

TRS-80[®]

Volume 4 Issue 1 JULY/AUGUST, 1982

Price \$1.50

Microcomputer News

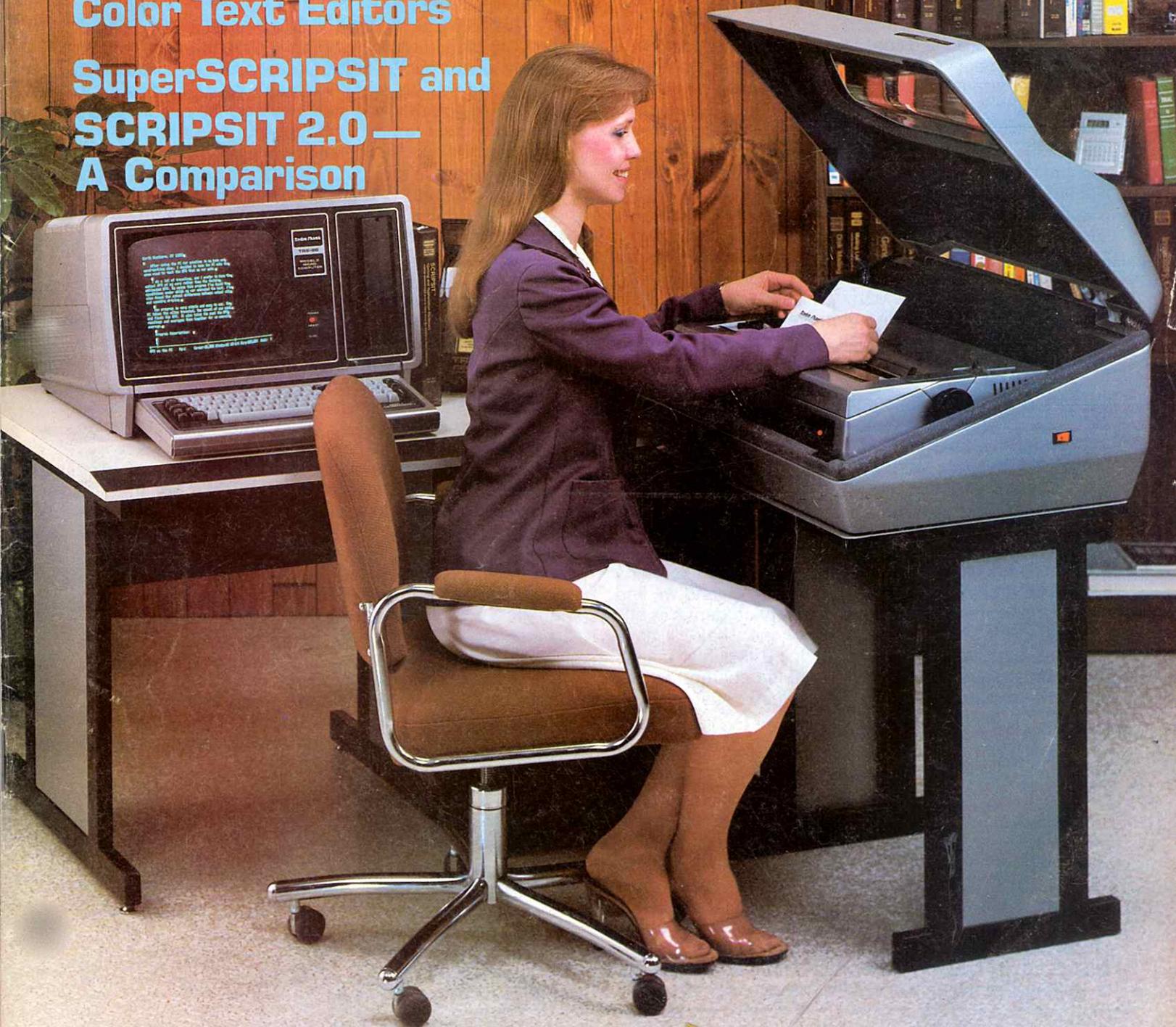
Information Published for TRS-80 Users.

Word Processing With the Keiths

The Sieve of Eratosthenes

Color Text Editors

**SuperSCRIPSIT and
SCRIPSIT 2.0 —
A Comparison**



Fort Worth Scene



The May issue of 80-US contained the following article by Radio Shack's Ed Juge. Since there has been a lot of misinformation and since the policy has recently been restated to our own people, I thought you might like the information; here is Ed's May "Tandy Topics" article, reprinted with permission from 80-US:

SERVICE POLICIES

You've heard me discuss many times, the fact that we can't repair modified TRS-80s . . . we reserve the right to void your warranty if a unit has been opened or modified . . . we can't be responsible for non-Radio Shack parts installed, etc. Officially, our policies have been pretty much as stated. Unofficially, we've allowed our service center personnel to determine when special cases warranted special handling. We ended up with some inconsistencies, though, where one repair facility would agree to work on a unit after another facility had refused. The good news is that we think we've learned quite a bit about what we can do effectively, and what we can't.

So about the first of March, a letter went out to all of our Service Centers, giving them specific guidelines on what to do with TRS-80's which aren't completely "stock." And since it will undoubtedly make a lot of you very happy . . . and since I think it shows a real effort on the part of our Support Operations management to increase our level of service, I'd like to tell you about these guidelines.

Up front, let me say that there are some "maximum parts/labor charges" for certain completely stock equipment. Those charges will not apply to units which are "non-stock" in any way. With that caveat, we'll proceed . . .

First of all, let's look at the case of RAM chips. Those of you with "stock" TRS-80's containing Radio Shack chips . . . we will replace only the defective chips we find, not a whole set. If you've installed "foreign" RAMs, and we find one or more of them defective, the service center is still required to replace a complete set or none. They will, however, call and give you the option of doing it yourself or having us do it. There will be a checkout fee charged for our time, even though no service is performed.

FOREIGN DISK DRIVES

We have said many times that we can't work on Model III's which contain non-Radio Shack drives. Well, our folks tell us that they will accept such units for service . . . partly! What that means is that if the drives work properly, we will simply overlook them. If the drives are at fault, we will disconnect them, check or repair the "stock TRS-80" portion of your computer, and let you handle any drive problem with the vendor or the dealer who installed them.

I still issue a strong warning to the non-technical reader: This leaves you in the dangerous position I've described before, that Radio Shack will guarantee the operation of our equipment, as the drive vendor may guar-

antee his drives. But you'd better be sure someone guarantees that they will work together, unless you have the knowledge and equipment to do it yourself. Now my sales pitch . . . a few freight bills or "figure it out as we go" repair bills by a third party can more than eat up your savings.

MODIFICATIONS

This one's a bit harder. If a modification doesn't appear to be hurting anything, we'll disregard it. If we can't properly check out the unit, or if the modification seems to be causing the problem, we'll ask your permission to remove it. If you decline, you'll pay only the checkout fee. Otherwise, we'll remove it, return the parts to you, and effect repairs.

Now, there is one big "gotcha" . . . when there are "cuts and jumpers," or damage to the board such that our repair people back here in Fort Worth wouldn't accept it for rebuilding on an exchange basis. If your computer qualifies, then your only options may be to repair it yourself, or authorize us to install a new circuit board on a non-exchange price basis.

Our repair people will always return any non-Radio Shack parts, whether you request them or not. We've always done this if you asked when you brought the unit to us, but some of you asked only when you picked your computer up . . . after the parts had been thrown away or mixed up with other defectives to the point they could not be identified.

There are a couple of other clauses in the new policy that you should be aware of. If our people feel that the computer poses any kind of safety hazard to the service technician, they may refuse to service it at all. And our after-service warranty will specifically exclude the operation of any non-Radio Shack parts or modifications which we leave installed. Also, TRS-80 equipment containing non-Radio Shack parts or modifications cannot be covered by our Maintenance Agreements.

So having opened, modified, or installed your own parts no longer automatically excludes Radio Shack service on your TRS-80. It can, however, cause that service to cost you more; and a trip to our Repair Center will not necessarily assure a working computer if your non-stock parts are faulty.

* * * * *

DATE CHANGE

As you may have noticed, this issue of TRS-80 Microcomputer News is dated July/August rather than just July. This change was made to make our dating method conform more closely with your expectations. We have received several complaints that the News always arrives at the end of the month rather than the beginning. This change will result in the issues getting to you at the beginning of the month they are dated.

This change will not affect the actual number of issues (free or paid) which you will receive (this issue counts as one issue, not two). Our mailing list system counts the number of issues we send you, without regard for when they are sent. So, if you are entitled to six free issues, you will still get six free issues. If you have paid for twelve issues, you will get twelve issues.

Reproduction or use without express written permission from Tandy Corporation of any portion of the Microcomputer News is prohibited. Permission is specifically granted to individuals to use or reproduce material for their personal non-commercial use. Reprint permission for all material (other than William Barden's article) with notice of source is also specifically granted to non-profit clubs, organizations, educational institutions, and newsletters.

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

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

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

Back issues of Microcomputer News prior to January 1981 are available through your local Radio Shack store as stock number 26-2115 (Suggested Retail Price \$4.95 for the set). Back issues of 1981 copies are 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 for publication should be submitted on magnetic media (tape, disk, or CompuServe). If you submit material on tape or disk and it is accepted for publication, we will send you two cassettes or diskettes for each one you sent us. Cassettes will come from our box of mixed blank cassettes. If you submit material on CompuServe, and we think we may use the material, we will extend your Microcomputer News subscription by six months for each article accepted.

If you are submitting material over CompuServe, please include your name and address or your subscription number so we can find you.

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 70007535.

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.

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 enhance readability, and we separate each program statement onto a separate line. While these techniques increase program readability, they also require more memory, and may execute more slowly than the original program did.

Trademark Credits

CompuServe™	CompuServe, Inc.
Dow Jones™	
NEWS/RETRIEVAL	
Service*	Dow Jones & Co., Inc.
Time Manager™	Image Producers
VisiCalc*	VisiCorp, Inc.
Program Pak™	Tandy Corporation
SCRIPSIT™	Tandy Corporation
TRSDOS™	Tandy Corporation
TRS-80*	Tandy Corporation

Contents:		
Assembly Language Programming		8
by William Barden, Jr.		
Color Computer		
Bugs, Errors, and Fixes		
Scripsit (26-3105)	46	Spectaculator (26-3104) 46
Programs		
Asteroids by Richard Zepp	46	Super Text by Matthew McGovern 43
Fire Works by Franklin Pigounias	47	Text Editor by Brian Lievallen 43
Scripsit and the LP VIII		
by Garrett McGowan	45	
Computer Customer Service		
More about Scripsit 2.0		12
Data Bases		
CompuServe		14
New Menu System		
Racing Information		15
Dow Jones		16
Movie Reviews		
Profile		19
Merging Profile with Scripsit		
VisiCalc		23
Additional Use of @LOOKUP by Lawrence Scanlon		
Educational Products		
Compress Graphics Using DEBUG by Dennis Tanner		34
Computers for Neophytes		15
Network 3 — New Capabilities for the Classroom		32
Feature Story		
Word Processing with the Keiths		5
Fort Worth Scene		
General Interest		2
Micro Research Problem		17
Model I/III		
Bugs, Errors, and Fixes		
Accounts Payable (26-1554)	30	Disk Scripsit (26-1563) 30
Accounts Receivable (26-1555)	30	Manufacturing Inventory Control (26-1559) 30
Advanced Statistical Analysis (26-1705)	31	Medical Office Systems (26-1568) 30
Cassette Portfolio (26-1506)	29	Stockpak (26-1507) 30
COBOL (26-2203)	31	
Disk Payroll (26-1556)	30	
Product Line Manager's Page		25
SuperSCRIPSIT and SCRIPSIT 2.0 — A Comparison		
Programs		
Poor Man's Editor by Dallas Powell		35
Model II		
Bugs, Errors, and Fixes		
Accounts Payable (26-4505)	38	Statistical Analysis (26-4540) 39
Inventory Control (26-4502)	38	TRSDOS (26-4910) 39
Scripsit 2.0 (26-4531)	38	TRSDOS-HD (26-4150) 38
Product Line Manager's Page		36
Scripsit Ideas (XERCH by Peter Beeson, Underscore by J.G. Lowder)		
Medical Office System		37
Programs		
More Scripsit Ideas by C. M. Mendenhall		37
More Computer Clubs		
Notes on Previous Newsletters		15
March 1982		22
Peripherals		
Product Line Manager's Page		
Using the Digitizer to Calculate Area Measurement and Curve Integration		18
Pocket Computer		
Programs		
Convert 35MM to 16MM by Robert Edwards		11
Cost Comparison by Vernon Scally		41
Factorials by Thomas Cox		41
Measurement Conversion by Frederic Bishop		40
Mileage Calculations by Alan Leiwant		42
MPG on PC by Bruce Taub		42
Projectile Trajectory by Larry Mermelstein		40
Resistor Values by Robert Edwards		41
View From the 7th Floor by Jon Shirley		4

Prices shown in TRS-80 MICROCOMPUTER NEWS are in U.S. Funds.

