to answer such problems:

1. Why does the video, sitting on my CB, say "what?" when a CB pal says "Break 19"?
2. Is the top of my video screen a good place to put my computer tapes?
3. When someone turns on a light why does the computer say "What"?

I read the newsletter with wonder.

I used micromusic with two music classes: one had 29 learners and the other one had 33 learners. All wanting to use the computer at one time. At the time I thought "would not the network 2 controller be great," then I saw the process.

"First a 32K or 48K Model III disk system, then one to sixteen Model I (Level II?) computers. . . . Do the following 12 steps twice." page 18 July 1981.

Now picture this in a 20 to 45 minute class period with 33 people pushing keys. Then it says "At each system follow these steps."

As my Model I Level I said "What and How? or Sorry?" Them were the good old days.

Compu-Artist

Brian Durham Fort Worth, Texas

I am thirteen years old and I own a Level I. Several of my friends own a TRS-80 (Level I and Level II) and we exchange many of our own programs. My friend, David Henson, gave me a graphics program that he had written. I added several minor changes and I am sending it to the Newsletter. This program is usable for Level I or Level II.

```
1 REM *** COMPU-ARTIST ***
2 REM BY DAVID HENSEN (MINOR CHANGES BY BRIAN DURHAM)
10 CLS
12 A=RND(128)-1
     : B=RND(128)-1
     : C=RND(48)-1
14 D=RND(48)-1
     : E=RND(20)
20 FOR X=A TO C STEP E
22 FOR Y=C TO D STEP E
24 SET(X,Y)
26 NEXT Y
     : NEXT X
28 GOTO 12
```

H and T Pads

Mike Stark San Diego, CA

In the July newsletter you asked, where are the Level I users? Well here is a Level I and Level II Model I user with a program that might be of interest to anyone who works with audio. I work for a radio station, and matching the various pieces of equipment with differing impedances and "best" operating levels is a day in and day out task. A lot of the matching is done with amplifiers and even more is done with resistive pads. To optimize the signal to noise ratio in a very long audio chain both balanced and unbalanced pads might be needed. The program I have written is designed to both calculate balanced H pads and unbalanced T pads. It also displays the appropriate resistances as the pads would be constructed. I hope this might be of use to some of your readers.

```
10 A=0
     : B=0
     : C=0
     : D=0
     : E=0
     : F=0
     : G=0
     : H=0
     : I=0
     : J=0
     : K=0
     : L=0
     : M=0
     : N=0
     : O=0
     : P=0
12 Q=0
     : R=0
     : S=0
     : T=0
     : U=0
     : V=0
     : W=0
     : X=0
     : Y=0
     : Z=0
20 REM * MIKE STARK 4264 46TH ST. SAN DIEGO, CA. 92115*
25 REM * * VERSION 3.0 * *
30 CLS
     : P. "    H & T PADS"
     : P.
40 I. "WHAT IS THE FIRST IMPEDANCE"; F
50 I. "WHAT IS THE SECOND IMPEDANCE"; G
60 I. "HOW MUCH ATTENUATION IN DB DO YOU WANT"; Q
70 CLS
     : P. "WELL LET ME SEE...THOSE PADS WOULD LOOK LIKE
     THIS: "
80 X=10
     : Y=(Q/20)
     : GOS. 3100
     : K=P
     : GOS. 2000
     : GOS. 1000
90 R=F*(D+1-(2*K/S))/(D-1)
     : T=G*(D+1-(2*K*S))/(D-1)
     : U=G*(2*K*S/(D-1))
100 M=R/2
     : N=T/2
     : CLS
```

(Continued on Page 22)

H and T Pads (From Page 21)

```
15Ø GOS. 4ØØØ
2ØØ P. AT 15, "ATTENUATION"; Q; "DB"
3ØØ P. AT 147, "T PAD";
     : P. AT 2Ø5, R
     : P. AT 215, T
     : P. AT 273, U
4ØØ P. AT 532, "H PAD";
     : P. AT 59Ø, M
     : P. AT 6ØØ, N
     : P. AT 658, U
     : P. AT 718, M
     : P. AT 728, N
5ØØ P. AT 768, F; "OHMS INPUT IMPEDANCE  "; G; "OHMS OUTPUT
    IMPEDANCE"
999 END
1ØØØ X=K
     : Y=2
     : GOS. 31ØØ
     : D=P
     : RET.
2ØØØ X=(F/G)
     : Y=.5
     : GOS. 31ØØ
     : S=P
     : RET.
3Ø25 IF X=Ø T. Y=Ø
     : RET.
3Ø3Ø IF X>Ø T. 3Ø45
     : ST.
3Ø45 Y=X/4
     : Z=Ø
3Ø55 W=(X/Y-Y)/2
3Ø6Ø IF W=Ø T. RET.
     : IF W=Z T. RET.
3Ø7Ø Y=Y+W
3Ø75 Z=W
3Ø8Ø G. 3Ø55
31ØØ E=Ø
     : P=1
     : IF Y=Ø T. RET.
     : IF X<Ø T. 3114
     : REM SQR
3112 G. 3116
3114 IF I.(Y)=Y T. 3122
3116 IF X<>Ø T. 313Ø
     : P=P*E
     : RET.
3122 P=1-2*Y+4*I.(Y/2)
     : X=-X
313Ø GOS. 317Ø
3135 X=Y*L
314Ø GOS. 324Ø
     : P=P*E
     : RET.
317Ø IF X<=Ø T. ST.
     : REM LOG
318Ø A=1
     : B=2
     : C=.5
     : E=Ø
3188 IF X>=A T. 32Ø2
319Ø IF X<C T. 32Ø8
3192 X=(X-.7Ø71Ø7)/(X+.7Ø71Ø7)
3194 L=X*X
3196 L=(((.598979*L+.961471)*L+2.88539)*X+E-.5)*.693147
3198 IF A.(L)<1E-6 T. 322Ø
32ØØ RET.
32Ø2 X=C*X
     : E=E+A
     : G. 3188
32Ø8 X=B*X
     : E=E-A
     : G. 319Ø
322Ø L=Ø
     : RET.
324Ø L=I.(1.4427*X)+1
     : IF L<127 T. 3258
     : ST.
3254 E=Ø
     : RET.
3258 E=.693147*L-X
326Ø A=1.32988E-3-1.41316E-4*E
3262 A=((A*E-8.3Ø136E-3)*E+4.1657E-2)*E
3264 E=(((A-.166665)*E+.5)*E-1)*E+1
3266 A=2
3268 IF L>Ø T. 3276
327Ø A=.5
     : L=-L
3274 IF L=Ø T. RET.
3276 F. X=1 TO L
     : E=A*E
     : N. X
     : RET.
4ØØØ IF R<=Ø G. 5ØØØ
4Ø1Ø IF T<=Ø G. 5ØØØ
4Ø2Ø IF U<=Ø G. 5ØØØ
4Ø3Ø IF M<=Ø G. 5ØØØ
4Ø4Ø IF N<=Ø G. 5ØØØ
4Ø5Ø RET.
5ØØØ P. "THAT IS NOT ENOUGH ATTENUATION TO MATCH THOSE
    IMPEDANCES"
5Ø1Ø G. 4Ø
6ØØØ END
     : REM * AUDIO LEVEL MATCHING PADS
```

Bingo

Arie W. Vander Pol Turlock, CA

In response to your plea in the July 1981 newsletter I am herewith submitting the very first program I wrote on my Level I, Model I TRS-80 for consideration for publication in the newsletter. It is a program which randomly draws the 75 BINGO numbers one at a time and displays them on a board. It is based on the 52 card-drawing routine described in the Level I manual. We have a lot of fun with it and it eliminates the possibility of a player who is also drawing the numbers from getting an advance peek at the number. The TRS-80 never cheats! It could be made a little shorter by spinning up the array in a For-Next loop but then it is already fairly short. Suit yourself.

I am still programming in Level I but hope to upgrade to Level II soon (any sales coming up?). At present I am writing my third attempt at a family budget program. I keep running out of memory on my 4K machine. However, I take this as a challenge to try and shoehorn all I want to do into a 4K memory in Level I language. It does not detract at all from the fun of programming. One just has to be more creative. Upgrading to 16K would be the easy way out!

One note on the BINGO program, a "NO" answer to the question at line 3010 sends the program into a loop and you will have to use the "Break" key to start over. Please feel free to use the program at your discretion.

BINGO NIGHT

```
5 CLS
1Ø PRINT AT 16, "TRS-8Ø BINGO NIGHT"
15 PRINT AT 192, "B"
2Ø PRINT AT 32Ø, "I"
25 PRINT AT 448, "N"
3Ø PRINT AT 576, "G"
35 PRINT AT 7Ø4, "O"
4Ø FOR P=12 TO 36 STEP 6
45  FOR N=1 TO 127
5Ø   SET(N,P)
55   NEXT N
6Ø  NEXT P
```

(Continued on Page 23)

Bingo (From Page 22)

```
7Ø FOR C=1 TO 75
     : READ A(C)
     : NEXT C
8Ø DATA 1, 2, 3, 4, 5, 6, 7, 8, 9, 1Ø, 11, 12, 13, 14, 15
9Ø DATA 16, 17, 18, 19, 2Ø, 21, 22, 23, 24, 25, 26, 27, 28,
     29, 3Ø
1ØØ DATA 31, 32, 33, 34, 35, 36, 37, 38, 39, 4Ø, 41, 42, 43,
     44, 45
11Ø DATA 46, 47, 48, 49, 5Ø, 51, 52, 53, 54, 55, 56, 57, 58,
     59, 6Ø
12Ø DATA 61, 62, 63, 64, 65, 66, 67, 68, 69, 7Ø, 71, 72, 73,
     74, 75
13Ø P=75
14Ø V=RND(75)
145 M=5
15Ø IF A(V)=Ø GOTO 14Ø
16Ø IF A(V)<=15 GOTO 3ØØ
17Ø IF A(V)<=3Ø GOTO 4ØØ
18Ø IF A(V)<=45 GOTO 5ØØ
19Ø IF A(V)<=6Ø GOTO 6ØØ
2ØØ IF A(V)<=75 GOTO 7ØØ
21Ø A(V)=Ø
22Ø P=P-1
23Ø GOSUB 3ØØØ
26Ø IF P<>Ø GOTO 14Ø
27Ø P. A. 832, "ALL THE NUMBERS HAVE BEEN DRAWN--DUMMY"
299 END
3ØØ X=191
31Ø GOTO 4ØØØ
4ØØ X=259
41Ø GOTO 4ØØØ
5ØØ X=327
51Ø GOTO 4ØØØ
6ØØ X=395
61Ø GOTO 4ØØØ
7ØØ X=463
71Ø GOTO 4ØØØ
998 P.A. 896, "BINGO!--WE HAVE A WINNER!!!"
999 GOTO 999
3ØØØ Y=1
     : N=Ø
3ØØ5 P. A. 896,
3Ø1Ø IN. "DO YOU WISH ME TO DRAW ANOTHER NUMBER (Y/N)"; Z
3Ø2Ø IF Z=Ø GOTO 998
3Ø25 P.A. 896,"
3Ø3Ø RETURN
3999 END
4ØØØ P.A. X+A(V)*4, A(V);
4Ø1Ø GOSUB 6ØØØ
4Ø2Ø M=M-1
4Ø3Ø IF M=Ø GOTO 21Ø
4Ø4Ø GOTO 5ØØØ
4999 END
5ØØØ P. A. X+A(V)*4, "  ";
5Ø1Ø GOSUB 6ØØØ
5Ø2Ø GOTO 4ØØØ
5999 END
6ØØØ FOR I=1 TO 5ØØ
6Ø1Ø NEXT I
6Ø2Ø RETURN
6999 END
```

Binary Search (From Page 7)

This routine looks for the target string (TR$) in an array of sorted strings (the A$ array with N members). A match does not have to be exact. If there are two or more strings which match the target string, all of them are printed out, and the user is given a choice. Furthermore, in my program there is no vagueness about the size of the variables 'N' and 'N1', as there is with Mr. Belemecich's routine.

Note that line 30 should contain the routine or a call to the routine that loads the A$ array with values. This same routine must also assign N a value equal to the number of strings in A$.

