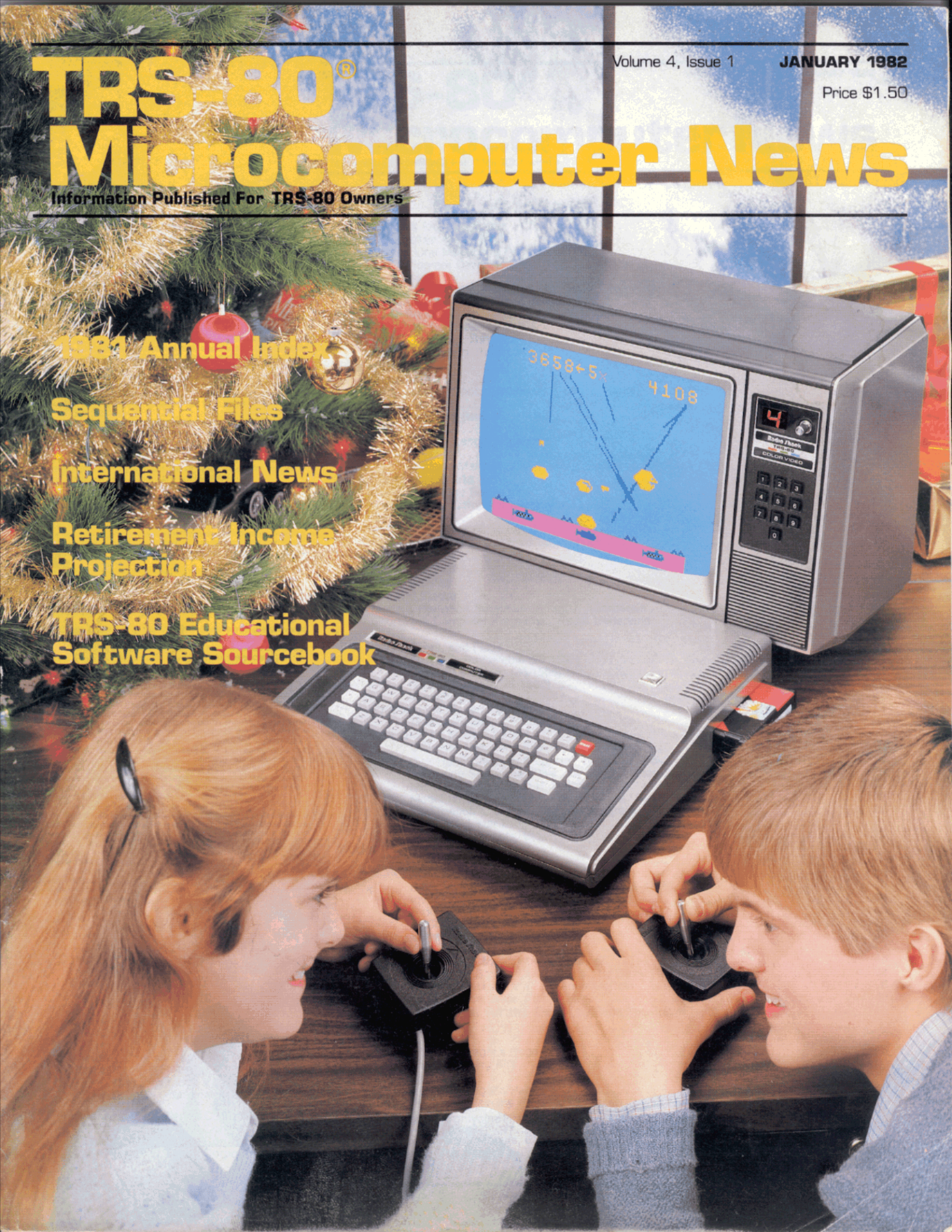
Volume 4, Issue 1 JANUARY 1982
Price $1.50

TRS-80® Microcomputer News

Information Published For TRS-80 Owners



1981 Annual Index

Sequential Files

International News

Retirement Income Projection

TRS-80 Educational Software Sourcebook

Pictured on the front cover are two children still enjoying a game of Polaris (26-3065) with their favorite Christmas gift—The TRS-80 Color Computer (26-3002) and Joysticks (26-3008).

TRS-80® Microcomputer News

Volume 4, Issue 1 **JANUARY 1982**

Price $1.50

Information Published For TRS-80 Owners

TRS-80 Microcomputer News is published monthly by Radio Shack, a division of Tandy Corporation. A single six-month subscription is available free to purchasers of new TRS-80 Microcomputer systems with addresses in the United States, 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 ($18.00 in Canada), check or money order in U.S. funds. All correspondence related to subscriptions should be sent to: Microcomputer News, P.O. Box 2910, Fort Worth, Texas 76113-2910.

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

Back issues of Microcomputer News prior to January, 1981 are available through your local Radio Shack store as stock number 26-2115 (Suggested Retail Price $4.95 for the set). Back issues of 1981 copies are not available.

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

Material may be submitted by mail to P.O. Box 2910, Fort Worth, Texas 76113-2910, or through CompuServe. The Microcomputer News' CompuServe user ID number is 70000,535.

Notes to Program Users:

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

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

Comments on our program listing style:

In order to make the program listings we publish easier to read, we have adopted a style of inserting spaces to enchance readability, and we seperate each program statement onto a seperate line. While these techniques increase program readability, they also require more memory, and may execute more slowly then the original program did.

When you are entering a program for your own use, you may wish to eliminate many of the extra blanks (see your owners manual for required blanks), and you should certainly move multiple statements up to a single line where possible.

Trademark Credits

Apple®	Apple Computer
CompuServe ™	CompuServe, Inc.
DIF ™	Personal Software, Inc.
Dow Jones ™	Dow Jones Co.
PET ™	Commodore Bus. Machines
Personnel Manager ™	Image Producers
Project Manager ™	Image Producers
ReformaTTer ™	Microtech Exports
Time Manager ™	Image Producers
VisiCalc ™	Personal Software, Inc.
Program Pak ™	Tandy Corporation
SCRIPSIT ™	Tandy Corporation
TRSDOS ™	Tandy Corporation
TRS-80®	Tandy Corporation

Contents:

View From the Seventh Floor

Jon Shirley
Vice President, Radio Shack Computer Merchandising

It's hard to believe that a new year is upon us already. During 1982 we will reach our fifth anniversary in the personal computer business which will, I guess, make us old timers in this new industry. 1981 was an interesting year. The list of companies that brought out personal (or very small business) computers is incredible. Off the top of my head I can remember IBM, DEC, Xerox, Data General, TI, Monroe, Pentel (who?) and several I forgot. Apple re-introduced the Apple III and Commodore introduced so many new cpu's that we gave up counting long ago. The Japanese were virtually unseen except for NEC in America but there were about two new computers a month introduced in Japan. Will this be the year of the Japanese kamakazi computer?

The software world was a little disappointing with no new VisiCalc's bursting on the scene and most of the new computer entries offering virtually no new software at all.

We did not introduce a new computer last year *but* we did add over 100 products to our computer product line including a lot of new software. We now offer more supported software than any other small computer manufacturer. We also offer more peripherals, accessories and supplies than anyone else. In 1981 we added over 100 Computer Centers and nearly that many service centers. We nearly doubled our hotline staff and cut their response time to minutes. And you will notice that with this issue of the Microcomputer News we have tripled its size. So we've been busy. Behind the scenes our R and D group has been busy, too, and you can look forward to a very interesting 1982 from Radio Shack. End of commercial.

In October I went to Boston, just in time for their Indian summer, to speak at a forum held by the Boston Computer Society and to visit the associated computer show. The Boston Computer Society was founded by one Jonathan Rotenberg five years ago when he was a 13-year-old high school freshman. Today the Boston Computer Society is one of the largest in the country and publishes a fine magazine. You can judge Jonathan's abilities by the fact that our forum included the presidents of Apple, Commodore and Microsoft plus representatives of Atari, Sinclair and IBM. If you are in the New England area and do not belong, I strongly suggest you check into the BCS, it's a great group and I will treasure my new lifetime membership.

Incidentally one of the speakers, Mr. Jim Finke, who is president of Commodore, really put us all in our place when he stated that the entire personal computer industry this year will equal the potato chip industry in volume of sales. He went on to say that if the projected 50 to 100% growth of our industry keeps up, by next year we will reach about half of the pet food business. By 1983, if the rapid growth continues, we will creep up on the pantyhose industry. Sort of made us all feel a little deflated.

Two magazines hit my desk the other day in the same three foot stack of mail and I was struck by their similar front cover look. One was Personal Computing, published by Hayden Publishing, which has been around for a while. In the issue was an article titled "What do you call **your** computer?" It took four pages to explain that there was no difference between a personal computer and a small business computer, which I think missed the point that there certainly is a difference depending on what software is available. That, however, is not what caught my eye. The article stated that until this summer the difference was easily answered: Radio Shack and two other famous names were personal computers while North Star, Vector and some others were small-business computers. Obviously the writer did not know of the TRS-80 Model II, and I wonder what the writer does know since he also stated that the TRS-80 Model I was a good "starter" machine, but . . . it was a 6502 machine which made it a little slow. I wonder if that is creative journalism?

The other magazine is a new offering from the folks who bring us BYTE (McGraw Hill, Inc.), and it is called Popular Computing. If you have tried to read through a copy of BYTE and discovered it was a little over your head, and it's way over mine, fear not, Popular Computing is for the beginner. If you are just getting started take a look at it—I think it will be a good publication and not just because two ex-Radio Shack people are among the editors.

Caveat Emptor revisited. My column in the September issue was at least read by someone. I received a letter from Mr. Warren G. Rosenkranz, President of VR Data Corp. Mr. Rosenkranz took exception to my column and stated that my comments were "harsh and anti-competitive." He correctly stated that his company and several others have given many people more reason to buy a TRS-80 due to the availability of low cost peripherals. He also stated that his company stands behind what it sells and that their disk drives can be troubleshot by the neophyte. His letter indicated that copies were sent to no less than nine computer magazines.

I really think Mr. Rosenkranz missed my point. Caveat Emptor means "buyer beware" and all I was saying in this column was that before you buy, be sure you know all the facts. For example, how does a neophyte align a disk drive that requires a very expensive professional oscilloscope? How about UL and FCC which Mr. Rosenkranz did not address in his letter? How about the fact that we cannot service these customer modified machines? None of these items were addressed in the ad in '80 Microcomputing that caused me to write the column in the first place. As long as we have a free market in America, and I hope that is forever, there will be price competition. And the price of any item must relate to value received. We do not have the lowest price on disk drives for the simple reason that we have over 6200 locations in America where you can buy one or take one back for service.

As I said at the end of my response to Mr. Rosenkranz "VR Data may be a fine company with a fine product, I really have no way to know, but are you willing to say that **all** of your competitors in the same business offer equal value and quality and will be able to stay in business?" So, dear reader, the choice is, as always, yours . . . but caveat emptor!

Until next month.

Sequential Files

JUST WHAT IS A FILE, ANYWAY?

Radio Shack's new book "TRS-80 Data File Programming" (62-2085, $11.95) defines a data file as "stored alphanumeric information that is separate and distinct from any particular BASIC program. It is located (recorded) on either a magnetic disk, diskette, or cassette tape. . . . A data file is program-independent. It . . . can be accessed and used by many different programs . . . each performing a different activity with that file's data."

For this article we will examine a file structure known as sequential files.

SOME VOCABULARY

File - A group or collection of data stored on tape or disk. Example: Name and Address file

Record - One complete set of information within a file. Example: One person's name and address

Field - One item of information within a record. Example: An individual's street address.

Delimiter - One or more characters used by the computer to mark the end of one data item and the beginning of the next.

EOF - End of File. Most of our computers offer an EOF function which lets you test for end of file. If you attempt to read past the end of file, an error will occur.

WHAT IS A SEQUENTIAL FILE?

A sequential file is a data file which is created and accessed one record after another. To get the tenth item in a sequential file, you must read through the first nine records before you can read the tenth. If you then wanted to read the ninth item, you would have to start again with record 1 and read sequentially until you got to nine.

Sequential files are the only type file available to computers users who only have tape systems. If you have a disk system, you have access to at least one other file type, random (direct) access, which we will briefly discuss next month.

Sequential files are used primarily for information which does not change frequently, like a name and address file.

HOW DO I BUILD AND USE SEQUENTIAL FILES?

In TRS-80 systems, (All models including Color and Pocket) there are two primary commands which are used for reading and writing sequential files: PRINT # and INPUT #.

In addition, for disk systems, and Color Computer Tape files you have two additional commands: OPEN and CLOSE.

OPEN when it is available, is used to give you access to a sequential file.

The CLOSE statement is used to free the file buffer for other uses, and to make sure that all information which should have been written to the tape or disk has been written.

The PRINT# statement is used to write information into a sequential file.

The INPUT# statement is used to read data from a sequential file.

Most sequential files can not be extended, or made larger than they currently are. Further, a sequential file can be OPEN for either an Input or an Output operation but not both. These limitations on sequential files require a special update procedure.

To update a sequential file:

1. OPEN the old file for Input.
2. OPEN a new sequential file for Output.
3. Read records in from the old file and make any needed changes.
4. Write out the corrected data record.
5. After all information, both new and old has been written to the new file, CLOSE both data files.
6. You may want to retain the old file for archival or BACKUP purposes. If you do not need to retain the old file, KILL it.

To extend a sequential file:

1. OPEN the file to be extended for Input.
2. OPEN a new temporary file for Output.
3. Read the old file directly into the new file.
4. When you reach the EOF for the old file, you are ready to extend the file by continuing to write to the new file.
5. When the extension is complete, KILL the old file and rename the temporary file to the old file name.

HOW IS DATA STORED IN A SEQUENTIAL FILE?

Data is written to tape or disk in essentially the same format as you would see the same information if it were printed to the video display rather than to tape or disk. Note, that commas and semi-colons which seperate data items will effect the way the information is stored on disk. Numbers are written in their full expanded format, and not in their compressed memory format.

Since the data is PRINT#ed to tape or disk in a specific sequence, it must be read in that same sequence. For example:

If you PRINT# the data using:

```
PRINT#1, A, B$, C
```

you will have to read that data with a similar INPUT# statement, such as:

```
INPUT#1, D, C$, F
```

SPECIFIC HINTS FOR SEQUENTIAL FILES:

If you are writing several strings to a sequential file, seperate one string from another by using an explicit comma:

```
PRINT #1, A$; ","; B$
```

this forces the comma delimiter between data items and allows the information to be retrieved properly.

If you are trying to write a string data item which contains a comma as a valid character, you will need to surround the data item with explicit quotes in the data file:

```
A$="JONES, JOHN J."

PRINT#1, CHR$(34); A$; CHR$(34)
```

The quotes tell the computer to accept all characters up to the next set of quotes as valid data.

For further information on sequential data files, look at how the various programs in this issue handle their data files. We have tried to include a fairly general sampling of programs which use sequential access techniques. You should also check your computer's operating manual for information which is specific to your machine. Radio Shack books of interest include:

"Programming Techniques for Level II BASIC"
(62-2062)
"TRS-80 Data File Programming"
(62-2085)

Accounts Payable System

(Model II, 26-4505)

Answers to Commonly Asked Questions

This discussion will cover Accounts Payable (APS) 26-4505. APS is the program with the second highest number of phone calls to Computer Customer Services.

Read the APS manual first. You should make sure that your copy of APS has been upgraded to correct all known errors by entering all corrections to the program. Your Computer Center can provide you with a list of these corrections, or you may want to ask the Computer Center to help you make the corrections. Then take your list of General Ledger accounts, as provided by your General Ledger program, and a backup copy of Accounts Payable and begin the program.

The APS will print checks, cash requirements, aging reports, and totals for posting to your company's General Ledger.

SETUP CYCLE

1. SETUP
   A. Initial Setup
      1. Enter Company Information
      2. Assign General Ledger Codes.
2. ADD VENDORS
3. ADD INVOICES

MONTHLY WORKING CYCLE

1. ADD/CHANGE VENDORS
2. ADD/EDIT INVOICES
3. INVOICE SELECTION
4. POST INVOICES
5. INVOICE SELECTION
6. CASH REQUIREMENTS
7. AGING REPORT
8. PRINT CHECKS
9. END OF PERIOD

On power-up, enter the TRSDOS date to correspond to the date of the activity you plan to enter. The printer should be ready before running the program. When TRSDOS Ready appears type in APSETUP and press ⟨ENTER⟩. This will be the initial setup. You are allowed to change your information later. Enter your company information and indicate which accounting method you wish to use - Cash or Accrual. You will be asked to enter a password. This password is assigned to your data files, and may be the same as or different from your diskette Master password. This password should not contain punctuation characters or blank spaces! Next, enter the General Ledger Codes (GL Codes). GL Codes 1 & 2 are already assigned for the Accounts Payable account debits and credits. GL Code 3 is your Cash (banking) account. GL Code 4 is Purchase Discount. The General Ledger Codes 5 - 100 may be defined by you. The GL Code is automatic. You enter a five letter abbreviation for the GL Account Name, and the GL Account Number to match the same account in the General Ledger program (26-4501).

You should understand the difference between GL Codes and GL Account Numbers, and the difference between GL Code Names and the GL Account Names. Some examples are:

GL Codes	GL Code Names	GL Acct. Number	GL Acct. Names
GLC 5	MERC1	1300	Merchandise Dept 1
GLC 6	MERC2	1301	Merchandise Dept 2
GLC 7	MERC3	1302	Merchandise Dept 3
GLC 8	OFFS	1320	Office Supplies
GLC 9	OFFE	1330	Office Equipment
GLC 10	SLTAX	2100	Sales Tax Payable
GLC 11	SHIPN	2200	Ship & Freight
GLC 12	MISC	1350	Miscel Expenses

You should print a General Ledger Code list and have it available when you enter invoices later.

If you get an error message of "Bad File Mode" while information is being stored, you did not follow the rules concerning the password. If you get a "Disk I/O" error, then the write enable tab is not on the diskette.

You may change the name of the GL Codes and GL Acct. No. if you need to. This must be done immediately after End of Period processing by going back to SETUP and indicating that you want to Change System Information.

To start the main program type in "APS" at TRSDOS Ready. The Main Menu displays the current status for invoices, vendors, and amount payable. Begin by adding vendors. The vendor name and number should be unique, because the program sorts on these two fields. If it finds either of these fields duplicated, the file organization is confused and this renders the diskette useless.

You are not allowed to enter YTD purchases and payments or the amount and date of last payment when adding the vendors. The YTD Purchases and Credits field is updated by posting invoices. The YTD Payments and Debits field is updated by posting payments. When you first begin to use this Accounts Payable system you must enter all outstanding invoices, which will update the YTD Purchases and Credits field. If you begin to use APS after the beginning of the year, the YTD figures in APS **plus** the YTD figures from your previous records will be the correct YTD figures. After Year End Close Out, the YTD Purchases and Credits and YTD Payments and Debits fields will be set to zero. The APS system will then move the new year beginning balances into the YTD purchases and credits field.

One suggestion: Add all of your vendors, then make a BACKUP that has only the Company Setup and Vendor Information. Should you be forced to start over for some reason, this will save you several hours of entering this information into the system again.

Invoice Maintenance allows you to Add, Search, Edit, and Delete invoices. When adding the invoice, you enter the vendor number you assigned to this company, the invoice number, and due date.

If the vendor allows a discount when paid by a certain date, enter the discount date. The discount amount may be either a percentage or a dollar amount which you enter. If you enter "5" for a five percent discount, the discount will be calculated on the gross amount you enter for this invoice. It is possible that you may not be allowed a discount on freight. To calculate the proper discount, you can enter the discount as a percentage, then add on the items which are allowed a discount amount. When asked if this invoice is correct, answer

NO, then enter the discount as a dollar amount instead of as a percentage. Use the dollar value the computer calculated and then add the other (non-discounted) items.

When adding an invoice APS requests the GL Code, not GL Acct. No. The program allows four entries per invoice. If your invoice uses more than four GL Codes you will need to create two or more separate invoices (invoice numbers 1000A, & 1000B for example).

The Edit function creates a second invoice that offsets an old posted invoice. If the invoice has not been posted, delete the incorrect invoice and re-enter it correctly. The adjusted invoice created by editing allows you to change the discount date and following entries. Since the old invoice is not cancelled, on the adjusting invoice you should enter only the amounts you need to increase or decrease the original entries to get the correct amount.

The APS manual, pages 41-43, describes some special entries such as debit memos for the return of merchandise before you pay the invoice, and for return of merchandise after you pay the invoice and then receive a check from the vendor.

The invoice selection function is a two menu process. Select and Exclude invoices are opposites just as Hold and Release are opposites. You may use these functions to select a group of invoices on a particular due date or all the invoices for an individual vendor. You may select all posted invoices to print checks or you might select all invoices currently unpaid. If you have several invoices selected for various reasons, you may want to exclude all invoices then select a new group by a different status. You may post without selecting the invoices but you must select the invoices before printing checks. The Hold and Release options are specifically for printing checks. If an invoice is selected but is held, the invoice will not be paid.

The second menu shows the range of invoices you may choose (Select, Exclude, Hold, Release). The All Invoices option will choose all invoices. The posted invoices will choose only those that are currently posted. The Vendor/Invoice allows the operator to choose a vendor and choose the invoices one at a time or all of the invoices for this vendor. The Discount Date allows the operator to choose the invoices with a discount on and before this date. You may skip the discount date and choose by Due Date by pressing ⟨ENTER⟩ for Discount Date. The Due Date chooses the invoices due on and before this date. You may choose by both Discount Date and Due Date or by just one.

If you press ⟨ENTER⟩ for discount date and then enter a date for Due Date, all invoices due on or before the due date you entered will be selected EXCEPT those invoices with a 00/00/00 (i.e. no discount) discount date. For Example:

To select all invoices due on or before March 1, 1982 (including invoices both with and without discounts) follow these steps:

1. Select invoices by pressing ⟨ENTER⟩ for discount date and type 03/01/82 for due date. This will select all invoices due on or before March 1, 1982 which have discounts.
2. Make an additional selection by typing 00/00/00 for discount date and then type 03/01/82 for due date. This will select all invoices due on or before March 1, 1982 which have no discount.

If you have already selected a posted invoice by the Vendor/Invoices option, and then select all posted invoices, this invoice will not be selected twice. You can use Invoice Selection for posting then exclude all and use it again for Printing Checks.

Now would be a good time to get a list of invoices. This may be done by Complete List which prints all invoices or by Selected List which prints only selected invoices. You might want a list of your vendors as well. This can be Complete or it will select vendors who have invoices selected.

Posting is the next required step. Before you begin this step PLEASE make a BACKUP. Posting and printing checks are the two functions that involve the maximum amount of disk activity, with several files opened and closed, reads and writes back and forth to the data files, etc. If you should encounter an error condition, type in the command CLOSE and press ⟨ENTER⟩, then remove the diskette and turn the computer off. Contact the store that sold you the computer system and ask if any program changes have been made for this error. They will need to know which function you were performing, the Error Code number and the program line number. Example: "I was posting invoices and got an Error 5 in line 350".

The posting process will distribute the dollar values to the GL Codes involved and to the vendors. This will increase the amount payable on the Main Menu. Printing checks decreases the amount payable.

If you are on an accrual accounting method, you should wait until the end of the month and enter information from the General Ledger Recap - End of Period Report into the General Ledger or do the transfer if you have two drives.

A couple of reports you might want to do after posting are Cash Requirements and Aging. The Cash Requirements report gives a list of all posted invoices in order by Due Date except for those invoices being held. The Cash Requirements report takes a very long time to complete. APS may appear to hang up on the last vendor, but it is sorting the information. Do Not Hit the BREAK Key or the RESET Switch! Remember, patience is a virtue which you need when working with computers.

Before you print checks, you may print a preview of invoices to be paid. You may preview the checks and decide to exit because you need to hold some of the invoices or post more invoices. After printing the checks you have began a cycle that must be completed in an exact sequence. In this sequence, the checks are printed, a check register is printed showing check number, vendor name and number, the net amount of the invoice and checks, and whether a discount was taken. You then update the vendors balances.

If your accounting method is cash, you will record the General Ledger distributions only when the invoices are paid. When an invoice is paid, take the results from the posting summary and create a document in the General Ledger for the appropriate expense accounts and cash account.

The final step of the month for APS is to run the End Of Period (E-O-P). If you plan to use the automatic transfer from APS to GL, the transfer must be done before the E-O-P processing and must only be done once per month. At TRSDOS Ready type in APGL and press **[ENTER]**. Your APS should be in drive 0 and GL should be in drive 1. When the transfer is complete, exit to TRSDOS. Remove the GL diskette from drive 1. Go to the APS E-O-P Menu and do function # 1. Exit from APS. Put the GL diskette in drive 0 and run the GL program. Post the documents that were created by the transfer.

MOST FREQUENTLY ASKED QUESTIONS

Question 1: When storing Setup Information I get an Error 52 or 54, why?

Answer: You used a bad password. The password should not contain blanks or punctuation characters.

Question 2: The Main Menu shows I have a negative number of invoices, why?

Answer: You probably received an Error Code and failed to stop and go to a previous backup.

Question 3: The Main Menu shows I have a negative number selected invoices, why?

Answer: You deleted some invoices before they were posted, yet they were selected. You should exclude your invoices before deleting them.

Question 4: I have an invoice which I plan to pay on the installment plan, or I want to make partial payments. Can I?

Answer: The program only prints checks for the full amount of the invoice. When you enter the invoice, create separate invoices with the dollar amount you plan to pay during each month. If you do not know how much the payment will be, add the invoice for the full amount but do not post it. Each time you plan to print a check for this invoice create another invoice, (100X, 100Y, 100Z), with the amount you plan to pay this month. Delete the original invoice after it has been completely paid.

Question 5: My End of Period Report bottom line does not print "Net AP $0.00," why?

Answer: During the month when you purchase and post more than what you pay out, the result will be a net Credit amount to Accounts Payable. During the months when you pay more invoices than you record purchases, the net result to Accounts Payable will be a debit figure. The only time you will have a "Net AP $0.00" is when you record purchases (Credit AP) and payments (Debit AP) of exactly the same amount.

Question 6: "I try to print checks but APS does not go beyond Preview of Checks, why?"