View from the 7th Floor

by Jon Shirley,
Vice President,
Radio Shack Computer Merchandising

I am told that this is a word processing issue. I am a big user of Scripsit both at home and in the office, and if any of you readers have a TRS-80 without any word processing, you are missing a great tool. (All right, all you Pocket Computer owners, I know there is no Scripsit for you.) With the issuance of Scripsit for the Color Computer all of you can now enjoy WP. If you have not ever tried WP, run to our nearest store and try a little hands-on.

At its most simple, WP will let you hack away at the keyboard with one or two fingers, make lots of errors and still produce a perfect printed document. At its most complex (which is still easy to use), it will let you create form letters, fill in the blanks automatically, and print them out unattended. WP is absolutely the best thing that ever happened since the typewriter; may it fade away.

One of the most useful features of Model II Scripsit is user-defined keys. This lets you define a key stroke sequence (escape 1 through 0 and shifted 1 through 0) to do anything that you could do from the keyboard. I use this feature a great deal to hold words or phrases that I use often. For example, "Radio Shack TRS-80 Model II" is escape 3 on some of my disks. But you can also store commands in the user key sequence. For example, a user key could move the cursor a certain number of spaces or insert blanks.

The user key can insert both text and commands, so a key could insert an often-used word and the commands to make it underlined. The keys can also call other user keys to nest operations. One example of this use would be to do a phrase built up of smaller phrases stored in other keys. Each key is limited to 255 keystrokes.

One application for the user key is for the secretary who handles a lot of correspondence. A single user key could hold a sequence such as **<ENTER> <ENTER> <ENTER> <TAB>** Sincerely, **<ENTER> <ENTER> <TAB>** Mr. Phil Alphabet **<ENTER> <TAB>** Vice President **<ENTER>**. That sequence will finish a letter. The sequence could be longer and finish with the initials of the secretary at the lower left of the letter. If one individual does letters for several different people each one would be assigned a different user key.

Another application is in editing a letter. If you had something to insert in a lot of places that could not be accessed by the global search command, you could have a sequence that invoked insert, put in the new text and a space, then hit delete to end the sequence. All this is pretty simple stuff. Experimentation can lead you to some very complex routines.

OFFICE AUTOMATION SHOW AND CONFERENCE

I recently attended the Office Automation Show and Conference in San Francisco. All the big names were there all with their answers for the computerized, paper-less office of the future. They all seemed to be offering the same stuff, and they all seemed to have the same problem—their stuff is not selling. Offices are getting automation, but it's personal computers that are doing the automating. The reason for this is that personal computers are relatively cheap, that they are reasonably user friendly, and that there is enough good software to make them cost justified.

It's a little crazy to see all those big companies with their XXXnet (everyone has one), when the hookup alone costs more than most microcomputers. To make matters worse, some require workstations that sell for over \$15,000 and printers that sell at over \$40,000. So I will go out on a limb and predict that the revolution in the office will arrive through the back door with zillions of little computers sneaking in while the data processing department tries to decide which "net" to buy that won't be obsolete next year.

A GLIMPSE OF THE FUTURE?

I did see one really neat product and while I usually do not talk about other equipment, I will pass this one on to you. It is a small portable computer with all this: 16 bit processor, 256K RAM, 256K bubble memory, 320x240 flat screen display, nice 57-key keyboard, built-in 300/1200 baud direct connect auto answer/auto dial modem, and IEEE 488, RS232 interfaces. When I said small I meant it, 15" by 11½" by 2"! (Yes, really, 2" thick). Is it a dream? No, it's very real and has only one small drawback; it's about \$8,500 without software. I must admit that this is the first computer of another brand that I really wanted to buy. I have no idea why, other than it looked so neat. However . . . at that price I think I'd better wait.

What this little product does show is the future. Flat screen displays with speed and resolution will come down in cost. I do not think bubble memory will, but CMOS RAM will and solve that problem. One thing about this business, it is not boring. Evolution and revolution are the name of the game, and we can all look forward to some real science fiction stuff in the years ahead. Just think about having VisiCalc on your 32 bit wrist watch computer. On the other hand, my old Model I has gone well over a year now with no failures and no CRC errors, despite its frequent use so I guess I can wait a little longer for the future.

Until next month.

Word Processing with the Keiths

Kathleen Keith of Rineyville, Kentucky, is a professional writer who thinks that Scripsit is fantastic. There are two signs in her computer room. The first hangs over a souvenir Bible page that she obtained in 1955 on the 500th anniversary of Gutenberg's development of movable type. It says:

THE FIRST BREAKTHROUGH FOR LITERACY
AND PRINTING — MOVABLE TYPE

The second sign, which hangs over her 64K two disk Model II and Daisy Wheel printer, says:

THE SECOND BREAKTHROUGH FOR LITERACY
AND WRITING — WORD PROCESSOR

HER WRITING PROCEDURE

"As a writer I have been following this procedure in using the word processor. I CREATE a document and type this episode for the six to ten pages I can do that day, usually stopping in the middle of some action. When I am finished for the day I COPY it onto another diskette and I also PRINT it out on the Daisy Wheel II single spaced, to file away. I guess writers are used to the hard copy and can't yet do without it. The next day I go back through what I had typed and correct all the typographical and spelling errors and grammatical boo-boos and continue writing.

"Many of the publishers say that because of increasing inflation, the money offered to writers is being whittled down. With the opportunities provided by the marvels of the data that the writer has put onto the magnetic disk, this can eliminate a highly paid human typesetter, and the necessity for me to proofread the galley proofs that he or she has produced . . ."

WHAT SCRIPSIT HAS MEANT TO ME — A SUMMARY

"Not many of my writer friends around here have heard of word processors for writers. At a recent writers conference held in Kentucky, I offered this summary of what my word processor has meant to me.

"This spring my husband and I had returned from an extensive trip through Mexico gathering background material for me to prepare my novel on the culture of the Mayan peoples before the Spanish came. In my research I had read a number of theses on the subject. I found that many times in the first pages of the thesis, after giving recognition to all the experts who had assisted the writer, there were many dedication pages which read in essence: 'To my Wife, the Typist.' Well, I told the group that I haven't got a wife to be a typist. (Laughter from the audience) While I don't mind typing the first draft of my work in the flush of creation, what I hate is retyping everything after I've edited it, especially the good parts that really didn't need any reworking. (General nods of agreement) I told them that it was only after considerable soul searching that we finally

decided to get the Daisy Wheel II Printer despite the additional cost. This was simply because it was the only printer that produced typewriter quality print. This is absolutely necessary to have in order to present a salable package to a publisher. Our philosophy of purchasing the entire word processing system was that no one would blink an eye if we purchased a new car today. The word processing system (computer and all!) cost less than a new car so I am continuing to drive my car with 111,000 miles on it.

"I went on to explain that my thought processes have been freed from thinking about the mechanics of what's happening when I type, and my fingers can almost keep up with what I am thinking. My husband says I type three times as fast as the electric typewriter and it's reassuring not to hear that typewriter noisily banging away on the paper. Trying to explain why I needed the additional disc drive to copy the discs was more difficult but everyone was curious about the whole word processing idea."

KATHY KEITH'S COMMANDMENTS FOR A HOME COMPUTER ROOM

- I Thou shalt not set thy cup of coffee on thy Computer Desk nor any glass with any liquid in it.
- II Thou shalt not bring anything with a magnetic field into thy Computer Room. Thy bulk eraser shall be kept in another room or 50 feet away from all disks or computer.
- III Thou shalt make two BACKUP copies of any software thou dost purchase before thou shalt start to play with it.
- IV Thou shalt not press the Reset switch whilst thou hast a SCRIPSIT document open in thy computer's drive.
- V Thou shalt not let thy cat take a nap with thee in thy computer room, in particular if the cat hast longhair.
- VI Thou shalt keep thy windows and door closed for temperature and humidity control. Thou shalt vacuum thy computer room regularly and immediately spray sensitive areas with anti-static spray. (Nylon rug and video screen.)
- VII Thou shalt take all measures to keep dust out of thy computer, vacuuming out floor vents for thy heating system, including return air vents. Thou shalt examine the filter on thy furnace and replace at the slightest shadow. And if thou hast a wood burning stove in thy house, thou shalt filter incoming air to thy computer room and use such additional mechanical devices as an air cleaner near thy computer's air vents.
- VIII Thou shalt not let any Ham operate his Amateur Radio Walkie Talkie in thy computer room.
- IX Thou shalt cover thy computer and printer with dust covers when not in use, every day!
- X Thou shalt not hit the switch with thy knee on thine additional disk drive, for if thou dost turn it off, thy disk

will be afflicted with glitches from which it cannot be cured.

A TWO COMPUTER FAMILY!

A few weeks after we received Mrs. Keith's letter her equally ardent TRS-80 user husband, Gilbert Keith, amused and delighted us with the following.

HAPPINESS IS HAVING YOUR OWN PRINTER

"Happy day when the little LP VIII arrived. Waiting for it had been a trauma and an eternity. The joy and pleasure it has given me since then is not counted in dollars and cents, as the pleasure it affords me and the fact it talks to me is worth all the anxious days spent in waiting for it.

"After my wife Kathy had installed her Model II, I knew I just had to have the Model III so I started looking for a printer. Checking out all the different ones on the market was a chore. None seemed to be just what I wanted. Maybe being spoiled by Kathy's Daisy Wheel II did it. The only solution was to purchase the LP VII and wait to see what would be on the market later that I could be happy with. With Scripsit, Mailing List and VisiCalc, the limitations of the LP VII became apparent as the days passed by."

ANOTHER ADDITION TO THE FAMILY

"I don't know how I missed the LP VIII in all of my checking out of the printers. At the Radcliff, Kentucky, Radio Shack store one day, the LP VIII was called to my attention and you know what? The order was placed that day. Just reading about it in the catalog was enough to convince me. I later found out my gut feeling was right. Two days later the Line Printer VII was sold, leaving me without a printer. I was in misery. Being told the waiting time for the LP VIII was to be one month, I called several stores, even the Tandy store in Ft. Worth, to see if a LP VIII was available. One day, at the local store I saw the UPS truck stopping and I went out to see what was in it. I was ecstatic as there were three LP VIII's on the truck designated for the local store. With no further ado I grabbed one, and with coattails flying, headed for my car. 'BABY' had found a new home."

AN INCOMPATIBLE BABY?

"First, I'm not a programmer and understand very little about DOS and BASIC. But I have spent many happy hours seeing what I could do with 'BABY.' My main use for the computer is Scripsit. When the local computer center in Louisville was asked if 'Baby's' different type styles could be used with Scripsit, I was informed that they had not had time to check it out yet. The more I thought about it, the more it bugged me. One night while watching a movie on TV, my mind was in a whirl. There had to be a way! Surely Scripsit and 'BABY' were compatible. Suddenly, it came to me and I left Kathy watching the movie. I rushed to my computer room and started to work.

"It wasn't long before my old letters and writings were coming in all different shapes and forms. I was able to print two columns on one paper. Weird, same letter on the same page side by side. I'm sure SuperSCRIPSIT will be able to do all this and much more. (Editor's note: How right he is!)

"I will show the way it is accomplished. Now all I have to do is to wait for SuperSCRIPSIT and the dictionary. It will not be as hard, as I have the 'best' computer and printer I could ever want. . . ."

Using Scripsit And The LP VIII Type Styles

Gilbert Keith
Rineyville, Kentucky

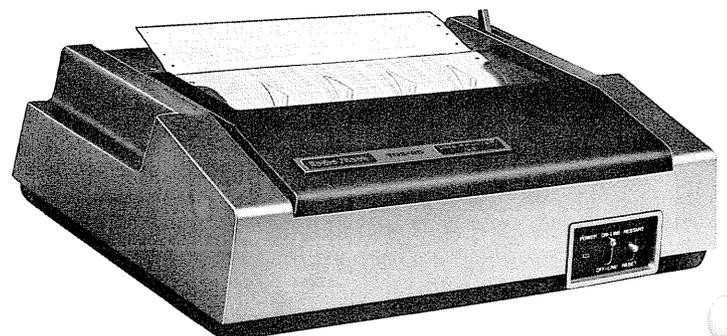
First, we will use the SCRIPSIT program entirely. After you have written and saved your document on Scripsit, decide what style of type you want to print it in. There are four you can use (Monospaced, Proportional, Condensed Monospaced, Elongated).