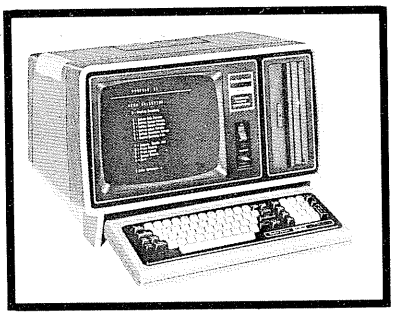
Here is my routine:

```
1Ø REM -- BINARY SEARCH ROUTINE, F. F. ROWE, OAK RIDGE,
     TENN.
2Ø CLEAR 5ØØØ
     : DEFINT A-Z
     : CLS
     : DIM A$(3ØØ)
3Ø REM -- MORE PROGRAM STATEMENTS HERE (READ IN STRINGS,
     ASSIGN VALUE TO N)
4Ø PRINT
     : INPUT "STRING TO SEARCH FOR"; TR$
5Ø L=LEN(TR$)
     : LO=1
     : HI=N
6Ø MD=(LO+HI)/2
     : IF TR$=LEFT$(A$(MD),L) THEN 15Ø
7Ø IF TR$>LEFT$(A$(MD),L) THEN LO=MD+1 ELSE HI=MD-1
8Ø IF HI>=LO THEN 6Ø
9Ø REM -- COME HERE IF STRING NOT FOUND
1ØØ IF LO>N THEN LO=N 'MAKE SURE WE ARE STILL IN ARRAY
11Ø IF HI<1 THEN HI=1
12Ø PRINT "STRING NOT FOUND BETWEEN '" A$(HI) "' AND '"
     A$(LO) "'"
     : GOTO 4Ø
15Ø REM -- FOUND A MATCH. LOOK FOR OTHERS
16Ø FOR J=MD-1 TO 1 STEP -1
     : IF TR$<>LEFT$(A$(J),L) THEN 17Ø ELSE NEXT
17Ø LO=J+1
     : FOR J=MD+1 TO N
     : IF TR$<>LEFT$(A$(J),L) THEN 18Ø ELSE NEXT
18Ø HI=J-1
     : IF LO=HI THEN J=LO
     : GOTO 22Ø
19Ø REM -- FOUND MORE THAN ONE MATCH. GIVE USER A CHOICE.
2ØØ FOR J=LO TO HI
     : PRINT J-LO+1; A$(J)
     : NEXT
21Ø J=Ø
     : INPUT "WHICH OF THESE"; J
     : IF J=Ø THEN 4Ø ELSE J=J+LO-1
22Ø REM -- MATCH!!!
23Ø PRINT "CHOSEN STRING
     :  "; A$(J)
24Ø REM -- OTHER PROGRAM STATEMENTS WOULD FOLLOW
```

SAMPLE PROGRAM RUN

Assume that array A$ contains the following 5 elements (N = 5):

```
A$(1)="AA"
A$(2)="AAA"
A$(3)="AB"
A$(4)="GHR"
A$(5)="XYZ"

STRING TO SEARCH FOR? A
1 AA
2 AAA
3 AB
WHICH OF THESE? 2
CHOSEN STRING:  AAA

STRING TO SEARCH FOR? AB
CHOSEN STRING:  AB

STRING TO SEARCH FOR? B
STRING NOT FOUND BETWEEN 'AB' AND 'GHR'

STRING TO SEARCH FOR? X
CHOSEN STRING:  XYZ
```



Model II

Product Line Manager's News

The first joint activity between Datapoint and Tandy was announced in November, 1979. Texas Peripherals began production in July, 1980 and is now the primary supplier to Tandy of 5 1/4" floppy disk drives, and to Datapoint of 8" disk drives.

On September 9, 1981 we announced a joint activity of a different type and far more important to both companies. Datapoint has been the pioneer in the development of a networking system for small to medium scale computers. Tandy on the other hand has sold more personal computers than anyone. Two frequently asked questions have been "How can we use multiple TRS-80s together?" and "What is the logical upgrade for TRS-80 users?"

We now have an answer—ARCNET.

Tandy will utilize the Datapoint ARCNET (Attached Resource Computer NETwork) protocols---and a new Datapoint LSI circuit network interface component—to provide high speed, very low cost common data bases and peripherals systems using common-resource networking. Datapoint developed ARCNET as a part of its highly reliable, field proven ARC system, in use since 1977.

The network is built from four relatively inexpensive hardware components---thanks to Datapoint's LSI design. First, the ARCNET interface card is needed in each computer on the network to enable it to communicate with the network—this card installs in existing card slots in the rear of the Model II. The card will retail for about $400 and be built by Texas Peripherals. This price is substantially less than competitive network offerings. The second element is coaxial cable which you buy by the foot to meet your needs. It is estimated that 30 feet of cable with connectors will cost about $30. Third, you need a junction box or port hub for up to four processors that will cost less than $200, or (fourth) for larger systems and distances an active junction box or port hub that will cost about $2000. Literally, this is all you need to simply plug TRS-80s together. In fact, you can just keep on plugging until you have up to 255 processors in the system. A very big computer resource from very small computers.

The ARCNET operates at 2.5 million bytes per second so that it is transparent to the user, i.e., the user normally does not know that anyone else is on the network. From a technical viewpoint the ARCNET local network for Tandy products is identical to the Datapoint network component of ARC. The interface card installed in the TRS-80 Model II uses the same LSI, the active and passive hubs are the same as is the cable. The software is very similar but has been configured to operate with our standard TRSDOS disk operating system to provide software compatibility with existing Model II software and hardware. A Tandy and a Datapoint ARCNET can communicate by the use of emulation software.

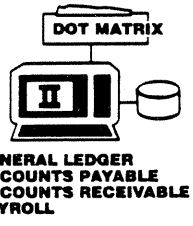
ARCNET will permit multiple TRS-80 Model II computers, and future TRS-80 computers, to be linked into effectively large scale networks or systems never before possible with personal computers. In fact, ARCNET is the lowest cost network implementation for the office of the future. The owner of a TRS-80 Model II does not obsolete his investment. His existing Model II's are easily modifiable to become part of the network.

Lets trace how a typical TRS-80 user could grow using ARCNET. Many TRS-80 users are now doing their accounting on a floppy disk based TRS-80 Model II system with a printer.

1 ACCOUNTING DEPARTMENT

This requires a sizeable quantity of floppy disks to change programs and to store data. By adding our new 8.4 megabyte hard disk a single Model II can operate all the standard accounting functions of General Ledger, Accounts Receivable and Payable and Payroll without the need for this quantity

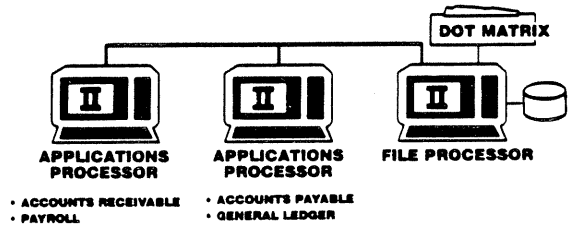
2 ACCOUNTING DEPARTMENT



of floppy diskettes. As the business grows, however, the time required to enter all the data will exceed the capability of one operator at one keyboard. Today, while a second Model II could be added, that would not be a good solution as the data files on the hard disk and the printer could not be shared by both computers.

The solution is ARCNET. By installing an ARCNET interface board in the existing TRS-80 it becomes a file processor and by adding two more TRS-80s as applications processors, a low cost interconnection hub and some cable we have a starter ARCNET system.

3 ACCOUNTING DEPARTMENT



The accounting tasks will now be split between the two applications processors, each of which is a TRS-80 computer running it's own program but sharing the hard disk files and the printer. Of course you may not need the built-in floppy disks on all the TRS-80s in the network, so we will provide a version of the Model II without disk. The accounting department has now doubled its processing ability at a cost significantly less than adding another entire TRS-80 system and without any reprogramming.

Since ARCNET can be reconfigured by the user at any time by plugging in another TRS-80, as you exceed the capacity of this network you can simply add another TRS-80 application processor or file processor.

(Continued on Page 25)

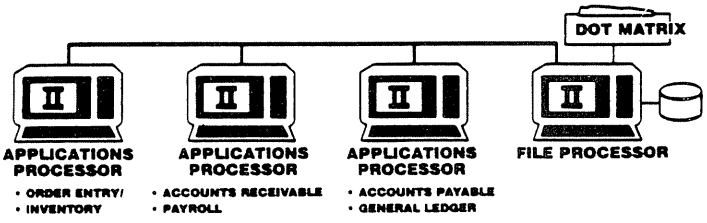
Model II (From Page 24)

In concept and operation ARCNET is very similar to Ethernet but, unlike Ethernet, ARCNET is based on a proven, reliable technique with over four years of field operation. Today, adding one computer to an Ethernet system would cost 3 to 5 times the cost of the same addition to ARCNET. We selected the ARCNET local network because of it's proven ability to handle large business networked applications using hundreds of processors.

Our small business, who happens to be a Widget distributor, now decides that accounting is under control and it's time to get the inventory on the computer.

4

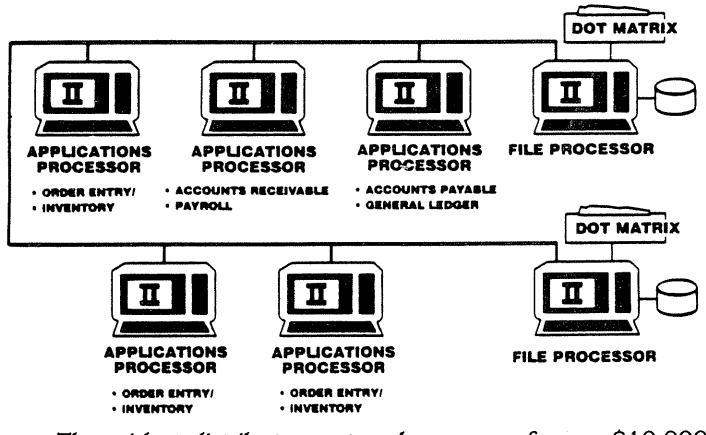
ACCOUNTING DEPARTMENT



By plugging in one more Model II our distributor adds order entry/inventory control. At this point the total cost of this system is still less than $20,000. As the widget business thrives, more TRS-80 work stations are needed to handle the increased order volume. Two more TRS-80s can be added to the network but by now the size of the files and the printing volume pass the ability of the original file processor. Another TRS-80 file processor with hard disk and printer are also added and our final configuration looks like this.

5

ACCOUNTING DEPARTMENT

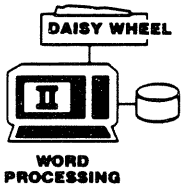


The widget distributor system has grown from a $10,000 system to about a $45,000 system in easy, affordable steps. The additional equipment was added without down time as any unit in the ARCNET system can be added or removed without affecting the system. And since the TRS-80s in the system are interchangeable, the loss of any one unit for service does not affect the ability of the system to operate. Perhaps most important of all, this growth has been accomplished without rewriting any software. In fact all the software needed for this system is in our new RSC-6 computer catalog. The system as shown has 5 independent applications processors which can not only be used for the specific applications shown but can be used interchangeably for any of the applications shown when required. There are about 19 megabytes of disk storage, which can be increased to 76 megabytes without adding any additional file processors, and there are two 60 line per minute printers.

Let's go back to our original starter TRS-80 system but now it is in an office of a large corporation as a word processor. The hard disk is the same but the printer is a letter quality daisy wheel.

6

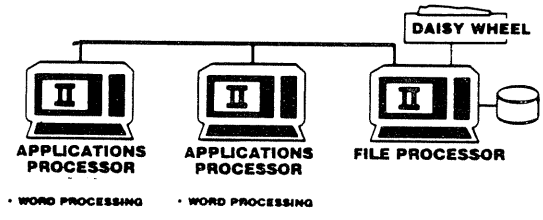
ADMINISTRATIVE OFFICE



Typically word processing is very keyboard intensive with only light duty printing. By going to an ARCNET local network, additional secretaries can have TRS-80 workstations and share the same daisy, wheel printer which can be equipped with an automatic paper sheet feeder.

7

ADMINISTRATIVE OFFICE

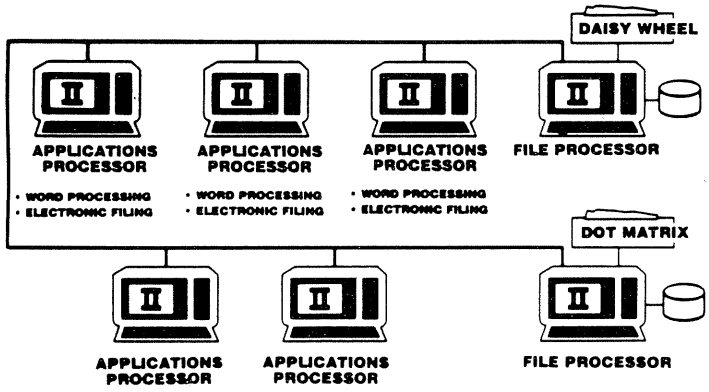


Another application which could be used on this network is electronic filing, which would be available to all the TRS-80 workstations. Again we have the software today to support these applications and both our word processing and electronic filing software includes a high degree of security to prevent unauthorized access. Clearly TRS-80 and ARCNET can now be integral components of the office of the future.

Just as with our widget distributor this office system can grow as requirements change, and a typical growth pattern would be to add a second file processor as the number of workstations increase, but this time a high speed dot matrix printer is added to produce draft documents to leave the letter quality printer free to produce final documents.

8

ADMINISTRATIVE OFFICE



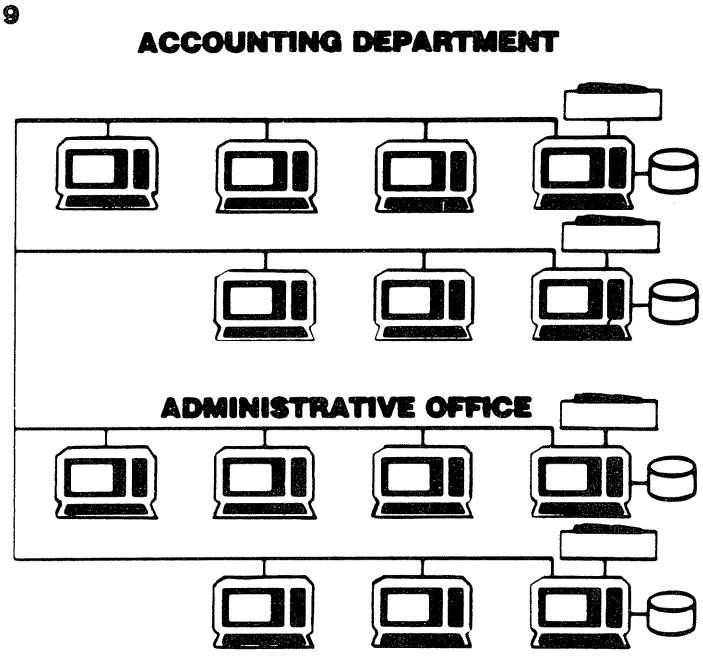
(Continued on Page 26)

Model II (From Page 25)

In the system shown five workstations share the resources of the two hard disks and the two printers. Of course additional workstations can be added at any time.