Answer: You may not have any posted invoices. You may not have selected any invoices. You may have the invoices on hold.

MODEL I/III QUESTIONS

Question 7: The May, 1979 Newsletter contained several line changes to Model I Tape Mailing List (26-1503) which allow the Sort to be optional, as well as allowing a sort by Zip Code. I made these changes but now the Search function does not seem to be reliable. What can I do?

Answer: The following additional changes were provided by one of our customers to correct this problem and allow the search function to work reliably:

Old Line: `95Ø J=1: K=N`

New Line: `95Ø I=1: K=N`

Old Line: `96Ø IF K<J THEN ...`

New Line: `96Ø IF K<I THEN ...`

Old Line: `97Ø I=INT((J+K/2):B$(1)= ...`

New Line: `97Ø B$(1)= ...`
(Delete the first statement in the line)

Old Line: `98Ø IF B$(Ø)<B$(1) THEN K=I-1:GOTO 96Ø`

New Line: `98Ø IF B$(Ø)<>B$(1) THEN I=I+1:GOTO 96Ø`

DELETE LINE 990 ENTIRELY

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

Model II Scripsit 1.0 Underline

Anne McCoy
SECURITY MUTUAL LIFE INS., CO.
P.O. Box 82248
Lincoln, NB 68501

I have just finished reading the October, 1981 issue of TRS-80 Microcomputer News. The Answers portion of the Model II SCRIPSIT section caught my eye. John Hawkins of K-Fox Radio wanted to print just an underline or line rather than underscore characters using Model II's SCRIPSIT 1.0.

I battled with this very problem several months ago and after many hours came up with what I feel could be an easier method to print a line such as a fill-in-the-blank type of line or several lines.

I first type the normal SHIFT HYPEN code for underline begin. Then I type SHIFT HYPHEN code to end the underline **next** to the begin code. I then use the left arrow to go back one space to the end code. Using the F1 key I "open up" the codes. Then using the right arrow I space over however many spaces I want my blank line to be. Pressing the F1 several times will make several lines of "blank lines." Then I press the F2 key to "close" the underline.

I have programmed a user control key to automatically make a 5-space blank line for me. We use this function quite often when a figure or word is to be filled in later. My code for the user control key is: SHIFT HYPHEN/SHIFT HYPHEN/ LEFT ARROW/F1/RIGHT ARROW/RIGHT ARROW/RIGHT ARROW/RIGHT ARROW/RIGHT ARROW/F2/RIGHT ARROW. The last right arrow allows me to continue typing in the correct position after the blank line. The code for the user control key lookes like this on the screen: —— LEFT F1 RIGHT RIGHT RIGHT RIGHT RIGHT F2 RIGHT. What appears on the screen when using this key looks something like this: _———————_

The arrow keys **must** be used rather than the space bar or it will not work. If you think my solution merits printing in the Microcomputer News, you may use it. I think it may be less confusing than the CONTROL 9 6 2 etc.

New Services

The Multiple Choice, Standard & Poor's General Information, Archer Commodities

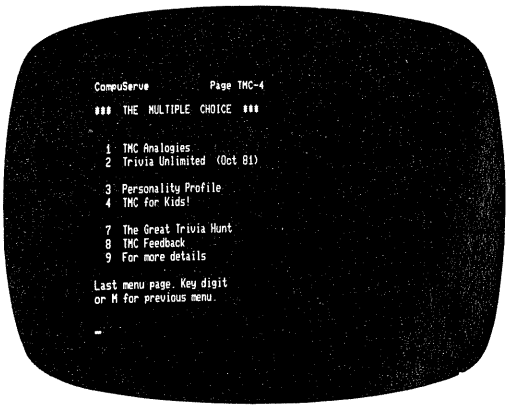
Editor's Note: The CompuServe Information Service is one of the largest information and entertainment services available to owners of personal computers and computer terminals. With each issue of TRS-80 Microcomputer NEWS, various features of CompuServe will be discussed. The CompuServe Information Service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

One of the newest providers of information to the CompuServe Information Service sees to it that you always have a choice.

QUIZZES, TRIVIA AND MUCH MORE

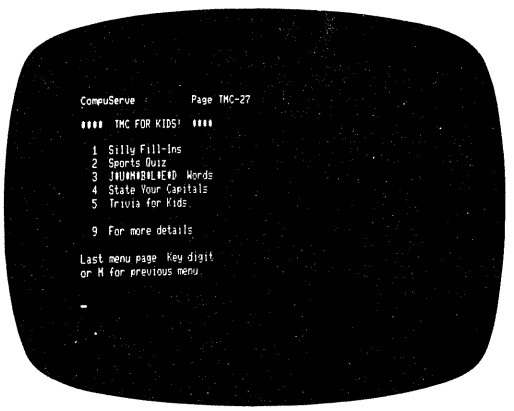
The Multiple Choice, an organization that provides recreational and educational multiple choice tests, is now an information provider to the CompuServe Information Service.

Trivia buffs will especially enjoy The Multiple Choice. Once in the program, they can choose a test with topics ranging from sports to movies to history for sharpening their trivia skills.



Students on the CompuServe network may gain that extra edge by quizzing themselves on a variety of subjects, including science and literature. And for those comtemplating something a little tougher, The Multiple Choice offers an analogy quiz that is similar to those taken by college students seeking admission to graduate school.

The Multiple Choice plans to add additional tests in the future including a children's series with programmable skill levels. Programmable skill levels ensure that the test material offers the greatest challenge and therefore, the greatest progress.

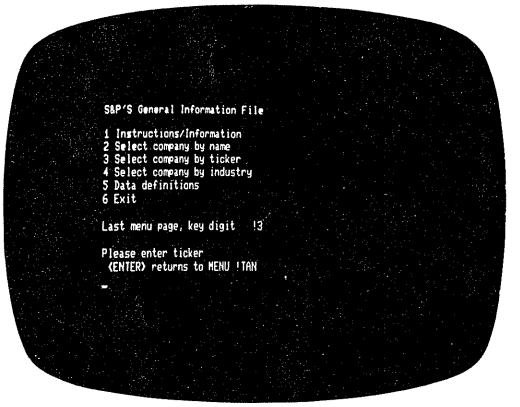


Offerings on The Multiple Choice are designed to involve the entire family with the computer in a family learning atmosphere.

CompuServe subscribers can access the service provided by The Multiple Choice through main menu item 8, Education.

STANDARD & POOR'S GENERAL INFORMATION FILE

Standard & Poor's General Information file is an easy and inexpensive way to obtain information on many companies.



Information, available for 25 cents per company access, is plentiful and varied. The S & P datafile for each company listed includes a Business Summary, Important Business Developments, Product/Service Line contributions to revenue/profits and selected financial items. More general information includes a corporate address and telephone number, list of directors and principal officers and stock transfer agents.

The specific categorical information contained in Standard & Poor's Business Summary section includes company products, size, recent financial developments and, in many cases, the long-term outlook.

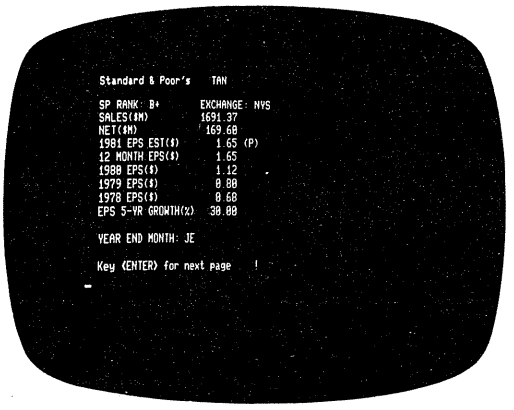
The Important Developments section describes recent events such as any recent contracts awarded to a company, or new product developments. The information is as current as possible.

On the Product/Service Line breakdown, Standard & Poor's indicates what percentage of total profits or sales is made up of a certain product or service. In the case of Norton Co., as an example, abrasives may make up 61 per cent of total sales and 47 per cent of total profits, while diversified products and subsidiary companies provide the remainder.

Financial information includes the latest available figures on net sales and net income, plus three years of earnings-per-share history, including current year estimates when available.

Standard & Poor's also provides a five-year earnings-per-share growth rate and assigns a Standard & Poor's Earnings and Dividend Rank, which is a letter ranking system (A-, B+, etc.) based on history.

Users may access information on the database using three methods: company name, ticker symbol (e.g., HRB for H & R Block), or unique industry code. The latter, for example, might be Industry Code 4300 which provides an alphabetized

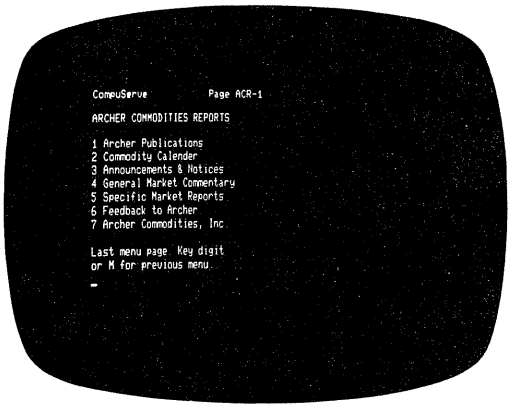


list of all companies in a particular industry.

The Standard & Poor's General Information File database is made up of major, publicly held companies — IBM, Tandy, McGraw-Hill and H&R Block, just to name several included in the 3,000-member list. A great variety of individuals can benefit from using the File: Investors, to obtain information on specific stocks; marketing executives, searching for possible service areas; corporate planners; job seekers; and financial institution managers.

ARCHER COMMODITIES

Archer Reports consist of a wide-ranging variety of information on all aspects of commodities trading. Containing such categories as Commodity Calendar, General Market Commentary and Specific Market Reports, Archer Commodities provides expertise and personal attention.



CompuServe customers receive, in addition to current market reports and commentary, educational and informational material on future trading plus special announcements from Archer Commodities. Subscriber requests for charts, market newsletters and quotation equipment are handled by Archer on an individual basis through an interactive feedback procedure.

Archer Commodities, Inc. is a commodity futures brokerage firm located in Chicago, and is a wholly-owned subsidiary of Heinold Commodities.

OTHER NEW SERVICES ON CompuServe

Popular Science magazine has joined CompuServe as a provider of information.

Found under main menu item 7, Home Information, Popular Science offers two kinds of information: Popular Science Energy news and Popular Science Product news.

A feedback feature in the Popular Science section allows CompuServe customers to participate in various contests sponsored by Popular Science. It also allows two-way communications between the magazine staff and its readers.

The Aviation Safety Institute, located under main menu item 6, Special Services, has started an Aviation Safety Special Interest Group (SIG).

A bulletin board has been established to allow pilots and other people to report aviation hazards, to discuss databases and to swap ideas and information relative to aviation safety.

With the activity surrounding air traffic controllers and aviation safety in general, you might want to look at the aviation safety SIG area.

CompuServe has added many new games under main menu item 3, Entertainment. Some of the more recent ones include Eliza, the on-line psycho-analyst; Go-Moku, a game similar to tic-tac-toe; Hammurabi, a game which puts you in the ruler's seat in governing an ancient city; Furs, where you play a 17th Century fur trader; Fantasy, which utilizes armies and weapons for conquering territory; and Dots, where you play against the computer in drawing lines to form boxes.

For those who follow the financial information on CompuServe, two more financial news columns have been added recently.

Investment News and Views, a Battery Lane publication, talks about the latest in investment news and activities.

Also, The Small Business Report, edited by Mark Stevens, provides news geared toward the small business person.

CompuServe adds and updates services every week. Check What's New under main menu item 5, CompuServe User Information, or Go NEW-1, to read about the most recent additions.

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

National Crisis Center for the Deaf

Follow-Up Letter From an Amateur Radio Operator

I have just come across the article "National Crisis Center for the Deaf" in your June, 1981 issue. You may be interested in knowing that there is software developed for communications with TRS-80, as well as other microcomputers, using the TTY/TTD (baudot) format.

Amateur Radio hobbyists have, historically, been among the pioneers of any communications frontier, and it has been no different with the development and utilization of computers in this field. I know of many Amateurs that are using their Radio Shack TRS-80 microcomputer in their radio communications hobby; some with programs allowing keyboard selectable speeds and modes including ASCII, baudot, and even Morse Code!

The construction of a proper modem for the various audio frequencies and frequency shifts is not difficult, and I am sure that most any Amateur using RTTY (radio teletype) would be glad to share his expertise. In fact, a group known as "Handi-Hams" is an international organization that does just that - offer assistance to those with **any** handicap including hearing, vision or physical. (Would you believe a braille teletype?!)

Scott Thompson, KB6CC
Public Information Assistant
The American Radio Relay League, Inc.
4024 W. Monte Vista Ave. Visalia, CA 93277

Free Text Search

A User Tells You All About This Valuable Database

Dow Jones Information Services would like to extend its best wishes for the New Year to all the TRS-80 Microcomputer News readers. Traditionally, this is a time for New Year's resolutions, and we at Dow Jones would like to express to you our ongoing resolve to expand our service to provide quality information which is useful and of interest to our subscribers.

Also a brief New Year's reminder to all of our customers. If you have any questions or comments regarding our service, such as "where can I find information on gold prices; how do I get information on "such and such," please call our Customer Service Representatives. They can help you find the information you want and very often figure out problems for you. The toll free number is:

1-800-257-5114

In New Jersey and Canada, please call:

609-452-1511

Customer Service is available 9:00 a.m.—11:00 p.m. Eastern time Monday through Friday, and 9:00 a.m. to 5:00 p.m. Eastern time on Saturdays. A quick call can often save you time and frustration.

We also received a New Year's greeting from Madge Midway. You might recall her first article from October's issue on being a new subsciber. She sent us a letter this month on our Free Text Search data base. One of the features of the TRS-80 Videotex software that enables our Radio Shack subscribers to use this data base is that the software does not divide words in the middle, thus every line is easily readable, even though Free Text Search is an 80 column wide data base. With that brief introduction, here is Madge.

Dear Dow Jones:

This is Madge Midway again, sending you greetings for the New Year from myself and all of us Midways.

I am writing to you hoping you might publish this letter as you did my last, because I have made a recent discovery on your service that I know would benefit so many fellow subscribers.

For several months I have been using your service, quite satisfied I might add, with the quality, speed of access, and information provided. I have especially enjoyed all the recent data base additions. However, it was not till just recently that I discovered the Free Text Search (FTS) of your News data base.

Of course, I have seen it mentioned in the literature, but I never bothered to call for a guide. It seemed rather an expensive data base, and I did not know how it would help me. My son, who uses Free Text Search for school reports finally convinced me to look into it further. I am so excited by the capabilities FTS offers, I wanted to write a short article that would simplify its use and and intrigue others to try it. The savings of time that this service offers far offsets any price considerations. But first let me describe how the service works.

FTS is a method of searching the Dow Jones Information Services News data base from the present back to June 1979. It contains over 125,000 articles that have appeared in the *The Wall Street Journal, Barron's,* and the Dow Jones News Service. It allows you to be as specific as you want using any combination of words for any specified period of time. I have always been somewhat reluctant to use this search capability; a bit afraid of the necessity to read through a complete guide. But uncomplicated searches can be done quite simply and provide rewarding results.

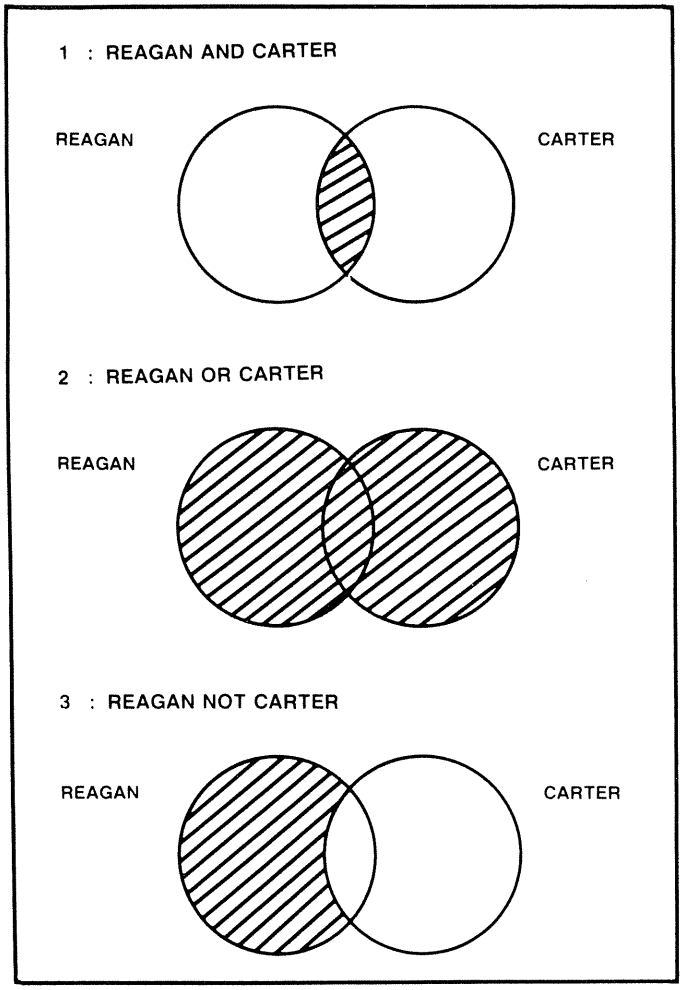
For example, what if you want information on a company that is not publicly traded such as Ciba-Geigy, Mitsubishi etc? Perhaps you are investigating companies for prospective employment, or a company is a potential client, or competitor. The traditional method of search would be to go through back newspaper copies and clip any related stories. Can you imagine how long that would take to cover 2½ years? In minutes through FTS you can have all those stories printed for you (if you have a printer, of course); otherwise, you can read them on your video display.

Another use for FTS is to conduct a subject search. You might want to see all articles relating to Canadian oil exploration, or creative financing in the real estate market. Once again, the amount of time saved using FTS is unbelievable. And let us say you remember there was an article in *The Wall Street Journal* during a certain month this year and you would like to send it to a friend. You can pin point your search to specify time periods as well as subject matter.

By now I would think you can see how useful Free Text Search can be. I would like to take you through the simplest procedure so you can try your own search, and like my experience, call Dow Jones for a Free Text Search manual at no charge, to become more adept in its usage.

Once on line with Dow Jones Information Services, select Free Text Search from the menu. This will connect you to this data base.

There are two basic tools used in Free Text Search - **terms** and **operators.** A term is a piece of information to use



in the search. A name or subject would be a term, for example - Reagan. When you enter Free Text Search you will see the query number 1 displayed on your screen. Enter your term and press (RETURN) and in a few seconds, the computer will display the number of stories containing that term. Often this is a large number, and you need to narrow down the possibilities by placing conditions on the terms, or linking terms together. This is accomplished by an **operator.** The important logical operators are - and, or, not. The examples displayed above show you graphically what each of these operators accomplish.

In query 1, Reagan **and** Carter will yield the number of documents with both Reagan and Carter mentioned in the same article.

Query 2, Reagan **or** Carter will yield documents with **either** Reagan or Carter mentioned.

Query 3, Reagan **not** Carter will yield documents with Reagan and only Reagan mentioned. Articles where both Reagan and Carter are mentioned will not be retrieved.

There are also **positional** operators allowing you to tailor your search still further by the positioning of the terms. There are: same, with, adj (for adjacent to).

Reagan **same** Carter - retrieves documents with Carter and Reagan in the same paragraph.

Reagan **with** Carter retrieves those articles with Reagan and Carter in the same sentence.

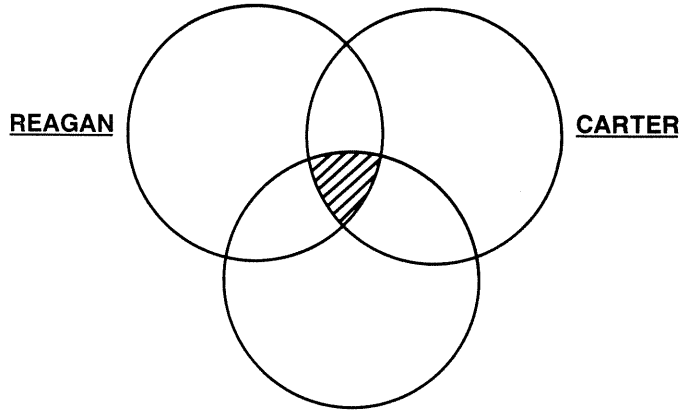
Reagan **adj** Carter retrieves those articles where the term Carter is directly adjacent to Reagan.

In the Search Mode the query numbers can also be used to economize in your search statement. For example, if search statement one is Reagan, statement 2 could be written Reagan and Carter or more simply, 1 and Carter (this becomes more attractive as the statements get longer and more complicated).

Logical and positional operators can be linked as in the case below, ADJ is processed first i.e. Reagan and Carter and (Economics adj policy).

1_: REAGAN AND CARTER AND ECONOMIC ADJ POLICY

RESULT 1



ADJ is the only operator that implies order. A adj B means that A and B are adj and that B follows A.

Now let us assume you have narrowed your search to a workable number of documents. The print statement allows you to view the results of your search, whether it be on a video screen, or through a printer. There are four parts in a print statement:

The print command, the search query number, the document section codes, and the document numbers. A sample print statement might look like this:

```
..PRINT 2 HL,DD,SO/DOC = 1-5
```

PARTS OF THE PRINT STATEMENT	MEANING
..PRINT	Print Command
2	Search Query Number
HL,DD,SO	Document Section Codes
/DOC = 1-5	Document Numbers

The print statement must contain the two periods, (print can be shortened to P) and the search query number must be included. There are a number of section codes - those listed above refer to headlines (HL) display date (DD, date article appeared) and Source (SO). There are several other codes that are listed in the Guidebook. The number after /DOC = refer to those particular articles you wish to see. The above example will display the headlines, date and source of the five most recent documents. After viewing this you might select two documents to view the entire text. Then you enter:

..P 2 ALL/DOC = 2,4

To return to the search mode from Print mode, simply ..SEARCH (or ..S) after the prompt and press RETURN. To sign off from FTS type in

&OFF (RETURN)

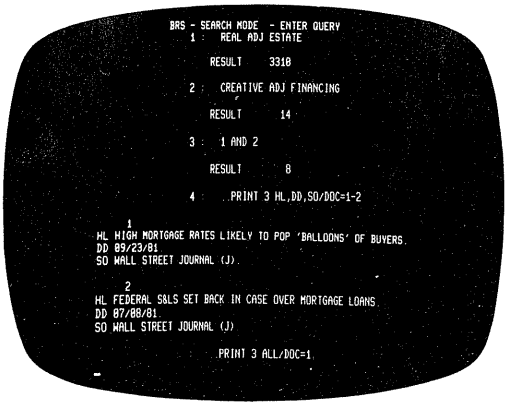
This will bring you back to the standard Dow Jones mode, still on line, to enter other data bases, or type disc (disconnect) to log off the service.

I hope this simple example has succeeded in whetting your appetite to try this data base. You might run through the example below, or call Dow Jones toll free at 1-800-257-5114 and request a Free Text Search Guide at no charge. I did not go into any of the more sophisticated search structures here using dates, numbers, etc. The more precise you become in your search, the less time it takes. Just a few tips:

Before signing on to the system, you should have a clear idea of what path of inquiry you wish to follow. Also, seek accuracy and precision in a document hunt. When searching, do not be alarmed if you input in the wrong format. The computer will give you another chance.

Free Text Search is available Monday 8:30 a.m. to 8:00 p.m. Eastern time and Tuesday through Friday 8:00 a.m. to 8:00 p.m. It is not available on week ends or holidays. The cost during both prime and non prime hours is $1.33/minute.

Try the example below. The search and headline printing took about three minutes, with document printing adding two minutes. Then try your own simple search. Happy searching and Happy New Year.



TRS-80 Disk Head Cleaning Kits

The Shack's Low-Cost Method Keeps Your Drives Running Like New

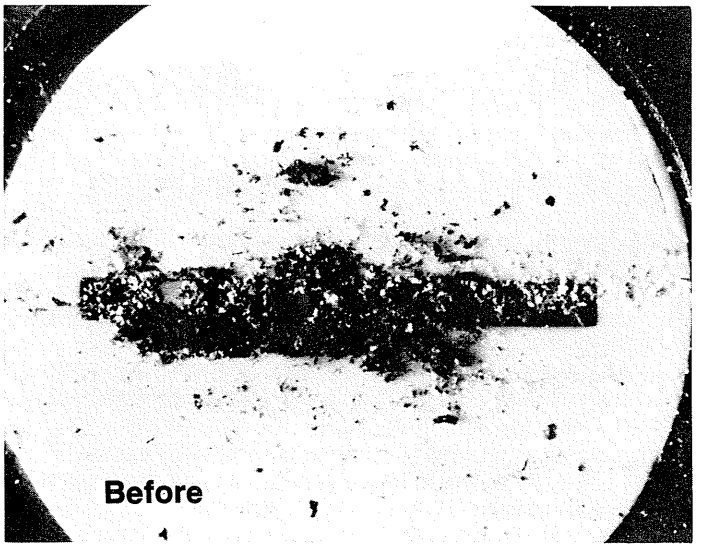
Within the last 18 months, several different brands of diskette drive head cleaning products have been introduced to the users of floppy disk computer systems. Prior to these kits, nothing was available, and not much was mentioned about users cleaning the floppy drive heads. This task was left to service technicians. With the dramatic explosion in the number of floppy systems in use, and the high cost of remedial maintenance, the need for user-oriented preventive maintenance programs became obvious, and the head cleaning kits appeared on the market.