Boot up the SCRIPSIT diskette and at TRSDOS Ready type BASIC (**ENTER**). When READY appears, type the following:

```
10 LPRINT CHR$(27);CHR$(N);"!"
```

N is the number of the type you want. Then press (**ENTER**). With the printer on-line, type RUN. The printer should make a little grunt and do nothing. Next type: CMD"S" and press (**ENTER**). You should be back to TRSDOS. Type: SCRIPSIT (**ENTER**) and you are ready to load any document you want to print. Remember, you must set the format line for the type style you are printing. The "LM" and the "RM" must be set. For printing condensed with two columns on a page, first I set the format line like this: >PL=66,LM=15,TM=3,BM=60,RM=60 (**ENTER**). For printing, I roll back the paper to the top of the page and reformat thus >PL=66,LM=70,TM=3,BM=60,RM=120. When printed you should have two columns on the same page. Try this out on some scratch paper and when you find out the right format line, SAVE it for that type of print. You should have a lot of fun experimenting with SCRIPSIT and the LP VIII. I know I have . . .

```
10 LPRINT CHR$(27);(20);"!"
30 LPRINT CHR$(27);CHR$(14);"LARGE"
50 LPRINT CHR$(27);CHR$(15);"SMALL"
70 LPRINT CHR$(27);CHR$(17);"PROPORTIONAL"
90 LPRINT CHR$(27);CHR$(20);"CONDENSED"
110 LPRINT CHR$(27);CHR$(19);"ORDINARY"
130 LPRINT CHR$(08);CHR$(143);"BOLD LETTERS"
150 LPRINT CHR$(15);"UNDERLINE"
170 LPRINT CHR$(14);"WITHOUT UNDERLINE"
190 LPRINT "BOLD LETTERS"; CHR$(08); CHR$(143);
    "BOLD LETTERS"
```



Kathleen Keith's Scripsit Backup System

These four steps are required to set up an effective Word Processing System to give peace of mind to users and insure against loss of documents along the way.

This set of 14 (or 16) disks will be enough Backup copies of SCRIPSIT and the Dictionary. Give Scripsit disks a name for their intended purpose.

A writer uses 2 disks called NOVEL. Day to day LETTERS are a second set. (It may also be necessary to create a set of BUSINESS disks).

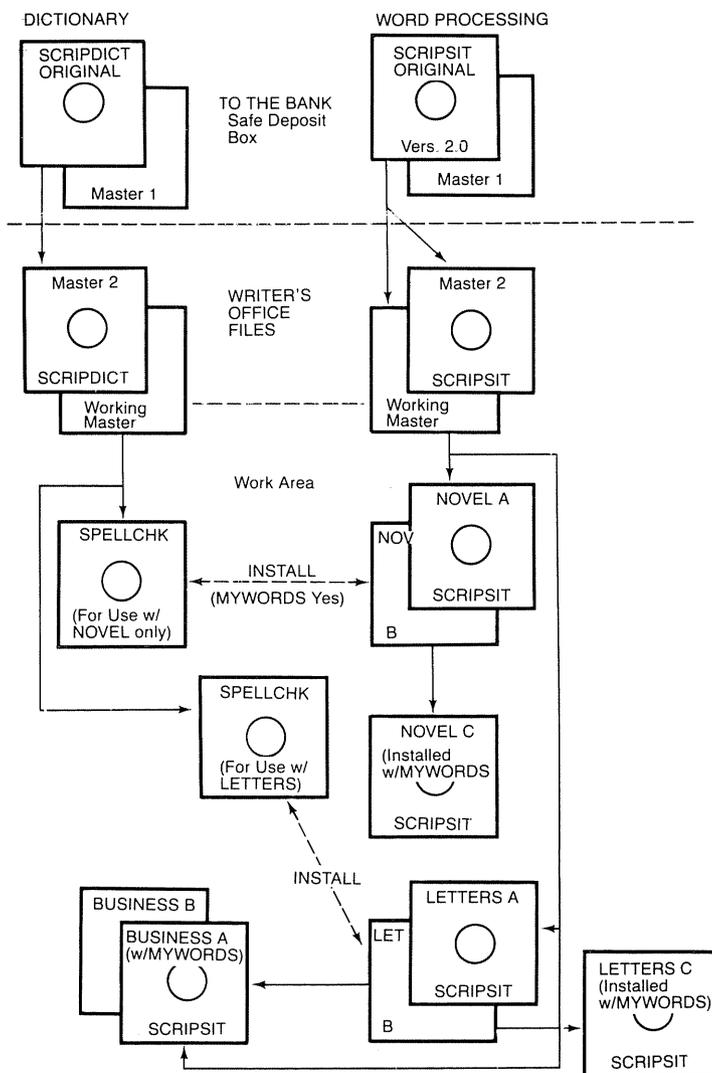
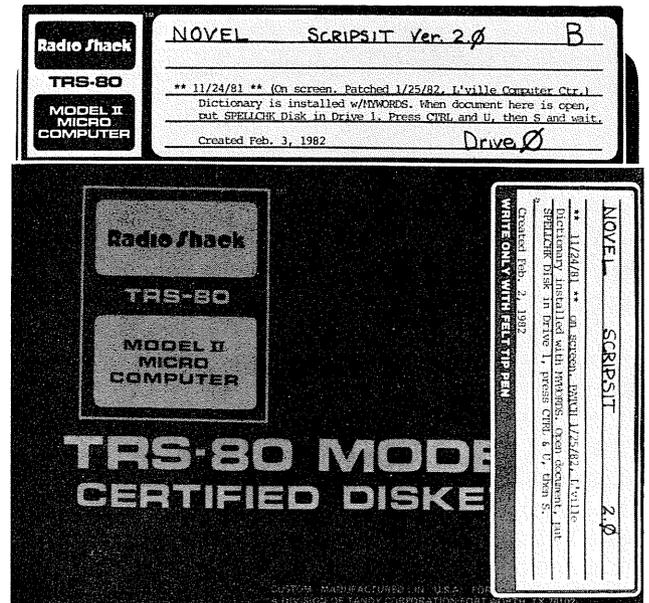
4. When Dictionary (now called SPELLCHK) has been installed on NOVEL Scripsit disk B, having chosen the MYWORDS Option, make a Backup of this NOVEL disk. (Before wiping NOVEL A, check that all documents are copied onto NOVEL B. Then it can be reused in this system.)

Label SPELLCHK disk to be used with NOVEL Scripsit ONLY.

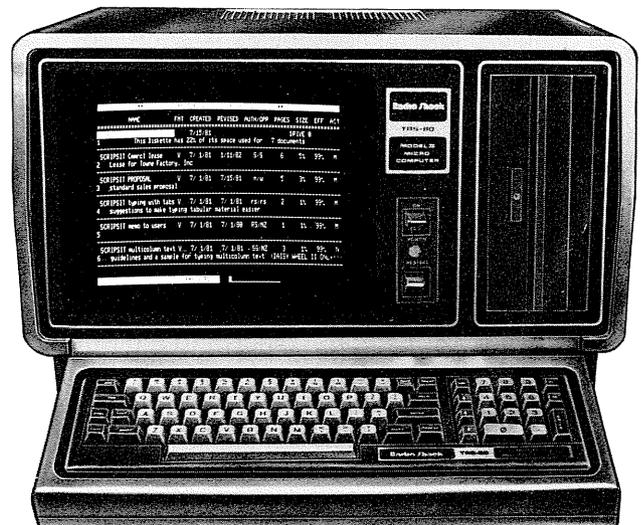
Make separate SPELLCHK Dictionary for use with LETTERS Scripsit disks. Make backup of LETTERS B, and check that all documents on LETTERS A disk have been copied onto LETTERS B.

If deemed necessary, make up Scripsit disks for BUSINESS also, but SPELLCHK Dictionary for LETTERS can be used with it.

Label all disks carefully!



A set of disks are put into a soft-sided binder to protect them. This binder includes two SCRIPSIT disks which were named NOVEL when they were backed up but carry written labels of A and B also. The other disk in this binder is the Spelling and Hyphenation Dictionary called SPELLCHK which was installed on a NOVEL Scripsit Disk. This SPELLCHK will have the MYWORDS Option which will include the words unique to this Novel. (These words include geographic names and foreign words.)



Sieve of Eratosthenes

by William Barden, Jr.
© William Barden, Jr. 1982

What do the ancient Greeks, a hack computer writer, and the Radio Shack Color Computer have in common? Give it some thought before answering, and please, no bad jokes about "Euripides? I mend-a these . . .".

Oh, oh. There's the buzzer. I'm sorry, Mr. Smith, you lost the chance for a fun-filled weekend in Fort Worth, but we're giving you two fun-filled weekends as a consolation prize . . . The answer to the question, of course, is prime numbers.

Now wait. I know you might not like mathematics. However, you don't need to know much about math for this topic, and it has some fascinating aspects. We're going to use a 2000-year old algorithm to generate prime numbers in the wink of an eye by assembly-language on the Color Computer. Furthermore, we'll use the high-resolution graphics of the CC to display the numbers. Interested? Read on . . .

THE PRIME SUBJECT

Prime numbers have interested numerologists through the ages. A prime number is any number that can't be divided evenly by any number except itself and the number one. For example, 5 is a prime number, because 2, 3, or 4 can't be divided into 5 without remainder; only 5 and 1 can be divided into 5 evenly. The number 11 is also a prime number; divisors of 2, 3, 4, 5, 6, 7, 8, 9, and 10 result in remainders, and numbers over 11 result in fractions. The number 12 is not prime because it can be divided evenly by 2, 3, 4, or 6.

The first few prime numbers are 1, 2, 3, 5, 7, and 11.

How would you go about finding all the prime numbers up to a given number? There's been a great deal of recent research into prime number theory — such topics as the total number of primes in a given range of numbers, tests for primes; and special classes of prime numbers, such as "Mersenne primes." One old, but effective method for finding primes, however, was developed over 2,000 years ago by Eratosthenes, a Greek mathematician and astronomer (who, among other things, accurately calculated the size of the moon!).

THE "SIEVE OF ERATOSTHENES"

Eratosthenes' method is shown in Figure 1. All the numbers to infinity are listed in a row. (I gave Bruce an infinitely long list, but he may cut it down to manageable proportions for the newsletter . . .)

Now a "multiplicand," starting with 2, is multiplied by 2, 3, 4, 5, 6, and all integral multipliers. The result is "struck out" of the list of all numbers. In the case of a multiplicand of 2, for example, 4, 6, 8, 10, 12, and so forth would be

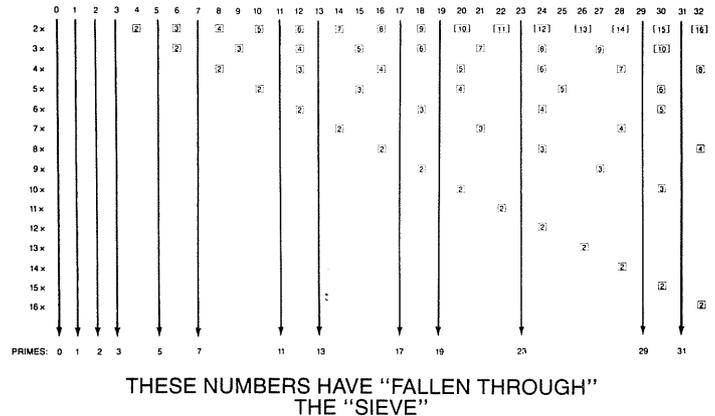


Figure 1. The "Sieve of Eratosthenes"

struck out or deleted from the list of all numbers. This process is repeated for multiples of 3 (6, 9, 12, etc.), multiples of 4 (8, 12, 16, 20, etc.), multiples of 5 (10, 15, 20, 25, etc.), and all numbers up to the last number in the list.

After the multiplicand has been processed, the numbers remaining in the list "fall through" the "sieve" and are primes, as all multiples have been considered.