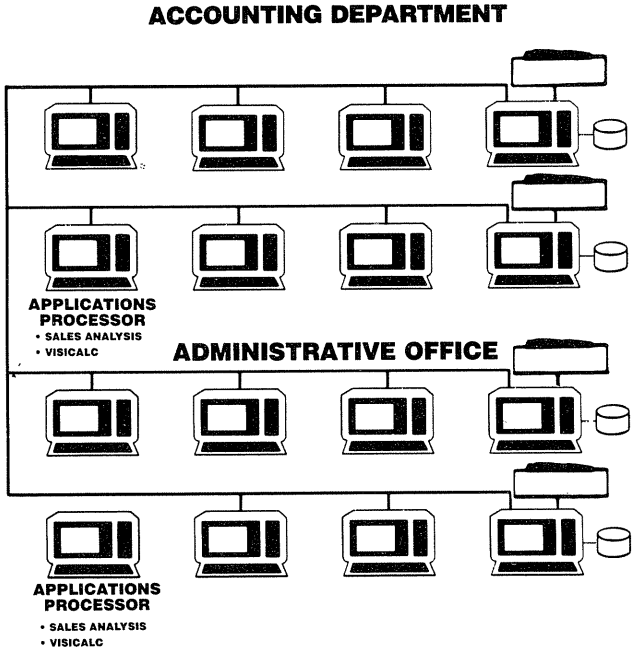
If this office was not in a large company but was the office of our widget distributor the two systems could be connected into one network for only the cost of the cable.

9



The entire company now has a network with a common database accessible to all users but with security restrictions as needed. To make use of this database let's go one more step and add two more TRS-80 workstations for the sales manager and president.

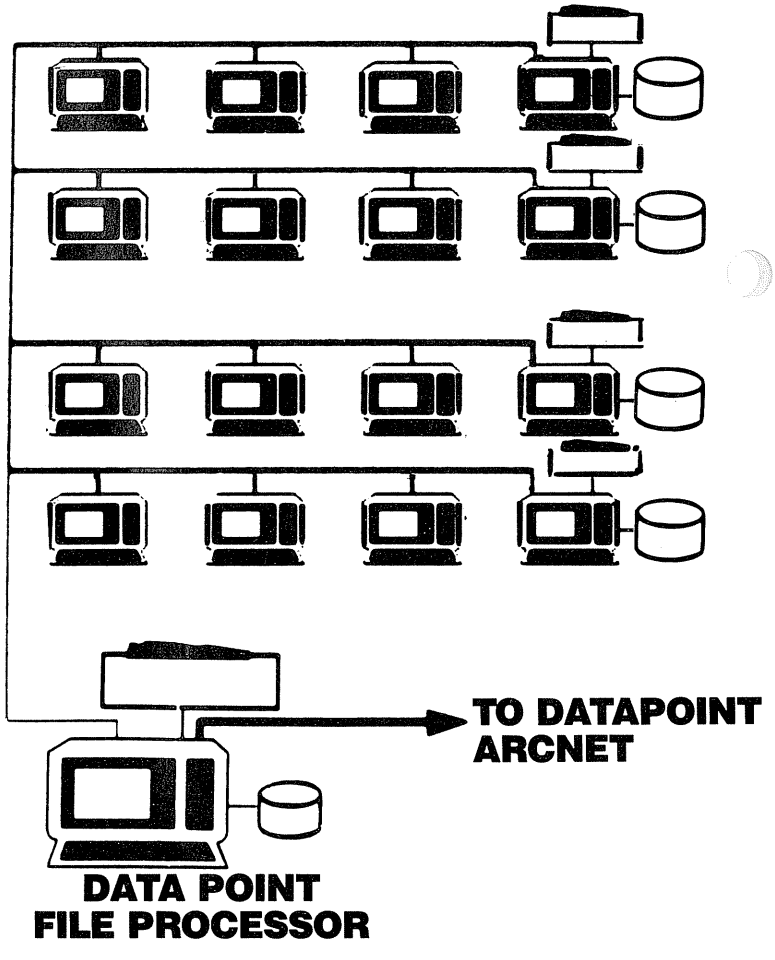
10



These stations can operate our sales analysis program which can query both the order entry and the accounts receivable data bases to provide customer, salesman, or product sales results. They make the power of VisiCalc available to the manager for forecasting and planning as well as making word processing available to ease writing of the keynote speech for the annual convention of widget distributors. And next year we will offer an electronic mail system to help make this a paper-less company.

And if our widget distributor should grow to need even more computing power, the ARCNET network can also include Datapoint file processors and computers. This means you can now add common peripherals like Datapoint's 137 megabyte disk and 900 line per minute printers to your TRS-80 system, or in reverse you may add TRS-80 application processors to your Datapoint network. Obviously, the size and power of the TRS-80 system has taken a quantum leap forward and Datapoint users have new flexibility in system configuration.

11



In addition the Model II, with it's bisynchronous communications ability, can also serve to tie an ARCNET local network to an IBM or other mainframe.

To sum up the advantages of ARCNET:

- It gives both our existing base of TRS-80 owners as well as new buyers a clear path to expand as their needs dictate without obsoleting of present software or hardware and with the assurance that our future hardware and software will be compatible.

Model II (From Page 26)

- It provides a common data base to all users allowing implementation of sophisticated software to help managers monitor their business.
- It provides more computing power than is possible with a minicomputer of similar cost.
- It provides system redundancy since the failure of any one unit will not halt operations.
- It can be installed at a low cost yet its high speed reliable operation is already well proven.
- And most important it provides an easy growth path to a large powerful data processing system in affordable steps.

In the decade of the 80's, networking of computers will become commonplace. We believe the Datapoint networking experience has resulted in a reliably proven, low cost networking system which will speed the implementation of networks into common use. Therefore, we will implement ARCNET in some of our future product offerings which implies a high degree of compatibility between these future products and the TRS-80 Model II.

Initial deliveries of TRS-80 ARCNET will be in the middle of 1982. While this announcement is earlier than normal, we think it's desirable for two reasons—one, clearly it indicates Tandy's commitment to high speed, low cost networking, and two, it reveals a growth pattern for the TRS-80 Model II computer—the best selling computer in its class.

Model II Bugs, Errors, and Fixes

26-4703 RSCOBOL 1.3b on TRSDOS 2.0a

See Model I/III Bugs, Etc. for information on calculating the RAM space needed to run a COBOL program.

Any operation other than OPEN (such as READ, WRITE, CLOSE) on a file which has not been opened, will not display an error message. Instead you will be returned directly to TRSDOS.

To fix this problem (On RSCOBOL version 1.3b on TRSDOS 2.0a ONLY!), build the following DO file (If you attempt to patch directly from a TRSDOS command line, you have no audit trail to check in case of trouble):

```
BUILD PATCHCOB [ENTER]

PATCH RUNCOBOL A=45DB, F=FD2A3A4B, C=C3Ø87ØØØ [ENTER]
[ENTER]
PATCH RUNCOBOL A=7ØØ8, F=ØØØØØØØØØØØØØØØØØØØØ,
                       C=FD2A3AA4FD7E1EFEØ2DA [ENTER]
[ENTER]
PATCH RUNCOBOL A=7Ø12, F=ØØØØØØØØØØØØØØØØØØØØ,
                       C=2846FEØ5D22846C3DF45 [ENTER]
[ENTER]
PATCH RUNCOBOL A=6E4C, F=7C6D, C=A86F [ENTER]
[ENTER]
PATCH RUNCOBOL A=61F4, F=6E, C=7Ø [ENTER]
[ENTER]
PATCH RUNCOBOL A=6FA6, F=ØØØØØØØØØØØØØØØØØØØØØØØØ,
                       C=ØØØEØØ7ØØØØECØØ138ØØØ7ØØ [ENTER]
[ENTER]
PATCH RUNCOBOL A=6FB2, F=ØØØØØØØØØØØØØØØØØØØØ,
                       C=Ø9ØØØCØØØ9ØØØ6ØØØ3ØØ [ENTER]
[ENTER]
[ENTER]
```

As you are building the DO file, check it very carefully to ensure that everything is correct. You may want to check a printout of the file.

After completing the patches (on a BACKUP of your COBOL disk) by executing:

```
DO PATCHCOB [ENTER]
```

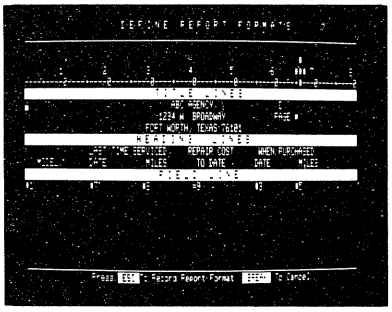
you will now observe an Error Code 91 if you try to perform an operation on a file which has not been OPENed.

26-4705 BASIC Compiler

In some cases an ISAM malfunction is occurring. It is possible, after calculating the record length and the number of records in a file, to not find a record which is actually in the file. The problem has been reported as OPENing two binary files and two ISAM files at the same time. An OVERFLOW error persists regardless of the stack size or record length. Another manifestation of the problem is OPENing WRITEing, and CLOSEing alternately between two ISAM files. In this case, another form of the error will occur after several records have been written.

To correct the problem, BUILD a DO file to make the following patches (see the example of building a DO file to patch RUNCOBOL, or check your Model II owner's manual for information on BUILD and DO.):

```
PATCH RSBASIC/LIO A=3666, F=2Ø24, C=ØØØ3
PATCH RSBASIC/LIO A=369B, F=ØØ37, C=4256

PATCH RSBASIC/OLF R=92,  B=12Ø, F=ØØØØØØØØØØ, C=DDE1C3Ø956
PATCH RSBASIC/OLF R=155, B=67,  F=8B35, C=7756
PATCH RSBASIC/OLF R=155, B=1Ø3, F=2Ø24, C=ØØØ3
PATCH RSBASIC/OLF R=155, B=156, F=ØØ37, C=4256
```



Profile

SMALL COMPUTER COMPANY

Written for the users of PROFILE II and PROFILE +
PROFILE II Editor
P.O. Box 2910
Fort Worth, Texas 76113-2910

READING PROFILE II FILES FROM BASIC

We're frequently asked about PROFILE II data formats—how to access them from BASIC programs.

You might want to do this for many reasons. In fact, for every time you've ever said, "I sure wish PROFILE II could . . . .," there may be a BASIC program waiting to be written that can greatly enhance the usefulness of your system.

Here are some typical examples:

1) Use PROFILE II to maintain lookup files which a BASIC program will read, such as a zip code table for postal rates, or catalog numbers for text descriptions.
2) Support a master/detail relationship between two PROFILE II files with a batch posting program. For example, you could post new orders into customer and inventory files at the end of each day.
3) Perform any sort of global editing on PROFILE II records.
4) Apply to PROFILE II records the elaborate tests and calculations that are so easy for BASIC to do; the results can be stored back into the PROFILE II records.

Actually, PROFILE II files are very easy to use from BASIC—they can be read as ordinary "direct" (random access) files. (We'll assume that you're familiar with BASIC programming under TRSDOS, as well as the information presented in the PROFILE II manual.)

First, all data is maintained in ASCII. Within each PROFILE II segment, the records consist simply of one ASCII field followed by the next, just as you define them with the Create option.

When you create a file using option 1 at the Main Menu, you provide a descriptive name and a length for each field. These fields are stored in your PROFILE II records in exactly the sequence in which you define them. Let's take this example:

FIELD NUMBER	FIELD HEADING	FIELD LENGTH
01	Account Number	00006
02	Last Name	00015
03	First Name	00012
04	Balance Due	00009
05	Remarks	00043

These five items can be fielded in a BASIC program like this:

```
FIELD x, 6 AS AC$, 15 AS L$, 12 AS F$, 9 AS B$, 43 AS R$
```

Simply follow the hardcopy of your PROFILE II file, assigning the correct length to each variable.

Within each field, you'll find the data aligned left and padded on the right with blanks, or aligned right and padded left in the case of numeric data. This will depend on the field type used for data entry in your PROFILE II screen format.

If you're writing to a PROFILE II record from BASIC, you can use LSET or RSET to align your data appropriately within the field. In our example, "Balance Due" is probably a dollars-and-cents item which should be aligned right with RSET.

As you probably already know, the four possible segments of a PROFILE II file have the following filenames:

Segment 1: xxx/KEY
Segment 2: xxx/DAT
Segment 3: xxx/DA2
Segment 4: xxx/DA3

where "xxx" is the name of your file, with enough zeroes on the end to make up eight characters. All PROFILE II segments are kept by TRSDOS with a record length of 256, unless you have created them with some other record length.

If you're accessing Segment 2, 3, or 4, just FIELD the file appropriately, then issue a "GET x,y," where x is the buffer number of the file and y is the PROFILE II record number that you want. If you need fields from Segment 1, there's one more step involved.

We know that Segment 1 records are 85 bytes each. They are stored three to a physical disk record, making 255 bytes, with the 256th byte reserved. Thus, the first record in a file begins in sector 1, relative byte 0. The second record begins at relative byte 85 of the same sector; the third, at relative byte 170. The fourth Segment 1 record will be found at the beginning of sector 2, and so forth.