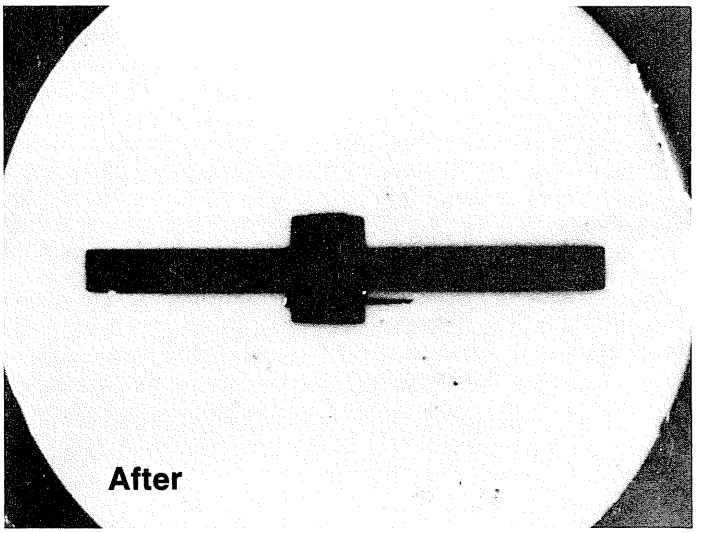
In past years, the older systems used low bit densities. These systems were more forgiving than the high density drives that are the standard for modern floppy systems. In addition to high technology drives, the floppy systems are being used in all types of environments. The opportunity for contamination is much greater, and the rapid increase in the number of drives used increases the chance for failures due to dirty heads. The problem of data errors due to dirty heads has always existed, but with more drives in use, these errors occur more frequently and people are now aware of the overall need to keep the systems clean. One frequent source of head contamination is cigarette smoke, and many computer installations are requesting users to not smoke while in the computer area. To further emphasize the importance of clean drive heads, diskette manufacturers clean the heads on their media certification equipment at least once during each eight hour work shift!

There are several different methods used to clean heads, but most kits use the same basic approach. All are similiar in that one or more cleaning diskettes are supplied in the kits. The material used inside the cleaning diskettes falls into three categories. The most common type of cleaning disk is made of random-woven porous polyester material. The kits also contain a bottle of cleaning solution and a dispenser cap. The procedure for using this type of kit is to saturate a portion of the cleaning disk with the solution and insert the diskette into the drive the same way media diskettes are inserted. As the cleaning disk rotates, the wet/dry action of the disk on the surface of the head cleans and buffs off the contaminants. This is the type system used by the Radio Shack Head Cleaning kits (26-407 for Model I, Model III, and the Color Computer, 26-4909 for Model II). Another type of commercial cleaning diskette uses a dry fabric and no cleaning solution. The third type of cleaning disk is an abrasive material on a polyester disk.

Testing by Radio Shack and Radio Shack disk drive suppliers indicates that the cleaning diskette chosen by Radio Shack is safe and effective for cleaning heads. Further, the specific product supplied by Radio Shack was the **only** product approved by **all** the different suppliers of Radio Shack drives. Testing showed that the wet/dry cleaning method is the most efficient, since the polyester disk closely matches the physical characteristics of the media disk. For ordinary head contamination, using the Radio Shack cleaning disk, 80% of the heads were clean within 20 seconds, and all were clean within 30 seconds. This product is safe for general use, and efficient for user-oriented maintenance.

Here are microphotographs of a disk head before and after being cleaned using Radio Shack's head cleaning kit:





In using the cleaning diskettes, the question of frequency of cleaning often comes up. If the computer systems are in a normal environment, such as an office or computer room, the drives should be cleaned after approximately 40 operational hours. The kit should also be used if there are data errors indicated or any problem reading or writing data. In a more contaminated environment, such as a production area or processing facility where dust, vapor or smoke can reach the drives, the heads should be cleaned more frequently.

Please note that it is important to keep track of how many times you have used a particular cleaning diskette. The material cleaned from the disk head may not be visible, so you should not plan on using the cleaning diskette until it "looks dirty." The thirteen cleanings that Radio Shack recommends for each of its cleaning diskettes ensures that you will not damage the disk head by trying to clean with a dirty cleaning diskette.

Regular use of the cleaning kits will help insure trouble free operation and help to eliminate expensive service calls. Since each kit will provide 26 cleanings, and the cost of a kit is only 29.95, use of the cleaning kits is significantly less than a service call.

Head cleaning is extremely important for today's systems, and perhaps will be mandatory for successful performance of future systems. With the wide variation in the media available today, and with the necessity for a near perfect match of head design and media performance, contamination can not be tolerated. The TRS-80 head cleaning kits are a major step in the effort to control head contamination. Regular use allows the computer to operate to maximum performance capability with a minimum in maintenance expense.

Model I/III Disk Head Cleaner Program

We recently received the following from R/V DARK STAR—Marel Sooke, CompuServe ID 70665, 1300:

The following program will spin the disk drives for 30 seconds for use with the disk head cleaner on the Model I TRS-80.

```
1Ø CLS
     : INPUT "REMOVE ALL DISKS FROM DRIVES, THEN
     PRESS ENTER"; A$
2Ø INPUT "HOW MANY DRIVES DO YOU HAVE"; D
3Ø IF D>4 THEN 2Ø
4Ø FOR C=Ø TO D-1
5Ø PRINT "INSERT CLEANER INTO DRIVE"; C;
     : INPUT "THEN PRESS ENTER"; A$
6Ø FOR R=1 TO 39ØØ
     : POKE 143Ø5, Ø
     : NEXT R, C
```

Fort Worth Scene

This month I would like to start by telling you a little about how programs get into the Microcomputer News, and how they get to looking like they do.

When we receive a program listing, we look it over very quickly. The purpose of this first inspection is just to see what the program is and what it is supposed to do. If the program looks reasonable, we have it entered into Scripsit and placed in a file of available programs.

As the time approaches to put out another Microcomputer News, we scan the file of available programs. What we are looking for is something new, different, or something which will fit the general topic of the current issue.

Once we find a likely program, we move it to our working disks. We also run the program at this point to make sure it does what the author said it would do. We occasionally find a program that the author says will work, but we can not get it to execute. We will spend a short amount of time trying to locate any minor problems. If the program still does not work, It gets refiled (probably never to be looked at again.)

If we have a working program, we then begin the process of reformatting the program to fit the Microcomputer News format. This format dictates that each statement in a program line should start on a new line in the News, and each colon will have a space after it. This means that a line like:

```
10 PRINT:PRINT:PRINT:PRINT:PRINT
```

will come out in the News as:

```
10 PRINT
   : PRINT
   : PRINT
   : PRINT
   : PRINT
```

We reformat the program lines in order to make the contents of each line more readable when you try to enter the program yourself. Not only do we add spaces after colons, we also insert them anywhere in the program listing that we think spaces will improve the readablity of the program. For instance:

```
20 IFI$>="0"ANDI$<="9"ORI$="-"ORI$=
"."THEN160
```

might become:

```
20 IF I$ >= "0" AND I$ <= "9" OR I$="-"
OR I$="." THEN 160
```

Once the program has been reformatted, it should be in a form which is easy to read and reasonably easy to enter into your computer. After we finish all of the reformatting, and before we go to press, the News staff re-runs all programs using the reformatted material as a starting point. This helps us to find program lines which we have accidentally left out or which may have been garbled in the production phase of creating an issue.

Why am I telling you all of this? Simple. The first form of line 20 requires less memory space than the second form, while the second form is more readable. But, if you find that programs are taking up too much memory space, you may wish to remove the extra spaces (the program may run a little faster after you remove the spaces, too) and move all those extra statements up to one line.

Now that you know how a program gets into the Microcomputer News, you should also know who supports them. Because of increased work loads, Computer Customer Services cannot support programs which are published here.

The Microcomputer News staff will try to ensure that all programs published work (at least on the surface) as the authors say they will. However, typographical and other errors do occur and you should test any program before you begin using it. One problem we face, in testing programs, is that some of the topics which we present are new to us, so we may not have the expertise to detect a program problem or error. If you discover a problem in one of our published programs, please write and let us know so we can let other readers know.

If you read fine print, you will notice that our disclaimer/copyright/anything else column at the front of the News grew a little bit this month. We are now suggesting that you contact the author of a particular program when you need help. To help you do this, we are printing the addresses of the authors of most articles. The author should be your best source of information. If you can not contact the author, or if you are not happy with the information you received, let us know. The Microcomputer News staff will, within the limits of our size and time, try to find answers for any questions we receive. In many cases we will publish the answers to these questions either in separate articles, or in a section called "Notes on Previous Issues".

Happy Computing.

Variable Length Segments

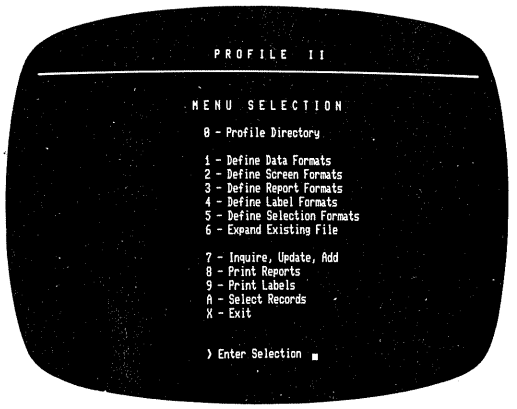
The small Computer Company
P.O. Box 2910, Ft. Worth, TX 76113-2910

Getting a program product into the production stage requires "freezing" its development at some point; new ideas and improvements must be curtailed to allow documentation and the final testing to proceed.

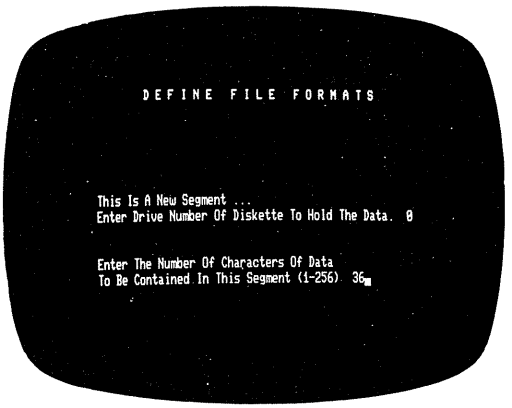
One of the ideas that almost made it into the production version of PROFILE II was variable lengths for segments two, three and four. Actually, this valuable feature was included in the PROFILE II programs, but because it arrived too late to be documented and tested, it was "turned off." Those of you who are regular readers of the TRS-80 News may already be familiar with the following patch which reenables variable length support:

```
PATCH CREATE/EFC A=51C7, F=C3ED51, C=ØØØØØØ
```

Once this patch has been installed, when you choose PROFILE II choice 1 (Define File Formats), you'll be asked for



the lengths of segments 2-4. This question will only be asked the first time each segment is defined. By defining each segment to be exactly the length required, you can achieve the maximum capacity for your system.



The PROFILE II manual states that for multi-segment files the maximum number of records possible is approximately 1800. This is predicated on the normal segment 2-4 length of 256 characters and a totally empty diskette for each additional segment. However, the maximum file size can be greatly increased by using segment 2-4 sizes of less than 256 characters. Let's try a few examples to see how this works.

Suppose a file will require 250 characters of data on a Model II system that has a 1-drive expansion unit. Normally PROFILE II would require the KEY segment to contain 85 characters and segment 2 to contain 256, making a total of 341 characters. This wastes 91 characters per record. If the above patch were applied, the second segment could be defined as being only 165 characters in length.

In general, the largest segment should be placed on the diskette with the most free space. In this example, that means placing the key segment (85 characters) on the drive 0 diskette and segment 2 on the drive 1 diskette. Now, an empty data diskette has room for about 480,000 characters of information. Dividing this figure by our record length of 165 characters, we find there is room for 2909 records on drive 1—a substantial improvement over the 1800 records possible with a fixed-length segment 2. There are other cases in which a normal PROFILE II won't make the best use of available disk space. An example of this would be a 2-segment file on a system with three available disk drives. In this case, to obtain the maximum amount of room, the variable length feature allows you to define the file in 3 segments instead, placing each segment on its own diskette.

Consider a file which requires 341 characters for data. Normally this would be a two-segment file (85-character first segment and 256-character second segment), limiting you to a maximum of 1800 records. A better solution would be to split the second segment into two 128-character segments, and place each on its own diskette. This allows you to double the number of records contained in this file.

It is not even necessary to place each segment on its own diskette at the time you define the file. All segments could be allocated to the drive 0 diskette and moved later as the files grow. To do this you will need to be aware of the file names used by PROFILE II. These are:

Segment 1 (key segment)	filename/KEY
Segment 2	filename/DAT
Segment 3	filename/DA2
Segment 4	filename/DA3

where filename is the name you called the file, padded with zeros to a length of eight characters. Assuming that the file is called SALES, the following statements can be issued from TRSDOS READY to move the second segment from drive 0 to drive 1:

```
MOVE SALESØØØ/DAT:Ø :1
```

and, if this step is successful:

```
KILL SALESØØØ/DAT:Ø
```

Similarly, segment 3 can be moved to drive 2 with:

```
MOVE SALESØØØ/DA2:Ø :2
```

and

```
KILL SALESØØØ/DA2:Ø
```

PROFILE II does not care which drives contain the various files; the system will automatically search all available drives to find the data segments and formats required for each program.

One note of caution: If you are using variable length segments the need for accurate planning is critical, since you must provide segment lengths at the beginning of the Define File process. Once a segment has been defined, PROFILE II will never again ask the segment length question.

A MATHEMATICAL SOLUTION

For those who are a bit mathematically inclined, there is a systematic method for determining the best way to distribute a record over the available drives.

The basic problem involved in determining ideal allocation is this: whichever one of the segments fills its drive first, the space that remains on the other drives is wasted.

Therefore, if we are looking for the ideal usage of the free space, we find that the proportion of the record stored on a particular drive should correspond to the proportion of total system free space available on that disk.

(Here comes the math. If you want, skip below to where we've solved the problem.) Algebraically speaking,

$$\frac{Bn}{TB} = \frac{Gn}{TG}$$

Where:

Bn = The number of characters on drive n
TB = The total number of characters in the record
Gn = The number of grans free on drive n
TG = The total number of grans free

Because the unknown quantity is the number of characters to place on drive n, we will multiply both sides of the equation by TB to solve for Bn. We now have:

$$Bn = \frac{TB \times Gn}{TG}$$

This, in fact, is the formula. (Welcome back to those who skipped the math.) Let's try an example and see how it works. Let's assume that we need 239 characters in our record, that we have three drives available, and that the free space is 100, 300, and 300 grans respectively.

How many characters of drive 0? Well, the total number of characters (TB) is 239, the number of grans free on drive 0 (Gn) is 100 and the total number of free grans (TG) is 700. Using our formula:

$$Bn = \frac{239 \times 100}{700}$$

$$Bn = 34$$

The number of characters for drive 0 is 34. If we use our equation for drives 1 and 2, we find it solves to 102 characters for each of them. We now have two problems. One is simple: through rounding, we have lost a character. The other problem is the first segment. Remember that while segments 2 through 4 may have variable length, PROFILE's first segment must be 85 characters in length. The three segments that we have come up with are 34, 102 and 102 characters.

The solution? Well we can solve both problems by placing two segments on drive 1. These will be our 85-character first segment, and a segment of 18 characters, making 103 (our lost character regained).

We now have:

Drive 0: segment 2, 34 characters
Drive 1: segment 1, 85 characters;
segment 3, 18 characters
Drive 2: segment 4, 102 characters.

34 + 85 + 18 + 102 = 239

This allocation will permit the maximum number of records given the available free space that we started with.

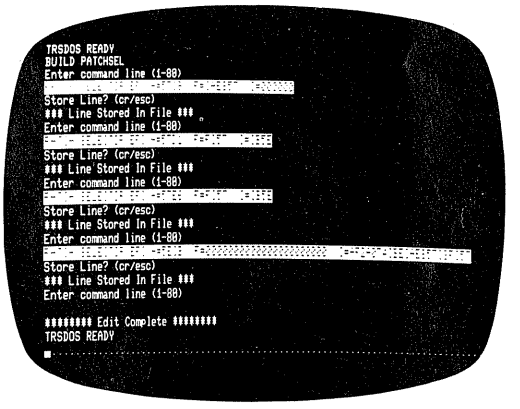
This formula does not work all of the time. Sometimes, all the segments are smaller than 85 characters. It then becomes impossible to break out a key segment from one of them. In this case, we must force the first segment onto the drive with the largest amount of free space and determine the best sizes for additional segments on the remaining drives. (We can use our formula to do this.)

The formula also fails to work when there are four drives and each of the segment lengths is larger than 85. Although it is possible to break out an 85 character segment, this would result in creating a fifth segment. (PROFILE, of course, allows only four segments.) Here, what we must do is force the 85 character segment onto the drive with the smallest amount of space. Again we can use our formula to spread the remaining characters over the remaining drives.

Well, all this sounds like a job that could be done by a BASIC program! If you think you have a program that will use the above formula to distribute a PROFILE record of any length across a given arrangement of free space, send it to us. We will publish the shortest correct answer that we receive (address is at the top of the column).

BUG IN SELECTOR/EFC

Last month we reported a problem in SELECTOR/EFC that causes the program to work improperly when the first record in the file is a blank or deleted record. (SELECTOR/EFC is the program that selects records for SCRIPSIT.) We solved this bug in our standard fashion: we set out a diskette containing the problem and a bowl of cookies. The next morning we found these patches. They solve the problem quite well.




```
PATCH SELECTOR/EFC A=57C9, F=CAE857, C=ØØØØØØ
PATCH SELECTOR/EFC A=57D1, F=FC57, C=CB5E
PATCH SELECTOR/EFC A=57E6, F=FC57, C=CB5E
PATCH SELECTOR/EFC A=5ECB, F=ØØØØØØØØØØØØØØØØØØØØØØØØ,
                           C=AF2AØ74CBECAE857C3FC57
```

(And you wondered what elves did now that shoes are mass produced.)

* * *

On behalf of all of us at The small Computer Company, the editors and writers of the Profile column would like to wish you the very best in this New Year.

Decision Making

Mr. Smith Buys a House and VisiCalc Helps Him Select the Best Financing

The hour had come for Smith to buy a house. Not only had the accountant pointed out certain tax advantages, but Mr. Smith and Mrs. Smith had both come to the conclusion that before long they would be needing a bit of extra room.

Smith consulted the papers. Weekend after weekend, he and Mrs. Smith toured homes in his area —but how could they be compared? Interest rates varied by percentage points. There were various financing options, different down payment possibilities. How could everything be organized for an intelligent decision?

Finally, the Smiths contacted a real estate agent who could help them. She turned to her TRS-80 Model III computer and used the VisiCalc™ program with a VisiCalc template she had prepared especially to help in analyzing difficult decisions like the Smiths'.

In only a few moments and after a few questions, she was able to print out for Smith a detailed mortgage amortization for several different financing situations and price levels. Smith could take that print out to his accountant and discuss alternatives in a reasonable fashion.

Here is what the real estate agent did:

She had organized a VisiCalc worksheet (called a Template) so that she could ask her clients the questions which would determine the values of the analysis positioned below on the sheet.

For each home to be analyzed, she needed to know:

- Asking price
- Down payment requirement (percent)
- Interest rate
- Length of time for the mortgage (in years)

To help her clients make informed decisions, in the template she also covered closing costs, annual homeowner's tax rates and any other fees which were pertinent. Knowing her clients' income, she could also calculate the percentage of gross income represented by the monthly mortgage payment.

SETTING UP THE TEMPLATE

The formula for calculating the principal + interest payment of a mortgage is:

$$P + I = \frac{\text{Amount of Loan} * (i / 100)}{1 - ((1 + i /100)^{-n})}$$

where i = % interest per MONTH
n = number of payments

This formula is the crux of the entire analysis. For example:

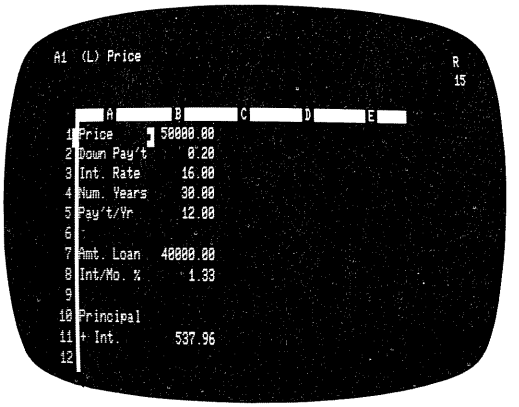
Asking Price	$50,000	
Down Payment Req.	20 %	
Interest Rate	16 %	
Length of Mortgage	30 years	{359 payments}
Payments/Year	12	

From this we can figure:

Amount of Mortgage	$40,000	{Asking price - Amount down}
Interest/Month	1.33 %	{interest rate/ payments per year}

Our VisiCalc template will look like the accompanying chart.

	A	B	C	D
1	Price	50000		
2	Down Pay't	.20	20 percent	
3	Interest rate	16	16 percent	
4	Number of Years	30	30 years	
5	Pay'ts/year	12		
6				
7	Amount of Loan	40000	B1 - (B1 * B2)	
8	Interest/Mo %	1.33	B3 / B5	
9				
10	Principal &			
11	Interest		B7*(B8/100) / (1-((1+(B8/100))^ -B4*B5-1))	
12				



With the above information, we can calculate an amortization schedule:

Below the given information, we'll want to list vertically the payment number. The 254 rows of the VisiCalc program won't let us take the amortization out for the full thirty years, but five years is good enough for our purposes. (Successive series of five years may be positioned side by side if you have sufficient memory. You can also use DIF™ to store the last calculated values of your template on disk, then bring them into a fresh model to calculate the second five years.) Consequently, Column A will list payment numbers 1 through 60.

(A14 would be 1;
A15 would be A14+1.
A16 would be A15+1, etc. Replicate the rest).

To the right of our payment column, we'll need six more columns. Label them CURRENT AMNT, INTEREST, PRINCIPAL, P & I, ENDING BAL, and EQUITY (it's always nice to know where you stand!).

Note: if you're creating this template on a Model II, you'll be able to see all the columns. If your machine is a Model III or Model I, please cursor across the screen with us.

Your column headings should be:

CURRENT AMNT INTEREST PRINCIPAL P & I ENDING BAL EQUITY

Now, for Payment 1, the following references will hold:

Current Amount: B7
{which = 40000, or the amount of the loan after we've removed the down payment}

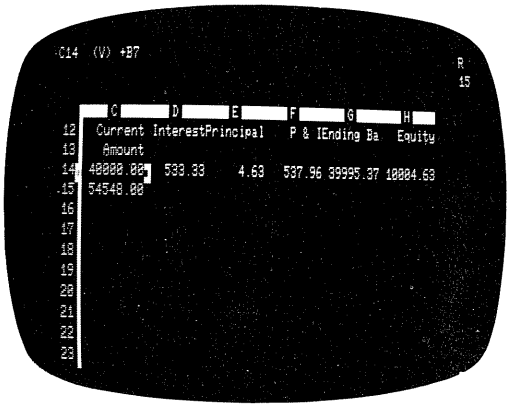
Interest: (Amount Financed)*(B8/100)
$$\{B8 = \frac{\text{(interest rate)}}{\text{(\# pay'ts per year)}}\}$$

Principal: B11-Interest
{B11 is principal + interest. obviously, subtracting interest (calculated just above) will leave us with principal . . . }

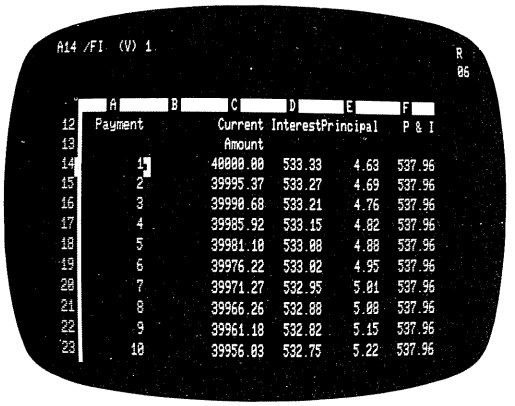
Principal & Interest: Principal + Interest
{This is a check to make sure our calculations stay constant through the life of the loan}

Ending Balance: Current Amt - Principal

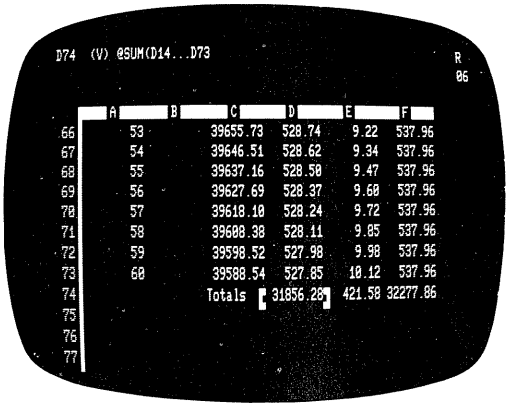
Equity: Initially, down payment + principal



For Payment 2, the CURRENT AMOUNT is simply the ENDING BALANCE brought forward from the row above; and because down payment has already been counted in the EQUITY, you may simply add Payment 2's principal to the previous payment's EQUITY to find the Payment 2 EQUITY amount.



Replicate the above formulas down the columns to the extent of your template (or memory). At the bottom of the columns, perform an @SUM on INTEREST, PRINCIPAL and P & I.



Hint: Because the majority of your calculations are first across rows and secondarily down columns, you should reset your VisiCalc template to ROW calculation by typing:
/GOR (Global Order of calculation by Row).

ADDING THE FINAL TOUCHES

While our VisiCalc real estate template is essentially complete now, taking closing costs, "points" and other fees into account will allow us to arrive at a more reasonable estimate of funds necessary for closing a deal. (These costs won't affect the amortization schedule, but can be invaluable in deciding whether a client can actually afford to close on a deal otherwise affordable.)

Knowing the client's income will allow a calculation to determine what percentage the monthly payment is of the client's gross income. Knowing an approximate tax bracket will allow the calculation of percentage of the monthly payment of take-home income. That tax bracket figure is also useful in approximating the tax consequences of a purchase too.

Before detailing your VisiCalc template to such an extent — or making ANY business decision using the output of a program you have entered into a computer — it is certainly advisable to contact an accountant to be sure of your facts and figures. No program can take the place of your good judgement.