This is a "brute force" method, as many numbers are struck out more than once. The number 24, for example, is deleted because it is a multiple of 2, 4, 6, 8, and 12. On the other hand, it is guaranteed to work.

ENTER THE COLOR COMPUTER

Eratosthenes would probably have given up stuffed grape leaves for a Color Computer. Early mathematicians performed astonishing calculations by laborious manual methods; imagine what they could have done with a modern small computer that can perform hundreds of thousands of additions per second! Can we use Eratosthenes method on the CC? If so, will it do the job in a reasonable time?

First, let's implement the sieve in BASIC. Figure 2 shows an Extended Color BASIC program that finds the prime numbers from 0 through a given number.

We've used a special way of compressing the number range in this routine. We could have used a one-dimensional array to hold the number range, but this would have used an enormous amount of space, as each array element is 5 bytes. A number range of 1000 would have required 5000 bytes.

What is the most efficient way of representing a number range? About the most efficient is letting one bit represent each number. The first bit would represent a number of 0, the next a number of 1, and so forth. This scheme

```

100 ' PRIME GENERATOR
110 S=1536
120 INPUT "# OF BYTES":E
130 E=E*8-1
140 PRINT "NUMBERS FROM 0 THROUGH "; E;" WILL BE TESTED FOR PRIMENESS"
150 PMODE 4,1:SCREEN 1,0
160 PCLS
170 FOR A=S TO S+E/8
180 POKE A,170
190 NEXT A
200 POKE S,10
210 FOR I=3 TO (E+1)/2 STEP 2
220 FOR J=2 TO (E+1)/2
230 IF I*J>E THEN GOTO 290
240 B=PEEK(S+INT((I*J)/8))
250 B=INT(2*(7-(I*J-INT((I*J)/8))*B))
260 A=(A OR B)
270 POKE S+INT((I*J)/8),A
280 NEXT J
290 NEXT I
300 IF INKEY#="" THEN GOTO 300
310 FOR A=S TO S+E/8
320 B=PEEK(A)
330 FOR I=7 TO 0 STEP -1
340 C=INT(2^I)
350 IF (B AND C)=C THEN GOTO 370
360 PRINT (A-S)*8+(7-I)
370 NEXT I
380 NEXT A

```

Figure 2.

is shown in Figure 3. You can see that each byte represents eight numbers, and that 125 bytes would represent numbers from 0 through 999, a reduction of 40 to 1 over an array. The bytes representing the number range can be anywhere in RAM.

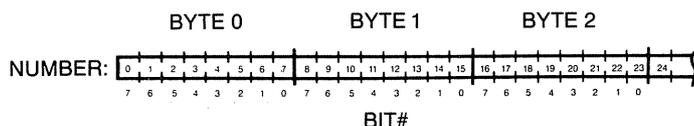


Figure 3. Mapping Numbers by Bits

We'll start off with all bits set to 0, representing numbers that are possible primes. We'll then find multiples and set the corresponding bits to 1, representing "non-primes." The numbers set to 0 after all processing is done will be primes. We should see something like Figure 4 when we're done.

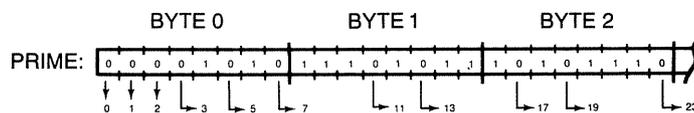


Figure 4. Bit Map After Primes Have Been Found

When I say, "we should see," I mean that literally. Why not put the bytes representing the numbers into a graphics page and display the entire number range? We should then be able to see the entire number range, the whole sieving process, and the resulting primes on the screen!

The program first sets variable S to 1536. This number is the start of the graphics screen area in RAM.

Next, variable E is INPUT. E represents the number of bytes used, which will represent a number range of 0 through $8 * E$, as there are eight bits per byte. The last number in the range will be $E * 8 - 1$, and E is reset to this number in line 130.

Line 150 sets the highest resolution graphics mode in Extended Color BASIC, and switches to a graphics screen. The screen is then cleared by a PCLS, setting all bits to 0. This clear may not be obvious, but in a two-color mode, each graphics element is represented by one bit, and the PCLS sets all elements to the background color or 0.

The loop at lines 170 through 190 performs a somewhat dishonest maneuver. It sets all even numbers to 1, indicating "non-prime." We know that all even numbers will be non-prime, and this reduces the overall processing time.

At the end of the loop, we store a 00001010 into the numbers 0 through 7, setting 0, 1, 2, 3, 5, and 7 to primes, another dishonest tactic. From here on, however, we'll behave ourselves.

The heart of this routine is in lines 210 through 290. This is the code that finds all multiples of 3, 5, 7, 9, 11, and so forth, and sets the bit to a "non-prime." Variable I is the "multiplicand" in the loop, and varies by 2. Variable J is the multiple, and varies by 1. The product of $I * J$ is the number to be "struck out." The first time through the loop we'd have the non-primes 6, 9, 12, 15, etc., the second time the non-primes 10, 15, 20, etc., and so on. Another time-saving trick we're using here is to step only through one-half the range ($(E + 1) / 2$), to avoid unnecessary processing of large numbers.

The only complication here is the "bit-finagling" we have to perform to find the proper bit and byte in RAM. The location in RAM (in the graphics page) is given by:

$$(S + \text{INT}((I * J) / 8))$$

The term $I * J$ is the number. The integer value of this product divided by 8 is the byte location of the byte containing the number. Variable A in line 240 holds the eight bits of the byte from the PEEK at the location in the graphics page.

We now have the byte containing the bit, but what about the bit itself? The term:

$$(I * J - \text{INT}((I * J) / 8) * 8)$$

finds the "bit location" within the byte. This number will be 0 through 7 representing the leftmost to rightmost bit. If we subtract this value from 7, we now have 7 through 0, representing the power of two of the bit location. This power of two is then used in 2^n to yield 128, 64, 32, 16, 8, 4, 2, or 1, which is the actual bit value involved. This bit value (variable B) is "ORed" with the PEEK value to set the bit to indicate a non-prime condition. A POKE at line 270 restores the byte in RAM.

At the end of the loop in line 300, we've gone through all multiplicands and multiples, and have "struck out" all non-primes.

Don't forget that while we've been doing this, we've seen the whole sieving process on the graphics screen, thanks to the magic of the Color Computer and dolmadakia (stuffed grape leaves). The resulting display represents all remaining prime numbers.

Line 300 looks for a keypress before printing out the primes. The loop at lines 310 through 380 goes through all bytes a bit at a time and checks for a bit value of 0. If the bit value is a 0, the number is indeed prime and is PRINTed out by:

```
360 PRINT (A-S)*8 + (7-I);
```

MAKING ERATOSTHENES EVEN HAPPIER

The sieving process works fine, but has one drawback — it takes hours for large number ranges. Although the old Greek would have been ecstatic over this display, I suspect he would have given up even his baklava for the

assembly-language version of his sieve. Would you believe 32 seconds to find all primes from 0 through 49,151? Read on . . .

The assembly-language version of this program is shown in Figure 5. It is called from BASIC by a `USR` call and assumes that the RAM area used for storage is graphics page 0 starting at 1536. It also assumes that all even numbers have been cleared to 1.

```

3F00      00100      ORG      $3F00
00110      *****
00120      * PRIME NUMBERS BY 'SIEVE' METHOD *****
00130      * INPUT: LAST CONTAINS LAST NUMBER TESTED *
00140      * OUTPUT: PRIMES IN 1536 AREA *
00150      *****
3F00      0000      LDD      0          LAST NUMBER TESTED
3F02      CC      0003      LDD      #3          START NUMBER
3F05      FD      3F53      STD      CURI         INITIALIZE
3F08      FC      3F53      00190      PRI030      LDD      CURI         GET CURRENT MULTIPLICAND
3F0B      FD      3F55      00200      STD      CURN         INITIALIZE CURRENT NUMBER
3F0E      FC      3F55      00210      PRI040      LDD      CURN         GET CURRENT NUMBER
3F11      F3      3F53      00220      ADDD     CURI         FIND NEXT NUMBER
3F14      25      27          BCS      PRI080      GO IF OVER 64K
3F16      FD      3F55      00240      STD      CURN         STORE
3F19      10B3      3F00      00250      CMPD     LAST      TEST FOR LAST
3F1D      22      1E          BHI      PRI080      GO IF CURRENT GT LAST
3F1F      44          LSR     0          LSR
3F20      56          RORB    0          RORB
3F21      44          LSR     0          LSR
3F22      56          RORB    0          RORB
3F23      44          LSR     0          LSR
3F24      56          RORB    0          RORB
3F25      C3      0600      ADDD     #1536     FIND DISPLACEMENT
3F28      1F      01          TFR     D,X        POINT TO ADDRESS
3F2A      FC      3F55      LDD      CURN         NOW IN X
3F2D      C4      07          ANDB    #7          CURRENT NUMBER
3F2F      5C          BCS     0-7        GET BIT # 0-7
3F30      4F          CLR     CLEAR MASK   NOW 1 THROUGH 8
3F31      1A      01          ORCC    #1          CLEAR MASK
3F33      46          RORA    SHIFT MASK   SET C
3F34      5A          DECB   DECREMENT COUNT
3F35      26      FC          BNE     PRI070     GO IF NOT DONE
3F37      AA      04          ORA     ,X          SET BIT
3F39      A7      04          STA     STORE IN BYTE
3F3B      20      D1          BRA     PRI040     GO FOR NEXT NUMBER
3F3D      FC      3F53      00460      PRI080      LDD      CURI         LOAD MULTIPLICAND
3F40      C3      0001      00470      ADDD     #1          BUMP MULTIPLICAND
3F43      C3      0001      00480      ADDD     #1          TWICE FOR ODD #
3F46      FD      3F53      00490      STD      CURI         STORE
3F49      F3      3F53      00500      ADDD     CURI         TWICE # FOR LIMIT CHK
3F4C      10B3      3F00      00510      CMPD     LAST      TEST FOR DONE
3F50      23      B6          BLS     PRI080     GO IF CURRENT LE LAST
3F52      39          RTS     RETURN
3F53      0000      00540      CURI     FDB      0          MULTIPLICAND
3F55      0000      00550      CURN     FDB      0          CURRENT NUMBER
0000      0000      00560      END
00000      TOTAL ERRORS

```

Figure 5.

One parameter is passed to the assembly-language program, the value of the last number to be tested. This value is put into locations `$3F00` (most significant byte) and `$3F01` (least significant byte). The actual program starts at location `$3F02` and is not relocatable; reassemble with a different `ORiGin` if you have different memory requirements.

The algorithm used for this program is very similar to the algorithm discussed above. Multiplicands from 3—3, 5, 7, and so forth, are used in the outer loop from edit line 190 through 520 (notice the indentations in the listing comments). The inner loop from lines 210 through 450, multiplies the multiplicand by 2, 3, 4, etc., to find the number to be “struck out.”

The byte address is found as in the BASIC program — by dividing the number by 8. A bit mask is established as in the BASIC counterpart and used to set the appropriate bit to a 1 value.

DETAILED DESCRIPTION OF THE PROGRAM

An immediate value of 3 (`#3`) is loaded into the D register (A and B taken together to form one 16-bit register) and stored into location `CURI`. Location `CURI` holds the current multiplicand.

The `LDD` at instruction `PRI030` loads the current multiplicand and stores it in location `CURN`. `CURN` holds the current number for the “strike out.” The inner loop at `PRI040` loads `CURN`, adds `CURI`, and stores it in `CURN` for

the next strike out number. This effectively multiplies the multiplicand by 2, 3, 4, 5, etc., each pass through the loop. The `BCS` instruction branches to the next multiplicand at `PRI080` if a carry occurred. A carry will only occur if `CURN` is incremented above 65,535, and this ensures that the loop terminates properly for large multiplicands.

The loop also terminates for a current multiple greater than `LAST`, the last number to be tested.

The three sets of shifts (`LSRA/RORB`) finds the byte displacement by shifting the D register 3 bits to the right and dividing by 8. A value of 1536 added to the displacement points to the actual byte address containing the bit for the number. This pointer value is transferred to the X register in preparation for changing the bit value in the byte.

`CURN` is now `ANDed` with 7 to find the number of the bit that represents the number from 0 (leftmost) to 7 (rightmost). This number is incremented by one (1 through 8), the A register is cleared, and the Carry condition code is set by `ORCC #1`. All of this is in preparation for finding the “mask” for an `OR` operation that will be similar to the BASIC `OR`.