This means that the /KEY segment of a 300-record file will occupy only 100 sectors on the disk, as opposed to 300 sectors for each of the other segments.

How does this affect your BASIC program? Well, to locate a "logical" record number in the /KEY file, you'll have to compute which physical sector of the file that record is to be found in, and which 85 bytes in that sector constitute the logical record you want. This is called "deblocking," and here is how to do it.

Let's say you know the logical record number that you want, and let's call it LR. You'll need to find the physical record number, PR, and the sub-record, SR, within that physical record. The two formulas go like this:

```
PR = INT((LR-1)/3)+1
SR = LR-3*(PR-1)-1
```

At this point, PR contains the argument you'll need for your Statement, and SR tells which 85-byte portion of the sector contains your logical record. (SR will always equal 0, 1, or 2.) Now you can read and field the record like this:

```
GET x, PR
FIELD x, (85*SR) as XX$, 6 AS AC$, 15 AS L$...
```

XX$ will not be referenced again; it's used simply to skip past the bytes which are not part of your sub-record.

If you're using PROFILE +, or if you've applied the patches from the June, 1981 Newsletter, you may have defined Segments 2, 3, or 4 as some length other than 256. This won't require that you deblock those segments in BASIC; variable-length segments are managed by TRSDOS according to the record length you specified when you created the file. Simply add the correct record length to your OPEN command and proceed normally.

A few warnings:

(1) Check LOF(x) to make sure you're not exceeding the number of records to which the file was expanded by PROFILE II. (If you're reading Segment 1, the number of logical records is three times the value LOF gives.) (Continued on Page 29)

Profile (From Page 28)

(2) Never write CHR$(0)'s to the beginning of a /KEY file record. This will cause PROFILE II to treat that record as unsued.
(3) Be sure to CLOSE all PROFILE II segments that you've accessed before leaving your BASIC program.

Write us at the address at the head of this column and let us know how you like this material and how useful it is to you.

See you next month—

Notes on Previous Newsletters

November, 1980

Data Base Manager

Charles M. Mendenhall, Jr. Greensboro, NC

I have been receiving the TRS-80 Microcomputer News since I purchased my Model III last October and have enjoyed it very much. I would like to see more programs from readers published as I have found good use for many of those you have published.

I found the "Data Base Manager" program on page 7 of the November 1980 issue particularly good. I did find that it was difficult to use unless you kept a record of the information on each record. To remedy this information, I have modified the program to print an index of the records by record number, showing the first field of the record as an identifier. The modification is relatively simple—changing steps #270 and 280, deleting step 290 and adding steps #1200 to 1360.

```
270 PRINT"<Q>UERY A RECORD  <D>ELETE A RECORD  <L>IST ON
    PRINTER
280 PRINT"<A>DD/UPDATE RECORD  <I>NDEX OF DATA  <E>XIT
    PROGRAM

355 IF Z$="I" THEN 1200

1200 L1=1
     : P=1
1210 L2=L1+19
1220 K=L1+9
     : M=L1+10
1230 CLS
     : PRINT@ 15, "INDEX OF DATA"
     : PRINT@ 40, "PAGE NO."; P
     :  N=192
1233 PRINT STRING#(63,CHR$(131))
1235 OPEN "R", 1, "DMB/DAT:1"
1240 GOSUB 1010
1250 FOR II=L1 TO K
1260 GET 1, II
1270 PRINT@N, "<" II ">"; B$(1);
1280 N=N+64
1290 NEXT II
1295 N=226
1300 FOR II=M TO L2
1310 GET 1,II
1320 PRINT@N, "<" II ">"; B$(1);
     : N=N+64
     : NEXT II
1330 CLOSE
1340 PRINT
     : PRINT
     : PRINT "PRESS <N>EXT PAGE OR <R>ESUME PROGRAM"
1350 C$=INKEY$
     : IF C$="" THEN 1350
1352 IF C$="N" THEN 1360
1355 IF C$="R" GOTO 190 ELSE PRINT "REDO!!"
     : GOTO 1340
1360 L1=L1+20
     : L2=L1+19
     : P=P+1
     : GOTO 1220
```

I hope you will continue to pass along good programs as they are available to you.

June, 1981

AUDATA

Gary Bannister Indianapolis, IN

I was very pleased to find the AUDATA program by Mr. Terrell in the June 1981 issue of Microcomputer News. This is one program I have been looking for for some time.

While playing with the program, I have solved the problem of the 'confused' data that occurs on every other run. Simply add the following line to the source code BEFORE ASSEMBLY:

```
00765   LD   (IY+0),A   ;PUT IT BACK
```

This is what happened:

The program performs a memory test at GETBYT by seeing if a memory location can indeed be written to. It does this by simply examining the data, complementing it, putting it back, and checking to see if the complemented data has been written. The complemented data is complemented AGAIN to bring it back to its original form for processing, BUT IT IS NEVER WRITTEN BACK INTO MEMORY! Adding line 765 does just this.

If this line is added BEFORE assembly, there will be no problems. The assembler will take care of the necessary memory juggling, with no one the wiser but you and me!

The FIELD OVERFLOW ERROR that occurs at line 01060 has happened to me before. I have been told that it is a bug peculiar to early versions of EDTASM (I think mine is Ver. 1.2). It is apparently cured in later versions, but if the code assembles correctly, ignore it.

I hope this is some help to your readers.

Reduce Fractions

Joe Rose Greenwood, IN

I thoroughly enjoyed my first issue of RADIO SHACK Microcomputer News. I entered most of the programs in the newsletter into my Model III, but I encountered two problems with the very short program called "Reduce Fractions." First of all as I typed the program in I came across line 140 which reads:

```
140 INPUT"1,ELSE TYPE 0;Q
```

One simple quotation mark will fix that, and it should read:

```
140 INPUT"1,ELSE TYPE 0";Q
```

Then the second problem arose when I ran the program. To my surprise it never printed the answer. I believe the change should be in line 110. It reads:

```
110 IF X=B THEN PRINT"IN LOWEST TERMS":GOTO 130
```

Mr. Beasley's program seemed to run ok when I edited line 110, and deleted,:GOTO 130. So it now reads:

```
110 IF X=B THEN PRINT"IN LOWEST TERMS"
```

Thank you very much for your great newsletter.

July, 1981

Different Approach

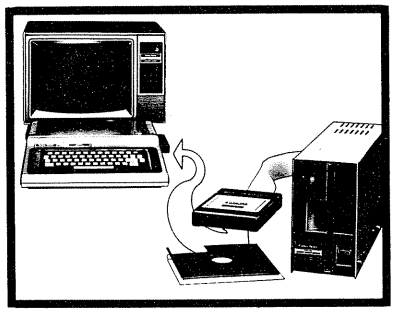
Sam Seltzer Sonyea, New York

I read, with interest, your "Different Approach" to the sorting of two-dimensional string arrays which appeared in the July issue of TRS-80 Microcomputer News.

I had some difficulty in printing out N$ after composition. In all cases, LEN(N$) was extremely variable, and in no case was the entire name printed out to the correct length. I finally discovered what I believe to be a typographical error, which you probably have caught by this time, anyhow. However, in case you have not, I thought I would call it to your attention:

```
Line 300 reads: N$=MID$(N$(X),7,II-I) and should
actually read:  N$=MID$(N$(X),7,I-7).

VAL(II-I)=LEN(AD$) whereas VAL(I-7)=LEN(N$).
```

(Continued on Page 37)



Color Computer

Product Line Manager's News

Cassettes and the Color Computer

Tomas Rokicki Wolfe City, Texas

The disk is a fantastic medium for computer data storage. Fast, relatively small, convenient, and able to store huge quantities of information, it is the hobbyist's dream. But the six hundred dollar price tag for the initial disk drive for the Color Computer is a bit more than my budget for computer expenses this month. Many of us must make do, therefore, with cassette data storage. Fortunately, the Color Computer has extensive cassette capabilities. Not so fortunately, they are not explained in any great technical detail in either of the BASIC manuals. A Technical Reference Manual is due in the next few months. Meanwhile, this short report should help.

First, one may ask, why does the Color Computer require a file to be opened and closed? Why can't data simply be written to cassette, as with the Model I? This is not the hassle or inconvenience it may seem. Actually, it is a great advantage. Read on.

When you open a cassette file for output, the computer writes a file header. This file header contains the name of the file and the type of information in the file (data, BASIC program, machine language program.) This allows the computer to later locate the right file. With Model I, you had to position the tape for each file.

When you output information to the cassette using the PRINT #-1 command, the computer does not immediately write that data to the cassette. Instead, it puts the data in a cassette buffer. This buffer is a block of memory locations in the bottom kilobyte of memory. As soon as this buffer is full, the computer writes the entire buffer to cassette at one time. The Model I wrote each piece of data independently, each with its own file header, which makes for much less efficient use of tape.

This brings up a small problem. What if the last piece of data you output to cassette does not fill the buffer? This is what the CLOSE command is for. This command writes the buffer out to cassette whether it is full or not, and then ouputs a special end-of-file (EOF) marker. The 'end' command does the same thing. This allows you to open two files, one after the other.

Data input from cassette is handled in much the same way. When you open a file for input, the computer searches the tape for a file with the name given. (If no name is specified, the computer searches for the first file.) The computer then makes sure that the file is the right type. Then it is ready for data input.

With the INPUT #-1 command, the computer reads in an entire buffer of data from the cassette. It then assigns data to the variable being input. On following inputs, the computer checks to see if the buffer has any data left in it. If it does, the computer gets the data from the buffer. If it doesn't, the computer reads a whole new buffer of data from cassette.

This is where the EOF(-1) command comes in handy. If the last piece of data read was the last data in the file, the computer sets EOF(-1) equal to true (or -1). Otherwise, EOF(-1) is false (0). This way you can check for the end of the file without using a special marker value or counting the number of times you input data.

This open-close file structure, in conjunction with the 1500 baud (that's bits per second) data handling, makes the cassette operation of the Color Computer much faster, easier, and more efficient than the Model I or Model III.

Another area that is neglected in the Color BASIC manuals is the machine language storage and retrieval from cassette. Actually, the commands are quite simple. The CSAVEM command, available to Extended BASIC owners, has four parameters. In order they are:

1. File name. Must be a string variable or enclosed in quotes. Can be the null string (" ").
2. Beginning of file. This is the start of the memory that you want to save on tape. Range: 0 to 65535.
3. End of file. This is the last memory location you wish to save on tape. Range: Beginning of file to 65535.
4. Start of program. This is put on tape, and when the file is loaded, this address is used as the default EXEC parameter. Range: 0 to 65535.

As an example, you have graphics display program in machine language which begins on graphics page 5. You wish to save the graphics on pages 1 to 4 also for initialization. You would then execute the following command:

```
CSAVEM "GRADIS", 1536, 9215, 768Ø
```

"GRADIS" is just the filename I chose to give this example. You would use your own file name.

1536 is the beginning of graphics page 1. 9215 is the end of graphics page 5. 7680 is the beginning of graphics page 5, and is the start of your machine language routine. Remember, there are 1536 bytes per graphics page.

No parameters are optional; all must be specified. This command is also good for saving arrays, graphics, text screens, or practically any other time you wish to save memory. If it is not used for machine language programs, the start of program parameter may be set to 350. A return from subroutine is located in this memory location, so an accidental EXEC will not blow the system.

CLOADM is even simpler. It has two parameters. The first is the file name; it may normally be omitted. The second is the offset address. This value is added to the beginning of file value, and the data is stored beginning there. If this parameter is used, the first must also be used; however, a null string can be used. The second parameter has a range of 0 to 65535. For instance, let us say that you wanted to load the above graphics and program in graphics pages 3 to 7 instead of 1 to 5. You would then type this in:

```
CLOADM "GRADIS", 3Ø72
```

A favorite trick of mine is to CLOAD the text screen directly so that the program user has something to read while the program is loading. One thing you have to remember is that there must be enough memory left after you offset, to store the program. Also, your machine language code must be relocatable.

Finally, for you masochists, I have included a disassembler program in BASIC. This will allow you to disassemble the ROM's, including the Program Paks. (To disassemble the Program Paks, you have to be sneaky, and this is definitely NOT recommended by Radio Shack. If you do not know EXACTLY what you are doing, DON'T! This can cause serious damage to both your Color Computer and the Program Pak!) To use this program, wait until the computer is initialized. Then enter the starting address, either in decimal, or in hexadecimal with an H preceding the number. A good starting point is 40999 decimal, the RESET sequence. The program prints out standard 6809 mnemonics. If you see an 'ERROR' on the screen, you are probably in a table or message string. The branches are referenced to actual location. I figured that would do, since I did not feel up to generating a label table.