We'd Like to Know . . .
How You're Using the VisiCalc Program

If you have an interesting application for the VisiCalc program and would like to share it with the other readers of the Microcomputer News, please send it on to:

VisiCalc Page Editor
c/o Microcomputer News
P.O. Box 2910
Fort Worth, TX 76113-2910

(Sorry, we can't answer individual requests for programming help.) Technical questions should be directed to:

Computer Customer Services
400 Atrium - One Tandy Center
Fort Worth, Texas 76102

or call: 1-800-433-1679 (Outside Texas)
1-800-772-5914 (In Texas)

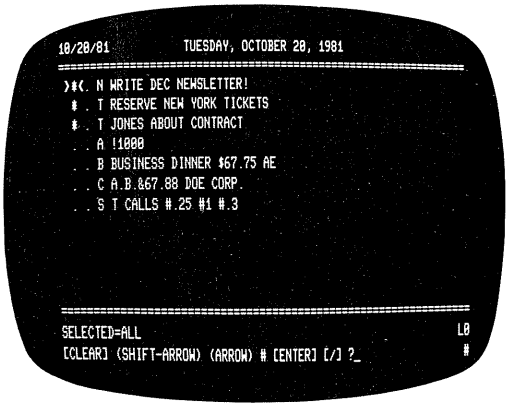
Time Manager™, Revisited

Last month I introduced you to Time Manager, a personal productivity aid that will help you with everthing from remembering to buy milk and bread on your way home from work, to the scheduling of your workday and keeping track of appointments and "to do's." I am so enthusiastic about this time saver that I would write about it for the rest of '82 but we have some other products and news to cover in future articles.

We talked last month about scanning past entries by keyword and category or examining future commitments in the same fashion. In this way project history, records of customer contacts or other information can be "filed" and instantly accessed before a meeting or phone call. We also talked about notepads for storing phone numbers, vendor or account codes, history of phone conversations, command syntax for using Time Manager, cross-reference lists, and more. And we talked about Time Manager's ability to instantly sort and schedule the key activities of your workday.

Let's look at one of the most powerful aspects of Time Manager—its account totalling feature. Use it for job costing, tracking purchases, expense report preparation, time studies, client billing, or even employee performance review. You will find many additional applications for Time Manager and we would like to hear about them.

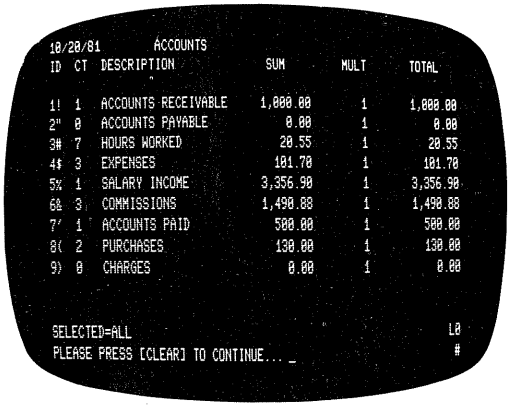
As daily "entries" are made (for example for expenses or purchases), one of several account symbols is added. Here is a typical day level screen using account totaling:



The first item flagged for selective totals is an accounts receivable entry for $1000, the second an expense transaction for an American Express charge, the third commissions for $67.88 and the last several phone calls in portions of an hour.

So now you finally get around to doing an expense report. Hmm? Where are the receipts, did you record all of your expenses?

Time Manager only needs the date of your last expense submitted, so let's put it to work. Ask for a global total for all dates since that last expense account submittal:



It took less than two seconds and a check sum is ready for you, in this case $101.70. If you wish to insure that all receipts are accounted for, it's easy enough to clear the totals and then combine global totaling with keywords like AE for American Express, or Hotel, etc. to insure that all receipts are gathered for appropriate expense categories. For example, a grand total be shown for all American Express charges for the time period that has elapsed since your last expense report.

Time Manager's scan feature would also allow you to scan forward and display each 'AE' entry in detail by day including the particular customer name and the type of expense like lunch, dinner, etc. This would provide the date of the charge and needed information for detailing your expense report plus provide a RUNNING total as you scanned the dates in question. I picked a expense report example, but it could just as well have been a scan by job number for which you wish to determine total hours of labor expended, or a product whose material costs need to be determined and so on . . . . . .

Let's go back and look at the account total screen. The ID column is the set of symbols used by Time Manager to provide selective totals for the accounts defined in column 3. These accounts are supplied with Time Manager but you will learn how to change them to fit your needs. The column CT displays the number of times an amount has been added into the account. The Total column differs from the Sum in that it represents the product of the multiplier and the Sum. An example of the use of the multiplier might be to affect various billing rates for services where Time Manager is used for time billing.

I'll have the opportunity to tell you more about Time Manager in other articles. That's because his cousins—Personnel Manager™ and Project Manager™ can utilize Time Manager for updating other important records. For example by entering a salary review in Time Manager and the raise amount on the day of the review you can update the individual's personnel records in Personnel Manager. Or, by marking a project stage in Time Manager as complete, you can update the project being tracked with Project Manager.

Model I/III Bugs, Errors and Fixes

26-1506 CASSETTE PORTFOLIO

In all versions of Cassette Portfolio SELL transactions may not exactly match the original purchase if fractional units are involved. This results in the lot being left on file (i.e. units left after selling all units may not exactly equal zero).

The problem is corrected by following these steps:

1. In BASIC, load the program by typing:

```
CLOAD "TTRAN" <ENTER>
```

2. Make the following changes to the ends of lines 123 and 126:

Old Line: `123 ...IFD!>S(J,Ø)THEN116`

New Line: `123 ...IFD!>S(J,Ø)+.ØØ5THEN116`

Old Line: `126 ...IFS(J,Ø)=ØTHENGOSUB132`

New Line: `126 ...IFS(J,Ø)<.ØØ5THENGOSUB132`

3. Save the changes in the program by typing:

```
CSAVE "TTRAN" <ENTER>
```

with the recorder set to record, and a fresh tape.

4. We would suggest making a second (Backup) copy of the corrected program.

26-1552 GENERAL LEDGER

We have found two problems with version 1.1 of General Ledger for both Models I and III. First, the program hangs up when you attempt to post transactions. Second, General Ledger Account Input does not stop after 100 accounts.

These problems are corrected by following the steps listed below:

1. BACKUP the diskette(s) and make the changes on the BACKUP copy of the program.
2. The following corrections will correct the first problem.
3. In BASIC, load the program by typing:

```
LOAD "GLTXPOST" <ENTER>
```

4. Make these changes by retyping the appropriate lines, or by using the EDIT command (refer to the Edit section of the owners manual):

Old Line:
```
285 CLOSE2:OPEN"R",3,"DETAIL:1"
    :OPEN"R",2,"GLJOUR:Ø":CLOSE2
```

New Line:
```
285 CLOSE2:OPEN"R",3,"DETAIL:1"
    :OPEN"R",2,"GLJOUR:Ø":ME=LOF(2)*6
```

Old Line: `3Ø1 ME=1:ONERRORGOTO3Ø4`

New Line:
```
3Ø1 IFME>2ØØTHEN3Ø5ELSENE=1:
ME=Ø:ONERRORGOTO3Ø4
```

Old Line: `3Ø2 PUT2,ME+1Ø:GOTO3Ø2`

New Line:
```
3Ø2 PUT2,NE:NE=NE+1:ME=ME+6:IFNE<1ØØ
THEN3Ø2ELSE3Ø5
```

Old Line:
```
3Ø4 IFERR<>122THENRESUME:ELSEME=
    (ME-3Ø)*6:RESUME3Ø5
```

New Line: `3Ø4 IFERR<>122THEN 99ØØELSERESUME3Ø5`

Old Line: `99ØØ RESUME`

New Line:
```
99ØØ ER=ER+1:IFER<5THENRESUMEELSE
PRINT@965,"ERROR"ERR/2+1"IN LINE"
ERL"-PRESS <ENTER> TO RESTART"
:FL=1:GOSUB1ØØ:PRINT@96Ø,
CHR$(31);:GOTO1Ø
```

DELETE Line 290

5. Save these changes by typing:

```
SAVE "GLTXPOST" <ENTER>
```

6. The following corrections will correct the second problem.
7. In BASIC load the program by typing:

```
LOAD "GLMAINT" <ENTER>
```

9. Make this change by retyping the line, or by using the EDIT command (refer to the Edit section of the owners manual):

Old Line:
```
47Ø MID$(GL$(IP),1,63)=AC$:K1=K1+1
    :GOTO41Ø
```

New Line:
```
47Ø MID$(GL$(IP),1,63)=AC$:K1=K1+1
    :IFK1>99THEN3ØØELSE41Ø
```

10. Save this change by typing:

```
SAVE "GLMAINT" <ENTER>
```

11. At TRSDOS Ready, make a BACKUP copy of the corrected diskette.

26-1555 ACCOUNTS RECEIVABLE

In all versions of Model I/III Accounts Receivable, when a correction entry is made to a payment, the correction replaces the previous payment on the statement.

This problem can be corrected by following the steps listed below:

1. BACKUP the diskette(s) and make the changes on the BACKUP copy of the program.
2. In BASIC, load the program by typing:

```
LOAD "ARS" <ENTER>
```

3. Make the following corrections:

(*NOTE: On versions prior to 3.0, this line is line 1650*)

Old Line:
```
2Ø2Ø PR#=PR#-VE#:CB#=CB#+VE#:CV#=CV#+VE#
    :G#(VJ)=G#(VJ)-VE#:G#(2)=G#(2)-VE#
```

New Line:
```
2Ø2Ø PR#=PR#-VE#:CB#=CB#-VE#
    :CV#=CV#-VE#:G#(VJ)=G#(VJ)-VE#
    :G#(2)=G#(2)-VE#
```

4. Type: `SAVE "ARS" <ENTER>` to save the changes in the program.
5. At TRSDOS Ready, make a BACKUP copy of the corrected program.

26-1604 VERSAFILE

Important Note on Using Key Words:

When you select a key word that will be used in setting up your file, you must use the word exactly as it is referred to in line 10000; This is, you must observe the requirement that a space must come both before and after the key word when you use it in the program.

The reason for this requirement is that "VERSAFILE" does not distinguish between characters that mark the beginning or end of a unit of data (the exception being periods "." and question marks "?".)

Example:

The following statement will be placed in the " IS " file:

THIS STATEMENT IS LEGAL.

This statement will NOT be placed in the " IS " file:

THIS IS, HOWEVER, NOT GOOD.

Versafile does not acknowledge " IS, " as being a keyword and will place the sentence in the general file. Versafile does, however, recognize " IS " (without the comma) and the sentence in the first example will be placed in the " IS " file.

We would also like to point out that Versafile recognizes the difference between upper and lower case characters. Because of this, if your key words are in upper case, like "IS," and you enter your text in lower case, all sentences will be placed in the general file (e.g. " is " is not the same as " IS ".)

Revisions, Enhancements, Etc.

The following is a list of current stock numbers for Computer Software revisions, enhancements and some miscellaneous items. These items can be ordered through your local Radio Shack store. If no price is shown, the item is available at no cost upon proof of purchase of the original program.

MODEL I

700-2007 LPC Driver N/C
With some Radio Shack printers and software the "top of form" is increased by one line on each page. The LPC driver corrects this problem.

700-2009 Profile (26-1562 ver. 3.2) N/C
Corrects printing problems that existed in earlier versions of Profile.

700-2220 VisiCalc (26-1566) DIF Documentation N/C
Programmer's Guide to the Data Interchange Format. Not included with any Model I VisiCalc packages.

70-2222 Payroll Disks & Manual (26-1556) $25.00
Includes the 1982 tax tables and a program change to allow the W-2 program to be user formattable. (There are no W-2 changes for 1981.)

700-2223 Manufacturing Inven. Control & Manual N/C
Ver. 3.0 corrects sort error when there are over 300 Raw Materials in a finished good. Disk & Manual

700-3004 TRSDOS 2.3 Manual (26-2104) $ 5.00
This manual incorporates all the changes that have been made to the previous TRSDOS 2.3 or 2.1 manual

700-5210 FORTRAN Sample Source Files for 26-2201 N/C
Diskette containing the Sample Source Files for 26-2201

700-6001 Mailing List Exchange $ 60.00
Upgrades Disk Mailing List (26-1551) to Business Mailing List (26-1558).

MODEL III

700-2006 Disk Scripsit (2-1563 Ver.3.2) on TRSDOS 1.3 N/C

700-2008 Versafile (26-1604) on TRSDOS 1.3 N/C

700-2010 Profile (26-1562 ver. 3.2) on TRSDOS 1.3 N/C

700-2012 Microfiles (26-1565) on TRSDOS 1.3 N/C

700-2212 Business Mailing List (2 Up Labels) N/C
Provides two across labels for 26-1558.

700-2214 General Ledger Disks & Addenda (26-1552 ver. 3.0) on TRSDOS 1.3 N/C

700-2219 Standard & Poor's Disks & Manual (26-1507) on TRSDOS 1.3 N/C

700-2220 VisiCalc DIF Documentation N/C
Programmer's Guide to the Data Interchange Format. Not included with Model III VisiCalc packages prior to enhanced version of VisiCalc (26-1569)

700-2232 TRSDOS 1.3 Disk & Manual N/C
For owners of Model III Disk Systems who never received the final Disk System Owner's Manual and TRSDOS 1.3 diskette.

700-3210 VisiCalc Replacement Disk (26-1566) $25.00
For owners who have destroyed or have unusable VisiCalc program diskettes.

700-3211 Scripsit Replacement Disk (26-1563) $25.00
For owners who have destroyed or have unusable Scripsit program diskettes.

700-6200 VisiCalc Exchange Disk for Enhanced Ver. of Program $99.05
Owner must provide exchange card included in the VisiCalc manual and pay the difference between the earlier program and the enhanced Mod III VisiCalc ($99.05) to obtain the enhanced version.

RSBASIC Lister Program

William J. Schauert
245 Maple Lane
Munster, IN 46321

I recently purchased the Compiler BASIC system for my Model III microcomputer. Although I have not had the opportunity to use all of the many features of this program, I find it be a truly "big system" compiler. I am sure I will be writing most of my programs in RSBASIC from now on.

My purpose in writing is that I have discovered a problem in the LIST utility included on the Model III diskette. When one tries to list compiled code using this program any items after a comma on a program line are omitted. The cause of the problem appears to be the fact that when Compiler BASIC reads the input line, the comma serves as a delimiter and marks the end of the input string. Line 130 of the original program seems to be the offender. I have included a listing showing my solution to the problem. By making the disk input a LINE INPUT, the program will read all characters including those after a comma.

I have modified the LIST/BAS program to put the page number, as well as, a title line which included the listing time and date. I have included a listing of this program also. I would like to share it with others, so they may spruce up their listings also.

To use the new LIST/BAS one must first use the TRSDOS FORMS command to set the number of lines per page to 66. This will allow the lister program to control when a new page will be printed. I like to print my listings in the 80 characters per line mode (I can read bigger print better), so this version will automatically increment the line counter by an additional line if the program line is more than 80 characters (my printer will wrap around at column 80). If one expects to have more than 160 characters per line, an additional test should be made at line 330 for lines greater the 160 characters. The program will also print the listing to the CRT display.

No test for printer ready is used so the user must have the printer ready or the program will wait for the printer to become ready.

Thank you for the opportunity to express my opinion, and keep up the good work!

```
RSBASIC compiler lister, verison 1.Ø wjps                page 1
LIST/LST:Ø                  date: 1Ø/25/81           time: ØØ:13:45

ØØØØ     1Ø    'This program is for printing the disk listing files
ØØØØ     2Ø    'generated by the BASIC compiler. These listing files
ØØØØ     3Ø    'use a record format called "variable length" records.
ØØØØ     4Ø    'The format is "length byte -- data". An example of
ØØØØ     5Ø    'the string "ABCD" is : .ABCD
ØØØØ     6Ø    '                       Ø4444
ØØØØ     7Ø    '                       51234
ØØØØ     8Ø    'Notice that the length byte is "5" which includes itself
ØØØØ     9Ø    'and there is no <CR> following the string.
ØØØØ     1ØØ   '
ØØØØ     11Ø   INPUT LENGTH=32,PROMPT="FILE ? ";A$
ØØ15     12Ø   OPEN #1,A$,MODE=R,TYPE=S
ØØ2B     13Ø   LINE INPUT #1 ;B$ : IF EOF(#1) <> Ø THEN 15Ø
ØØ43     14Ø   PRINT B$ : LPRINT B$ : GOTO 13Ø
ØØ5A     15Ø   CLOSE : END
SYMBOLIC MEMORY MAP
SCALARS
ØØ81    A       STRING*255    ØØ84    B    STRING*255
CROSS REFERENCE LISTING
SCALARS
A               11Ø    12Ø
B               13Ø    14Ø    14Ø
FINAL SUMMARY
  157 (ØØ9D)  BYTES OF PROGRAM
  512 (Ø2ØØ)  BYTES OF LOCAL DATA
   15 SOURCE LINES
   19 SOURCE STATEMENTS
*** COMPILATION COMPLETE ***

RSBASIC compiler lister, verison 1.Ø wjps                page   1
LIST/LST:1                  date: 1Ø/25/81           time: ØØ:Ø2:55


ØØØØ     ØØ1ØØ 'This program is for printing the disk listing files
ØØØØ     ØØ11Ø 'generated by the RSBASIC compiler. It expects the
ØØØØ     ØØ12Ø 'file to be in the variable length format.
ØØØØ     ØØ13Ø '        William J. Schauert 24-OCT-81
ØØØØ     ØØ14Ø '
ØØØØ     ØØ15Ø '
ØØØØ     ØØ16Ø 'set up program equates
ØØØØ     ØØ17Ø LNEMAX=51
ØØØB     ØØ18Ø PAGENO=Ø
ØØ11     ØØ19Ø LNECNT=Ø
ØØ17     ØØ2ØØ 'page header string
ØØ17     ØØ21Ø HDRLP$="RSBASIC compiler lister, version 1.Ø wjps
         page ###"
ØØ1C     ØØ22Ø TTLLP$="######################      date: ########     time
         : ########"
ØØ21     ØØ222 HDRCRT$="RSBASIC lister, version 1.Ø wjps        page ###"
ØØ26     ØØ224 TTLCRT$="################  date: ########  time: ########"
ØØ2B     ØØ23Ø '
ØØ2B     ØØ24Ø PRINT CHR$(28); CHR$(31)
ØØ47     ØØ25Ø PRINT"RSBASIC compiler lister, version 1.Ø wjps"
ØØ51     ØØ26Ø PRINT
ØØ59     ØØ27Ø INPUT LENGTH=21,PROMPT="FILE SPEC: ";FSPEC$
ØØ69     ØØ28Ø OPEN #1,FSPEC$,MODE=R,TYPE=S
ØØ7F     ØØ29Ø GOSUB 1ØØØ
ØØ82     ØØ3ØØ LINE INPUT #1; PRGLNE$
ØØ8C     ØØ31Ø IF EOF(#1)<>Ø THEN 36Ø
ØØ9A     ØØ32Ø PRINT PRGLNE$ : LPRINT PRGLNE$
ØØAE     ØØ33Ø IF LEN(PRGLNE$)>8Ø THEN LNECNT=LNECNT+2 ELSE LNECNT=LNECNT+1
ØØCB     ØØ34Ø IF LNECNT=>LNEMAX THEN LNECNT=Ø:LPRINT CHR$(12): GOTO 29Ø
ØØED     ØØ35Ø GOTO 3ØØ
ØØFØ     ØØ36Ø CLOSE
ØØF4     ØØ37Ø STOP
ØØF6     Ø1ØØØ 'print page header
ØØF6     Ø1Ø1Ø PAGENO=PAGENO+1
ØØFE     Ø1Ø2Ø PRINT STRING$(63,"-")
Ø113     Ø1Ø3Ø LPRINT : LPRINT
Ø123     Ø1Ø4Ø PRINT USING HDRCRT$,PAGENO
Ø13Ø     Ø1Ø5Ø LPRINT USING HDRLP$,PAGENO
Ø13D     Ø1Ø6Ø PRINT USING TTLCRT$,FSPEC$;DATE$; TIME$
Ø158     Ø1Ø7Ø LPRINT USING TTLLP$,FSPEC$;DATE$;TIME$
Ø173     Ø1Ø8Ø PRINT : PRINT
Ø183     Ø1Ø9Ø LPRINT : LPRINT
Ø193     Ø11ØØ RETURN
SYMBOLIC MEMORY MAP
SCALARS
Ø3Ø9  FSPEC    STRING*255  Ø2F4  HDRCRT  STRING*255
Ø2E8  HDRLP    STRING*255  Ø33D  LNECNT  REAL
Ø32D  LNEMAX   REAL        Ø335  PAGENO  REAL
Ø3ØC  PRGLNE   STRING*255  Ø2FA  TTLCRT  STRING*255
Ø2EE  TTLLP    STRING*255
CROSS REFERENCE LISTING


RSBASIC compiler lister, verison 1.Ø wjps                page   2
LIST/LST:1                  date: 1Ø/25/81           time: ØØ:Ø3:47

SCALARS
FSPEC                27Ø    28Ø    1Ø6Ø    1Ø7Ø
HDRCRT               222    1Ø4Ø
HDRLP                21Ø    1Ø5Ø
LNECNT               19Ø    33Ø    33Ø     33Ø     33Ø     34Ø     34Ø
                     17ØX   34Ø
PAGENO               18Ø    1Ø1Ø   1Ø1Ø    1Ø4Ø    1Ø5Ø
PRGLNE               3ØØ    32Ø    32Ø     33Ø
TTLCRT               224    1Ø6Ø
TTLLP                22Ø    1Ø7Ø
FINAL SUMMARY
  813 (Ø32D) BYTES OF PROGRAM
 1816 (Ø718) BYTES OF LOCAL DATA
   41 SOURCE LINES
   45 SOURCE STATEMENTS
*** COMPILATION COMPLETE ***
```

MERGEIT/BAS

A Powerful Programming Tool

Douglas B. Hatch
8 Joshua Huddy Drive
Colts Neck, NJ 07722

Enclosed you will find "MERGEIT/BAS." I wrote it in the process of aiding a local RADIO SHACK store update their inventory programs.

PROGRAM DESCRIPTION

MERGEIT/BAS was designed to compare two BASIC programs, creating a file that when merged against the first BASIC program would produce the second. This is a very powerful tool for users of someone else's software. If the programmer sends you an update, running MERGEIT/BAS would produce a list of changes necessary to upgrade the old version to match the updated version. If you have included personal modifications in the old program a listing of the MERGEIT/BAS file would help point out these modifications. Thereafter, the user would delete any lines within the MERGEIT/BAS file that would change these personal modifications. Finally, merging the old program with the file created by MERGEIT/BAS would produce the desired program modifications.

SYSTEM REQUIREMENTS

1. Disk BASIC and one drive.
2. Both programs must be on disk and saved in ASCII (i.e. saved with the ",A" option).
3. Printer (optional).

PROGRAM EXECUTION:

1. Under DOS, set the date.
2. type "BASIC"
3. Hit ⟨ENTER⟩ for "How many files?"
4. Hit ⟨ENTER⟩ for "Memory size?"
5. CLOAD or LOAD "MERGEIT/BAS"
6. Save a copy of "MERGEIT/BAS" on disk if one does not exist there already.
7. Type "RUN"
8. Answer all prompts as desired.
9. Program terminates.

INTERPRETING THE RESULTING MERGEIT/BAS FILE:

Obtain a listing of the MERGEIT/BAS file by typing "Y"es for the "Do you want a listing . . . " or by loading and listing it as a BASIC program. If you had specified "Y"es for "include comments or not (Y/N)," the listing should contain comments at the end of every "changed" or "inserted" line. "Deleted" lines will appear as " ' * deleted by . . . . . -MM/DD/YY".

Lines ending with an "inserted by . . . . . - MM/DD/YY" comment inform you that this line number was newly created

(Continued on page 27)

1981 Microcomputer News Index (4-Page Pull Out)

GENERAL TOPICS

1981 Microcomputer News Index

MODEL I/III

MODEL I

MODEL II

MODEL III

COLOR COMPUTER

POCKET COMPUTER

RADIO SHACK PROGRAMS

CUSTOMER PROGRAMS AND IDEAS MODEL I/III

MERGEIT/BAS (From page 22)

by the updated program. This inserted line may have once been in your old version, but was deleted by you for personal preferences. Check out this possibility, deleting it from the MERGEIT/BAS file if the line is to remain intact in the original file.

If these comments do not appear in the MERGEIT/BAS file there are two known possibilities. First and less obvious, if during file creation, MERGEIT/BAS can not fit its comments on the end of a line, the comment is not included. Second, you might have specified "N"o to the question, "include comments or not (Y/N)".