The loop at `PRI070` shifts the mask (in the Carry flag and the A register) right until it is aligned at the proper bit position. The count is the B register, and is decremented until it is 0.

After the shift, the byte containing the bit is `ORed` with the A register and restored into the byte address to strike out the non-prime number.

The outer loop at `PRI080` loads `CURI` and adds 2 to the value. `CURI` is initially 3, and will increment to the next odd number each time through the outer loop. The outer loop is terminated when the multiplicand is greater than half of the `LAST` value; all numbers have been struck out in this case.

USING THE ASSEMBLY-LANGUAGE PROGRAM

Figure 6 shows the assembly-language program in the form of `DATA` statements embedded in the BASIC driver. The assembly-language code is moved to the `$3F00` area before execution. Before loading the program, be certain to perform a “`CLEAR 200,&H3EFF`” to protect the `$3F00` area.

The program first asks for the `#` of bytes; enter 1/8th of the number range to be tested. For example, if you want to test 49,152 bytes, enter “6144.” This, by the way, is the maximum number range that you’ll be able to see on the screen.

The program will then strike out all even numbers; this will be a minute or so of slow BASIC processing on the screen. After the initial pass, however, things move like a Greek folk dance. You’ll see the multiples of 3 zap diagonally across the screen, followed by multiples of 5, followed by multiples of 7, and so forth. As numbers are struck out, you’ll see the remaining primes come into force. If only Eratosthenes were present . . .

After about 32 seconds, only primes remain on the screen. Press any key to get a listing of all primes; you may change line 360 to an “`PRINT #-2`” if desired, for a hard copy.

This is only one minor topic of many fascinating mathematical recreations that can be investigated on the Color

```

30 * PRIME GENERATOR
40 DATA 0,0,204,0,3,253,63,83,252,63
41 DATA 63,253,63,83,252,63,83,243,63,83
42 DATA 37,39,253,63,83,16,179,63,0,34
43 DATA 30,68,86,68,86,68,86,195,6,0
44 DATA 31,1,252,63,83,196,7,92,79,26
45 DATA 1,70,90,90,252,170,132,167,132,32
46 DATA 209,252,63,83,195,0,1,195,0,1
47 DATA 253,63,83,243,63,83,16,179,63,0
48 DATA 35,182,57,0,0,0,0
50 FOR A=#H3F00 TO #H3F56
56 READ B
70 POKE A,B
80 NEXT A
110 S=1536
120 INPUT "# OF BYTES";E
130 E=E*8-1
140 PRINT "NUMBERS FROM 0 THROUGH ";E;" WILL BE TESTED FOR PRIMEHOSS"
150 PHODE 4,1,SCREEN 1,0
160 PCLS
170 FOR A=S TO S+E/B
180 POKE A,170
190 NEXT A
200 POKE S,10
210 DEFUSR0=#H3F02
220 POKE #H3F00,INT(E/256)
230 POKE #H3F01,E-INT(E/256)*256
240 A=USR0(0)
300 IF INKEY$="" THEN GOTO 300
310 FOR A=S TO S+E/B
320 B=FEEK(A)
330 FOR I=7 TO 0 STEP -1
340 C=INT(2^I)
350 IF (B AND C)=C THEN GOTO 370
360 PRINT (A-S)*8+(7-I)
370 NEXT I
380 NEXT A

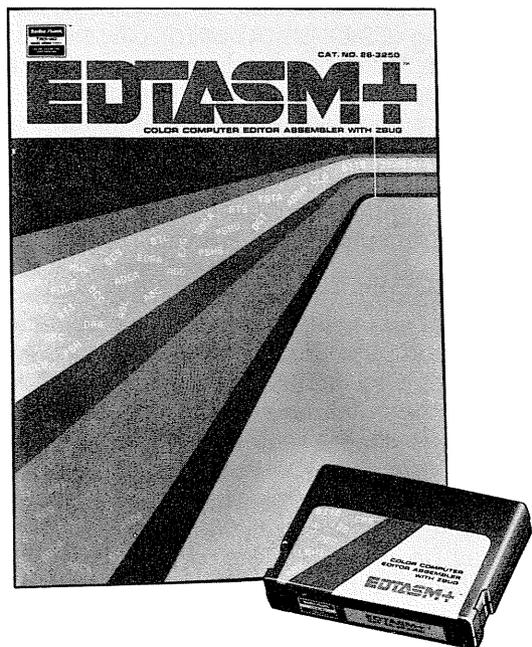
```

Figure 6.

Computer and for which assembly-language is desirable. A good place to get a background on topics in recreational mathematics is in Martin Gardner's column in "Scientific American"; Mr. Gardner is semi-retired now, and his "Mathematical Games" column has now become Hofstadter's "Metamagical Themas" (Hofstadter is the author of the best-selling "Gödel, Escher, Bach"), but past and present columns contain much valuable information on mathematical topics that can be implemented on small computers.

(Not to put myself in the same esteemed company, but I might mention another text on microcomputer math topics — "Microcomputer Math" from Howard W. Sams, and Co. by yours truly.)

Try the above program. You may be pleasantly surprised by the results and it may spur your interest in this assembly-language area. If all else fails, find a Greek mathematician and take him out to lunch. . . .



Convert 35 mm to 16 mm

Robert Edwards
 Steed Rd. Box 581-A
 Chenango Forks, NY 13746

A Program for Pocket Computer 1

```

10: "A"
   : PAUSE "CONVERT 35 mm TO 16 mm"
   : BEEP 1
   : CLEAR
20: INPUT "FOOTAGE AND FRAMES "; A
30: B=INT(A)
   : C=(A-B)*10
40: D=B*16+C
   : E=D/40
50: F=INT(E)
   : G=E-F
60: H=G*40
70: PRINT F; "FEET "; H; "FRAMES "
80: GOTO 20
90: "B"
   : PAUSE "CONVERT 16 MM TO 35 MM"
   : BEEP 1
   : CLEAR
100: INPUT "FOOTAGE AND FRAMES ";A
110: B=INT(A)
   : C=(A-B)*100
120: D=B*40+C
   : E=D/16
130: F=INT(E)
   : G=E-F
140: H=G*16
150: PRINT F; " FEET ";H;" FRAMES "
160: GOTO 100
170: END

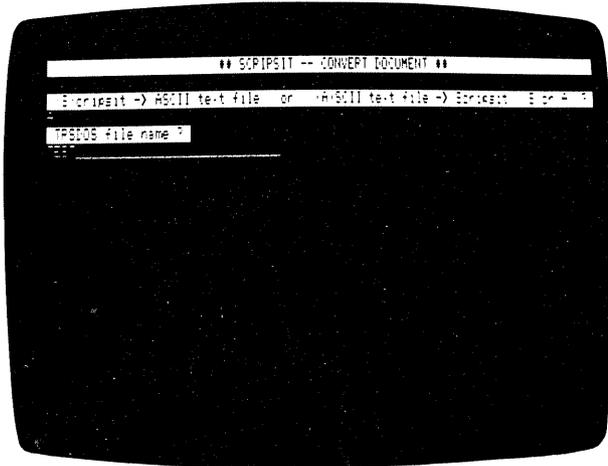
```



More about SCRIPSIT 2.0

This article is for all of you SCRIPSIT 2.0 users who know you have a wonderful word-processing program and want to learn more about its finer points. In this article we will try to explain the use of some of the features that might be unfamiliar to you. We will assume that you have some skill with SCRIPSIT 2.0, but if you haven't used SCRIPSIT 2.0 very much, or are unsure of your abilities, you will want to practice what you learn here before trying these techniques on actual documents. It will also be helpful if you read the sections on the CONVERT, ASSEMBLE, and COPY utilities in the SCRIPSIT 2.0 reference manual. We are going to cover these utilities using SCRIPSIT 2.0 with VisiCalc and Profile.

First a little instruction on a utility that is very important to this article, the Convert utility. The Convert utility allows you to convert an ASCII TRSDOS file to a SCRIPSIT document or a Scripsit document to an ASCII TRSDOS file. This feature allows us to use data or text created by other software packages such as VisiCalc, Profile, or even your own BASIC programs. To convert an ASCII disk file into a SCRIPSIT 2.0 document, or a SCRIPSIT 2.0 document into a ASCII disk file do the following. From an open document press **(ESC) (U) (ESC)** which displays the Convert prompt or press **(ESC) (U) (C)** to go directly to the utility. The Convert utility screen will be displayed.



To convert an ASCII disk file into a Scripsit document, press **(A)**, or to convert a SCRIPSIT 2.0 document into an ASCII file, press **(S)**. The second prompt asks for a "TRSDOS file name?." This is either the name of a file already on a disk or the disk file name under which the current document will be saved. If you are unsure of the requirements for a TRSDOS file name, be sure to read the TRSDOS section about FILE NAMES in your Model II owner's manual. To use the convert utility you must either have extra drives with a TRSDOS formatted disk in one of

them, or a "Minimum Scripsit Disk" (refer to the Scripsit reference manual to create this minimum disk) on which to put files and from which to take files.

VISICALC® PRF FILES

One of the print options of VisiCalc is the ability to "print" a sheet to a disk file. This file is called a PRF file and is an ASCII text file, which can be easily converted to a Scripsit document. To create a PRF file refer to the print commands section of your VisiCalc manual (/PF). Be sure that the width of the margins of the VisiCalc sheet are not wider than the margins of your original document. If they are, don't worry, you can fix them later with Scripsit. After you have created and printed your sheet to a disk file, you can use the information in a report, contract, list of specifications, or any other document without having to retype anything. To use the PRF file with a document you have created, open the document and place the cursor at the very end of the document. From the open document use the convert utility as described earlier (use A and your file name). After you enter the file name and press **(ENTER)** you will see the information appear on your screen as it is being added to your document. When the PRF file has been loaded into your Scripsit document and the cursor reappears you can use SCRIPSIT 2.0 to edit this information so it will appear exactly as you want. You can change the margins, delete information or insert print control codes, etc.

Another method (although more complicated) of using the PRF files with Scripsit is to create a new document, Convert the PRF file into the empty document, and use the Assemble utility of SCRIPSIT 2.0 to combine the two documents. Pressing **(ESC) (U) (A)** will display the Assemble Document menu. Enter the document name (file name) of the one to be added to the document that is currently open. Press **(ENTER)** and it will be included in this document at the end of the document. You can use this process to put as many different sheets of information together as needed. By editing the information, you can create a custom document with a minimum of effort.

PROFILE MERGE FILES

We will refer you to the Profile article elsewhere in this issue of Microcomputer News on how to create a regular merge file with PROFILE to use with SCRIPSIT 2.0. Here we will go over some of the less obvious uses of the Merge utility. Sometimes you may find that a file you have already created is just what you want for a new document except for a couple of items. A file might contain, for example, the first and last names for a mailing list but you don't want titles such as Dr., Mrs., Mr., etc. Using the Convert and Assemble utilities and the advanced editing features of SCRIPSIT 2.0, you can fix that problem. Also, you can combine two files that have the same variable names and the

same number of variables in each grouping. This would be a convenient way to combine two mailing lists. To do this, first open a new document and use the Convert utility discussed before to get the file into SCRIPSIT. Answer (A) to the first prompt. The TRSDOS file name you will use when you convert your PROFILE created file is the same one you would use if you were going to do a merge instead. (Refer to the section in the PROFILE manual on "Merging with SCRIPSIT 2.0"). Now look at your new document. It will look much the same as a merge file for SCRIPSIT 2.0. If you want to append a file so that you don't have to do a merge twice, open a new document, follow the same procedure, and then use the Assemble utility to add the other document to the first one. In looking at the assembled document you will see the variable names from the second merge document. Delete the second set of merge variable names using your normal editing functions to modify the information. Make sure there is an (ENTER) after each variable and a second (ENTER) after the last variable in each group of variables. If you don't have the correct (ENTER)'s then the document will not print. You are now ready to print your merge document.

LABELS, ENVELOPES, AND COMMON FORMS

Have you ever used SCRIPSIT 2.0 to write a business letter then manually (or with a typewriter) addressed the envelope? Didn't that seem like a waste of time? A better way is to set up a document for both a large envelope and a small envelope with the return address and X in the address block for the recipient's name, street address, city, state, and zip code. Then after you type your document, and before you are ready to print, you can assemble your envelope document with this one. Remember, that when you assemble a document, the second document is on a separate page and will print as a separate page. Fill in the name and address in place of the x's (you can define a user key to do this automatically.), and after the letter is printed you can slip an envelope into the printer, print the next page, and the envelope will be addressed and ready to send.