(Continued on Page 31)

Color (From Page 30)

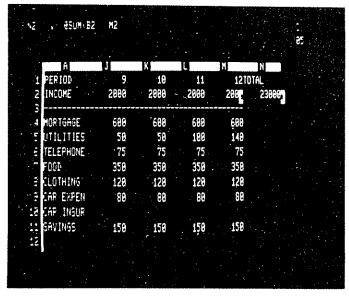
A last word—about the 32K memory expansion Radio Shack just came out with. It gives you effectively three times the power-up memory for program and data storage as the 16K machine does. Heck, with it, you can have eight graphics pages, a one kilobyte machine language routine, 5000 bytes of string storage, and still have 13,927 bytes left for program and data storage.

```
1Ø ' 68Ø9 DISASSEMBLER FOR THE TRS-8Ø COLOR COMPUTER 16K
   EXTENDED BASIC REQUIRED  TOMAS ROKICKI  BOX 244 WOLFE
   CITY  TEXAS  75496       AUGUST 24, 1981

2Ø ' THIS PROGRAM WILL PAUSE EVERY 13 LINES AND WAIT FOR A
   KEY TO BE PRESSED.  THIS WILL GIVE A PERSON TIME TO
   COPY THE  DATA OR JUST READ IT.  TO DISABLE THIS
   FEATURE, PRESS C   WHEN IT HAPPENS THE FIRST TIME.

3Ø ' THIS PROGRAM OUTPUTS STANDARD 68Ø9 MNEMONICS, EXCEPT
   FOR THE BRANCH INSTRUCTIONS.  THEN IT OUTPUTS ABSOLUTE
   ADDRESSES, SINCE THE ROM IS FIXED IN MEMORY.  THIS ALSO
   LETS ME GET AWAY WITHOUT USING  LABELS.
4Ø ' THIS PROGRAM MAY BE REPRODUCED FOR PERSONAL USE UNDER
   THE CONDITION THAT THE FIRST REMARK IS LEFT IN.  IT MAY
   NOT BE COPIED FOR SALE TO THIRD PARTIES.
5Ø CLS
   : PRINT " I N I T I A L I Z A T I O N"
6Ø CLEAR 1ØØØ
   : DIM N$(255), I$(15)
   : FOR X=Ø TO 255
   : READ N$(X)
   : NEXT
   : FOR X=Ø TO 15
   : READ I$(X)
   : NEXT
   : CLS
   : PRINT@33, "68Ø9 DISASSEMBLER  VERSION 1.2"
   : PRINT
   : INPUT "OUTPUT TO PRINTER (DEFAULT IS   NO)"; A$
   : IF MID$(A$,1,1)="Y" THEN ZZ=1
7Ø INPUT "ENTRY ADDRESS (DEFAULT IS 4Ø999)"; A$
   : IF A$="" THEN PI=4Ø999 ELSE IF MID$(A$,1,1)="H" THEN
   PI=VAL("&"+A$) ELSE PI=VAL(A$)
8Ø BY=PEEK(PI)
   : IF BY<>16 AND BY<>17 THEN TE=Ø
   : BY$=N$(BY)
   : TY=VAL(BY$)
   : BY$=MID$(BY$,2)
   : IF BY$="" THEN 13Ø ELSE 1ØØ ELSE TE=1
   : BY=PEEK(PI+1)
   : BY$=N$(BY)
   : X=INSTR(BY$,"/")
   : IF X=Ø THEN 13Ø ELSE BY$=MID$(BY$,X+1)
   : IF PEEK(PI)=17 THEN X=INSTR(X,BY$,"/")
   : IF X=Ø THEN 13Ø ELSE 18Ø
9Ø TY=VAL(BY$)
   : BY$=MID$(BY$,2)
1ØØ ON TY+1 GOSUB 32Ø, 16Ø, 19Ø, 2ØØ, 26Ø, 27Ø, 28Ø, 29Ø,
   3ØØ, 31Ø
   : GOTO 14Ø
11Ø IF B$<>"C" THEN B$=INKEY$
   : IF B$="" THEN 11Ø
12Ø N=Ø
   : GOTO 14Ø
13Ø OP$="ERROR"
   : BY$=""
   : LN=1
14Ø IF INSTR(1,BY$,"/") THEN
   BY$=LEFT$(BY$,INSTR(1,BY$,"/")-1)
15Ø N=N+1
   : IF N>13 THEN 11Ø ELSE A$=STRING$(15,32)
   : MID$(A$,1,4)=HEX$(PI)
   : FOR X=1 TO LN
   : GOSUB 43Ø
   : MID$(A$,4+2*X,2)=R$
   : NEXT
   : A$=A$+BY$+" "+OP$
   : PI=PI+LN
   : PRINT A$
   : IF ZZ THEN PRINT#-2, A$
   : GOTO 8Ø ELSE 8Ø
16Ø LN=2+TE
   : OP$=HEX$(PEEK(PI+TE+1))
   : RETURN
17Ø LN=4+TE
   : X=TE
   : TE=TE+1
   : GOSUB 26Ø
   : TE=X
   : GOTO 24Ø
18Ø BY$=MID$(BY$,X+1)
   : GOTO 9Ø
19Ø LN=1+TE
   : OP$=""
   : RETURN
2ØØ BY=PEEK(PI+TE+1)
   : IF BY>127 THEN 21Ø ELSE LN=2+TE
   : GOSUB 44Ø
   : BY=31 AND BY
   : IF BY>15 THEN OP$=R$+"-"+HEX$(32-BY)
   : RETURN ELSE OP$=R$+"+"+HEX$(BY)
   : RETURN
21Ø GOSUB 44Ø
   : BY=31 AND BY
   : OP$=I$(15 AND BY)
   : IF OP$="X" THEN 17Ø ELSE IF OP$="" THEN OP$="ERROR"
   : BY$=""
   : LN=1
   : RETURN ELSE LN=2+TE+VAL(OP$)
   : IF VAL(OP$)=Ø THEN 24Ø ELSE IF VAL(OP$)=2 THEN 23Ø
   ELSE OP$=MID$(OP$,2)
   : BY=PEEK(PI+2+TE)
   : IF BY<128 THEN OP$=OP$+"+"+HEX$(BY)
   : GOTO 24Ø
22Ø OP$=OP$+"-"+HEX$(256-BY)
   : GOTO 24Ø
23Ø BY=PEEK(PI+2+TE)*256+PEEK(PI+3+TE)
   : OP$=MID$(OP$,2)
   : IF BY>32767 THEN OP$=OP$+"-"+HEX$(65536-BY) ELSE
   OP$=OP$+"+"+HEX$(BY)
24Ø IF INSTR(1,OP$,"R")<>Ø THEN
   MID$(OP$,INSTR(1,OP$,"R"),1)=R$
25Ø IF 16 AND (PEEK(PI+TE)) THEN OP$="("+OP$+")"
   : RETURN ELSE RETURN
26Ø LN=3+TE
   : OP$=HEX$(PEEK(PI+TE+1)*256+PEEK(PI+TE+2))
   : RETURN
27Ø GOSUB 16Ø
   : OP$="#"+OP$
   : RETURN
28Ø GOSUB 26Ø
   : OP$="#"+OP$
   : RETURN
29Ø LN=2
   : BY=PEEK(PI+1)
   : IF BY>127 THEN OP$=HEX$(PI-254+BY)
   : RETURN ELSE OP$=HEX$(PI+2+BY)
   : RETURN
3ØØ LN=3+TE
   : BY=PEEK(PI+1+TE)*256+PEEK(PI+2+TE)
   : IF BY>32767 THEN OP$=HEX$(PI+LN-65536+BY)
   : RETURN ELSE OP$=HEX$(PI+LN+BY)
   : RETURN
```

(Continued on Page 32)

Color (From Page 31)

```
31Ø LN=2
     : BY=PEEK(PI+1) AND 15
     : GOSUB 41Ø
     : OP$=R$
     : BY=(PEEK(PI+1) AND 24Ø)/16
     : GOSUB 41Ø
     : OP$=R$+","+OP$
     : RETURN
32Ø LN=2
     : BY=PEEK(PI+1)
     : OP$=""
     : IF (128 AND BY) THEN OP$="PC"
33Ø IF (64 AND BY) THEN IF MID$(BY$,4)="U" THEN OP$=OP$+",S"
    ELSE OP$=OP$+",U"
34Ø IF (32 AND BY) THEN OP$=OP$+",Y"
35Ø IF (16 AND BY) THEN OP$=OP$+",X"
36Ø IF (8 AND BY) THEN OP$=OP$+",DP"
37Ø IF (4 AND BY) THEN OP$=OP$+",B"
38Ø IF (2 AND BY) THEN OP$=OP$+",A"
39Ø IF (1 AND BY) THEN OP$=OP$+",CC"
4ØØ IF MID$(OP$,1,1)="," THEN OP$=MID$(OP$,2)
     : RETURN ELSE RETURN
41Ø IF BY=Ø
       THEN R$="D"
       ELSE IF BY=1
          THEN R$="X"
          ELSE IF BY=2
             THEN R$="Y"
             ELSE IF BY=3
                THEN R$="U"
                ELSE IF BY=4
                   THEN R$="S"
                   ELSE IF BY=5
                      THEN R$="PC"
                      ELSE IF BY=8
                         THEN R$="A"
                         ELSE IF BY=9
                            THEN R$="B"
                            ELSE IF BY=1Ø
                               THEN R$="CC"
                               ELSE IF BY=11
                                  THEN R$="DP"
                                  ELSE R$="??"
42Ø RETURN
43Ø R$=HEX$(PEEK(PI+X-1))
     : IF LEN(R$)=1 THEN R$="Ø"+R$
     : RETURN ELSE RETURN
44Ø X=(BY AND 96)/32
     : IF X=Ø
          THEN R$="X"
          ELSE IF X=1
             THEN R$="Y"
             ELSE IF X=2
                THEN R$="U"
                ELSE IF X=3
                   THEN R$="S"
45Ø RETURN
46Ø DATA 1NEG, , , 1COM, 1LSR, , 1ROR, 1ASR, 1ASL, 1ROL,
     1DEC, , 1INC, 1TST, 1JMP, 1CLR
47Ø DATA P, P, 2NOP, 2SYNC, , , 8LBRA, 8LBSR, , 2DAA, 2ORCC,
     , 5ANDCC, 2SEX, 9EXC, 9TFR
48Ø DATA 7BRA, 7BRN/8LBRN, 7BHI/8LBHI, 7BLS/8LBLS,
     7BHS/8LBHS, 7BLO/8LBLO, 7BNE/8LBNE, 7BEQ/8LBEQ,
     7BVC/8LBVC, 7BVS/8LBVS, 7BPL/8LBPL, 7BMI/8LBMI,
     7BGE/8LBGE, 7BLT/8LBLT, 7BGT/8LBGT, 7BLE/8LBLE
49Ø DATA 3LEAX, 3LEAY, 3LEAS, 3LEAU, ØPSHS, ØPULS, ØPSHU,
     ØPULU, , 2RTS, 2ABX, 2RTI, 2CWAI, 2MUL, ,
     2SWI/2SWI2/2SWI3
5ØØ DATA 2NEGA, , , 2COMA, 2LSRA, , 2RORA, 2ASRA, 2ASLA,
     2ROLA, 2DECA, , 2INCA, 2TSTA, , 2CLRA
51Ø DATA 2NEGB, , , 2COMB, 2LSRB, , 2RORB, 2ASRB, 2ASLB,
     2ROLB, 2DECB, , 2INCB, 2TSTB, , 2CLRB
52Ø DATA 3NEG, , , 3COM, 3LSR, , 3ROR, 3ASR, 3ASL, 3ROL,
     3DEC, , 3INC, 3TST, 3JMP, 3CLR
53Ø DATA 4NEG, , , 4COM, 4LSR, , 4ROR, 4ASR, 4ASL, 4ROL,
     4DEC, , 4INC, 4TST, 4JMP, 4CLR
54Ø DATA 5SUBA, 5CMPA, 5SBCA, 6SUBD/6CMPD/6CMPU, 5ANDA,
     5BITA, 5LDA, , 5EORA, 5ADCA, 5ORA, 5ADDA,
     6CMPX/6CMPY/6CMPS, 7BSR, 6LDX/6LDY,
55Ø DATA 1SUBA, 1CMPA, 1SBCA, 1SUBD/1CMPD/1CMPU, 1ANDA,
     1BITA, 1LDA, 1STA, 1EORA, 1ADCA, 1ORA, 1ADDA,
     1CMPX/1CMPY/1CMPS, 1JSR, 1LDX/1LDY, 1STX/1STY
56Ø DATA 3SUBA, 3CMPA, 3SBCA, 3SUBD/3CMPD/3CMPU, 3ANDA,
     3BITA, 3LDA, 3STA, 3EORA, 3ADCA, 3ORA, 3ADDA,
     3CMPX/3CMPY/3CMPS, 3JSR, 3LDX/3LDY, 3STX/3STY
57Ø DATA 4SUBA, 4CMPA, 4SBCA, 4SUBD/4CMPD/4CMPU, 4ANDA,
     4BITA, 4LDA, 4STA, 4EORA, 4ADCA, 4ORA, 4ADDA,
     4CMPX/4CMPY/4CMPS, 4JSR, 4LDX/4LDY, 4STX/4STY
58Ø DATA 5SUBB, 5CMPB, 5SBCB, 6ADDD, 5ANDB, 5BITB, 5LDB, ,
     5EORB, 5ADCB, 5ORB, 5ADDB, 6LDD, , 6LDU/6LDS,
59Ø DATA 1SUBB, 1CMPB, 1SBCB, 1ADDD, 1ANDB, 1BITB, 1LDB,
     1STB, 1EORB, 1ADCB, 1ORB, 1ADDB, 1LDD, 1STD, 1LDU/1LDS,
     1STU/1STS
6ØØ DATA 3SUBB, 3CMPB, 3SBCB, 3ADDD, 3ANDB, 3BITB, 3LDB,
     3STB, 3EORB, 3ADCB, 3ORB, 3ADDB, 3LDD, 3STD, 3LDU/3LDS,
     3STU/3STS
61Ø DATA 4SUBB, 4CMPB, 4SBCB, 4ADDD, 4ANDB, 4BITB, 4LDB,
     4STB, 4EORB, 4ADCB, 4ORB, 4ADDB, 4LDD, 4STD, 4LDU/4LDS,
     4STU/4STS
62Ø DATA R+, R++, -R, --R, R+Ø, R+ACCB, R+ACCA, , 1R, 2R, ,
     R+ACCD, 1PC, 2PC, , X
```



Peripherals (From Page 10)

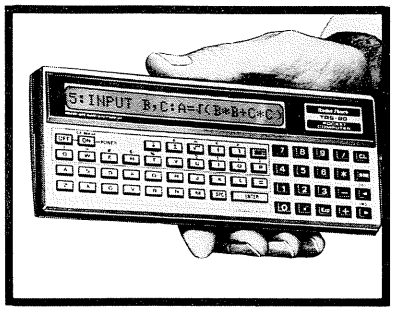
anguish. Note: if you are using Tymenet, the baud rate calculation is not being done, so this procedure should not be needed.