SAMPLE RUN:

This is an example of a typical MERGEIT/BAS run:

```
THE FILES SPECIFIED HAVE TO BE BASIC PROGRAMS,
   SAVED IN ASCII.

OLD PROGRAM FILESPEC : FILE1
NEW PROGRAM FILESPEC : FILE2
MERGE FILE FILESPEC  : FILE3

INCLUDE COMMENTS OR NOT (Y/N) ?Y

FILE3 CONTAINS  6  LINES TO MERGE AGAINST
FILE1 TO YIELD FILE2

DO YOU WANT A PRINTED LISTING OF FILE3 ? Y
```

MERGEIT/BAS Program Listings:
FILE1:

```
The original program

5 CLS
10 REM THIS IS A TEST
20 INPUT"ENTER YOUR NAME";N$
30 INPUT"ENTER YOU AGE";A
40 IF A<20 PRINT N$;", YOU CERTAINLY LOOK MATURE
   FOR YOUR AGE."
   : GOTO 100
50 IF A>40 PRINT N$;", YOU LOOK MUCH YOUNGER THAN
   YOUR AGE."
   : GOTO 100
60 PRINT N$;", YOU'RE LOOKING FIT AS USUAL."
100 PRINT"THE END"
```

FILE2:

```
The revised program

5 CLS
10 REM THIS IS A TEST
20 INPUT"ENTER YOUR NAME";N$
30 INPUT"ENTER YOUR AGE";A
40 IF A<20 PRINT N$;", YOU CERTAINLY LOOK MATURE
   FOR YOUR AGE."
   : GOTO 70
50 IF A>40 PRINT N$;", YOU LOOK MUCH YOUNGER THAN
   YOUR AGE."
   : GOTO 70
60 PRINT N$;", YOU'RE LOOKING FIT AS USUAL."
70 REM THIS IS A TEST TO SEE IF THE MERGEIT/BAS
   PROGRAM
80 REM WORKS WHEN THIS PROGRAM FILE2 IS MERGED
   WITH FILE1
90 REM
100 PRINT"THE END"
```

Resulting FILE3:

```
40 IF A<20 PRINT N$;", YOU CERTAINLY LOOK MATURE
   FOR YOUR AGE.":GOTO 70' CHANGED BY FILE2 -
   11/18/81
50 IF A>40 PRINT N$;", YOU LOOK MUCH YOUNGER THAN
   YOUR AGE.":GOTO 70' CHANGED BY FILE2 -
   11/18/81
70 REM THIS IS A TEST TO SEE IF THE MERGEIT/BAS
   PROGRAM' INSERTED BY FILE2 - 11/18/81
80 REM WORKS WHEN THIS PROGRAM FILE2 IS MERGED
   WITH FILE1' INSERTED BY FILE2 - 11/18/81
90 REM ' INSERTED BY FILE2 - 11/18/81
100 PRINT"THE END"' INSERTED BY FILE2 - 11/18/81
```

MERGEIT/BAS

```
10 REM MERGEIT/BAS - 6/18/80
20 CLEAR 1000
   : CLS
   : Q$ = CHR$(34)
30 IF LEFT$(TIME$, 2) = "00" THEN PRINT @ 530, "*
     SET DATE UNDER DOS *"
   : PRINT
   : PRINT
   : CMD"S"
40 PRINT
   : PRINT "THE FILES SPECIFIED HAVE TO BE BASIC
     PROGRAMS, SAVED IN ASCII."
   : PRINT
50 LINE INPUT " OLD PROGRAM FILESPEC : "; F1$
60 LINE INPUT " NEW PROGRAM FILESPEC : "; F2$
70 LINE INPUT " MERGE FILE FILESPEC  : "; F3$
80 PRINT
90 PRINT " INCLUDE COMMENTS OR NOT (Y/N) ?";
   : GOSUB 740
100 IF IN$ = "Y" THEN CM = 1 ELSE CM = 0 ' 1 =
     INCLUDE COMMENTS
110 REM * OPEN UP THE DISK FILES SPECIFIED *
120 OPEN "I", 1, F1$
130 OPEN "I", 2, F2$
140 OPEN "O", 3, F3$
150 REM * SET UP COMMENT VARIABLES *
160 IF CM = 0 THEN 210 ' USER DOESN'T WANT
     COMMENTS
170 DT$ = LEFT$(TIME$, 8)
180 I$ = "' INSERTED BY "+ F2$+ " - "+ DT$
190 C$ = "' CHANGED BY "+ F2$+ " - "+ DT$
200 D$ = "' DELETED BY "+ F2$+ " - "+ DT$
210 REM * COMPARE FILES *
220 MC = 0
   :  REM * RESET MERGE COUNT BEFORE BEGINNING *
230 R1 = 1 ' SET TO READ FROM 1 BEFORE BEGINNING
240 R2 = 1 ' SET TO READ FROM 2 BEFORE BEGINNING
250 REM *START OF MERGEIT/BAS FILE CREATION LOOP*
260 IF EOF(1) + EOF(2) = -2 THEN CLOSE
   : GOTO 450
270 IF EOF(1) THEN R1 = 0
   : L1$="65535"
   : ELSE IF R1 = 1 THEN LINE INPUT #1 , L1$
280 IF EOF(2) THEN R2 = 0
   : L2$="65535"
   : ELSE IF R2 = 1 THEN LINE INPUT #2 , L2$
290 A1 = ASC(L1$) ' GET ASCII OF FIRST CHAR IN
     LINE 1
300 A2 = ASC(L2$) ' GET ASCII OF FIRST CHAR IN
     LINE 2
310 IF A1 < ASC("0") OR A1 > ASC("9") THEN GOTO
     770
320 IF A2 < ASC("0") OR A2 > ASC("9") THEN GOTO
     800
330 IF L1$ = L2$ THEN R1 = 1
   : R2 = 1
   : GOTO 250 ' LINES SAME
340 MC = MC + 1
   : REM * INCREMENT MERGE COUNT - LINES DIFFER *
350 N1 = VAL(L1$) ' GET LINE NUMBER FROM FILE 1
360 N2 = VAL(L2$) ' GET LINE NUMBER FROM FILE 2
370 IF N1 = N2 THEN R1 = 1
   : R2 = 1
```

```
   : TX$ = L2$
   : CM$ = C$
380 IF N1 > N2 THEN R1 = 0
   : R2 = 1
   : TX$ = L2$
   : CM$ = I$
390 IF N1 < N2 THEN R1 = 1
   : R2 = 0
   : N1$ = STR$(N1)
   : TX$ = RIGHT$(N1$, LEN(N1$) - 1)
   : CM$ = D$
400 REM * SEE IF COMMENTS ARE WANTED AND CAN FIT
    IN LINE "
410 IF CM = 0 THEN 430 ' USER DOESN'T WANT
    COMMENTS
420 IF 255 - LEN(TX$) > LEN(CM$) THEN TX$ = TX$ +
    CM$
430 PRINT #3, TX$ ' WRITE LINE IN MERGEIT/BAS
    FILE.
440 GOTO 250 ' COMPARE NEXT PAIR OF LINES
450 REM * FILES CLOSED AND PROGRAM HAS CREATED
    MERGE FILE *
460 PRINT
470 IF MC = 0 THEN PRINT "* FILES IDENTICAL *"
   : END
480 IF MC = 1 THEN LN$ = " LINE" ELSE LN$ = "
    LINES"
490 PRINT F3$; " CONTAINS "; MC; LN$; " TO MERGE
    AGAINST"
500 PRINT F1$; " TO YIELD "; F2$
510 PRINT
520 PRINT"DO YOU WANT A PRINTED LISTING OF ";
    F3$; " ? ";
   : GOSUB 740
530 ON INSTR("YN",IN$)+1 GOTO 520, 540, 650
540 OPEN "I", 3, F3$
550 LPRINT F3$+ " - CREATED BY MERGEIT/BAS - "+
    LEFT$(TIME$,8)
560 LPRINT
570 LPRINT "OLD PROGRAM NAME : "; F1$
580 LPRINT "NEW PROGRAM NAME : "; F2$
590 LPRINT "# OF MERGE LINES : "; MC
600 LPRINT
610 IF EOF(3) THEN CLOSE
   : LPRINT CHR$(12)
   : GOTO 650
620 LINE INPUT# 3, L3$
630 LPRINT L3$
640 GOTO 610
650 PRINT
660 F1$ = Q$+ F1$+ Q$
   : F2$ = Q$+ F2$+ Q$
   : F3$ = Q$+ F3$+ Q$
670 PRINT "LOAD "; F3$
680 PRINT "LIST "; F3$; " , REMOVING UNDESIRED
    DELETES,"
690 PRINT "      OVERWRITES AND INSERTS."
700 PRINT "SAVE "; F3$; ",A"
710 PRINT "LOAD "; F1$
720 PRINT "MERGE "; F3$
730 END
740 IN$ = INKEY$
   : IF IN$ <> "Y" AND IN$ <> "N" THEN 740
750 PRINT IN$
760 RETURN
770 REM * FILE 1 IS NOT BASIC OR NOT SAVED IN
    ASCII *
780 PRINT
790 PRINT F1$;
   : GOTO 830
800 REM * FILE 2 IS NOT BASIC OR NOT SAVED IN
    ASCII *
810 PRINT
820 PRINT F2$;
830 PRINT" IS NOT A BASIC PROGRAM SAVED IN
    ASCII."
840 CLOSE
850 END
```

Cassette Telephone File

Scott Walker
2731 Kelley Chapel Road
Decatur, GA 30034

This program allows you to type in a person's name and telephone number and then save it on cassette.

Type in the program. When you see the READY sign CSAVE the program using a file name that you like. Then all you have to do is type 'RUN'. Type the person's name and telephone number, then press 'ENTER' and the program will then save the person's name and number on tape.

When you want to retrieve the name and the telephone number in the future you must load the program using CLOAD. Then when you get the 'READY' sign, type 'RUN 500' and then the program will be ready to scan the tape for the name that you are presently hunting.

I must let you know, it would be better if you saved the program that I am fixing to show you on a blank tape and save the names and numbers on another blank tape. If you do not and you rewind the tape all the way back to where the program is or where any other program is you will get an error. If you do get this error sign all you have to type is 'RUN 500 until it finds the segment of the tape where the names and numbers are found.

```
10 CLS
15 REM 'THIS IS THE FILE SAVING PROGRAM'
20 INPUT "NAME"; A$
30 INPUT "TELEPHONE #"; B$
40 PRINT#-1, A$, B$
50 PRINT
     : PRINT A$, B$
60 PRINT
     : PRINT
     : PRINT
     : INPUT "PRESS <ENTER> TO CONTINUE SAVING
     FILES, OR TYPE '/' TO END."; C$
70 IF C$="" THEN GOTO 10 ELSE END
80 IF C$="/" THEN END
90 REM 'THIS IS THE END OF THE FILE SAVING
     PROGRAM.'
500 CLS
505 CLEAR 1000
510 REM 'THIS IS THE PROGRAM TO RETRIEVE THE
     FILES."
520 INPUT "WHO'S FILE"; Z$
530 INPUT#-1, A$, B$
540 PRINT
     : PRINT A$, B$
550 IF A$=Z$ THEN END ELSE 530
560 REM 'THIS IS THE END OF THE PROGRAM TO
     RETRIEVE THE FILES."
```

Modification to Sorted Directory Program

Scott Bailey
8237 E. Coolidge Street
Scottsdale, AZ 85251

I am writing to let you know how much I have enjoyed your newsletter and also to comment on Dave McGlumphy's sorted directory program which was printed in your July issue.

The program works well but there are some minor changes I would like to share with you that I feel enhance the program's operation. The worst problem (?) with the program is that it would print blanks ahead of the programs on the final listing. Upon inspection, I found this was due to the fact that the program counted the drive numbers in the data statements as programs and left room in the A$ array for them. Changing the following 3 lines as follows will correct the problem:

```
18Ø Q9=Ø
     : FOR J=1 TO N
     : READ D$
     : PRINT@ 362, J;
22Ø Q9=Q9 + 1
     : A$(Q9)=D$ + DNBR$
23Ø NEXT J
     : N=Q9
```

This will correct the final printout, but the program will not correctly print out the number of programs.

There are times when I like to have listings of my programs sorted by disk number and alphabetically under each disk. The versatility of the sorting program makes this easy; just change line 220 as follows and insert line 255:

```
22Ø Q9=Q9 + 1
     : A$(Q9)=DNBR$ + D$
255 FOR J=1 TO N
     : A$(J)=MID$(A$(J), 3) + LEFT$(A$(J), 2)
     : NEXT J
```

This moves the disk number into the sort position so the programs sorts by the numbers; if they match it moves on and sorts alphabetically. Line 255 moves the disk numbers back to the end of the strings.

To be able to change back and forth, add lines 235 and 237, then change line 250:

```
235 PRINT
     : PRINT
     : PRINT"SORT ALPHABETICALLY OR BY DISK
     NUMBER?";
     : I$=INKEY$
     : I$=""
237 I$=INKEY$
     : IF I$<> "A" AND I$<> "D" THEN 237 ELSE
     UP=3 + 2 * ( I$ = "D" )
     : PRINT I$;
25Ø UL=14
     : UO=Ø
     : UN=N
     : UA=VARPTR (A$(1))
     : GOSUB 35Ø
     : PRINT"DONE."
     : PRINT
```

This moves the sort pointer to either the first character (the drive number) or the third (the file name).

RADIO SHACK has my permission to print any part or all of my letter in their newsletter. I hope that there it will be of some aid to other TRS-80 users. Keep up the good work!

Extending Sequential Files

John Whinery
401 Antelope Street
Scott City, KS 67871

This program is for Model I owners with Disk BASIC who would like to be able to extend a sequential file without reading the complete file into another file, and then change the name of it.

Although some Disk BASICs have this capability built into them, the way they keep track of the "End of File" and "Next Record Number" may not be compatible with RADIO SHACK Disk BASIC.

This program will demonstrate how RADIO SHACK BASIC handles "EOF," "NRN" and "ERN" while a file is open. As you add to the file, it will display the first 14 bytes of the "File Control Block." FCB + 5 and FCB + 8 are EOF bytes, (relative byte in sector). FCB + 12 & 13 is ERN, (relative sector in file). FCB + 10 & 11 is NRN, (relative sector in file). Watch these bytes change as data is entered. To end session, type "END." The complete file will then be printed. Run the program several times.

ERN	(file closed)	0,1,1,1 . . . 1* 1,2,2,2 . . . 2
NRN & ERN	(file open)	0,0,0,0 . . . 0* 1,1,1,1 . . . 1
EOF	(relative byte)	0,1,2,3..255 * 0,1,2,3 . . . 255

```
5ØØ REM EXTEND/BAS   JOHN WHINERY SCOTT CITY, KS
51Ø CLEAR 1ØØØ
52Ø OPEN "R", 1, "DUMMY"
53Ø FIELD 1, 1 AS A$          'DEFINE BUFFER
54Ø V = VARPTR(A$)            'GET VARPTR
55Ø CLOSE
56Ø BUF = PEEK(V+1) + 256 * PEEK(V+2)
     'BUF POINTS TO #1 BUFFER
57Ø FCB = BUF - 32
     'FCB = FILE CONTROL BLOCK
58Ø OPEN "O", 1, "DATA"
59Ø POKE FCB + 5, PEEK(FCB + 8)
     'SET EOF BYTE
6ØØ ERN = PEEK(FCB + 12) + 256 * PEEK(FCB + 13)
     'ERN=END RECORD NUMBER
615 IF PEEK(FCB + 8) > Ø THEN ERN = ERN - 1
     'IF EOF > Ø ERN = ERN - 1
62Ø MSB% = ERN / 256
     : LSB% = ERN - (MSB% * 256)
     'CALCULATE NRN
625 POKE FCB + 1Ø, LSB%
     : POKE FCB + 11, MSB%    'SET NEXT RECORD #
63Ø Y=Ø
     : FOR X=FCB TO FCB+13   'YOU CAN DELETE
64Ø PRINT Y, PEEK (X)        'LINES 63Ø TO
65Ø Y = Y + 1                'END OF PROGRAM
66Ø NEXT                     'AFTER YOU HAVE
67Ø INPUT A$                 'TESTED THIS PROGRAM
68Ø IF A$ = "END" THEN 71Ø
69Ø PRINT #1, A$             'ADD DATA TO FILE
7ØØ GOTO 67Ø                 'UNTIL A$="END"
71Ø CLOSE
1ØØØ CLEAR 1ØØØ              'VERIFIES THAT FILE
1Ø1Ø OPEN "I", 1, "DATA"     'HAS BEEN EXTENDED
1Ø2Ø INPUT# 1, A$
1Ø25 PRINT A$
1Ø3Ø IF EOF(1) THEN 2ØØØ
1Ø5Ø GOTO 1Ø2Ø
2ØØØ CLOSE
```

Model III Disk Inventory —Version Two

Dorian Henao
1521 Ocean Avenue
Brooklyn, NY 11230

I am writing to you once again (using SCRIPSIT, of course) because I found a program that you might want to print in your newsletter.

The program is like the Disk Inventory System by Bud Baker (July, 1981), but allows the user to update, search by the beginning or ending characters of a program name, directory of any disk, and other niceties.

The Inner Workings of the Program

Lines 000-020 - Initialize the system, and prints main menu.
Lines 025-195 - Do various functions:
Lines 025-030 - Ask you for the drive you want to work with.
Lines 035-075 - Prints the directory for the drive you specified, and reads the name of the programs by peeking at the screen. (Sneaky!)
Lines 080-135 - Update various files which allow the computer to know which disk is which. In case it is the first time that you have used the program, the computer creates the needed files. The files are: Number, Dn, File, Fileo.
Lines 140-195 - These are various subroutines which the lines above calls. These includes two different sorts, etc.
Lines 200-225 - Second Menu. Gives the user various choices. Basically, this is the heart of this program, and this is what makes this a different type of Disk Inventory System.
Lines 230-280 - Option One: Search for program by name.
Lines 285-295 - This is a pause before returning to line 205.
Lines 300-330 - Option Two: Search for program by ext.
Lines 335-340 - Option Three: Send a listing of all disks, and their content to the lineprinter.
Lines 345-355 - Option Four: Same as three, but output goes to the screen.
Lines 360-360 - Option Five: All programs to the printer in alphabetical order.
Lines 365-380 - Option Six: Directory of any disk
Lines 390-390 - Option Eight: End

Yes, yes I know you are saying "What happened to option seven?" Well my friends, option seven is return to main menu so it goes to line five.

Lines 395-510 - These are subroutines which are called when you choose option one in the MAIN menu.
Lines 395-415 - Ask you which drive to catalog, and checks for a file which is used internally.
Lines 420-430 - Do a directory of the disk you specified above.
Lines 500-510 - Remind the user that this program is for a system with two or more drives.

So that is a description of the entire program. You might be saying to yourself "Can it not be converted for a single drive system?" the answer is no. I tried. But the only way to keep all the goodies is to use various files, and these would have to go on every disk, which in turn would take up more space.

Keep the following in mind when using this program:
Do you have two or more disk drives?
Do you really have that many programs and can not find them?
Do you have all the time to type in the program and DEBUG IT?
Do ALL of your disks have between five and ten granuals free?

If your answer to all of these was YES then GOOD LUCK! I hope you use it well.

```
1 GOSUB 5ØØ
5 CLS
     : CLEAR 5ØØØ
     : PRINT CHR$(23); @266, " Do you want to
     <1> Catalog a new disk or update an old
     disk.
     <2> Look at a program file or search for a
     program."
9 PRINT@Ø, CHR$(28);
1Ø A$ = INKEY$
     : IF A$ = "" THEN 1Ø
15 A = VAL(A$)
     : IF A < 1 OR A > 2 THEN 1Ø
2Ø ON A GOTO 25, 2ØØ
25 TROFF
     : CLS
     : GOSUB 395
     : CLS
     : PRINT CHR$(23), @512, "Please insert the
     disk to be cataloged in drive #"; DK; " and
     press <ENTER>."
27 PRINT CHR$(28);
3Ø QQ$ = INKEY$
     : IF QQ$ = "" THEN 3Ø
35 DIM A$(5Ø)
     : CLS
     : GOSUB 42Ø
     : N = 15375
4Ø X = PEEK(N)
     : IF X <> 32 AND PEEK(N - 1) = 191 THEN I =
     I + 1
     : C-= Ø
45 IF C > 3Ø THEN 7Ø
5Ø IF N > 16382 THEN 7Ø
55 IF X = 32 THEN POKE N, 191
     : N = N + 1
     : C = C + 1
     : GOTO 4Ø
6Ø POKE X, 191
     : POKE N, X
     : A$(I) = A$(I) + CHR$(X)
     : N = N + 1
     : IF C > 3Ø THEN 7Ø
65 IF N < 16382 THEN 4Ø
7Ø CLS
     : FOR X = Ø TO I
     : IF A$(X) = "P" OR A$(X) = "NUMBER" THEN
     A$(X) = CHR$(191)
     : GOTO 8Ø
75 PRINT A$(X),
     : IF POS(Ø) + LEN (A$(I + 1)) + 9 > 64 THEN
     PRINT
8Ø NEXT
     : ON ERROR GOTO 85
81 OPEN "I", 1, "DN:Ø"
82 INPUT#1, DH
83 CLOSE
     : GOTO 9Ø
85 CLOSE
     : OPEN "O", 1, "DN:Ø"
     : PRINT#1, Ø
     : CLOSE
     : RESUME
9Ø ON ERROR GOTO 1ØØ
     : OPEN "I", 1, DK$
     : INPUT#1, DN
     : CLOSE
     : SK = DN
     : IF DH >= DN THEN 115
95 OPEN "O", 1, "DN:Ø"
     : PRINT#1, Ø
     : CLOSE
     : GOTO 115
1ØØ CLOSE
     : OPEN "I", 1, "DN:Ø"
```

```
     : INPUT#1, DN
     : CLOSE
     : OPEN "O", 1, DK$
     : PRINT#1, DN + 1
     : CLOSE
     : IF DN + 1 < = DH THEN 11Ø
1Ø5 OPEN "O", 1, "DN:Ø"
     : PRINT#1, DN + 1
     : CLOSE
     : DH=DN+1
11Ø RESUME
115 ON ERROR GOTO 14Ø
     : CLOSE
     : PRINT "This is disk #"; DN
     : OPEN "I", 1, "FILE:Ø"
     : INPUT#1, NP
     : DIM N$(2ØØ), D(2ØØ), O(2ØØ)
     : IF NP = Ø THEN 13Ø
12Ø FOR X = 1 TO NP
     : INPUT#1, N$(X)
     : NEXT
     : CLOSE
     : ON ERROR GOTO Ø
     : OPEN "I", 1,"FILEO:Ø"
     : FOR X = 1 TO NP
     : INPUT#1, D(X)
     : IF D(X)=SK THEN N$(X)=CHR$(191)
125 NEXT
13Ø CLOSE
     : ON ERROR GOTO Ø
     : X=NP
     : N=Ø
     : CLS
     : FOR L = X+1 TO X+I+1
     : N$(L)=A$(N)
     : N=N+1
     : D(L)=DN
     : NEXT
     : NP=NP+I+1
     : FOR I=1 TO NP
     : O(I)=I
     : NEXT
     : GOSUB 145
     : FOR X=1 TO NP
     : IF N$(O(X))<>CHR$(191) THEN NP=X
135 NEXT
     : OPEN "O",1,"FILE:Ø"
     : PRINT#1, NP
     : FOR I=1 TO NP
     : PRINT#1, N$(O(I))
     : NEXT
     : CLOSE
     : OPEN "O", 1, "FILEO:Ø"
     : FOR I=1 TO NP
     : PRINT#1, D(O(I))
     : NEXT
     : CLOSE
     : GOTO 5
14Ø CLOSE
     : OPEN "O", 1, "FILE:Ø"
     : PRINT#1, Ø
     : CLOSE
     : RESUME
145 L=I
     : C=Ø
     : FOR N=I TO NP
     : IF N$(O(X))=CHR$(191) THEN C=C+1
15Ø NEXT
     : FOR X=1 TO NP-C-1
     : IF N$(O(X))=CHR$(191) THEN GOSUB 16Ø
155 NEXT
     : GOTO 165
16Ø CLS
     : PRINT CHR$(23), "Pre Sort"
     : A=O(X)
     : FOR X=N TO NP-1
     : O(N)=O(N+1)
     : NEXT
     : O(NP)=A
     : X=X-1
     : RETURN
165 C=Ø
     : FOR X=1 TO NP-1
     : IF N$(O(X)) > N$(O(X+1)) THEN GOSUB 175
     : X=NP-1
     : NEXT
     : GOTO 165
17Ø L=X
     : NEXT
     : TROFF
     : RETURN
175 CLS
     : PRINT CHR$(23); "Sorting Data"
     : FOR I=X+1 TO 1 STEP -1
     : IF N$(O(I)) < N$(O(I-1)) THEN A=O(I)
     : O(I)=O(I-1)
     : O(I-1)=A
     : C=1
18Ø IF C=Ø THEN I=1
     : NEXT
     : GOTO 19Ø
185 C=Ø
     : NEXT
19Ø L=L-1
     : IF L<1 THEN L=1
195 RETURN
2ØØ ON ERROR GOTO 5
     : OPEN "I", 1, "FILE:Ø"
     : INPUT#1, NP
     : DIM N$(NP), D(NP)
     : FOR X=1 TO NP
     : INPUT#1, N$(X)
     : NEXT
     : CLOSE
     : OPEN "I", 1, "FILEO:Ø"
     : FOR X=1 TO NP
     : INPUT#1, D(X)
     : NEXT
     : CLOSE
     : OPEN "I", 1, "DN:Ø"
     : INPUT#1, DN
     : CLOSE
2Ø5 CLS
     : PRINT "Do you want to
     1.  Search for program (N) Char. from the
     beginning.
     2.  Search for program (N) char. from the
     end.
     3.  Print all disks to Line printer.
     4.  Print all disks to the screen"
21Ø PRINT "5.  List all programs to printer in
     alphabetical order with disk number.
     6.  Directory of any disk.
     7.  Restart
     8.  End"
215 QQ$=INKEY$
     : IF QQ$="" THEN 215
22Ø FL=VAL(QQ$)
     : IF FL<1 OR FL>8 THEN 215
225 ON FL GOTO 23Ø, 3ØØ, 335, 345, 36Ø, 365, 5,
     39Ø
23Ø CLS
     : PRINT "You give me any number of
     characters, and I will list all disks that
     contain a program that starts with those
     characters."
     : INPUT "Characters to search by"; A$
     : L=LEN(A$)
     : PRINT "Output to printer (Y/N)"
235 P$=INKEY$
```

```
     : IF P$="" THEN 235
24Ø IF P$="Y" OR P$="y" THEN P=1
     : GOTO 255
245 IF P$="n" OR P$="N" THEN P=Ø
     : GOTO 255
25Ø GOTO 235
255 CLS
     : P$=""
     : FOR X=1 TO NP
     : IF LEFT$(N$(X),L)=A$ THEN 265
26Ø GOTO 28Ø
265 IF P=1 THEN 275
27Ø PRINT N$(X), "Disk #"; D(X)
     : GOTO 28Ø
275 LPRINT N$(X), "Disk #"; D(X)
28Ø NEXT
285 PRINT@98Ø, "Press <ENTER> To Continue";
29Ø QQ$=INKEY$
     : IF QQ$="" THEN 29Ø
295 GOTO 2Ø5
3ØØ CLS
     : PRINT
     : PRINT TAB(5); "You give me any number of
     characters, and I will list all of the
     programs that end in these characters"
     : PRINT "search for "; STRING$(5,95);
     : INPUT A$
     : L=LEN(A$)
     : PRINT "OUTPUT TO PRINTER (Y/N)";
3Ø1 P$=INKEY$
     : IF P$="" THEN 3Ø1
3Ø2 IF P$="Y" OR P$="y" THEN P=1
3Ø3 GOTO 3Ø5
3Ø4 IF P$<>"n" OR P$<>"N" THEN 3Ø1
3Ø5 CLS
     : FOR X=1 TO NP
     : IF RIGHT$(N$(X),L)=A$ THEN 315
31Ø GOTO 33Ø
315 IF P=Ø THEN 325
32Ø LPRINT N$(X), "Disk #"; D(X)
325 PRINT N$(X), "Disk #"; D(X)
33Ø NEXT
     : GOTO 285
335 OPEN "I", 1, "DN"
     : INPUT#1, DN
     : CLOSE 1
     : FOR X=1 TO DN
     : LPRINT
     : LPRINT
     : LPRINT "---------- Disk #"; X; "
     ----------"
     : LPRINT
     : FOR N=1 TO NP
     : IF D(N)=X THEN LPRINT"           "; N$(N)
34Ø NEXT
     : NEXT
     : GOTO 285
345 FOR X=1 TO DN
     : CLS
     : FOR X=1 TO DN
     : PRINT "Disk #"; X
     : PRINT
     : PRINT
     : PRINT
     : FOR N=1 TO NP
     : IF D(N)<>X THEN 355
35Ø PRINT N$(N),
     : IF POS(Ø)>5Ø THEN PRINT
355 NEXT
     : PRINT
     : PRINT
43Ø IF DK=2 THEN CMD"D:2"
     : RETURN ELSE IF DK=3 THEN CMD"D:3"
     : RETURN
5ØØ CLS
     : PRINT CHR$(23)
     : PRINT@24, "NOTICE"
     : PRINT
     : PRINT "This program will not work with a
     single drive system"
5Ø5 PRINT CHR$(28);
51Ø FOR ZZ=1 TO 3ØØØ
     : NEXT
     : RETURN
     : FOR I=1 TO 2ØØØ
     : NEXT
     : NEXT
     : GOTO 2Ø5
36Ø CLS
     : FOR X=1 TO NP
     : LPRINT N$(X), "Disk #"; D(X)
     : PRINT N$(X), "Disk #"; D(X)
     : NEXT
     : GOTO 2Ø5
365 CLS
37Ø INPUT "Directory of what disk"; D
     : IF D>DN THEN PRINT "Disks have only been
     Cataloged up to "; DN
     : GOTO 37Ø
375 CLS
     : PRINT "Disk #"; D
     : PRINT
     : FOR X=1 TO NP
     : IF D(X)=D THEN PRINT N$(X)
38Ø IF POS(Ø)>5Ø THEN PRINT
385 NEXT
     : GOTO 285
39Ø CLOSE 1
     : CLOSE 2
     : CLOSE 3
     : CLS
     : CMD"S"
395 CLS
     : PRINT
     : PRINT CHR$(23); TAB(3) "On which drive
     (1-3) will you place your disk to be
     cataloged?";
     : PRINT@448, CHR$(28);
     : PRINT@192, "";
     : LINE INPUT DK$
4ØØ DK=VAL(DK$)
     : IF DK=<Ø OR DK>3 THEN CLS
     : GOTO 395
41Ø IF DK=1 THEN DK$="NUMBER:1"
     : RETURN
415 IF DK=2 THEN DK$="NUMBER:2"
     : RETURN ELSE IF DK=3 THEN DK$="NUMBER:3"
     : RETURN
42Ø IF DK=Ø THEN CMD"D:Ø"
     : FOR X=1 TO 5Ø
     : NEXT
     : RETURN
425 IF DK=1 THEN CMD"D:1"
     : RETURN
```

S I G I® and the TRS-80 Model II

The Educational Testing Service (ETS) based in Princeton, New Jersey, has developed a computerized system to help college students, or others planning additional education beyond high school, to make informed and rational career decisions. The initial development of their system, named the "System of Interactive Guidance and Information" (or "SIGI"), was funded by the Carnegie Corporation. Through extensive field testing and evaluation funded by the National Science Foundation, SIGI has been demonstrated to be extraordinarily valuable in helping students to make decisions. SIGI is being used on about eighty-five college campuses on minicomputers and mainframes, and about thirty sites already use the system on the TRS-80 Model II.