This also holds true for forms. You can set up a document (again with Xs) and then fill in the correct information on each form, print the form and you're ready to fill in the next form. Or if you have the desired information on Profile you can make the form your base document and use the MERGE utility to print your forms. This may sound complicated, but if you have ever had to fill out the same form over and over, with only the information that fills in the blanks changing each time, you will appreciate how much easier this method is.

Labels are also easy to print using a common form with Xs where the information goes. You could even set up a mailing list with a base document having the merge variables positioned correctly and another document with the names. You can't sort or select any specific information, but if you only need to print the labels then this procedure will work well. If you need to add information to the merge document, add the information, convert the merge file and then print the labels using the label base document and the Scripsit merge utility.

CONCLUSION

We hope these ideas on different ways to use SCRIPSIT 2.0 will be helpful in your every day business world. Some of the secretaries here at Radio Shack are just finally getting used to the incredible features of Scripsit. Some of us have fairly poor handwriting and occasionally our writing efforts may require several revisions. With Scripsit our handwriting becomes legible (because we now use a printer instead of scribbling) and as many times as we've revised this article, we sure are glad we have Scripsit so we don't have to retype the entire article each time we make a change. If you employers think Scripsit 2.0 is just for the secretarial pool, try using it yourself. Type a couple of memos instead of writing them down and having your secretary type them. Instead of having to do everything your secretary will only have to proof, do the final print, and distribute the memo allowing her (him) to make more productive use of the few hours in a work day. You might consider doing the first draft of a letter yourself, also. With this program the sometimes boring task of writing becomes fun.



Computer Customer Services Address and Phone Number

8AM to 5PM Central Time

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

Model I/III Business Group	817-870-2041
Model II/16 Business Group	817-870-2042
Languages and Compilers	817-870-2044
Color and Pocket Computer Group	817-870-2150
Hardware and Communications Group	817-870-2571
Educational Software	817-390-3302
Games, Books, and New Products	817-870-2271

New Menu System

Menu Structure Takes on New Look, Promotes Ease of Use

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.

As part of CompuServe's continuing effort to make its information service responsive to customers needs, a new menu structure has been put into effect.

All of the information contained on CompuServe has been organized under 6 primary interest sections:

1. Newspapers
2. Home Services
3. Business and Financial Services
4. Personal Computing Services
5. User Information
6. Index.

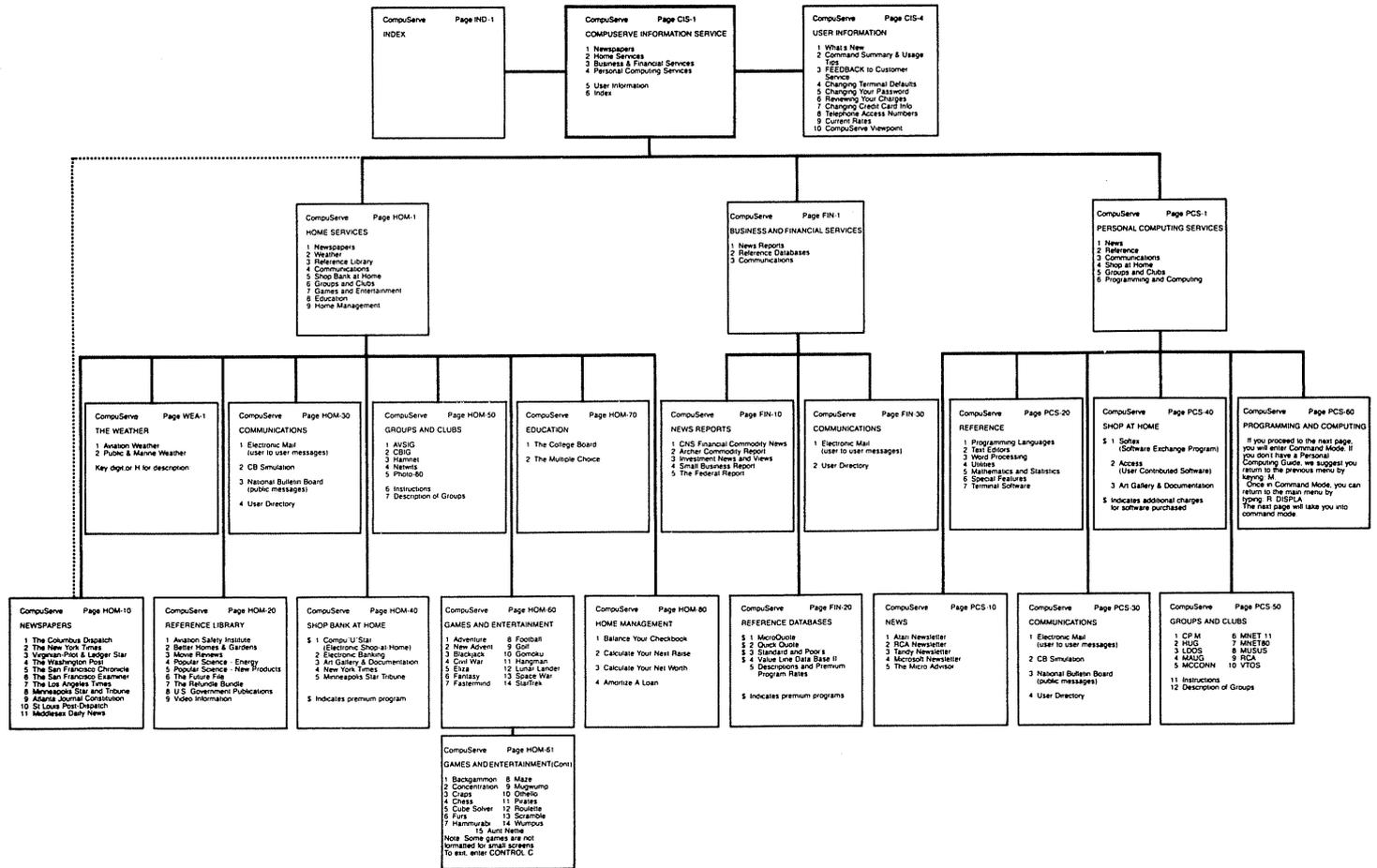
Newspapers.

Our electronic newspapers are now directly accessible from the new main menu as selection 1. All eleven newspapers can also be reached from the Home Services menu.

Home Services.

The Home Services sections contains general and family-related information and services. Currently, these include:

1. Newspapers
2. Weather: This includes both aviation and public/marine weather.
3. Reference Library: Under this menu you can find general interest newsletters from such information providers as the Aviation Safety Institute, magazines such as Better Homes & Gardens, and the Refund Bundle.



4. Communications: A communications area has been placed in each of the three main service sections to enable you to have access to these often-used offerings as easily as possible. This communications area includes Electronic Mail, CB, the National Bulletin Board and the on-line User Directory.
5. Shop/Bank at Home: Electronic banking and shopping offerings can be found under this choice including, among others, Comp-U-Star and Bank-at-Home.
6. Groups and Clubs: The groups and clubs available under Home Services are general interest groups that share an interest in non-technical subjects.
7. Games and Entertainment: The games menu contains your old favorites, like Adventure and Football, in addition to games that were previously available on other areas of the service. Interactive games, such as Decwars and Space War, can be reached through this menu.
8. Education: The education section contains the College Board that is a must for any prospective college student and the Multiple Choice where you can test your trivia skills and be challenged by many types of self development tests.
9. Home Management: In this section are grouped several very handy home-management programs for doing such otherwise tedious jobs as balancing your checkbook or figuring out a loan schedule.

Business and Financial Services.

The Business and Financial Services section includes all business and financial news reports, updated stock quotes and business reference databases including, among others, MicroQuote and Standard and Poor's.

Personal Computing Services.

The Personal Computing Services section includes news, programming languages, a word processor, a Software Exchange program for purchasing software, computer related groups and clubs and an area for programming activities.

User Information.

The User Information section contains material pertinent to your general use of the information service including Feedback, documentation and product ordering, What's New, a general command summary as well as instructions for verifying and/or altering your terminal defaults, password and credit card information. Detailed billing information, current access rates and telephone numbers are located here.

Index.

The Index section contains the most up-to-date version of our Subject Index.

* * * * *

Questions and comments about the CompuServe Information Service can be sent to Richard A. Baker, editorial director, CompuServe Information Service, P.O. Box 20212, Columbus, Ohio 43220 or through Feedback on the User Information menu.

Racing Information

The method of accessing Racing Information has changed: RIS may be accessed by going to page HOM-110 on CompuServe. When "Command" appears, type "SS4." Then type "RF." Then type the lowest message number that is on the system.

Michael F. Hollander
 Director of Operations
 Racing Information Systems
 User ID: 70001,557

Microcomputers for Neophytes

The University of Oklahoma, known for excellence in computing sciences and computer related mathematics, and its Saturday workshops in computer usage for secondary and elementary teachers, has introduced a new course in *microcomputer programming for the complete neophyte*. The course is a hands-on course using 30 TRS-80 computers for 40 students. Class meets for two hour-long lecture-demonstrations each week, with students expected to put in at least another four hours a week of lab plus homework. (Lab is open 9:00 a.m.-9:00 p.m. M Tu W Th F with breaks for class, which are held in the lab room.) The text used is *EXPLORE COMPUTING with TRS-80 and COMMON SENSE* (Prentice Hall). Grades are on an S/U basis, with either one or two hours of strictly undergraduate credit available, depending upon amount and difficulty of work undertaken.

Dr. Richard V. Andree said the course was created in 1981 primarily to help future teachers, but the course attracted twice as many students as could be accommodated, including three Ph.D.s and numerous graduate students who could not receive credit, but worked hard anyway. Only four of the forty students who enrolled were Education majors, the others came from all over the University.

Highly successful, the course was enlarged in Spring 1982 and will be further enlarged in Fall 1982 if additional microcomputers can be obtained.

New Computer Clubs

The BASICS
 Frederick Town High School
 % Gary Mayton
 117 Columbus Rd.
 Fredericktown, OH 43019

Johnson Amateur Computer Society
 34 South Fir Street
 Medford, OR 97501
 Computer Bulletin Board System
 phone number 503-535-6883

Movie Reviews

Into horror movies? According to Jay Brown, a popular movie review columnist who can now be read on the Dow Jones News/Retrieval service, those of you who are could belong to a beastly class of human beings who delight in potent doses of blood and gore.

In one of his latest movie reviews, the veteran columnist says: "This blood drenched horror number faithfully delivers a full ration of gore that should delight fans of this genre."

Brown recently wrote this in his review of the "The Beast Within" — a movie about a teenager who turns into a grotesque monster.

And all you thought Brown wrote about had to do with dramatic effects and the intricacies of theatrical style.

His writing is of a feisty sort and has most recently been added to the Dow Jones News/Retrieval service.

Another sample of his writing on this latest consumer oriented data base is his review of the movie, "Amin—The Rise and Fall" — which he rates as (FAIR).

He wrote: ". . . Joseph Olita, in the title role, is amazingly similar in appearance to the brutal Amin who is shown eating human flesh, cutting up his wife's corpse and preserving the heads of his enemies in his refrigerator."

But Brown does not always focus on the gore. In fact, there are times when he goes to as many as eight different movies in one week, when there is a glut of new releases.

For example, his review of the movie "Reds," begins with: "A brilliant and passionate thick slice of history with Warren Beatty splendidly performing as radical journalist John Reed who chronicled the Russian revolution."

And there's Brown's review of the movie, "On Golden Pond," in which he wrote: "The main reason to see this sentimental drama is the magnificent casting of Henry Fonda and Katherine Hepburn as an elderly couple trying to enjoy perhaps their last summer together while embroiled as usual in family conflicts."

Not to be forgotten is his review of the movie, "Absence of Malice," in which he wrote: "A brisk, intelligent drama that takes a penetrating look at the ethics of journalism in view of the quest for truth and justice."

After spending several years as a newspaper reporter and editor for several publications, Brown decided to branch off on his own and formed the Cineman Syndicates in 1975.

His reviews are being published by more than 70 newspapers and news services, including publications in Saudi Arabia, Caracas, and the Philippines.

"This started as a newspaper syndicate publication," Brown said recently. "I had noticed that nobody had a cap-sulized movie review . . . It was something that I sort of needed myself and I found it lacking," he said.