1. Load the software that you will be using to access CompuServe or turn on your Videotex unit.
2. Place the phone call.
3. Listen until you hear the CompuServe tone over the phone.
4. Establish a connection:
   a. Direct Connect Modem I—Turn on the Modem by flipping the switch to the Originate position.
   b. Videotex Terminal—Press the BREAK key.
5. If you are still listening to your phone, you will hear a second tone. This tone is generated by your equipment.
6. NOW press CTRL C (Shift Down Arrow C) or the BREAK key and THEN hang up the phone. (Model II users who are using TERMINAL should press the F1 key.)

Once the CTRL C is sent, the CompuServe system begins calculating the baud rate. If the system hears noise (e.g. the noise of the receiver being replaced on the hook), before the CTRL C or BREAK is sent, the noise may be incorrectly used to calculate the baud rate. If this happens, you will have to begin again from step one.

You should use this sequence with ANY direct connect modem, even a non-Radio Shack unit, which requires you to place the connecting phone call on your regular telephone.

I'll finish with a new set of promises. Coming soon is an article describing how to get the most out of our Model II SCRIPSIT 2.0 and your Line Printer 8. A description of new printwheels is in the works, and much more! As always, you will read about new products here, first.



Pocket Computer

Product Line Manager's News

Matrix Inversion - Updated

De Forrest Metcalf El Paso, Texas

This letter is motivated by the request which appeared in the July issue of Microcomputer News for a better mathematical program to compute the inverse of a matrix. The following program listing fits this description. Actually, it does much more than just invert a matrix—its primary forte is solving sets of simultaneous linear algebraic equations, for which there is great demand these days. Matrix inversion, per se, is not needed too very often and a program which can do just this and no more will not have much call for its services.

To invert a matrix with my linear system solver, the user simply makes a series of invocations of the program to compute the individual columns of the inverse matrix. The successive columns of the identity matrix are inputted as the given (right-hand-side) vectors for the successive runs, and the resultant solution vectors are the corresponding columns of the inverse matrix.

The linear system solver for which I am enclosing the text listing will handle up to (and including) 10 simultaneous equations and requires about 15 seconds to process the larger sets on my Model I machine. The program employs the preferred method of Gaussian elimination and back-substitution, supplemented by partial pivoting. The matrix inverter described in the July issue of Microcomputer News did not employ the pivoting technique and this accounts for its failure in attempting to invert the sample 4x4 matrix.

My linear system solver is something of a universal program in the sense that, with a few trivial changes, it will run on any of the Radio Shack computers. As listed, it is programmed for the Model I and Model III machines (with Level II BASIC capability) and runs in abour 2K of RAM. For the Pocket Computer, exclude all the comment and DATA statements. (Editor's note: We modified the program listings so they should run on either type computer. The main change was to change the variable AA to A.)

Computations involving matrices are often greatly benefited by going to double precision arithmetic. The user with a Model I/III machine having Level II capability can accomplish this by recoding the variable A as IA and then replacing line 150 with two lines as follows:

```
150 CLEAR 10
      : CLS
      : PRINT CHR$(23)
155 DEFINT B, C, D, E, I
      : DEFDBL A, F
      : DIM A(116)
```

Having introduced these changes the statement A = INT(A) on line 170 can be omitted (along with its preceding colon).

I am also enclosing another matrix program which may be of interest to you. It computes the eigenvalues for a symmetrical matrix. The eigenvalue/eigenvector business is very good these days and this program should be useful in many areas of today's high technology. It employs the method of Householder reflections to reduce the given matrix to a tridiagonal form (lines 160-350). Thus reduced, its characteristic polynomial is easily arrived at. Lines 360-410 compute the roots of this polynomial by the method of Newton-Raphson. These roots, of course, are the desired eigenvalues. Multiple roots can be dealt with by changing the first statement in line 400 to E = E-F*2 for double roots, to E = E-F*3 for triple roots, etc. Rough initial estimates of the eigenvalues are prompted for; these can be obtained by the method of Gerschgorin's disks.

Having computed all the eigenvalues, the associated eigenvectors can then readily be calculated by recourse to my linear algebraic system solver. Thus we have a complete solution to the algebraic eigenvalue problem insofar as symmetric matrices are concerned. This second program, like the first, is easily adapted to the Pocket Computer or to the use of double precision with Model I/III machines which support Level II BASIC. As listed, the eigenvalue program can be entered directly into any machine with 2.5K or more of RAM. It can deal with matrices of order up to (and including) ten.

Linear Equation Solver (Note to Pocket Computer users: Eliminate lines 100 - 160, and you can leave out the REM lines. No other modifications should be needed.)

```
100 REM -- LINEAR EQUATION SYSTEM SOLVER
110 REM -- P.C. SIMULATION.  PROGRAM WRITTEN BY...
120 REM -- D. F. METCALF, EL PASO, TEXAS
130 REM -- NOTE
     : TO USE THE INTERNAL TEST DATA,
140 REM -- CHANGE "INPUT" TO "READ" IN LINES 200 AND 220.
150 CLEAR 10
     : CLS
     : PRINT CHR$(23)
     : DIM A(116)
160 DATA 1, 10, 1,   2, 0, 1,   3, 3, 2,   3.5, -7, 10.5
170 INPUT "# OF EQNS."; A
     : A=INT(A)
180 IF A<2 OR A>10 GOTO 170
190 PRINT "ENTER MAT. ELEM. BY ROWS"
     : FOR B=1 TO A
200 FOR C=1 TO A
     : GOSUB 600
     : INPUT A(E)
     : NEXT C
     : NEXT B
210 PRINT "ENTER R.H.S"
220 FOR B=7 TO A+6
     : INPUT A(B)
     : NEXT B
230 REM -- START GAUSSIAN ELIMINATION.
240 FOR C=1 TO A-1
     : F=0
     : FOR B=C TO A
250 REM -- SEARCH FOR PIVOT ROW.
260 GOSUB 600
     : IF F<ABS(A(E)) LET F=ABS(A(E))
     : D=B
270 NEXT B
     : IF F=0 STOP
280 E=C-A+16
     : IF C=D GOTO 340
290 REM -- SWAP ROWS (ALSO R.H.S.).
300 FOR B=E TO 16
     : F=A(A*C+B)
310 A(A*C+B)=A(A*D+B)
     : A(A*D+B)=F
     : NEXT B
320 F=A(C+6)
     : A(C+6)=A(D+6)
     : A(D+6)=F
330 REM -- FORWARD ELIMINATION ROUTINE.
```

(Continued on Page 34)

Pocket Computer (From Page 33)

```
34Ø FOR D=C+1 TO A
     : F=A(A*D+E)
     : IF F=Ø GOTO 38Ø
35Ø F=F/A(A*C+E)
     : FOR B=E+1 TO 16
36Ø A(A*D+B)=A(A*D+B)-F*A(A*C+B)
     : NEXT B
37Ø A(D+6)=A(D+6)-F*A(C+6)
38Ø NEXT D
     : NEXT C
     : A(A+6)=A(A+6)/A(A*A+16)
39Ø REM BACK SUBSTITUTION ROUTINE.
4ØØ FOR B=A-1 TO 1 STEP -1
     : F=Ø
     : C=Ø
     : GOSUB 6ØØ
41Ø FOR C=B+1 TO A
     : F=F+A(E+C)*A(C+6)
     : NEXT C
42Ø A(B+6)=(A(B+6)-F)/A(E+B)
     : NEXT B
43Ø REM -- DISPLAY SOLUTION VECTOR, THEN EXIT.
44Ø PRINT "THE SOLUTION VECTOR IS."
45Ø FOR B=1 TO A
     : D=B+6
     : PRINT B; A(D)
     : NEXT B
     : END
58Ø REM -- INTERNAL SUBROUTINE TO CALCULATE E, THE
59Ø REM -- SUBSCRIPT CORRESPONDING TO MATRIX ELEMENT A(B,C)
6ØØ E=A*B+C-A+16
     : RETURN
```

The July, 1981 Microcomputer News article contained a 2x2 matrix:

$$\begin{bmatrix} 6 & 7 \\ 5 & 6 \end{bmatrix}$$ Inverse: $$\begin{bmatrix} 6 & -7 \\ -5 & 6 \end{bmatrix}$$

We will use this matrix to demonstrate the matrix inversion process:

```
RUN
# OF EQNS.? 2

ENTER MATRIX ELEMENTS BY ROWS.
? 6
? 7
? 5
? 6

ENTER R.H.S
? 1
? 0

THE SOLUTION VECTOR IS . . .
1     5.99999
2    -4.99999


  RUN
# OF EQNS.? 2

ENTER MATRIX ELEMENTS BY ROWS.
? 6
? 7
? 5
? 6

ENTER R.H.S
? 0
? 1

THE SOLUTION VECTOR IS . . .
1    -6.99999
2     5.99999
```

Notice that we had to run the program twice, once for each column in our original square matrix. Two runs are required since the program alters the original matrix values during the solution process.

The R.H.S. of the solution is a column from the identity matrix of the appropriate size. Our identity matrix is:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

In any identity matrix, all of the matrix elements are zero, except for the elements of the diagonal which runs from the upper left corner of the matrix to the lower right corner.

To get the values of the first column of the inverse matrix, we entered the elements of the original matrix row by row: 6 7 5 6 , then we entered the values for the first COLUMN of the identity matrix: 1 0 . The result was:

```
THE SOLUTION VECTOR IS . . .
1    5.99999
2   -4.99999
```

Rounding (as we did in the July article) we have the first COLUMN of the inverse matrix:

```
 6
-5
```

To get the second column of the inverse matrix we ran the program again, input the information from the original matrix by row: 6 7 5 6 , and then we entered the values from the second column of the identity matrix: 0 1 . The result was:

```
THE SOLUTION VECTOR IS . . .
1   -6.99999
2    5.99999
```

Rounding again, we got:

```
-7
 6
```

Putting the two columns together we get the inverse matrix, which is:

$$\begin{bmatrix} 6 & -7 \\ -5 & 6 \end{bmatrix}$$

Here is the Eigenvalue program:

(Note: Pocket Computer users should eliminate lines 100 -125. You should also change the SQR in line 200 to the square root symbol)

```
1ØØ REM -- EIGENVALUE CALCULATION FOR SYMMETRIC MATRIX.
1Ø5 REM -- P.C. SIMULATION.  PROGRAM WRITTEN BY...
11Ø REM -- D. F. METCALF, EL PASO, TEXAS
115 REM -- NOTE
     : MATRIX ELEMENTS ARE ENTERED BY ROW, UP TO
12Ø REM -- AND INCLUDING THE DIAGONAL ELEMENT IN EACH ROW.
125 CLEAR 1Ø
```

(Continued on Page 35)

Pocket Computer (From Page 34)

```
     : DIM A(87)
13Ø INPUT "MATRIX SIZE "; A
     : A=INT(A)
     : IF A<3 GOTO 13Ø
14Ø B=(A*A+A)/2
     : FOR L=33 TO B+32
15Ø INPUT A(L)
     : NEXT L
     : A(13)=A(33)
     : C=32
16Ø FOR B=1 TO A-2
     : G=B+1
     : C=C+G
     : J=C
     : F=Ø
17Ø FOR I=B+2 TO A
     : J=J+I-1
     : D=A(J)
     : A(I+12)=D
18Ø F=F+D*D
     : NEXT I
     : H=A(C)*A(C)
     : D=H*1E-14
19Ø IF F<=D LET A(B+22)=A(C)
     : GOTO 33Ø
2ØØ D=SQR(F+H)
     : IF A(C)>Ø LET D=-D
21Ø A(B+22)=D
     : A(G+12)=A(C)-D
     : J=C-B
     : D=Ø
22Ø FOR L=G TO A
     : J=J+L-1
     : E=J
     : F=Ø
23Ø FOR K=G TO L
     : E=E+1
     : F=F+A(E)*A(K+12)
     : NEXT K
24Ø IF L>=A GOTO 26Ø
25Ø FOR K=L+1 TO A
     : E=E+K-1
     : F=F+A(E)*A(K+12)
     : NEXT K
26Ø D=D+F*A(L+12)
     : A(L+22)=F
     : NEXT L
27Ø D=D*.5
     : H=A(G+12)*A(B+22)
28Ø FOR L=G TO A
     : A(L+12)=A(L+12)/H
29Ø A(L+22)=A(L+22)+D*A(L+12)
     : NEXT L
3ØØ J=C-B
     : FOR K=G TO A
     : J=J+K-1
     : E=J
31Ø FOR L=G TO K
     : E=E+1
     : F=A(E)
32Ø F=F+A(L+12)*A(K+22)+A(K+12)*A(L+22)
     : A(E)=F
     : NEXT L
     : NEXT K
33Ø A(G+12)=A(C+1)
     : NEXT B
34Ø C=C+A
     : A(A+22)=A(C)
     : A(A+12)=A(C+1)
35Ø FOR J=A+2Ø TO 23 STEP -1
     : A(J+1)=A(J)
     : NEXT J
36Ø INPUT "ESTIM "; E
     : FOR K=1 TO 2Ø
     : H=A(13)-E
     : D=1
     : B=-1
     : C=Ø
37Ø FOR I=2 TO A
     : G=A(I+12)-E
     : L=A(I+22)
     : F=H*G-D*L*L
38Ø G=B*G-C*L*L-H
     : D=H
     : H=F
     : C=B
     : B=G
     : NEXT I
     : F=H/B
39Ø IF ABS(F)<3E-7 GOTO 41Ø
4ØØ E=E-F
     : NEXT K
     : PRINT "NO CONV."
41Ø PRINT "ANS. IS ";E
     : GOTO 36Ø
```

Editor's comments: No one on the newsletter staff is familiar with eigenvalues. However, it is obvious that not all matrixes will be in a form to allow you to use the above routine. Therefore we dug down into our stack of programs and found an eigenvalue program for the pocket computer which appears to be less restrictive than the one above.