Since SIGI has recently been converted for use with the Model II through a grant from the Kellogg Foundation, the cost of the hardware for SIGI is within the reach of even more colleges and schools. And Radio Shack offers a fifteen percent discount on hardware used to run SIGI and purchased by SIGI licensees, so the cost of SIGI hardware has been further reduced. The hardware required includes a TRS-80 Model II, a one drive expansion unit, and a printer.

Field testing for SIGI has been taking place at six colleges and one high school. The staff at the field-test sites are available to talk to prospective users about their experiences with SIGI. They may be contacted at the following numbers:

Connie Webster
Case Western Reserve University
216/368-3754

Skip Sturman
Dartmouth College
603/646-2215

Sherry Traupane
Florida Department of Education
(Marianna High School)
904/488-0400

Sallie Downer
Russell Sage Junior College of Albany
518/445-1753

Don Casella
San Francisco State University
415/469-2530

Joe Johnston
University of Missouri-Columbia
314/882-2351

Ronald Lambert
Vassar College
914/452-7000, ext. 2600

The following information is reprinted by permission from the SIGI Information Bulletin:

SIGI at a Glance

- is an interactive computer-based aid to career decision making . . .
- serves primarily students in, or about to enter, two-year and four-year colleges . . .
- complements the work of guidance counselors . . .
- was developed on the PDP-11 computer . . .
- has been converted for other minicomputers, some mainframes, and certain microcomputers . . .
- includes six interrelated subsystems listed below. (Each subsystem raises a major question and helps the student answer it. These questions and answers form distinctive steps in decision making.)

For further information about SIGI, call 609/734-5165 or write to:

SIGI Office
Educational Testing Service
Princeton, New Jersy 08541

	SUBSYSTEM	WHAT THE STUDENT DOES	QUESTIONS ANSWERED
	Introduction	Learns concepts and uses of major sections listed below.	Where do you stand now in your career decision making? What help do you need?
I.	VALUES	Examines 10 occupational values and weighs importance of each one.	What satisfaction do you want in an occupation? What are you willing to give up?
II.	LOCATE	Puts in specifications on 5 values at a time and gets lists of occupations of interest.	Where can you find what you want? What occupations should you look into?
III.	COMPARE	Asks pointed questions and gets specific information about occupations of interest.	What would you like to know about occupations that you are considering? Should you reduce your list?
IV.	PREDICTION	Finds out probabilities of getting various marks in key courses of preparatory programs for occupations.	Can you make the grade? What are your chances of success in preparing for each occupation you are considering?
V.	PLANNING	Gets displays of program for entering each occupation, licensing of certification requirements, and sources of financial aid.	How do you get from here to there? What steps do you take to enter an occupation you are considering?
VI.	STRATEGY	Evaluates occupations in terms of the rewards they offer and the risks of trying to enter them.	Which occupations fit your values best? How do you decide between an occupation that is highly desirable but risky and one that is less desirable but easier to prepare for?

Computers in the Bunkhouse

Those "How I Spent My Summer Vacation" essays will never be the same, at least not for children attending the fifth annual National Computer Camp. This unique educational and recreational experience is the original computer camp in the USA. Last year's sessions were widely successful, and plans are being made for another summer of learning the technological ropes for kids between the ages of 10 and 18. This year the camp has been expanded to two locations, Simsbury, Connecticut and Atlanta, Georgia. Campers may elect one- or two-week sessions from July 11 to August 6, 1982.

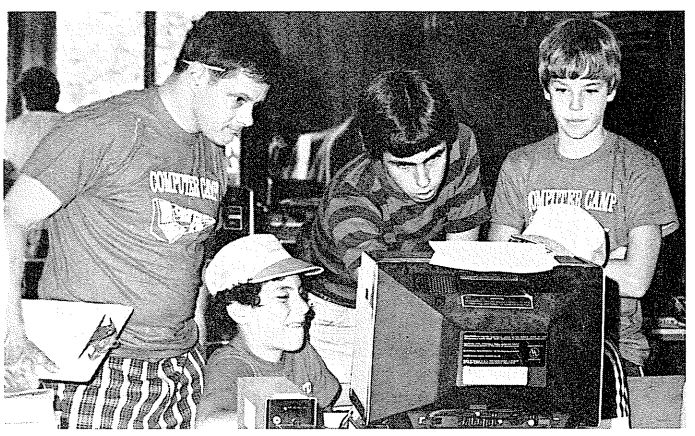
The computer enthusiasts attending the computer camps include kids with a wide range of computer experiences, including no experience whatsoever. The campers are grouped according to computer experience and age. The camp director, Michael P. Zabinski, Ph.D., is assisted by elementary and secondary school teachers who are experienced in classroom computer instruction.

As does any good camp program, this one keeps its campers going morning, noon and night. For the morning sessions, which run from 9:00 AM to noon, campers divide into groups of 12 campers each, with an instructor, an assistant and several computers. Later on in the afternoon, the campers are encouraged, but not required, to participate in outdoor activities such as swimming, volleyball, tennis, water polo, and shuffle board. But "at least half" choose not to, says Zabinski, preferring the mental exercise their computers allow. In addition to open computer rooms, evenings also bring olympics, space adventure tournaments, Las Vegas night, and game-a-thons. Sound like ordinary camp fare? Rest assured that the playing field is always the video screen. To divert attention from the small screens, occasional big screen showings of popular films are also offered in the evening.

Director Zabinski, who is a professor at Fairfield University, Fairfield, Connecticut, says he founded the camp in 1978 because he's worked with teachers interested in computers for several years and wanted a way to teach the kids. Dr. Zabinski is also the author of "Introduction to TRS-80 Level II BASIC" a computer textbook available from Radio Shack (26-2116 $9.95).

For further information contact:

Michael Zabinski, Ph.D., at (203) 795-3049
or write to
National Computer Camp
Box 624D
Orange, CT 06477



reading•is•fun

Radio Shack's new home education series **reading•is•fun** is based on the idea that educational reading can be fun, that fun reading can be educational, and that working with the Color Computer can be both educational and fun. **reading•is•fun** programs operate on a 4K, 16K, or 32K TRS-80 Color Computer tape system. Lesson materials in the series are appropriate for any child reading at levels 4 to 6.

The **reading•is•fun** series takes classic stories like Moby Dick, 20,000 Leagues Under the Sea, Dracula, and The Hound of the Baskervilles (more titles will follow) and presents these stories using a motivational approach which has found growing acceptance in classrooms.

Each title in the series includes a reading book presenting an illustrated-format version of a story, a read-along audio tape, and a computer tape for use with the TRS-80 Color Computer. The child READS the book, LISTENS to the audio tape, and then runs the program on the computer tape to take spelling and vocabulary DRILLS based on the story.

READ

The illustrated-format readers were developed by an established educational publisher using the Dale-Chall Readability Formula to insure that the text is appropriate for reading levels 4 to 6. The format is designed to motivate the child to read the illustrated book itself, but these introductions to classic stories in a shortened illustrated form may also make the child want to read the originals!

LISTEN

The read-along audio tape gives a word-for-word dramatic presentation of the story. It is designed to aid in comprehension while increasing the child's enjoyment of the story.

DRILL

Once the child has read the book and listened to the tape, it is time to use the Color Computer.

Using the Color Computer makes working the spelling and vocabulary drills seem like a game. The computer provides immediate feedback after each problem is worked, and a progress report is given after every ten problems. The program is designed so that it can be run over and over without presenting the same problem sequence.

Spelling exercises are furnished on one side of the computer lesson tape, with the Vocabulary program on the other side. The Spelling program displays a sentence from the story that was read and then pauses so that the child can read the sentence. Then a word in the sentence is highlighted, and disappears. To work a spelling exercise, the child enters the correct spelling of the word that was highlighted. The Vocabulary program presents a word from the story and gives a selection of definitions. A vocabulary problem is worked by selecting the number that corresponds to the correct definition.

When an exercise is answered correctly on the first try, a tone sounds and a message like "GOOD JOB!" or "SUPER!" appears on the screen. After an incorrect first try, the computer displays the message "PLEASE TRY AGAIN". If two tries are incorrect, the computer displays the correct answer.

At the Radio Shack Education Division, we are enthusiastic about the new **reading•is•fun** series, and believe that children will enjoy the READ, LISTEN, and DRILL activities, and that parents will find the program a constructive home learning aid.

Neighborhood Computer Literacy

Don Claytor is committed to computer literacy. He has set up the Computer Learning Center in Elmira, New York, to give young people familiarity with computers. Funded by the Private Industry Council, the Center provides free classes for students in grades eight through twelve after school and during the summer. Radio Shack Educational Regional Coordinator Bob Sochor trained the eighteen volunteers who form the teaching staff of the Center.

The idea for this project started on a small scale, with one 4K TRS-80 Model I in Don's home. With increased funding and community interest and participation, it has grown to fill a 5000 square foot building. This building now houses a 48K TRS-80 Model I with four disk drives, five 16K TRS-80 Model III's, a 64K TRS-80 Model II, and several printers.

Harold Faulisi, the manager of the local Radio Shack Computer Center, cooperated by helping Don choose the appropriate computer configurations for his application. Don says the service has been good—"no quibbles or quarrels."

Last spring, the Computer Learning Center had a ten-week session offering five different sections. Each of the eighty-five students was allowed to participate in one section. The sections were divided as follows:

1. Introduction to data processing. The students in this section took field trips to local businesses and industries to see how computers help people on the job.
2. Word processing. Students with typing skills learned word processing using Radio Shack's SCRIPSIT program.
3. Programming. Using materials including Radio Shack's Level II Reference Manual and some exercises from Radio Shack's Computer Education Series, students learned how to program the computer in BASIC.
4. Computer assisted instruction. Using educational software (including Radio Shack's K-8 Math Program), students enjoyed developing their skills in math and other areas.
5. Games. Students typed in programs from listings in various computer game books. This gave them the experience of working with programs, the satisfaction of completing a challenging task, and the enjoyment of playing the computer games.

A new after-school session began November 1. Working with ten time slots per week, about seventy students enrolled for "hands-on" computer activities. The Center has also been used during the day by the local school district to teach BASIC to junior high school students.

Don uses the computers and Radio Shack software to assist in his own administrative tasks as well. He uses Profile to keep track of student records, VisiCalc to work on the Center's finances, SCRIPSIT for correspondence, and Versafile to keep track of his file of newspaper clippings.

Don's plans for the future include efforts to get more private funding for the Center so that they can get more computers and handle more students.

For Don, the experience has been a satisfying one. "We've been trying to do this for ten years," he says. "Only since micros could we do this on a community-wide basis."

TRS-80® Educational Software Sourcebook

How to Obtain a Listing for Your Program

In response to the growing number (flood?) of inquiries for information about the availability of TRS-80 educational software from outside publishers and authors, Radio Shack will publish an educational software directory entitled the "Radio Shack TRS-80 Educational Software Sourcebook (26-2756)".

The Educational Software Sourcebook will offer your TRS-80 instructional material national exposure in the educational community. It will be sold to the educational community through over 4,000 Radio Shack stores and the more than 170 Radio Shack Computer Centers.

The Educational Software Sourcebook will be published annually. Listings are to be for the term of one year and must be resubmitted or renewed at the end of twelve months following registration. The listing fee for commercial software is $10.

Teachers, schools, and educational institutions which offer public access to classroom computer resources or computer assisted instructional material will be listed at no charge under a special resource classification. For the first edition of the TRS-80 Educational Software Sourcebook, all programs in the public domain, submitted by teachers or school groups, will be listed individually without fee.

Unlike the current TRS-80 Applications Software Sourcebook (26-2114), educational software authors and publishers will have a user site reference option. The intent is to furnish educators with software validation information. The user site reference can be any individual, institution, or agency selected by the publisher for this purpose. The reference can also be any internal department responsible for disseminating information on pilot test sites or validation procedures used by the publisher.

Each listing submission which includes the optional user site reference must be accompanied by a written statement from the individual, institution or agency being used as a reference, which gives Radio Shack permission to publish their name and address as the software publisher's reference for public inquiry.

The Radio Shack Educational Software Sourcebook will accept only listings for instructional or administrative TRS-80 computer programs. Requests to list computer software such as monitors or operating systems will be returned. Radio Shack reserves the right to reject any listing by returning the payment to the sender.

All listings are accepted with the express understanding that Radio Shack shall have no liability for errors which may occur in the printed Educational Software Sourcebook listing, including the failure to include a listing, and in no event shall Radio Shack be liable for damages to any person submitting listings in an amount greater than the listing fee ($10.00).

Application forms for a listing in the Radio Shack Educational Software Sourcebook can be obtained from:

TRS-80 Educational Software Sourcebook
Radio Shack Education Division
400 Atrium, One Tandy Center
Fort Worth, Texas 76102

Computing in the Netherlands

Radiation Suppression In a Model I

J. Noordanus
Bachlaan 6
1272 EK Huizen
The Netherlands

7 August 1981

In reading your column "View from the Seventh Floor," in TRS-80 Microcomputer News, Vol., 2, Issue 9, Nov. 1980 regarding the difficulties in retrofitting radiation suppression means in older TRS-80's I thought you and many users would be interested how I tackled this problem.

Continuous use of my micro provoked so much irritation in the family, when looking to VHF TV programs, that as a radio engineer I had to find a good solution.

To obtain cheap, simple and effective screening, at the source of the trouble, without poking into the interior of the computer or obstructing the cooling or damaging the appearance I used (do not laugh) galvanized aviary wire with a square mesh, 12 x 12 mm, soldered at the joints.

For the VHF frequencies involved this mesh gives good screening and the grey colour harmonizes with the TRS-80 colours. Modelling this wire-mesh around the CPU, leaving holes for connections and the keyboard is very easy and can be accomplished in a short time.

Interconnection of the different mesh parts is done by bending loose ends of one part around the squares of the other part, thus forming a rigid cage. This screen is connected to the CPU earth (ground) via the power supply plug (terminal 4). I made this cage in a basket form, in case of servicing the CPU can be lifted out.

Very important is that each outgoing cable passes (with 4 or 5 turns) through a ferrite ring of about 2-3 cm inner diameter. This dimension leaves room to pass the plugs, there is no need to remove these from the cable.

The video display unit (happily) needs no screening, if the in and outgoing cables are also passed through ferrite rings. If an expansion interface is used the situation is somewhat more complex.

The wire cages of CPU and expansion interface should be properly interconnected by the same wire mesh on the bottom, the sides and at the keyboard level. Ferrite rings are again used for all cables, flat ferrite bars are used around the band cable.

The attached photographs show you the end result of this exercise. Perhaps a more sophisticated solution would be a cage of perforated metal sheet, but this is more costly, unless mass fabricated.

Results: Interference on our TV set (with attic antenna) vanished and AM-FM reception with a portable radio is not disturbed at a distance of about 1 meter from the micro.

In the long run (about 5 months) it came out that also the stability of the TRS-80 system itself was improved. No strange (casual) happenings occurred any more.

Before informing our local electronics press I thought it wise to inform you first. You are free to publish these results in your Microcomputer News, if you provide copies and due acknowledgement. * * *

Computer Programs Sent by Shortwave

The Results and Plans for the Future

BACKGROUND

On September 10th 1981, Radio Netherlands made a unique experiment for international short-wave broadcasting. The station broadcasts on the short-wave broadcasting bands to various parts of the world, and this includes target areas in the Pacific, North America, Asia, Europe and Africa for transmissions in the English language. In addition to transmitters at Lopik in Holland, the station also uses relay facilities in Madagascar and Bonaire, Netherlands Antilles. This ensures reliable signal strength in chosen target areas. Since reception is generally good, it was decided that computer programs in machine readable form might be readable if broadcast over the air.

"Media Network" is the title of the weekly 30 minute communications magazine broadcast by the English Service of Radio Netherlands. It concentrates mainly on short-wave broadcasting developments, though topics such as satellite TV and video developments also feature from time to time. Since listeners in North America, Australia, and New Zealand have indicated an interest in home computer developments, the possibility of investigating this aspect of communications was studied. Many home computers have facilities for a cassette interface for the storing of information. It seemed reasonable therefore that a program could be recorded onto tape at the studios in Hilversum, Holland, broadcast over short-wave and recorded by the listener. If conditions were ideal, the listener's home computer should then be able to use the program transmitted.

As short-wave broadcasting suffers from background noise, due to both atmospheric and man-made sources of interference, it was decided to experiment with test transmissions. Listeners with access to a home computer were asked

to record the Media Network broadcast of September 10th, 1981, and attempt to load the program into their computer. This would determine whether the signal to noise ratio was good enough to enable broadcasting of computer program information on an international scale. To avoid failure due to bad radio propagation conditions, the computer information was also repeated the following Thursday September 17th in transmissions to the Pacific and North America.

Three home computers were selected for the experiment, these being the most popular types among the audience. These were the Apple®, TRS-80 Tandy Radio Shack, and Commodore PET™ systems. A simple direction and bearing program was devised in easy-to-use BASIC, by Professor John Campbell, Head of the Computer Science Department of Exeter University in England. The program was recorded in the three different versions to suit the cassette interfaces of the chosen machines.

No modifications were made to the transmission mode, it being the standard Amplitude Modulation (AM) system in regular use by international broadcasters. The recordings were recorded at 0 dB to ensure that at the time of transmission, the transmitters were as near the ideal 100% level of modulation as possible. This in fact was the case for the relay station transmitters at Bonaire and Madagascar, which are fed via satellite with a bandwidth of approx 5.5 kHz. The Lopik transmitters however were not 100% modulated due to the age of the senders, the level being nearer 70% on the day of broadcast.

RESULTS

235 listeners responded to the request for assistance and supplied feedback to the station. It was noticeable that quite a few monitors had access to test equipment, and provided useful data from oscilliscope screens as to the level of noise and modulation. A total of 42% of listeners were successful in copying a perfect, or near perfect, program which was read by their home computer. Reasons for failure can be summarized as follows:

1. Only one listener in Switzerland reported success with the Apple computer. All other reports indicated that the entire data was lost by noise.
2. Only listeners who had direct connections between receiver and cassette recorder were able to make satisfactory copies. No one who simply held a microphone in front of the short-wave receiver was successful.
3. 10% of listeners who recorded the program off air and were unsuccessful at the first attempt, found that by re-recording onto a second tape machine, and raising the level of the signal, resulted in acceptable copy.
4. 5 listeners reported that adjustment of the tone control, boosting the treble response was critical.
5. Bandwidth setting on the short-wave receiver was important. Those who used settings lower than 5 kHz reported a failure. It was interesting therefore that many listeners who were successful were only using modest or average quality equipment.
6. Incompatibility was experienced by some users. It was not stated in the broadcast that the TRS-80 program was for the Model I Level II machines. Radio Shack Color Computers are unknown in Europe because the colour standard is not NTSC.
7. Of the 98 successes, the following statistics are available:
   a) 1 success with the Apple transmission
   b) 36 successes were reported with PET systems
   c) 61 successes were reported with TRS-80 systems

Distribution shows that 82% of successes were from Europe. Countries represented were Denmark, England, Scotland, Switzerland, West Germany, Austria, Sweden, and Belgium. Countries outside the continent were Canada, USA (1) and Belize (monitoring via Bonaire). No response was received from those listening via the Madagascar transmitters to indicate success.

CONCLUSIONS

It is clear that transmission of computer data in machine readable form is a practical possibility via international short-wave radio. Whilst program data has already been transmitted via amateur radio over short-wave bands, the operators have had the advantage of point to point communications and the more efficient Single Side-band (SSB) mode of transmission. As far as we know, this is the first time that computer data has been transmitted in the broadcasting bands, intended for a mass audience. The success percentage indicates it may be a viable experiment worth further investigation.

It is clear that the best chance of reliable copy is obtained when:

a) recording is made directly from receiver to cassette.
b) levels of recording are set as high as possible without too much distortion.
c) the signal is processed at the transmitting end to ensure maximum boosting of treble frequencies.
d) the receiver is operated at the widest possible bandwidth, without allowing interference from adjacent broadcasting stations.
e) tone controls on recorders and receiver are adjusted for maximum treble boost.

LISTING OF COMPUTER PROGRAMME TRANSMITTED

BASIC program for range and bearing calculation:

```
100 DIM A(2),B(2)
110 R=3953
120 P=3.1416
130 PRINT " "
140 K=1
150 FOR J=K TO 2
160 IF J=1 THEN 190
170 Z$ = "XMTR"
180 GOTO 200
190 Z$ = "RCVR"
200 PRINT " "
210 PRINT Z$,"LAT?:","(DEGS","MINS)"
220 INPUT A(J),X
230 A(J)=P*(A(J)+X/60)/180
240 PRINT "N OR","S?"
250 INPUT S$
260 IF S$ = "N" THEN 300
270 IF S$ = "S" THEN 290
280 GO TO 240
290 A(J)=-1*A(J)
300 PRINT " "
310 PRINT Z$,"LONG.","(DEGS","MINS)"
320 INPUT B(J),X
330 B(J)=P*(B(J)+X/60)/180
340 PRINT "E OR","W?"
350 INPUT S$
360 IF S$ = "E" THEN 400
370 IF S$ = "W" THEN 390
380 GO TO 340
390 B(J)=-1*B(J)
400 NEXT J
410 PRINT " "
420 C=COS(A(2))
430 X=C*COS(B(2))
440 C=C*SIN(B(2))
450 D=SIN(A(2))
460 H=SIN(A(1))
```

```
47Ø G=COS(B(1))
48Ø J=SIN(B(1))
49Ø K=COS(A(1))
5ØØ W=G*X+J*C
51Ø E=H*W-K*D
52Ø F=G*C-J*X
53Ø G=K*W+H*D
54Ø IF ABS(G)=1 THEN 6ØØ
55Ø W=1-G*G
56Ø IF W > Ø THEN 58Ø
57Ø GO TO 6ØØ
58Ø H=ATN(G/SQR(W))
59Ø GO TO 61Ø
6ØØ H=G*P/2
61Ø IF ABS(G)=1 THEN 76Ø
62Ø IF W <= Ø THEN 76Ø
63Ø IF E>Ø THEN 74Ø
64Ø IF E<Ø THEN 7ØØ
65Ø IF F>Ø THEN 68Ø
66Ø X=-Ø.5*P
67Ø GO TO 78Ø
68Ø X=P/2
69Ø GO TO 78Ø
7ØØ X=ATN(F/E)-P
71Ø IF F < Ø THEN 78Ø
72Ø X=X+2*P
73Ø GO TO 78Ø
74Ø X=ATN(F/E)
75Ø GO TO 78Ø
76Ø PRINT "ANY","ANGLE","O.K."
77Ø GO TO 83Ø
78Ø D=18Ø*(P-X)/P
79Ø C=INT(D)
8ØØ IF D-C < Ø.5 THEN 82Ø
81Ø C=C+1
82Ø PRINT "ANGLE",C,"DEGS."
83Ø PRINT " AND"
84Ø D=R*(Ø.5*P-H)
85Ø C=INT(D)
86Ø IF D-C < Ø.5 THEN 88Ø
87Ø C=C+1
88Ø PRINT "RANGE",C,"MILES"
89Ø PRINT " "
9ØØ PRINT "NEW","XMTR?","(YES","OR","NO)"
91Ø INPUT S$
92Ø PRINT " "
93Ø IF S$ = "YES" THEN 1Ø2Ø
94Ø IF S$ = "NO" THEN 96Ø
95Ø GO TO 9ØØ
96Ø PRINT "NEW","RCVR?","(YES","OR","NO)"
97Ø INPUT S$
98Ø PRINT " "
99Ø IF S$ = "YES" THEN 14Ø
1ØØØ IF S$ <> "NO" THEN 96Ø
1Ø1Ø GO TO 1Ø4Ø
1Ø2Ø K=2
1Ø3Ø GOTO 15Ø
1Ø4Ø STOP
1Ø5Ø END
```

This program is available to all users without copyright providing the compiler John Campbell and Radio Netherland's MEDIA NETWORK are credited.