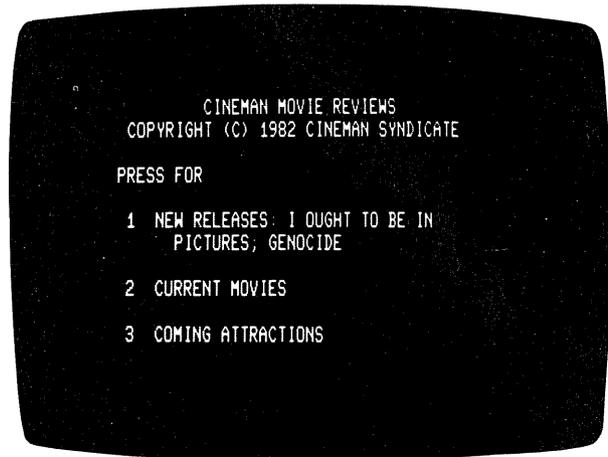
What was lacking was easy, quick and up-to-date access to movie reviews that are interesting, informative, and concise.

Brown said he sees approximately 100 films a year.

On the Dow Jones News/Retrieval service, Cineman reviews can be accessed by typing:

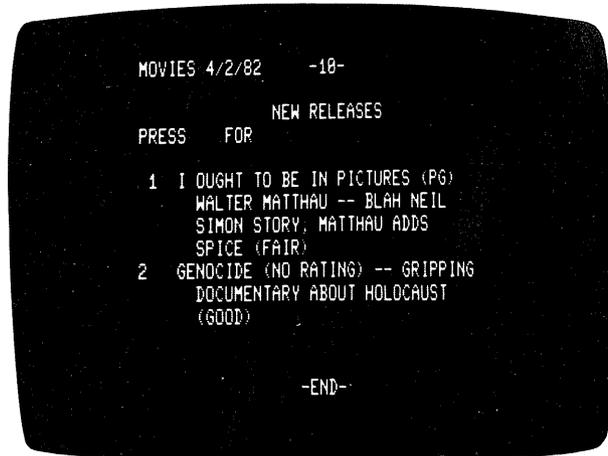
```
//MOVIES <ENTER>
```

and you will see the menu listed on screen 1, below.



Screen 1.

If you press Number 1, for New Releases, another menu similar to the one on screen 2, will be displayed.



Screen 2.

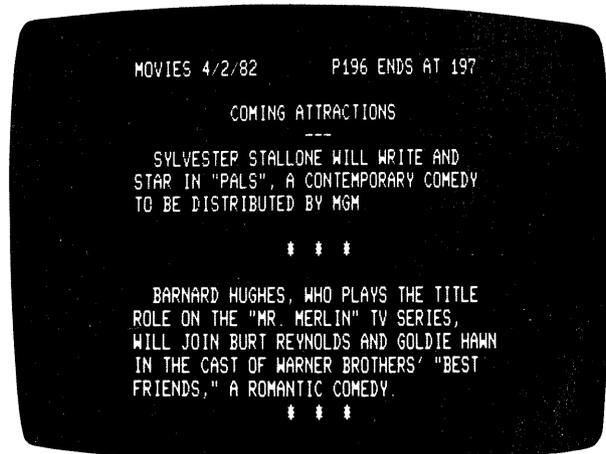
At the bottom of the screen, notice the word: -END-. That tells you that there are no more movies in that category.

To call up one of the reviews, press a number and **(ENTER)**. In this case, if you press number 1, the review shown in screen 3 is displayed.



Screen 3.

If you want the remainder of the review, press **(ENTER)**. The Current Releases category on the initial menu works the same way. And the Coming Attractions category is even more simple to access. If you press number 3 on the main menu screen (see Screen 1), you will immediately see a screen showing coming attractions, similar to screen 4 below:



Screen 4.

If you are not a Dow Jones News/Retrieval service subscriber but are interested in this or other data bases, call the Dow Jones Customer Service hotline number: 257-5114. It's toll free.



micro-RESEARCH Problems for Computer Assisted Solution

Richard V. Andree
micro-RESEARCH Problems
P.O. 2910
Fort Worth, TX 76113-2910

This is our first micro-RESEARCH column in what we hope will be a long series.

A micro-RESEARCH problem is an open-ended problem which does not require advanced mathematics to *understand*—most are understandable by 7th grade students, all by students taking 9th grade algebra. Yet the problems are *not easy* to solve without the assistance of a microcomputer. Most first attempt computer solutions can be speeded up considerably by the judicious use of *common sense* (frequently called "computer sophistication"). The micro-RESEARCH problems are present to challenge *your* programming ability. I hope you enjoy them.

I will provide you with a sample solution to each month's problem, but you will have to wait a month or two to get it (after all, I want you to work out your own solution, not copy mine!) If I get some really unique written responses, I may publish one or two of them also. There is one important thought here—in most, if not all cases, there is no one correct answer or method to solve a particular problem. Be creative, and generate your solution. The problems are open-ended. Once the stated problem is solved, numerous possible extensions will occur to the solver. The extensions justify the use of the adjective "RESEARCH" in the title, since the full answers are not available in standard publications, and frequently are actually unknown. Yet the exploration of these problems is well within the ability of secondary students who can program in BASIC.

micro-RESEARCH Problem #1.

Find all integer squares $S = N * N$ which contain each of the ten digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 *exactly once*.

You should find this problem easy to understand, and it can be attempted using a pocket calculator or a large table of squares. It takes little sophistication to recognize that only those squares containing exactly 10-digits need be considered. Furthermore, as soon as a repeated digit is discovered in a 10-digit square, it can be discarded without further checking. It takes some understanding of computer programming to display only the solutions. Can you do it?

Some obvious *extensions* (which are not as easy) include:

For "squares" read "cubes" or "fourth powers" or "factorials."

For "exactly once" read "exactly twice" or "exactly 3 times."

The problem may also be altered by using only the nine non-zero digits, 1, 2, . . . , 9.

What results can *you* come up with?

Using the Digitizer to Calculate Area Measurement and Curve Integration

The basic approach used here is to sum up strips of area equal to $\Delta X * Y$. To measure the area inside a figure such as Figure 1, the area under the upper portion of the figure is considered positive and the area under the lower portion negative.

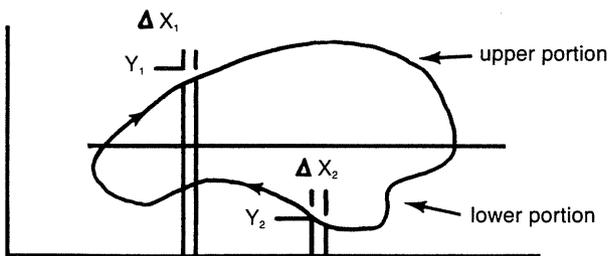


Figure 1

The Model II program below has the advantage of automatically closing the figure. This technique involves inserting a test for a specific set of coordinates. Line 398 checks to see if the value of the coordinates X and Y both equal zero. This should only be true when the first two coordinates are read in and when the program is terminated by pressing RESET on the digitizer.

Before running the program, press RESET on the digitizer so the first values for X and Y read from the Digitizer will be 0's. When all the coordinates have been input, press RESET again and the area of the figure will be printed.

```

100 CLS
   : CLEAR
150 DEFINI I-N
160 AR=0
240 DIM A%(8)
250 FOR I=0 TO 8
260 READ A,B
270 D=D+A*B
280 C=A+256*B
290 IF C>32767 THEN C=C-65536
300 A%(I)=C
310 NEXT I
320 IF D<>2050 THEN PRINT
   : PRINT"DATA ERROR"
   : GOTO 4030
330 REM
340 REM DATA LIST
350 REM
360 DATA 229,62,96,207,225,32,249,120,230,127,119
370 DATA 35,54,0,62,2,201,0
375 A=0
380 GOSUB 8000

```

```

390 XO=X
   : XP=X
   : YO=Y
   : YP=Y
395 GOSUB 8000
398 IF X=0 AND Y=0 THEN GOTO 450
400 DX=X-XP
401 AY=(Y+YP)/2-YO
410 AR=AR+(DX*AY)
420 XP=X
   : YP=Y
425 PRINT @160,AR;" "
440 GOTO 395
450 SA=AR
   : X=XO
   : Y=YO
460 AR=((X-XP)*(Y+YO)/2)
470 AR=AR+SA
475 CLS
480 PRINT"AREA = ";AR
490 END
5000 REM READ ONE CHARACTER FROM THE RS-232
5010 REM
5020 J=0
5030 DEFUSR0=VARPTR(A%(0))
5040 J=USR0(J)
5050 RETURN
5060 REM
6000 REM GET AN 8-DIGIT STRING OF COORDINATES
6010 REM
6020 GOSUB 5000
   : IF J<>88 THEN 6020
6025 PRINT@100,A
   : A=A+1 'COUNTER
6030 D$=""
   : FOR I=1 TO 8
   : GOSUB 5040
   : D$=D$+CHR$(J)
   : NEXT I
6040 RETURN
6050 REM
7000 REM CONVERT 4 HEX DIGITS TO DECIMAL
7010 REM
7020 DE=0
7030 FOR I=0 TO 3
7040 J=ASC(MID$(HX$,I+1,1))-48
7050 IF J>9 THEN J=J-7
7060 DE=DE+J*16^(3-I)
7070 NEXT I
7080 IF DE>32767 THEN DE=DE-65536
7090 DE=.01*DE
7100 RETURN
7110 REM
8000 REM GET ONE SET OF (X,Y) COORDINATES
8010 REM
8020 GOSUB 6000
8030 HX$=LEFT$(D$,4)
8040 GOSUB 7000
8050 X=DE
8060 HY$=RIGHT$(D$,4)
8065 IF X=0 AND Y=0 AND AR<>0 GOTO 450
8070 GOSUB 7000
8080 Y=DE

```

(Continued on Page 24)

Merging Profile Files With Scripsit

The Small Computer Company
Written for the users of PROFILE II,
PROFILE+ and PROFILE III+

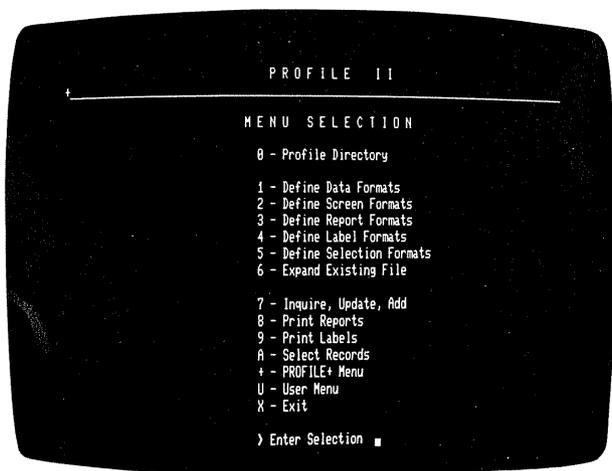
PROFILE Editor
P.O. Box 2910
Fort Worth, TX 76113-2910

We're often asked how, exactly, one merges PROFILE files with SCRIPSIT to create letters to one's accounts or mailing list. The merge process is remarkably simple, since PROFILE II, II+ (also called PROFILE+) and III+ all have built-in merge creation operations. These operations each build up to 5 special merge files compatible with SCRIPSIT, indicated in the directory by the extensions SR1, SR2, SR3, SR4 and SR5. Your merge document, therefore, will be "FILENAME/SR1" (or SR2, et cetera).

In the process of creating a merge file, the computer will usually write the file onto the Drive 0 diskette. Check that you have enough room on Drive 0—the calculation is given below; if not, we'll explain how to set up the file on another diskette with more space.

Let's assume that you own a small solar-powered toy company, SUN CHILD, INC. You have 3 salespeople, a 20-item catalog and 100 customers. You would like to give your best customers a special discount on a new solar calculator and want to send out personalized letters based on your PROFILE customer list.

PROFILE II & II+ : STEP BY STEP

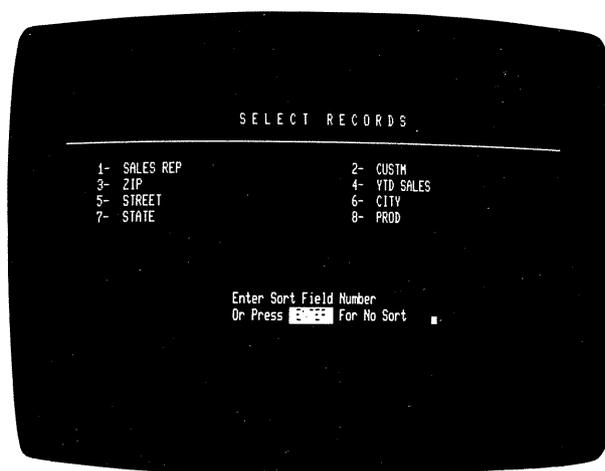


Screen 1: Profile II Menu

1. At the menu, press 5, "Define Selections." For the SUN CHILD company, you'd type in the filename SUN-SALE. Then press 1 for the selector number, and Y[es] to indicate that this is a new format.