Here is the second eigenvalue program as we have it:

The problem is to set up a characteristic equation according to Danilevski's method, and solve the equation by using Hitchcock's method to determine the eigenvalue.

A matrix is given as $A = [a_{ij}]$ (here, A is an n-element square matrix.)

Deform the matrix, first supposing $m = 1$, $m' = m + 1$.

Matrix $T = [t_{ij}]$, and $T^{-1} = [t'_{ij}]$ are set up as follows.....Eqn. (1)

When $j = m'$,

$$t_{ij} = -a_{im}/a_{m'm} \quad (i \neq m')$$
$$t_{m'j} = 1 / a_{m'm}$$

$$t'_{ij} = a_{im}$$

When $j \neq m'$,

$$t_{ij} = t'_{ij} = 0 \quad (i \neq j)$$

$$t_{ij} = t'_{ij} = 1$$

Thus, the matrix is deformed from TAT^{-1} into A.

If $m' \neq n$, $m + 1$ is changed to m, and $m' + 1$ to m', thus returning to eq. (1).

If $m' = n$, the deformation is terminated.

Now, matrix A is deformed as follows, except for the final row:

$$a_{ij} = 0 \quad (i \neq j + 1)$$
$$a_{ij} = 1 \quad (i = j + 1)$$

From the final row, the coefficient $[r_i]$ may be determined as follows.

(Continued on Page 36)

Pocket Computer

(From Page 35)

$r_i = -a_{kn}\ (k = n - i + 1)$

As a result, the following equation is obtained as the characteristic equation:

$L^n + r_1 L^{n-1} + \ldots + r_{n-1} L + r_n = 0$

Thus a characteristic equation is obtained which is solved by using Hitchock's method.

$$\begin{bmatrix} 38 & -83 & -21 & 76 \\ -27 & 54 & 86 & -18 \\ 71 & 21 & 85 & 15 \\ -18 & 65 & -74 & 46 \end{bmatrix}$$

Points about the following program:

- The equation cannot be calculated when a_{ij} is an imaginary number.
- Calculation is impossible when a_{mm}' is 0.
- The characteristic equation may not always converge.
- When the roots are equal roots, there is a possibility of failure to converge.
- At every repetition, the current state of convergence is displayed.
- As the displayed figure approaches 0, convergence is near its end.
- This program can be used in the range of $2 <= n <= 7$.

Here is a sample run using the 4×4 matrix given above (note that the information in parentheses (a_{11}, etc.) is the variable or element being entered):

Step	Input	Display shows:		
01.	SHFT A		ORDER	
02.	(n)	4 (ENTER)	?	
03.	(a_{11})	38 (ENTER)	?	
04.	(a_{12})	−83 (ENTER)	?	
05.	(a_{13})	−21 (ENTER)	?	
06.	(a_{14})	76 (ENTER)	?	
07.	(a_{21})	−27 (ENTER)	?	
08.	(a_{22})	54 (ENTER)	?	
09.	(a_{23})	86 (ENTER)	?	
10.	(a_{24})	−18 (ENTER)	?	
11.	(a_{31})	71 (ENTER)	?	
12.	(a_{32})	21 (ENTER)	?	
13.	(a_{33})	85 (ENTER)	?	
14.	(a_{34})	15 (ENTER)	?	
15.	(a_{41})	−18 (ENTER)	?	
16.	(a_{42})	65 (ENTER)	?	
17.	(a_{43})	−74 (ENTER)	?	
18.	(a_{44})	46 (ENTER)	89.40408949	0
19.		(ENTER)	−49.70138121	0
20.		(ENTER)	91.64864585	88.72786239
21.		(ENTER)	91.64864585	−88.72786239

(Ex. 2):

$$\begin{bmatrix} 38 & -83 \\ -27 & 54 \end{bmatrix}$$

Answer: $L_1 = 94.01042$, $L_2 = -2.01042$

```
1Ø "A"
     : CLEAR
     : PAUSE "EIGEN VALUE"
2Ø INPUT " ORDER  "; C
3Ø FOR G=1 TO C
4Ø FOR H=1 TO C
5Ø BEEP 1
     : PAUSE USING "#####"; G; H
     : INPUT I
6Ø GOSUB 6ØØ
7Ø A(K)=I
     : NEXT H
     : NEXT G
8Ø USING
     : FOR E=1 TO C-1
9Ø H=E
     : FOR G=1 TO C
1ØØ GOSUB 6ØØ
11Ø A(L)=A(K)
     : NEXT G
12Ø D=E+1
     : G=D
     : H=E
     : GOSUB 6ØØ
13Ø F=A(K)
     : FOR H=E TO C
14Ø GOSUB 6ØØ
15Ø A(K)=A(K)/F
     : NEXT H
16Ø FOR A=1 TO C
17Ø IF A=D GOTO 23Ø
18Ø G=A
     : H=E
     : GOSUB 6ØØ
19Ø F=A(K)
     : FOR H=E TO C
2ØØ G=A
     : GOSUB 6ØØ
21Ø J=K
     : G=D
     : GOSUB 6ØØ
22Ø A(J)=A(J)-F*A(K)
     : NEXT H
23Ø NEXT A
     : FOR A=1 TO C
24Ø F=Ø
     : FOR H=1 TO C
25Ø G=A
     : GOSUB 6ØØ
26Ø J=K
     : G=H
     : GOSUB 6ØØ
27Ø F=F+A(J)*A(L)
     : NEXT H
28Ø G=A
     : H=D
     : GOSUB 6ØØ
29Ø A(K)=F
     : NEXT A
     : NEXT E

3ØØ FOR A=1 TO C
31Ø G=A
     : H=C
     : GOSUB 6ØØ
32Ø J=K
     : G=C-A+1
     : GOSUB 6ØØ
33Ø A(L)=-A(J)
     : NEXT A
34Ø G=1
     : GOSUB 6ØØ
35Ø E=1
     : M=C
     : GOSUB 5ØØ
36Ø FOR E=1 TO C
```

(Continued on Page 37)

Pocket Computer (From Page 36)

```
37Ø A(F)=A(L)
     : F=F+1
     : L=L+1
     : NEXT E
38Ø A=Ø
     : B=Ø
39Ø IF 2>=C GOTO 45Ø
4ØØ GOSUB 8ØØ
41Ø IF (ABS F>=Ø.ØØØ1) + (ABS G>=Ø.ØØØ1) <> Ø LET A=A+F
     : B=B+G
     : PAUSE F, G
     : GOTO 4ØØ
42Ø GOSUB 9ØØ
43Ø FOR E=1 TO C-2
     : GOSUB 5ØØ
44Ø A(F)=A(I)
     : NEXT E
     : C=E
     : GOTO 38Ø
45Ø E=1
     : GOSUB 5ØØ
46Ø A=A(F)
     : IF C=1 BEEP 3
     : PRINT -A, Ø
     : END
47Ø E=2
     : GOSUB 5ØØ
48Ø B=A(F)
     : GOSUB 9ØØ
49Ø END
5ØØ G=E+13
     : H=G+1
     : F=M+H+1
 51Ø I=H+1
     : RETURN
 6ØØ L=G+15
     : K=15+C*G+H
 61Ø RETURN
 7ØØ GOSUB 5ØØ
 71Ø A(I)=A(F)-A*A(H)-B*A(G)
     : RETURN
 8ØØ E=1
     : GOSUB 5ØØ
 81Ø A(G)=Ø
     : J=Ø
     : K=1
     : A(H)=1
 82Ø GOSUB 7ØØ
 83Ø L=A(I)-A*K-B*J
     : E=E+1
 84Ø IF C<>E LET J=K
     : K=L
     : GOTO 82Ø
 85Ø GOSUB 7ØØ
 86Ø H=A(H)
     : I=A(I)
     : D=K*K-J*(L-H)
 87Ø F=(H*K-I*J)/D
 88Ø G=(I*K-H*(L-H))/D
 89Ø RETURN
 9ØØ BEEP 3
     : A=A/2
     : B=A*A-B
     : IF B=Ø GOTO 93Ø
 91Ø E= √ ABS B
     : IF Ø>B PRINT -A, E
     : PRINT -A,-E
     : RETURN
 92Ø PRINT E-A,Ø
     : PRINT -A-E, Ø
     : RETURN
 93Ø PRINT -A, Ø
     : PRINT -A, Ø
     : RETURN
```

Notes (From Page 29)

August, 1981

Checker Board

From L. F. Perry of Denton, Texas we got this:

I really enjoyed my first issue of your newsletter. I have had my color computer for only a month-and-a-half, and I must say that I never used anything as much as I have this computer. It has far surpassed my expectations, and surprised me at it's capabilities several times.

I do have a couple of, what I think are, valid complaints. Radio Shack needs to put out a manual on assembly language and machine language programming for the color computer. Why isn't Radio Shack going to offer more than a 32K up-grade for this computer? It should be easy for you to do it using the program-pak port. Another company is already offering this feature, plus disk drive. I would personally rather have Radio Shack equipment and not have to buy from someone else to upgrade my system. Lastly there is another thing that bugs me, the cost of your program paks. $39.95 for a game is out-of-line. I know that we all need to make a profit, but that is a little much! I am not going to buy anymore games programs unless the price goes down. I know several people who feel exactly as I do, so I know I am not alone.

(Editor's note: I asked the Color Computer Product Line Manager about the pricing of ROM cartridges. What I got was a very detailed discussion of the several differences between producing software on cassette versus ROM.

Cassette duplication is a relatively straight-forward operation once you have perfected the techniques. The cost of changing from one program to another is little more than the cost of a master tape. This allows cassette software to remain relatively inexpensive. Further, if an error is discovered, it is possible to reuse the cassettes. On the negative side, cassettes can be erased (requiring the purchase of a new one), sometimes fail to load, bind, and have other mechanical problems, as well as requiring some knowledge of the computer system to run.

First, ROMs are not reuseable. ROM cartridges require precision masks so the production ROMs can be produced, a much more elaborate (read expensive) production facility for the ROMs, additional parts (PC boards, sockets, etc.), extra inspections, and a greater amount of assembly time. It is these differences that add to the cost of ROM programs and require the higher prices. End of Note.)

Enough of my complaints. I enjoyed the graphics programs in the August Newsletter and especially liked the Checker Board program. There are several problems (bugs) in the version using the joysticks. If the joystick being read changes position, either by itself, or by the user, during a jump removal, and the blinking box is on a white square, when the button is pushed the white square will be filled with orange, and the checker that was to be removed stays put. Next, reading the joystick button the way it was listed in the joystick version causes the computer to mis-interpret the instructions and proceed to the next subroutine.

I changed the program to eliminate these two problems. The main thing that bothered me about the change to the joysticks was that a small square was left behind after being on a white square. I changed the program to make the whole square pointed to by the joystick "pulse." I think this improvement in the program not only made the board cleaner looking, but also made it easier to see the joystick position.

There were also some printing errors and I listed them in my list along with the correct changes.