For further information on the experiments please contact the following address:

Jonathan Marks
Producer: Media Network
English Section
Radio Netherlands,
P.O. Box 222,
Hilversum, HOLLAND.

Telephone IDD 31 35 16151 ext 344. available during office hours 07.30-16.30 GMT.

FUTURE PLANS FOR PHASE TWO OF RADIO NETHERLANDS EXPERIMENT

Following the first phase of these experiments, we intend to continue with the next broadcast of computer programs on January 28th, 1982. This will be broadcast on the following frequencies, **shortwave**.

TIME (Greenwich Mean Time)	FREQUENCIES IN kHz	
07.50	9715, 9770	Pacific
08.50	9715	Pacific
09.50	15560, 11930, 9895 6045, 5955	Europe
13.50	17605, 11930, 9895 6045, 5955	Europe
18.50	15220, 6020	Africa
20.50	21685, 17695, 17605, 15220, 9715	Africa & Europe

The above transmissions are on Thursday, January 28th. Below are two transmissions for North America which are broadcast on Friday, January 29th, 1982 according to Greenwich Mean Time. However, remember that in the target area it will still be Thursday evening.

02.50	9590, 6165	North America East
05.50	9715, 6165	North America West

WHICH COMPUTERS?

We intend to broadcast a sunrise-sunset program suitable for the short-wave listener for the following computers: SINCLAIR ZX-81, TRS-80 MODEL I LEVEL 2, and PET. We hope to include ATARI too, but since this computer is not available in Europe we are looking for co-operators in North America willing to help record this system into tape. Offers please to the above address.

HOBBYSCOPE BASIC CODE

There is a technical enthusiasts' programme broadcast by the Dutch domestic radio service, entitled "HOBBYSCOOP" (pronounced in English as Hobbyscope). The programme has a following of approximately 60% of the computer users in the Netherlands (= about 1200 home computer users) and they broadcast programs in machine readable form over the FM and Medium Wave transmitters of Hilversum 1 each Sunday night at 17.30 GMT. Because of the incompatibility of the various systems of cassette interfaces, Dutch computer enthusiasts together with the programme's producer, Hans Janssen, have developed an "ESPERANTO" for computers. This consists of a 1200 baud code, consisting of two tones, one at 1200 Hz, the other at 2400 Hz. To be able to decode the transmitted programs many computers (such as APPLE or PHILIPS) simply need a copy of the translation program. This is being provided for a small charge to interested listeners in the Netherlands. Other computers such as the TRS-80 need the addition of a small amount of electronics costing US $40.00 complete. These charges only cover the cost of components—the whole scheme is designed to make a workable system **NOT** to create profits. All designs, the code, and programs are **without** copyright.

Since the Apple system does not seem to work well on short-wave, we see an application for the Hobbyscope code for international use. Since Apple users only require a copy of the translation program, users who helped in Phase 1 have been selected. They will be sent a copy of the translation program, and asked to monitor the January 28th broadcast. A

(Continued on page 41)

RSCOBOL Screen Dump Utility

David Ray
605 Caravaca Drive
Garland, TX 75043

I am enclosing an assembly listing of an enhanced version of the screen dump routine which appeared in the April, 1981 edition for Model II users. This routine has been written specifically for call from a COBOL mainline program. Although the previously published listing is satisfactory for many applications, SCRNDUMP has the added feature of converting reverse video characters, which would otherwise be displayed as graphics characters, to their white on black counterparts.

The subroutine origin is at hex F400, which overlays the DEBUG code, therefore it is imperative that DEBUG not be on when the routine is called. Those who do not have the Z80 assembler can enter the program through DEBUG and use DUMP to store it.

In my COBOL systems, all keyboard input is done from a COBOL subroutine. This routine is structured such that at any point, the user can enter a CONTROL/N and SCRNDUMP will be called, printing the current contents of the screen.

```
                        ØØ1ØØ        ;
                        ØØ2ØØ        ;
                        ØØ3ØØ        ;       SCRNDUMP
                        ØØ4ØØ        ;
                        ØØ5ØØ        ;       David Ray
                        ØØ6ØØ        ;
                        ØØ7ØØ        ;       9-Aug-81
                        ØØ8ØØ        ;
                        ØØ9ØØ        ;
                        Ø1ØØØ        ;
                        Ø11ØØ        ;       Subroutine to print the contents of the
                                             screen to to the
                        Ø12ØØ        ;       Line Printer. May be called from COBOL as
                                             follows:
                        Ø13ØØ        ;
                        Ø14ØØ        ;                CALL "SCRNDUMP".
                        Ø15ØØ        ;
                        Ø16ØØ        ;Code is inc to cause reverse video characters
                        Ø17ØØ        ;to be converted to their standard equivalents,
                        Ø18ØØ        ;giving the effect on the hardcopy that no reverse
                        Ø19ØØ        ;video was used.
                        Ø195Ø        ;
                        Ø2ØØØ        ;The displayed data is processed one CRT line at a
                        Ø21ØØ        ;time, first loading line Ø (top line) into the
                        Ø22ØØ        ;area BUFF. Each character is then checked for
                        Ø23ØØ        ;reverse video, and if found to be that, it is
                        Ø24ØØ        ;XORed with 1ØØØØØØØ (binary) to yield standard
                        Ø245Ø        ;ASCII value for that character.  After all
                        Ø25ØØ        ;char have been so checked, the entire buffer
                        Ø26ØØ        ;is transferred to the printer using the
                        Ø265Ø        ;appropriate SVC.
                        Ø27ØØ        ;
                        Ø28ØØ        ;A form feed is printed before & after the screen
                        Ø29ØØ        ;dump.
                        Ø3ØØØ        ;
                        Ø31ØØ        ;       OVERLAYS DEBUG CODE BEGINNING AT F4ØØ
                        Ø32ØØ        ;
ØØØØ'                   Ø33ØØ        ASEG
                        Ø34ØØ        ORG     X'F4ØØ'
F4ØØ  C5        Ø35ØØ MAIN:: PUSH    BC          ; Save BC Register
F4Ø1  D5        Ø36ØØ        PUSH    DE          ; Save DE Register
F4Ø2  E5        Ø37ØØ        PUSH    HL          ; Save HL Register
F4Ø3  3E 12     Ø38ØØ        LD      A,18        ; 18-SVC for Char to Prnter
F4Ø5  Ø6 ØC     Ø39ØØ        LD      B,12        ; 12-ASCII value for Form Feed
F4Ø7  CF        Ø4ØØØ        RST     8           ; SVC
F4Ø8  21 F444   Ø41ØØ        LD      HL,BUFF     ; Location of 8Ø Char Buffer
F4ØB  1E ØØ     Ø42ØØ        LD      E,Ø         ; Loop Index
F4ØD  43        Ø43ØØ LOOP1: LD      B,E         ; Row Number (Ø thru 23)
F4ØE  ØE ØØ     Ø44ØØ        LD      C,Ø         ; Column Number (Start at zero)
F41Ø  16 5Ø     Ø45ØØ        LD      D,8Ø        ; Buffer Length
F412  3E ØB     Ø46ØØ        LD      A,11        ; SVC number
F414  CF        Ø47ØØ        RST     8           ; Do the SVC to get Video data
F415  Ø1 F444   Ø48ØØ        LD      BC,BUFF     ; Load BC with address of buffer
F418  16 ØØ     Ø49ØØ        LD      D,Ø         ; D register is loop index
F41A  ØA        Ø5ØØØ LOOP2: LD      A,(BC)      ; Load data from buffer to A reg
F41B  FE 8Ø     Ø51ØØ        CP      1ØØØØØØØB   ; Compare with 128 decimal
F41D  FA F423   Ø52ØØ        JP      M,SKIPOR    ; Jump if neg (not rev. video)
F42Ø  EE 8Ø     Ø53ØØ        XOR     1ØØØØØØØB   ; XOR to reset 7 bit which will
                Ø54ØØ                            ; convert the character from
                Ø55ØØ                            ; reverse to positive video
F422  Ø2        Ø56ØØ        LD      (BC),A      ; Replace character into buffer
F423  14        Ø57ØØ SKIPOR:INC     D           ; Bump Loop Index
F424  Ø3        Ø58ØØ        INC     BC          ; Bump Buffer Location
F425  7A        Ø59ØØ        LD      A,D         ; Prepare to test loop index
F426  FE 4F     Ø6ØØØ        CP      79          ; Compare with 79 for number of
                Ø61ØØ                            ; characters in buffer
F428  C2 F41A   Ø62ØØ        JP      NZ,LOOP2    ; Go to get next character
F42B  Ø6 5Ø     Ø63ØØ        LD      B,8Ø        ; Length of Buffer
F42D  ØE ØD     Ø64ØØ        LD      C,13        ; Printed Line Terminator (CR)
F42F  3E 13     Ø65ØØ        LD      A,19        ; SVC number
F43i  CF        Ø66ØØ        RST     8           ; Do SVC
F432  1C        Ø67ØØ        INC     E           ; Bump Index for Loop
F433  7B        Ø68ØØ        LD      A,E         ; Prepare to test loop index
F434  FE 17     Ø69ØØ        CP      23          ; Do Compare (lines Ø thru 23)
F436  C2 F4ØD   Ø7ØØØ        JP      NZ,LOOP1    ; Jump if not finished
F439  3E 12     Ø71ØØ        LD      A,18        ; Print character SVC
F43B  Ø6 ØC     Ø72ØØ        LD      B,12        ; Form Feed ASCII value
F43D  CF        Ø73ØØ        RST     8           ; Do SVC to print form feed
F43E  E1        Ø74ØØ        POP     HL          ; Restore HL register
F43F  D1        Ø75ØØ        POP     DE          ; Restore DE register
F44Ø  C1        Ø76ØØ        POP     BC          ; Restore BC register
F441  3E ØØ     Ø77ØØ        LD      A,Ø         ; Zero Accumulator or
                Ø78ØØ                            ; COBOL will stop run
F443  C9        Ø79ØØ RET                        ; Finished - Return
F444            Ø8ØØØ BUFF:  DS      8Ø          ; 8Ø Character Buffer area
                Ø81ØØ        END     MAIN


Macros:

Symbols:
BUFF    F444    LOOP1   F4ØD    LOOP2   F41A    MAIN    F4ØØI
SKIPOR  F423
No Fatal error(s)
```

Model II Bugs, Errors and Fixes

26-4502 INVENTORY MANAGEMENT

The Inventory Management system does not give you the capability to re-initialize the system or change parameters once the system has been run.

If you wish to re-initialize the IMS system, the following BASIC program will do it. This program should be used ONLY if there is no clean (un-initialized) master disk available for BACKUP. Note that this BASIC program is a correction of the program we published in the September, 1980 Microcomputer News.

1. At TRSDOS Ready, type:

```
BASIC -F:3 <ENTER>
```

2. When BASIC has loaded, enter the following program:

```
1Ø CLEAR 1ØØØ
2Ø KILL "SALEDATA/DAT"
   : KILL "CONTROL/DAT"
3Ø OPEN "R", 1, "CONTROL/DAT", 96
4Ø FIELD 1, 96 AS X$
   : LSET X$=STRING$(96,32)
   : PUT 1, 1
5Ø LSET X$=STRING$(96,Ø)
6Ø FOR F=2 TO 5
   : PUT 1, F
   : NEXT F
   : CLOSE
7Ø SYSTEM "IMS"
```

3. Type:

```
RUN <ENTER>
```

4. Type

```
SYSTEM
```

to get to TRSDOS Ready and make a BACKUP.

26-4510 VERSA-FILE

For an important note on using key words, please see Model I/III Versa-file (26-1604) on page 21 of this issue.

26-4512 PROFILE II

We received the following information from Allan L. Forsythe, Headmaster of the carroll school in Lincoln, Massachusetts:

"I attempted to set up a two segment program on drive 0. Segment one was a mailing list. All of segment 2 was to be reserved for a single field. I entered "256" for the field size. The program proceeded to allot zero space for the field. Furthermore, it refused to execute mode 7 (Update, Record Addition). When we redid the program using "255" characters for the field (leaving one space blank) the program ran fine."

Mr. Forsythe is correct, you cannot have a single field which is longer than 255 bytes. If you need to use 256 bytes, reserve two separate fields of 128 bytes and then display them next to each other.

26-4604 ACCOUNTS RECEIVABLE (THREE DISK)

Please note that while you are in Customer File Maintenance, you should not change the customer number if the customer has outstanding items in the A/R Open Item file. Use the Customer Account Inquiry application to determine whether there are existing A/R open items.

26-4715 BINARY SYNCHRONOUS COMMUNICATIONS 3270

The following information should be used to update or modify the information presented on page 19 of the 3270 BiSync package:

3270 KEYBOARD EQUIVALENTS
MODEL II EQUIVALENTS

3270 KEY	FROM PG.19	SHOULD BE
ENTER	⟨ENTER⟩	
CLEAR	⟨CTRL⟩ ⟨Z⟩	
RESET	⟨CTRL⟩ ⟨R⟩	
BACK SPACE	⟨ ← ⟩	
FORE SPACE	⟨ → ⟩	
UP LINE	⟨ ↑ ⟩	
DOWN LINE	⟨ ↓ ⟩	
BACK TAB	⟨CTRL⟩ ⟨B⟩	⟨CTRL⟩ ⟨E⟩
FORE TAB	⟨TAB⟩	⟨TAB⟩ or ⟨CTRL⟩ ⟨I⟩
NEXT LINE	⟨CTRL⟩ ⟨N⟩	
ERASE EOF	⟨CTRL⟩ ⟨F⟩	⟨CTRL⟩ ⟨G⟩
ERASE INPUT	⟨CTRL⟩ ⟨I⟩	⟨CTRL⟩ ⟨K⟩ (SEE NOTE 1)
FIELD MARK	⟨CTRL⟩ ⟨M⟩	⟨CTRL⟩ ⟨F⟩
DUP	⟨CTRL⟩ ⟨D⟩	
LOCAL PRINT	⟨CTRL⟩ ⟨P⟩	
HOME	⟨CTRL⟩ ⟨H⟩	⟨CTRL⟩ ⟨O⟩
PA-1	⟨ESC⟩ ⟨E⟩	⟨ESC⟩ ⟨D⟩
PA-2	⟨ESC⟩ ⟨F⟩	⟨ESC⟩ ⟨E⟩
PA-3	⟨ESC⟩ ⟨G⟩	⟨ESC⟩ ⟨F⟩
PF-1	⟨F1⟩	⟨F1⟩ or ⟨ESC⟩ ⟨1⟩
PF-2	⟨F2⟩	⟨F2⟩ or ⟨ESC⟩ ⟨2⟩
PF-3	⟨ESC⟩ ⟨3⟩	
PF-4	⟨ESC⟩ ⟨4⟩	
PF-5	⟨ESC⟩ ⟨5⟩	
PF-6	⟨ESC⟩ ⟨6⟩	
PF-7	⟨ESC⟩ ⟨7⟩	
PF-8	⟨ESC⟩ ⟨8⟩	
PF-9	⟨ESC⟩ ⟨9⟩	
PF-10	⟨ESC⟩ ⟨A⟩	
PF-11	⟨ESC⟩ ⟨B⟩	
PF-12	⟨ESC⟩ ⟨C⟩	

NOTE 1

Implementation of the 'ERASE INPUT' key requires a change to the User Configuration Mode (UCM). Make the following changes:

Page 48, statement 33400, DB 0 ;BREAK KEY
Page 49, statement
34200, DB ERAINP ;CONTROL/K IS ERASE INPUT

26-4910 TRSDOS 2.0a

There is a problem with the LOC (Get Current Record Number) function in BASIC. The problem is that after an LOF (Get End-of-File Record Number) is executed, the pointer to the current record number is left pointing to the end of the file instead of the current record number. Therefore, when an LOF is executed after an LOF, the number returned is always the end-of-file record number, until a GET is encountered, which restores the current record number.

To illustrate this problem, run the sample BASIC program listed below. If LOC were working correctly, columns 1 and 3 would match.

To solve this problem, enter the following patch from TRSDOS Ready:

```
PATCH BASIC A=65B6 F=21 C=23
```

After you have made the patch, run the sample program again. You should notice that columns 1 and 3 now match, as they should.

SAMPLE BASIC PROGRAM:

```
1Ø CLS
     : CLEAR 1ØØ
     : OPEN "D", 1, "TEST/DAT:Ø"
2Ø FOR X=1 TO 1Ø
     : FIELD 1, 14 AS TS$
3Ø LSET TS$="THIS IS A TEST"
4Ø PUT 1, X
     : PRINT X, TS$
     : NEXT X
     : CLOSE
5Ø OPEN "D", 1, "TEST/DAT:Ø"
6Ø FOR X=1 TO 1Ø
     : FIELD 1, 14 AS TS$
7Ø GET 1, X
     : PRINT LOC(1), LOF(1), LOC(1)
     : NEXT X
     : CLOSE
```

Revisions, Enhancements, Etc.

700-204 Accounts Receivable (26-4504)	N/C
26-4504 on TRSDOS 1.2a	
700-2017 FORTRAN on TRSDOS 2.0a (26-4701)	N/C
700-2018 EDTASM on TRSDOS 2.0a (26-4512)	N/C
700-3013 Profile II on TRSDOS 2.0a (26-4512)	N/C
700-5001 Accounts Receivable Source Code	$299.00
SOURCE CODE for 26-4504 on TRSDOS 1.2a	
700-6002 TRSDOS 2.0a Upgrade (26-4910)	$ 24.95
TRSDOS 2.0a Disk & Replacement pages for manual.	
700-6005 Scripsit 2.0 Upgrade (26-4531)	$100.00
For owners of Scripsit 1.0, 1.0a (26-4530) who want to upgrade to Scripsit 2.0 (26-4531).	

Clean Text

Alexander B. Spencer
7708 Turnberry Lane
Dallas, TX 75248

Enclosed is a short program written in BASIC for the Model II. The program is designed to clean a file of unwanted video control characters which might be embedded in ASCII text. Such a file is commonly captured using TERMINAL or other smart terminal program. The video control characters are usually unwanted and interfere with most word-processor programs that might be used to clean up the text for further use.

The program is designed to operate on files with record lengths up to 80. All video control characters and nulls are

removed. The record length may be extended to any length up to 80 characters if it is desirable to convert to fixed-length. This is done by right filling with blanks.

During operation the character codes are displayed along with the text of each line. Characters being omitted are printed in reverse video, allowing monitoring of the process.

Please note the use of ON ERROR GOTO instead of the usual IF EOF(n) THEN statement to check for read past EOF. This was done because I found that leading nulls (character code 00) produced a false exit using the IF EOF(n) statement whereas the ON ERROR GOTO statement was not triggered by them. Apparently the IF EOF(n) statement checks only for a null whereas the ON ERROR GOTO is only triggered by a complete trailing block.

I have found this program useful and others might also.

```
10 REM   ********* PROGRAM 'CLEAN/BAS' **********
20 REM
30 REM      * READS FILE SAVED BY TERMINAL AND
40 REM        FILTERS VIDEO CONTROL CHARACTERS
50 REM      * RIGHT FILLS RECORDS WITH BLANKS
60 REM        TO LENGTH REQUESTED (<80,>0)
70 REM      * IF REQUESTED LENGTH <=0,
80 REM        RECORD LENGTH UNALTERED
90 REM      * REGARDLESS OF LENGTH REQUESTED
100 REM       WILL NOT TRUNCATE RECORDS
110 REM     * SAVES EDITED RECORDS ON FILE
120 REM     * REQUIRES 2 BUFFERS
130 REM     * REQUIRES SCROLL/CIM (MAY, 1980)
140 REM     * REQUIRES -M:61400
150 REM
160 REM ***** SET PARAMETERS & CONSTANTS *****
170 CLS: CLEAR 500, 61400
180 F=1
190 PRINT CHR$(02);
200 RW=0: CD=0: RD=0
210 SYSTEM "LOAD SCROLL/CIM"
220 DEFUSR=&HF000
230 Z=USR0(3)
240 GOTO 290
250 REM ***** MISC ROUTINES *****
260 PRINT@(1,0), STRING$(80," ");: RETURN
270 PRINT@(0,0), STRING$(80," ");: RETURN
280 REM ***** HEADING & INPUT *****
290 PRINT@(0,33), "FILE FILTERING";
300 PRINT@(2,0), STRING$(80,"=");
310 PRINT@(1,20), "CLEAN FILE  "; CHR$(01);: LINE
     INPUT F1$
320 PRINT@(1,40), CHR$(02); "AND SAVE AS  ";
     CHR$(01);: LINE INPUT F2$
330 GOSUB 260
340 PRINT@(1,28), CHR$(02); "RECORD LENGTH FOR
     FILL  "; CHR$(01);: INPUT LL
350 GOSUB 260
360 Z$="CLEANING "+ F1$+ " AND SAVING AS "+ F2$+
     " WITH REC LEN OF"+ STR$(LL)
370 PRINT@(1,40-(LEN(Z$)/2)), CHR$(02); Z$;
380 PRINT@(3,15), "<F1> AT ANY TIME TO SET/CHANGE
     SINGLE TO CONTINUOUS"
390 REM ***** READ AND WRITE FILES *****
400 OPEN "I", 1, F1$
410 OPEN "O", 2, F2$
420 ON ERROR GOTO 810
430 LINE INPUT #1, T$
440 Z$=""
450 GOSUB 600
460 PRINT Z$;
470 PRINT: PRINT: PRINT: PRINT
480 I$=INKEY$: IF I$="" THEN 500
490 IF ASC(I)=01 THEN F=F*-1
500 IF F>0 THEN 570
510 PRINT TAB(12) "<ESC> TO DELETE--<F2> TO
     TERMINATE--ANY OTHER TO SAVE ";
520 PRINT TAB(32);: I$=INKEY$: IF I$="" THEN 520
530 IF ASC(I$)=27 THEN RD=RD+1: PRINT: PRINT:
     GOTO 420
540 IF ASC(I$)=02 THEN 810
550 PRINT: PRINT
560 CD=CD+D
570 RW=RW+1
580 PRINT #2, Z$
590 GOTO 420
600 REM ** SEARCH & DESTROY VID CONTROL CHARS **
610 D=0
620 TL=LEN(T$)
630 IF TL=0 THEN 800
640 FOR I=1 TO TL
650 C$=MID$(T$, I, 1)
660 T=ASC(C$)
670 IF T>31 AND T<128 THEN 720
680 IF T=8 THEN 720
690 D=D+1
700 PRINT CHR$(26); T; CHR$(25);
710 GOTO 740
720 Z$=Z$+C$
730 PRINT T;
740 NEXT I
750 IF POS(0)=0 THEN 770
760 PRINT
770 L=LEN(Z$)
780 IF LL<L THEN 800
790 Z$=Z$+ STRING$(LL-L, " ")
800 RETURN
810 RESUME 820
820 CLOSE 1: CLOSE 2
830 GOSUB 260
840 IF LL<1 THEN LL$= " UNCHANGED" ELSE
     LL$=STR$(LL)
850 GOSUB 270
860 Z$=F1$+ " CLEANED AND WRITTEN TO "+ F2$+ ",
     RECORD LENGTH"+ LL$
870 PRINT@(0,40-(LEN(Z$)/2)), Z$;
880 PRINT@(1,0), RW; " RECORDS CLEANED";
890 PRINT@(1,27), RD; " RECORDS DELETED";
900 PRINT@(1,54), CD; " CHARACTERS DELETED";
910 PRINT@(23,0),;
920 END
```

SCROLL/CIM

These Hex values should be entered using DEBUG, beginning at Hex address F000:

```
F000  21 08 F0 E5 2A 03 28 E9 45 3E 1B CF C9
```

The command line to save a copy of SCROLL/CIM is:

```
DUMP SCROLL/CIM START=F000, END=F010
```

HOBBYSCOPE (From page 38)

program in HOBBYSCOPE BASIC CODE will be transmitted on that date. Since the two tone system is at frequencies which are more easily transmitted via SW, we feel the success rate might be higher. If successful, this system might be used to serve the majority of users. Further technical details on this code will be available to interested parties on request shortly. Experiments on the amateur bands on both SSB and AM indicate success, and the use of sync pulses in the code ensures minimal disruption during disturbed conditions. Mixing of data obtained on repeat broadcasts is also possible. If the system works it could prove a universal esperanto for mass communication of computer data.

The More Serious Side of the CC

An Inside Look at Software Development

Welcome to the New Year. Considering the time of year, I thought it would be the proper time to give you some perspective on where Radio Shack is coming from and some insight into where we are going with regard to the Color Computer.

We all know that Radio Shack doesn't discuss what projects it is currently working on, and with that in mind, I'll try to talk in generalities.

First, a little background information . . . The Color Computer was designed to fill a consumer need for a simple to use, affordable computer for use in the home environment. The 1981 Radio Shack catalog listed eight software packages (Program Paks) which would work with the Color machine. Seven of these programs were entertainment oriented (games) while the eighth was for home budgeting. The 1982 Computer Catalog shows 24 software packages available (or soon to be available) for the Color machine. Fifteen of these are entertainment oriented (games, again), eight are self-improvement/home management type programs and one is a utility (the screen print program).

A few unbiased viewers, along with some concerned Color Computer owners, have expressed the feeling that the Color Computer lends itself easily to applications other than games and that Radio Shack is not aggressively expanding its horizons in pursuit of that market. On the surface, it could appear that these people are right. I agree that the Color Computer is capable of being lots more than just another game-cartridge machine. But I object to the thought that we are not entering the more "serious" applications market. Well, "where are those other serious applications that it is capable of ?" one might ask. Well, the most I can say right now is, we're working on them. An indication can be found in the RSC-6 catalog at your nearest Radio Shack Computer Center. With programs such as Color Scripsit (an inexpensive word processor for the home at $39.95) or Spectaculator (a what-if planner/spreadsheet, similar to VisiCalc at $39.95), or Investment Analysis (for use with Personal Finance giving yields on investments, etc. at $39.95), or Computer Learning Lab (a series of programs that show learning CAN be FUN at $49.95).

Without "breaking the rules" I can't tell you what we are working towards releasing next; but, one could logically assume that if there is a demand, we'll strive to fill that demand.

Another important factor in releasing software for sale is time. That's right, time, and lots of it. Let's take a mythical software package through development. In the beginning, an idea is developed. This idea can come from me, someone in the software development group, from the store personnel, or from you, the consumer. Once the idea is formulated, the various requirements for the program, in terms of hardware/software necessary are established. At this time, there is a formal set of specifications drawn up. The specifications will include sample screens for text and/or graphics, flow charts on how the program will operate, and an overview of what the program is trying to accomplish. It will also give Merchandising an idea of the cost involved in development, thereby establishing a tentative retail price for the product.

The next step is to decide whether to code the program "in-house" or to contract with an outside software house. To actually develop working, error-free software usually takes several months. To develop good programs, it usually takes about one month to generate 1000 bytes of program code. (The average game Program Pak contains about 4000 bytes of code.) At the same time, the documentation (manual) is being written. Both the program and manual are written from the original specifications.

Once the program code is complete, it goes into testing. The people testing the program check to make sure that the program functions according to the original specifications without any "crashes" during the running of the program. This means that if you press the fire-when-ready button, something should happen, in most cases anyway.The testing group also verifies that the manual has enough information so the user can RUN the package. At this point, any programming errors or omissions are referred to the programmer for correction. Once the corrections have been made, the testing is started all over again from scratch.

Once the testing group has approved the software, it is released to the Merchandising department. Here, a group of software analysts play consumer. They check the program for accuracy, try to destroy or "bomb" the program; they try cheating or anything else that might suit their fancy. They'll try to catch any omissions in the manual for features that are not explained or for something in the manual that the program does not support. At this point, the software is released for duplication and the manual is released for printing.

Duplication in the quantities that Radio Shack uses requires anywhere from 30 to 90 days. The printing of the manual also follows in this same time frame.

As you can see, if everything goes exactly as planned, the software package will take at least 6 months from start to finish. If any problems arise, the timetable is set back. Obviously, game programs are shorter, easier to "debug," plus never have problems with volunteers to test the program. (Sometimes, we can't get them to stop testing the program. Can you believe it ?)

With more serious software, like Color Scripsit or Spectaculator, the controlling program is longer, therefore it takes longer to develop, release, and subsequently bring to market. A prime example is Investment Analysis. We originally anticipated that this program would be available by the end of December. However, due to the depth of involvement of the program and its corresponding length, it has taken us longer than normal to bring it to the market. What was it about "the best laid plans of mice and men ?" Well, maybe now you get the picture.

If you call our Customer Service "Hotline" and ask what plans are in the works for the Color Computer, they'll probably tell you that "what you see is what you get." This is what we've told them to say. If today, I said there would be XYZ software package for the Color Computer ready for sale on January 30, 1982, some people would be angry if it wasn't ready on January 2.

In an attempt to summarize what I've been saying, suffice it to say that we ARE working on more game type programs, some will be Program Paks, some will be cassette-based; we ARE working on disk-based programs for our new Color Disk Drives; we ARE working on more serious applications for the people who want to take better advantage of their machine than just using it for entertainment! The closest time element we can give you is "sometime in the future . . . " But take heart, 'cause we ARE working on it.

Color Computer and Videotex Bugs, Errors and Fixes

Revisions, Enhancements, Etc.

The following is a list of current stock numbers for Computer Software revisions, enhancements and some miscellaneous items. These items can be ordered through your local Radio Shack store. If no price is shown, the item is available at no cost upon proof of purchase of the original program.

COLOR COMPUTER

700-2013 8 Bit Printer Driver for CC (LP VII, VIII)
Driver which allows graphic operation of LP VII and VIII. This program is needed by owners of the Color Computer with the standard 1.0 ROM only.

VIDEOTEX

700-2300 Dow Jones Service for any Videotex
This software is for Videotex owners whose Videotex packages did not contain the Dow Jones Service.

Auto Repeat, Revisited

David W. Gangwisch
10539 Lakemere
Dallas, TX 75238

The long awaited October issue of 'TRS-80 Microcomputer News' contained several pleasant surprises. The increase to 32 pages enhances what was already the single best feature of my Color Computer.

Alexander Benenson's article 'Auto Repeat' contains some very useful information. It helps to open up the mysteries of what is stored where. I would like to point out one addition and one correction, though. The addition concerns the triggers on the joysticks. The trigger on the right joystick reduces all addresses by one and the left decreases them by two. The correction concerns the keys CLEAR and ENTER. In experimenting with the program below I found that ENTER and CLEAR were exchanged on the chart.

```
10 PRINT@0
   : FOR X= 338 TO 345
20 PRINT X; PEEK(X)
30 NEXT X
   : GOTO 10
```

CLEAR affects address 339 and ENTER affects address 338. This program also displays the effects of the joystick triggers.

Shoot-Em Again

J.W. Myers
Route 2, Box 29
Bastrop, LA 71220

I was so intrigued by Mr. Alexander Benenson's article (Auto Repeat) in the October issue of Microcomputer News that I programmed a complete game from it to run on the TRS-80 4K Color Computer.

This game gives you three (3) levels of challenge — 3 hits to win, 6 hits to win and 9 hits to win. After approximately 35 seconds, if you have not won, the blue spaceship starts after your gun position. You have 12 more chances to get the necessary "hits" before he rams you.

I tried to leave his program in place, so my additions can be inserted in a program that has already been taped. The only exceptions are lines 110 and 112, I also deleted the REM lines.

The Program:

```
2 CLS(5)
     : PRINT@ 103, "LEVEL OF CHALLENGE ";
5 PRINT@ 167, "1. BEGINNER--3 HITS ";
10 PRINT@ 231, "2. AMATEUR---6 HITS ";
15 PRINT@ 295, "3. EXPERT----9 HITS ";
20 PRINT@ 359, "TYPE NUMBER WANTED ";
     : INPUT A5
25 IF A5=1 THEN A6=3
30 IF A5=2 THEN A6=6
35 IF A5=3 THEN A6=9
40 S$=CHR$(129)+CHR$(143)+CHR$(143)+CHR$(143)+
     CHR$(130)
50 E$=CHR$(128)+CHR$(128)+CHR$(128)+CHR$(128)+
     CHR$(128)
60 R$=CHR$(177)+CHR$(191)+CHR$(191)+CHR$(191)+
     CHR$(178)
70 O$=CHR$(241)+CHR$(255)+CHR$(255)+CHR$(255)+
     CHR$(242)
80 B$=CHR$(241)+CHR$(251)
     : BE$=CHR$(128)+CHR$(128)
90 BL$=CHR$(161)+CHR$(175)+CHR$(175)+CHR$(175)+
     CHR$(162)
100 A=247
     : CLS(0)
     : T=495
     : PRINT@ T, B$;
     : I1=1
     : I2=-1
     : FS=2
     : P1=0
     : P2=91
110 IF W=A6 THEN 500
112 OT=T
115 V=V+1
130 IF PEEK(343)=A THEN T=T-1
150 IF PEEK(344)=A THEN T=T+1
170 IF T>509 THEN T=480
180 IF T<480 THEN T=509
190 IF OT<>T THEN PRINT@ OT, BE$;
     : PRINT@ T, B$;
210 O1=P1
     : P1=P1+I1
     : IF P1>26 THEN P1=0
     : I1=RND(3)
220 O2=P2
     : P2=P2+I2
     : IF P2<64 THEN P2=91
     : I2=-RND(3)
223 IF V>275 GOSUB 450
230 PRINT@ O1, E$;
     : PRINT@ P1, S$;
240 PRINT@ O2, E$;
     : PRINT@ P2, BL$;
245 IF P2=>492 THEN 600
250 IF F=1 THEN 300
270 IF PEEK(345)=247 THEN F=1
     : FY=30
     : SOUND 1, 1
     : FX=(T-479)*2
280 IF F=0 THEN 110
300 OF=FY
     : FY=FY-FS
     : RESET(FX, OF)
310 IF FY<0 THEN F=0
     : PRINT@ T, B$;
     : GOTO 110
330 IF POINT(FX, FY)=1 THEN P=P1
     : Z$=S$
     : GOTO 380
```

```
340 IF POINT(FX, FY)=3 THEN P=P2
    : Z$=BL$
    : GOTO 380
350 SET(FX, FY, 7)
360 GOTO 110
380 W=W+1
385 FOR N=1 TO 3
    : PRINT@ P, Z$;
390 FOR Q=1 TO 50
    : NEXT
    : PRINT@ P, R$;
    : SOUND 100, 1
400 FOR Q=1 TO 50
    : NEXT
    : PRINT@ P, O$;
    : SOUND 150, 1
410 FOR Q=1 TO 50
    : NEXT
    : SOUND 200, 1
    : NEXT
415 CLS(2)
    : PRINT@ 235, "HITS="; W;
417 FOR Z=1 TO 500
    : NEXT
419 CLS(0)
420 PRINT@ T, B$;
    : F=0
    : GOTO 110
450 I2=3
460 U=U+1
470 RETURN
500 FOR Z=1 TO 100
501 CLS(8)
502 CLS(2)
503 CLS(5)
504 CLS(4)
    : NEXT
505 FOR T=133 TO 1 STEP -5
512 PRINT"GIN-GIN-GIN---YOU---WIN!WIN!WIN
515 SOUND T, 1
520 NEXT
530 GOTO 610
600 FOR Z=1 TO 100
601 CLS(4)
602 CLS(5)
603 CLS(2)
604 CLS(8)
605 NEXT
606 FOR T=200 TO 50 STEP -6
607 PRINT"BOO-BOO-BOO--YOU--LOSE-LOSE-LOSE"
608 SOUND T, 1
609 NEXT
610 FOR Z=1 TO 3000
    : NEXT
615 CLS(5)
    : PRINT@ 233, "TYPE P TO PLAY ";
620 K$=INKEY$
630 IF K$="P" THEN 40
635 W=0
    : U=0
    : V=0
640 GOTO 620
```

Core Editor

John Jamieson
513 Fillmore Street
Herndon, VA 22070

Before purchasing a color computer I did a considerable amount of research to convince myself that its price/performance ratio was best for my interests. After several months of use I am even more pleased with my decision.

Enclosed is a simple core editor that I have used to write machine language routines and to wander around in the BASIC object code. Please include it in your publication if you think others may find it useful.

```
1 ' CORE EDITOR
2 '    ENTER DEC ADDR AFTER ?
3 '    ALL OTHER NUMBERS IN HEX
4 '    UP ARROW STEPS FORWARD THRU MEMORY
5 '    DOWN ARROW STEPS BACK
6 '    LOAD ONE BYTE WITH ENTER
7 '    @ REQUESTS NEW ADDRESS
8 '
9 '
100 DIM NM$(300)
110 DATA NEG, ?, ?, COM, LSR, ?, ROR, ASR, LSL,
     ROL, DEC, ?, INC, TST, JMP, CLR
120 DATA P2, P3, NOP, SYNC, ?, ?, LBRA, LBSR, ?,
     DAA, ORCC, ?, ANDCC, SEX, EXG, TRF
130 DATA BRA, BRN, BHI, BLS, BCC, BCS, BNE, BEQ,
     BVC, BVS, BPL, BMI, BGE, BLT, BGT, BLE
140 DATA LEAX, LEAY, LEAS, LEAU, PSHS, PULS,
     PSHU, PULU, ?, RTS, ABX, RTI, CWAI, MUL, ?,
     SWI
150 DATA NEGA, ?, ?, COMA, LSRA, ?, RORA, ASRA,
     LSLA, ROLA, DECA, ?, INCA, TSTA, ?, CLRA
160 DATA NEGB, ?, ?, COMB, LSRB, ?, RORB, ASRB,
     LSLB, ROLB, DECB, ?, INCB, TSTB, ?, CLRB
170 DATA NEG, ?, ?, COM, LSR, ?, ROR, ASR, LSL,
     ROL, DEC, ?, INC, TST, JMP, CLR
180 DATA NEG, ?, ?, COM, LSR, ?, ROR, ASR, LSL,
     ROL, DEC, ?, INC, TST, JMP, CLR
190 DATA SUBA, CMPA, SBCA, SUBD, ANDA, BITA, LDA,
     ?, EORA, ADCA, ORA, ADDA, CMPX, BSR, LDX, ?
200 DATA SUBA, CMPA, SBCA, SUBD, ANDA, BITA, LDA,
     STA, EORA, ADCA, ORA, ADDA, CMPX, JSR, LDX,
     STX
210 DATA SUBA, CMPA, SBCA, SUBD, ANDA, BITA, LDA,
     STA, EORA, ADCA, ORA, ADDA, CMPX, JSR, LDX,
     STX
220 DATA SUBA, CMPA, SBCA, SUBD, ANDA, BITA, LDA,
     STA, EORA, ADCA, ORA, ADDA, CMPX, JSR, LDX,
     STX
230 DATA SUBB, CMPB, SBCB, ADDD, ANDB, BITB, LDB,
     ?, EORB, ADCB, ORB, ADDB, LDD, ?, LDU, ?
240 DATA SUBB, CMPB, SBCB, ADDD, ANDB, BITB, LDB,
     STB, EORB, ADCB, ORB, ADDB, LDD, STD, LDU,
     STU
250 DATA SUBB, CMPB, SBCB, ADDD, ANDB, BITB, LDB,
     STB, EORB, ADCB, ORB, ADDB, LDD, STD, LDU,
     STU
260 DATA SUBB, CMPB, SBCB, ADDD, ANDB, BITB, LDB,
     STB, EORB, ADCB, ORB, ADDB, LDD, STD, LDU,
     STU
270 FOR K=0 TO 255
280 READ NM$(K)
290 NEXT K
300 INPUT I
310 PRINT HEX$(I); " "; HEX$(PEEK(I)); " ";
     NM$(PEEK(I)); " ";
320 S$="&H"
330 Q$=INKEY$
340 IF Q$="" THEN 330
350 IF Q$="^" THEN I=I+1
    : PRINT CHR$(13);
    : GOTO 310
360 IF Q$=CHR$(10) THEN I=I-1
    : PRINT CHR$(13);
    : GOTO 310
370 IF Q$=CHR$(13) THEN 420
380 IF Q$="@" THEN PRINT CHR$(13);
    : GOTO 300
390 S$=S$+Q$
400 PRINT Q$;
410 GOTO 330
420 POKE I, VAL(S$)
430 I=I+1
440 PRINT CHR$(13);
450 GOTO 310
```

Retirement Income Projection

Use the PC to Find Out How Long Your Nest Egg Will Last

Here it is January, 1982, and it's hard to believe that it has been almost a year and a half since we introduced the Pocket Computer into our family of TRS-80 computers.

By now, many of you have probably purchased the printer and cassette interface to expand the versatility of your PC. So this month I thought I would pass along to you a program which makes good use of the printer. The program is called 'RIPP' for Retirement Income Projection Program and will allow you to see how long your retirement nest egg will last when it is invested at a certain rate of interest and you draw out monthly amounts (which may or may not be escalated each month by a factor for the annual inflation rate) to live on.

The program takes up all but 187 steps in the PC so it will take you a little while to load it in. After you do, (and save a copy to tape) I'm sure you will find it worthwhile.

```
2Ø:U=U+1
     : IF U>C GOSUB 245
     : GOTO 55Ø
3Ø:FOR X=1 TO 12
     : T=A*D
     : B=B*E
     : Y=Y+T
45:W=A
     : A=A+T-B
5Ø:IF A<=Ø GOTO 8Ø
55:Z=Z+B
     : NEXT X
65:GOSUB 2Ø5
     : GOTO 2Ø
8Ø:A=W+T+Z
     : GOSUB 27Ø
     : GOTO 55Ø
1ØØ:"A" CLEAR
     : BEEP 5
11Ø:INPUT "PRINCIPLE SUM? "; A
     : GOTO 12Ø
115:GOTO 11Ø
12Ø:INPUT "ANNUAL INCOME? "; B
     : GOTO 13Ø
125:GOTO 12Ø
13Ø:INPUT "NO. RETIREMENT YEARS? "; C
     : GOTO 14Ø
135:GOTO 13Ø
14Ø:INPUT "ANNUAL INT. RATE? "; D
     : GOTO 15Ø
145:GOTO 14Ø
15Ø:INPUT "INFLATION APR? "; E
165:USING "########.##"
     : PRINT "PRINCIPLE SUM: "
     : PRINT "$"; A
17Ø:PRINT "ANNUAL INCOME: "
     : PRINT "$"; B
     : B=B/12
175:USING "###"
     : PRINT "NO. OF YEARS: "; C
18Ø:USING "###.##"
     : PRINT "ANNUAL INT RATE: "; D; "%"
     : D=D/12ØØ
185:PRINT "INFLATION APR:  "; E; "%"
     : E=E/12ØØ+1
19Ø:GOTO 2Ø
2ØØ:PRINT "----------------"
     : RETURN
2Ø5:GOSUB 2ØØ
21Ø:USING "###"
     : PRINT "YEAR:"; U
215:USING "########.##"
     : PRINT "ANNUAL INCOME: "
     : PRINT "$"; Z
     : Z=Z-Y
22Ø:PRINT "FROM PRINCIPLE: "
     : PRINT "$"; Z
     : Z=Ø
225:PRINT "FROM INTEREST: "
     : PRINT "$"; Y
     : Y=Ø
23Ø:PRINT "ENDING BALANCE: "
     : PRINT "$"; A
235: RETURN
245:GOSUB 2ØØ
25Ø:PRINT "END PROJECTION!"
     : RETURN
27Ø:GOSUB 2ØØ
275:USING "###"
     : PRINT "FUNDS EXHAUSTED"
     : PRINT "IN MONTH: "; X
     : PRINT "OF YEAR: "; U
     : Z=Z-Y
28Ø:USING "######.##"
     : PRINT"TOTAL PAID(P+I): "
     : PRINT "$"; A
285:PRINT "INTEREST PAID:"
     : PRINT "$"; Y
29Ø:RETURN
5ØØ:" "PRINT "**** RIPP ****"
    PRINT "*RETIREMENT"
    PRINT "*PROJECTION"
51Ø:PRINT "*PROGRAM"
     : GOSUB 2ØØ
     : PRINT "THE TRS-8Ø"
     : PRINT "POCKET COMPUTER"
     : PRINT "WILL SHOW HOW"
515:PRINT "AN AMOUNT OF"
52Ø:PRINT "MONEY WILL LAST"
     : PRINT "WHEN INVESTED AS"
     : PRINT "AN ANNUITY FOR"
     : PRINT "INCOME, WITH OR"
53Ø:PRINT "WITHOUT AN"
     : PRINT "INFLATION RATE"
     : PRINT "ESCALATING THE"
     : PRINT "MONTHLY PAYMENT."
54Ø:GOSUB 2ØØ
     : GOTO 1ØØ
55Ø:GOSUB 2ØØ
     : PRINT "PRESS SHFT A"
     : PRINT "TO DO ANOTHER"
     : PRINT "PROJECTION."
     : GOSUB 2ØØ
     : END
```

To run the program, make sure the PC is in the DEF mode and connected to the PC printer (which should be turned on and initialized in the print mode.)

Select <SHFT> <SPC> and the printer will begin printing out the name of the program and a short description of what it does. When the printer stops, the display will show a request for 'Principle Sum'. This is the amount you have saved for your retirement income and the program can handle an amount up to 5 million dollars. Type in an amount (without commas) and press <ENTER>.

The next request is for the annual income you desire. For example, if you wish to receive $2500 per month, you would type in 2500*12 and press <ENTER>. The PC will convert this to $30000 per year. After this, you will be asked for the number of years you want this money to last. Then, the PC will want to know the annual interest your money will be earning

and this must be in a form such as 15.5 percent. The last request is for the annual inflation rate.

If you want to have your monthly income keep up with some inflation rate which you are going to estimate will occur on the average each year over the retirement period, enter a number here in the same form as the previous interest rate. The PC will automatically adjust your income upward each month by one twelfth of this rate, during the computation. If you do not want to use this feature, simply type zero and press <ENTER>.

At this point, the PC will print out a record of the data you entered and begin calculating each year's data. The figures for annual income, amount paid from principle, amount paid from interest and the balance at the end of the year will be printed for each year until either the end of the retirement period which you entered, or the principle sum is exhausted.

If the principle is exhausted before the end of the retirement period, the PC will tell you in which month of what year the funds ran out and the amount paid from principle and interest in that year. If there are funds remaining at the end of the period, the PC will show this amount as the ending balance in the last year.

By the way, you can use the 'Interest and Annuities' module of the Personal Finance package (Cat. No. 26-3518) to figure out how long it will take to accumulate a specific principle sum for retirement.

If you should get an ERROR 6 on the PC before the program finishes, it probably means that one of the amounts the PC was trying to display is greater than $9,999,999.99. In this case, it is most likely that the principle sum has grown larger each year because the interest rate was high and/or you did not put in an inflation factor. To recover from this, press the red <CL> button and then select <SHFT> <A> to re-enter the data.

To get you started, here is some sample data for the program:

PRINCIPLE SUM - 800000
ANNUAL INCOME - 40000
NO. OF YEARS - 15
ANNUAL INT RATE - 10
INFLATION APR - 12

You will notice that for the first eight years the amount paid from the principle is negative, that is the annual income is supported entirely from the earned interest and the principle is growing larger. However, in the ninth year this situation reverses due to the effect of the inflation factor and the principle begins to decrease each year to the point where at the end of the projected retirement period, the principle is about half of what it was when you started.

The program takes about 35 seconds to calculate and print each year's data, so don't be impatient, it's doing a lot of number crunching. Try experimenting with different annual incomes and interest rates to really see the effect on the principle and how long it will last. I think you will be surprised at some of the results from this program and I'm sure you will have fun with it.

That's it for this month, hope you all had an enjoyable holiday season . . . until next month . . . more Pocket Power!

Pocket Computer Bugs, Errors and Fixes

Revisions, Enhancements, Etc.

The following is the stock number for a Pocket Computer Software revision. This item can be ordered through your local Radio Shack store and is available at no cost upon proof of purchase of the original program.

700-2400 Real Estate Tape 2 (26-3510 Ver 1.1)
Corrects part B of REAL4 where the total amoritization was not including the first month's payment.

One Armed Dime Bandit

D.W. Bond
301 Grove Avenue
Waynesboro, VA

Having seen very few programs specifically applicable to your Pocket Computer published in the literature, you may be interested in the following adaptation of 'Program 5' from page 136 of your publication #62-2015. This program is run in the DEF mode.

(WANNA BET . . . without going to Las Vegas?)

```
1Ø  " "
     : AREAD A
     : BEEP 1
     : PAUSE "<ONE-ARM DIME BANDIT>"
15 PAUSE "3-OF-A-KIND WINS $Ø.6Ø"
2Ø A$(3Ø)="<...>"
     : A$(31)="<APPLE>"
     : A$(32)="<GRAPE>"
     : A$(33)="<LEMON>"
25 INPUT "NO. DIMES PURCHASED=?"; M
     : IF M=Ø GOTO 15Ø
3Ø R=Ø
     : S=Ø
     : T=Ø
     : M=M*.1
     : E=Exp8+1
     : USING "##.##"
35 PAUSE "YOU NOW HAVE $"; M
     : IF M=Ø GOTO 14Ø
4Ø IF A=9 GOTO 5Ø
45 INPUT "SPIN-THE-DRUMS (ENTER)"; A
     : IF A=Ø GOTO 15Ø
5Ø M=M-.1
     : V=Ø
55 FOR B=1 TO 3
6Ø X=439147+X
     : C=23*X
     : X=C-E*INT(C/E)
65 Y=INT(X/Exp7)
     : IF Y=9 GOTO 6Ø
7Ø PAUSE A$(3Ø); A$(3Ø); A$(3Ø)
75 V=3Ø+(Y<3)+(Y<6)+(Y<9)
8Ø IF B=1 LET H=V
85 IF B=2 LET J=V
9Ø IF B=3 LET K=V
95 R=R+(V=31)
     : S=S+(V=32)
     : T=T+(V=33)
1ØØ NEXT B
11Ø BEEP 1
     : PAUSE A$(H); A$(3Ø); A$(3Ø)
115 PAUSE A$(H); A$(J); A$(3Ø)
12Ø PAUSE A$(H); A$(J); A$(K)
     : Q=Ø
125 Q=(R=3)+(S=3)+(T=3)
13Ø IF Q=1 LET M=M+.6
     : PAUSE "GREAT...YOU WIN"
     : GOTO 35
135 PAUSE "SORRY...YOU LOSE"
     : GOTO 35
14Ø INPUT "TRY AGAIN ? (YES=1)";W
145 IF W=1 LET W=Ø
     : GOTO 25
15Ø PRINT "HOUSE OWES YOU $"; M
     : END
```

Any single digit from 1 to 8 will allow the program to run with a stop at line 45 each round. This stop may be eliminated by using the digit 9 with the AREAD input; the program will then run until all the players money has been used up.


TRS-80
General Ledger
Payroll

Radio Shack®

TRS-80 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.