Errors:

Lines 137 and 138 should read PAINT not PRINT.
Line 221 should have 0 = 5 not 0(zero) = 5.
Line 200 should be deleted.
Line 825 should read

Continued on Page 38)

Notes (From Page 37)

```
DRAW"BM 17,22; C2; U18; L9; BR8; R9; C4"
```

To keep the computer from jumping to a subroutine when a joystick button is not pushed, change these lines to read:

```
195 PQ=PEEK(6528Ø)
22Ø IF PQ=255 OR PQ=127 THEN 21Ø ELSE SOUND 2ØØ,3
96Ø IF PQ=255 OR PQ=127 THEN 9ØØ ELSE SOUND 2ØØ,3
```

Add these lines:

```
196 IF PQ=255 OR PQ=127 THEN 19Ø ELSE SOUND 2ØØ, 3
219 PQ=PEEK(6528Ø)
959 PQ=PEEK(6528Ø)
```

To keep a white square from accidentally being filled in, add these lines:

```
961 IF D=7 THEN 964 ELSE IF D=5 THEN 964 ELSE IF D=3 THEN
     964 ELSE IF D=1 THEN 964
962 IF D=2 THEN 966 ELSE IF D=4 THEN 966 ELSE IF D=6 THEN
     966 ELSE IF D=8 THEN 966
963 SOUND 1, 1Ø
     : GOTO 9ØØ
964 IF E=2 THEN 97Ø ELSE IF E=4 THEN 97Ø ELSE IF E=6 THEN
     97Ø ELSE IF E=8 THEN 97Ø
965 SOUND 1, 1Ø
     : GOTO 9ØØ
966 IF E=1 THEN 97Ø ELSE IF E=3 THEN 97Ø ELSE IF E=5 THEN
     97Ø ELSE IF E=7 THEN 97Ø
967 SOUND 1, 1Ø
     : GOTO 9ØØ
```

To cause the square at the joystick position to "pulse," change these lines to read:

```
72Ø LINE(TT-3Ø, UU-24)-(TT+2, UU), PRESET, B
74Ø LINE(TT-3Ø, UU-24)-(TT+2, UU), PSET, B
82Ø LINE(PP-3Ø, OO-24)-(PP+2, OO), PRESET, B
84Ø LINE(PP-3Ø, OO-24)-(PP+2, OO), PSET, B
93Ø LINE(EE-3Ø, DD-24)-(EE+2, DD), PRESET, B
95Ø LINE(EE-3Ø, DD-24)-(EE+2, DD), PSET, B
```

I hope you like the changes.
P.S. I couldn't find any use for line 45 so I deleted it.

Pocket Computer Bugs, Errors, and Fixes

Owners Manual

We recently received this information from J. W. Holzwarth of Ambler, Pennsylvania:

The Guess Number Game, Page 108 of the Pocket Computer Owner's Manual, does not duplicate the game of Master Mind ™. If this particular version was designed to keep from infringing on a copyright, then you, no doubt, are aware of the following: The program, as listed, will indicate the presence of more than one of a certain digit if more than one of that digit is present in the trial number but only one is present in the computer selected number.

Assume the computer selected number is 1234. If the trial number is 1111, the acknowledgement will be ABBB. The true Master Mind acknowledgement should be ACCC.

Assuming the same selected number, i.e. 1234, and the trial number is 9115, the acknowledgement will be BBCC. The true Master Mind acknowledgement should be BCCC.

To "correct" the program, make the following changes and additions:

```
21Ø A=B-9
     : IF A(C)=A(B) LET H=H+1
     : A(C)=11
     : C=16
231 A=E
232 FOR B=16 TO 13 STEP -1
234 GOSUB 1Ø
236 NEXT B
```

This version increases the execution time slightly; however, the program will now operate in a manner identical to the true Master Mind format.

Model I TERM changes

S. Nordhaus Tarzana, California

I would like to suggest the following changes and/or corrections to the TERM program which is given on pages 23-25 of the Model I RS-232C Interface booklet, cat. no. 26-1145.

The PATCH listed on page 23 in the above booklet has 4 or 5 errors which interfere with execution &/or assembly. Here are the corrected lines:

```
ADDR  Object Code          Source Code

5ØC4  E6 38                AND 38H
5ØC6  28 Ø5                JR  Z,NOFLT
5ØDØ  FE 6Ø                CP  6ØH
5ØD2  C3 1C 5Ø             JP  5Ø1CH
```

Also, Tape Editor/Assembler does not support uses of some of the labels, pseudo-ops, etc. in the above listing, so if you need to enter TERM in assembly language here are the other changes you need to make:

1. Replace labels that have a "$" in them (i.e. use DSP for DSP$; KBD for KBD$; CIO for CIO$ throughout).

2. If you are implementing PATCH, replace assembly line 26 with:

```
JP PATCH
```

and delete assembly line 27. If you do this, also change the second line of PATCH to:

```
LD A, (5Ø65H)
```

and, change the last line of PATCH to:

```
JP 5Ø1BH
```

(an unconditional jump to the object code line immediately following JP PATCH).

3. Replace the pseudo-op "BYTE" in assembly lines 69, 71-75, 140-147, and 151-154 with "DEFB" (a pseudo-op supported by the Editor/Assembler), and replace the pseudo-op "WORD" in assembly line 70 with "DEFW." Assembly lines 71-75 should be:

```
DEFB ØA4H    DEFB 77H    DEFB 47H    DEFB ØØH    DEFB ØA4H
```

respectively.

4. Assembly line 122 should be:

```
OR Ø4H
```

instead of 05H. Likewise the object code should be:

```
F6 Ø4.
```

5. You may want to change assembly line 151 to:

```
SPECTB DEFB Ø4H
```

instead of

```
Ø3H
```

to be able to generate control "A" (attn, EOT) with shift-down-arrow-A

6. As the last line of your assembly/source code, append END 5000H.

HALF-DUPLEX OPERATIONS

Enter the assembly language/source program as above but in between assembly lines 46 and 47 insert CALL DSP (which is CD3300 in object code). This merely causes what you type to be displayed on the screen. Also, the second line of PATCH should then be changed to LD A, (5068H) to settle up with the 3 new lines of object code just added.

TRS-80 Application (From Page 1)

"We have written a program which uses the VisiCalc matrix to keep track of energy consumption and costs for the City's own facilities. As consumption and cost figures are entered into the file, projections are made to the end of the budget year and these projected figures are compared to both last year's expenditures and the budget allocations for this year. It is possible to tell if consumption is up in relation to last year and, if energy continues to be used at the current rate, whether you will go over your annual budget.

"We have worked most of the bugs out of this system and are negotiating with the local utility company to see if they can provide us with data that can easily be entered into the computer by clerical staff. We are also starting to design a similar system to keep track of building inspections done for energy retrofit items. Through this system we hope to project not only numbers of inspections but also revenues from fees—obviously an important area for which to have accurate records."

Applications and Ideas in Abundance

"The third area in which we have been working on possible applications is solar engineering calculations. To date, we have acquired about a half dozen programs to do heat gain and loss calculations, including several programs which can calculate needs and performance of solar energy systems.

"Additional future applications include statistical analysis and a building monitoring program. We are currently exploring two new applications for the TRS-80. The first is the possibility of holding a conference at the beginning of next year on microcomputers and energy. At this conference, which would probably be held at the University of California at Davis, we might explore such topics as:

1. Using Microcomputers to do energy audits
2. Using Microcomputers for engineering calculations
3. Microcomputers and the Architect
4. Cost accounting and control and Microcomputers
5. Connecting Microcomputers to larger mainframe systems.

"The second project we are looking into is using the TRS-80 as an educational tool in energy. One possibility is to use the computer to present information at the second annual Retrofit Fair that Davis will be holding this fall. This could also include some on-line demonstrations by connecting the TRS-80 with other mainframe computers. I understand that the Department of Energy has developed several programs for this type of energy education effort and we are currently trying to locate them.

"Another possibility is to use the microcomputers in drafting and design classes in the local schools. As you may know, high school students who take drafting courses frequently design houses as a class project. We are trying to determine if there is any interest among the instructors in urging the students to design solar houses. They could then use some of the computer programs we have identified to help in their design and analysis. This would work out well since several of the local school districts have quite a few TRS-80s for student use."

Some Observations

"After using your system for seven months, I would like to make some initial observations. Some of these observations are based on my own experience and some are based on the experience of several associates who have recently purchased Model III machines after examining the one we're using.

"To begin with, we all have had little experience with computers in the past outside of a classroom environment. This has made the courses offered at the Radio Shack Computer Centers invaluable. All the courses are excellent—primarily because you are able to actually sit in front of a computer while you are learning.

"After becoming familiar with the computer's operation, the first professional applications have been those that had previously been done by hand. The first impact, then, is increased productivity—in our case the use of Scripsit for ordinance preparation. After some time, we began to explore the use of the computer for other applications in earnest. The first step that seemed reasonable was to take existing programs and adapt them to our needs. VisiCalc was tailor made for this. It was a relatively easy task to set up the matrix, particularly since this could be done without knowing BASIC or some other language.

"It is only recently that we have begun to modify some general BASIC programs for specific energy applications. Interestingly, this delay has not been primarily a matter of lack of expertise—we have many people interested in microcomputer programming who are willing to work with us, if we are able to determine what our needs are. What is difficult is making that jump from an idea of needs to a specific problem that is solvable by computer."

From the above comments, it appears that the Model III has a bright future in Davis as well as many other cities, as more people become aware of the TRS-80 reputation and capacity for hard work.

PRINT to LPRINT Conversion

Tom Doblmaier Summit, New Jersey

If you have BASIC lineprinter software which uses LPRINTS, this program may be of use to you. This program changes LPRINT statements to PRINT statements in your BASIC program. This is useful if you want a screen output without rewriting your program.

```
FOR X=17129 TO 20479
    : IF PEEK(X)<>175 THEN NEXT ELSE POKE X,178
    : NEXT
```

This program is executed from the command line. It peeks into RAM and if it finds a 175 (code for a LPRINT statement), it changes that code to a 178 (code for a PRINT statement).

Note:To change PRINTs to LPRINTs, change the program to read:

```
...IF PEEK(X)<>178.....POKE X,175...
```

LPRINT to PRINT Conversion #2

Jeffrey W. Hix Bellevue, NE

The following short program converts PRINT commands to LPRINT commands.

```
65399 END
65400'"CHANGER/PR",A  Submitted by Jeffrey W. Hix Bellevue,
    Nebraska  68005
65425 FOR QU=17128 32767
65450 Q1=PEEK(QU)
    : IF Q1=178 THEN POKE QU,175
65475 NEXT QU
65500 END
```

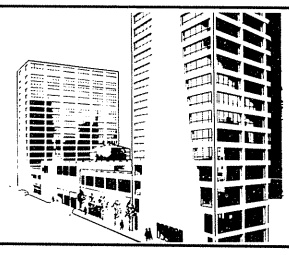
As listed it is subroutine SAVEd as "CHANGER/PR", A. Edit the numbers in 65450 to reverse the procedure and put on screen what would be LPRINTed on paper. This way I can edit my word processing (run as a program which this is) and then LPRINT it on paper when corrected. It was put together on a TRS-80 Model III (48K) 2 disk unit.

Radio Shack®
MICROCOMPUTER NEWS
P.O. BOX 2910
FORT WORTH, TEXAS 76113-2910

ADDRESS CHANGE
☐ Remove from List
☐ Change as shown
Please detach address label and mail to address shown above.

IF UNDELIVERABLE DO NOT RETURN

Fort Worth Scene



They're here! They're here!

Finally, back issues of the Microcomputer NEWS are available. We have reprinted ALL back issues of the newsletter from the very first one up through December of 1980. If you have been wanting a complete set of back issues, or if you just need some of the information, like Allan Emert's Machine Language Sort routine, now is your chance.

These reprints should be available from your local Radio Shack store or Computer Center. The stock number is 26-2115 and we have suggested a retail price of $4.95. If the store does not have the set, ask them to order it for you.

What about 1981? Well, back issues for 1981 are currently NOT available. We do plan to have a 1981 reprint set available sometime after the first of 1982, so please be patient.

One of my "hobbies" is going to school. I am currently enrolled in a COBOL programming course, and I thought I would pass along some information that I found out recently about moving COBOL from one computer to another.

I have a 48K two disk Model I system at home, which I used to enter my COBOL source program. Being behind, as usual, I decided (at 2:00 AM) to do the compile on my Model II at work.

My intention was to load the source file into Model I SCRIPSIT and use the serial printer driver to transmit it to my Model II. Unfortunately, the source file was too large to fit into SCRIPSIT in the 32K Model I we have in the office.

I quickly loaded Model I HOST for 32K and soon had the source file (plus a little garbage) saved on Model II as an ASCII file (thanks to Model II's TERMINAL program).

Now to get rid of the garbage. Again, no real problem. I loaded the Model II ASCII file into Model II SCRIPSIT 2.0, eliminated the garbage and saved the file back out onto disk as an ASCII file.

I had obtained a copy of the current 1.3b COBOL on TRSDOS 2.0a and everything looked good. I figured I could load the ASCII file into the Model II CEDIT (COBOL Editor) program, make sure everything was alright, and then compile. The CEDIT program promptly informed me that it could not read the ASCII file! Now what? A quick check of the directory informed me that my source code was stored with FIXED length records, and the COBOL files used VARIABLE length records.

Somehow I had to convert the fixed record length file into a variable record length file. I quickly scanned the Model II COBOL, TRSDOS, and BASIC manuals with no luck.

Fortunately, at this point I remembered TEXT EDITOR. It seemed that I had read that Text Editor would allow me to convert files for use with COBOL. Sure enough, a quick review of the manual, a few commands, and I soon had a COBOL compatible source file.

I learned (or relearned) two things:

1. Make sure all the programs you plan to use are on the same version of TRSDOS. (Hidden in the statement that I obtained the latest version of COBOL is a story of great frustration since my COBOL was on TRSDOS 1.2, not 2.0a).

2. Don't overlook a program just because you don't see an immediate use for it. I use SCRIPSIT constantly, and one of the ways I use it is as a very powerful text editor. I had read the Text Editor manual several months ago during the review process, but figured that I would never have any particular use for it. I now have a little more respect for Text Editor.

TRS-80 Microcomputer News is published monthly by Radio Shack, a division of Tandy Corporation. A single one year subscription is available free to purchasers of new TRS-80 Microcomputer systems with addresses in the United States, Canada and APO or FPO addresses. Subscriptions to other addresses are not available. The subscription rate for renewals and other interested persons is twelve dollars ($12.00) 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.

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.