2. You will see a "Define Selection Formats" screen. You can type in up to 16 names and numbers of fields you wish to use in your merge document. These names do not have to be the same as the names of your data fields. However, since they will be used as codes in the SCRIPSIT letter, you should make a note of the codes for later reference (this operation doesn't include a hardcopy option). For SUN CHILD, you would type in SALES (field 1), CUSTM (field 2), STREET (5), CITY (6), STATE (7) and ZIP (3). Press <ESC> to record the screen, and answer N[o] for password protection.

3. You'll be returned to the menu, where you will press selection "A" (or "4" if you're using the Limited Menu). The key (segment 1) fields will appear:



Screen 2: Profile II Select Records

You are asked for a sort field number. Since you want to sort by customer, press 2 for Field 2, and 25 for the number of characters.

4. A 16-entry selection format appears at the bottom of the screen. You want to select only those customers with total balances greater than \$100, so type in field 4, YTD (Year To Date) SALES. Press <ESC>.

5. When the "Select Record" screen appears, type in GT for "greater than" and 100.00:

YTD SALES GT 100.00

Press **(ESC)**. The computer will select and sort your records according to your criteria and return you to the menu. If you want to see that the new file is on Drive 0, you'd e[X]it from the menu, type "DIR" (Drive 0 is assumed), and press ENTER. The file would be SUNSALE0/SR1, and unlike any other non-merge file in the directory, it would have logical record lengths (LRL) of only 1 byte. It would, however, have many records. This is the only format SCRIPSIT can accept.

You can also see the field names and data in that file if you wish (provided that you have the program SELECT/BAS somewhere on your diskettes). At TRSDOS READY, type BASIC SELECT/BAS -F:1 and press **(ENTER)**. The screen will show:

```
ENTER FILE NAME? (type in SUNSALE)
```

and then

```
ENTER SELECTION NUMBER? Type in "1."
```

The program will display the field names and data to be used in the SCRIPSIT files.

SPECIAL INSTRUCTIONS: To calculate the amount of room you'll need for the merge file, take the lengths of all fields you want to pass to SCRIPSIT, multiply by the number of records you expect to pass, then divide by 1280 bytes per gran (granule). The result, in grans, is the maximum amount of room you'll need. (Numbers of grans available appear at the bottom of the DIRECTORY screen.) DISK SPACE FULL or Error 27 appears if you attempt to put the merge file on a diskette with too little room.

If you don't have enough room on Drive 0, you can create a file on a diskette that does have room; the computer will write the information to it rather than trying to set up a new file on Drive 0. At TRSDOS READY, type:

```
CREATE filename/SRn:drive# LRECL=1,NRECS=0
```

For instance:

```
CREATE SUNSALE0/SR1:1 LRECL=1,NRECS=0
```

This preallocates a merge file with no records on the drive you want. Just make sure you have the right PROFILE diskette in the system when you start the SCRIPSIT merge operation.

6. Move the Drive 0 PROFILE diskette to Drive 1. (According to the 2.0 SCRIPSIT manual, you're supposed to extract the file from the PROFILE diskette and store it on another diskette. This isn't necessary as long as the merge file is on the PROFILE Drive 0 diskette. If, however, you have only 1 drive, you can copy the PROFILE file onto the SCRIPSIT diskette:

At TRSDOS READY type:

```
COPY SUNSALE0/SR1:0 :0
```

You will have to switch the PROFILE and SCRIPSIT diskettes at the appropriate prompts. Put your SCRIPSIT diskette in Drive 0. Create a base letter, embedding the codes you set up in the PROFILE program in braces. For example:

```
{CUSTM}  
{STREET}  
{CITY}, {STATE} {ZIP}
```

Dear {CUSTM}:

We've just bought a truckload of the brand-new SPACE AGE.T solar-powered calculator, and we're so excited about it that we wanted to offer it to you, one of our most valued customers, at a pre-catalog discount.

Our catalog price — if we have any left to sell! — will be \$21.95. Your price is \$17.50 — more than 20% off!

I'm looking forward to your order. Call me at (212) 444-5555.

Best regards,

```
{SALES}  
SUN CHILD, INC.
```

7. Next, get the merge utility. If you're using SCRIPSIT 1.0, press CONTROL U, then M[erge]. For SCRIPSIT 2.0, do the same. ("Merge" is listed on the second utility screen, not the first, in the 2.0 version. Press any arrow to see it.)

8. The program will ask you:

Convert this document into merge information or use as Base document? B

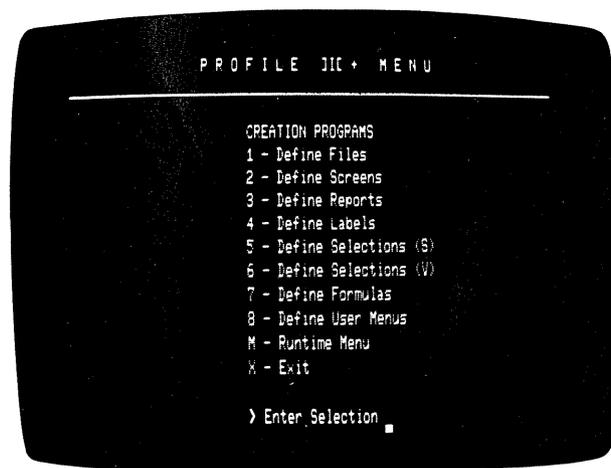
Leave the B — press **(ENTER)**.

Name of merge document (enter TRSDOS file name)?

For our purposes, SUNSALE0/SR1 and press **(ENTER)**.

9. You will be prompted for printing information. When you've filled in all the blanks, the computer will bring the SCRIPSIT letter up to the screen, fill in a customer name and address, and print out the letter while you watch. It will do the same for as many customer names as appear in the merge file.

PROFILE III+ : STEP BY STEP



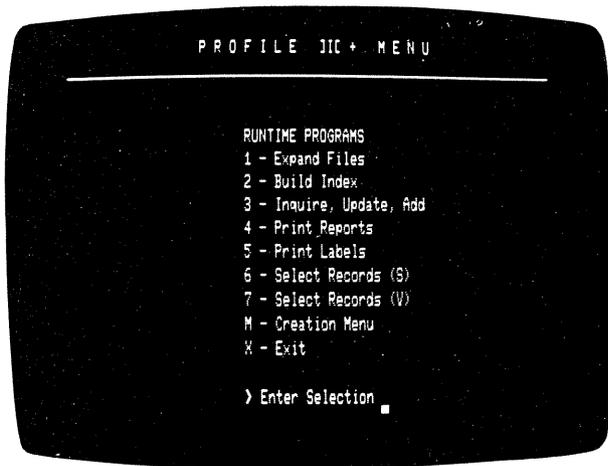
Screen 3: Profile III+ Creation Menu

1. The menu choice for "Define Selection" for SCRIPSIT is 5 on the CREATION diskette. Follow the prompts, using the same information as in the PROFILE II/II+ instructions above. You can make a hardcopy of your codes in this program.

The instruction for creating a file on a diskette in a drive other than 0 is slightly different:

```
CREATE filename/SRn:d (LRL=1,REC=0)
```

where "n" is the file number and "d" is the drive number.



Screen 4: Profile III+ Runtime Menu

2. The "Select Records (S)" menu choice is 6 on the RUNTIME menu. Follow the prompts, using the same information as above. The merge files are FILENAME/SR1, SR2, SR3, SR4, SR5 (and they too have record lengths of 1).

3. Move the PROFILE III+ RUNTIME Drive 0 diskette in Drive 1 and put your SuperSCRIPSIT diskette in Drive 0. Open or type out the base letter, but in this case, embed the codes in @s; however, DO NOT use the @ key on the keyboard to make the character—press SHIFT/0 instead: This does work as unlikely as it sounds, and is necessary because the regular @ is used to designate "record number" in the merge format.

4. Press CONTROL F (the real @ key and the letter F) for the merge operation. When prompted for the name of the merge document, type in SUNSALE0/SR1. The computer will read the merge file off the PROFILE diskette, prompt you for print parameters, and type the letters.

LETTERS

As you are probably aware, the new PROFILE+ enhancement package will be available from Radio Shack by the time you read this column. This package includes the programs most often requested by the large base of PROFILE II users.

For example, Mr. Robert P. Fischer of Lancaster, Pennsylvania, writes: "One problem with printing labels from PROFILE II is that two or more fields cannot be printed together without the blank spaces being printed between them. For instance, a last name field length of 20 and a first name field length of 30 requires a label 50 characters wide to print the complete name, even when the names may be much shorter than the space allotted."

PROFILE+ now includes a push-left field indicator in the label format operation, which moves names and other data together when there are unused spaces in the fields.

Another example: Mr. Sammy Robinett of Gorman, Texas, writes, "I have encountered a small problem. When using more than one Total Field in a report, I get a line feed

after each column is totaled so that the totals are skewed down the page instead of being on one line. Can you recommend a patch or anything else that I can do to correct this, or is this something that I will have to learn to live with?"

The problem in this case is that PROFILE II always puts new totals onto separate lines as:

```
100.00
250.00
23.00
```

However, if you have or buy the PROFILE+ enhancement, PROFILE+ will put the totals on one line if it sees that the fields won't overlap.

PROFILE II users can add these new operations and fields to their program with the PROFILE+ upgrade. Radio Shack catalog numbers are 26-4515 (\$299.00) for PROFILE II+, and 26-4517 (\$120.00) for the PROFILE+ upgrade for PROFILE II.

Mr. Al Kessler of Riverbank, California, sent us a print-out that explains how he squeezed 10 report formats out of PROFILE II, and could do even more—by renaming QUALITY2/MAP, KEY, DAT, DA2, DA3, PMP, PM2 and PM3 to QUALITY1/MAP, et cetera, before printing, then renaming the files to their original names afterwards. In the program, after all, you can only choose report formats 1 to 5!

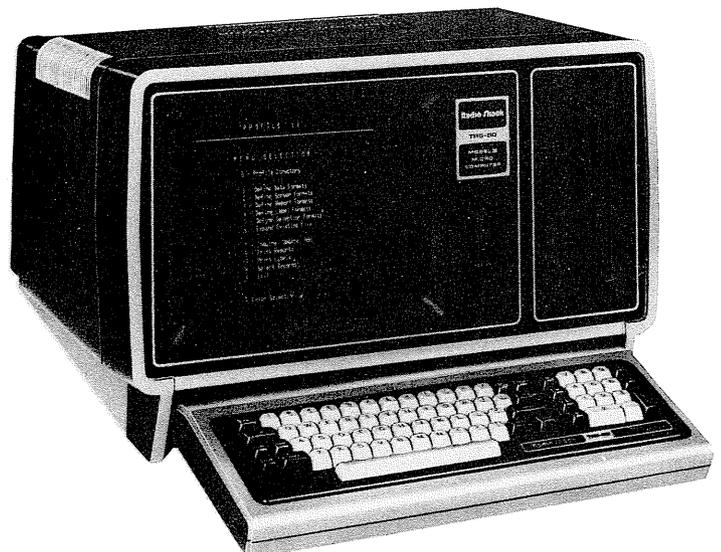
There is an easier way to do it. Set up report format number 5, and rename it to number 6 or A or whatever. Set up another format 5 and rename it number 7, B, et cetera. Then enter your user's menu—or write one—and in writing up the menu entry, list report format #6, #7, et cetera, instead of #1 through #5.

For instance:

```
1 PRINT SALES REPORT
PRINT/EFC (SALESMEN,6,PRINT SALES REPORT)
```

This bypasses PROFILE's normal 5 report limitation.

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



Notes on Previous Issues

March, 1982

Double Dump to Tape (Page 5)

Serge Tremblay of Chateaugay, Quebec, Canada sent his version of "Double Dump":

```
20000 FOR N=1TO3:OUT255,4:FOR M=1 TO 7000:
NEXT M:CSAVE "A":NEXT N
```

The OUT 255,4 controls the remote control for the cassette.

Mark Young of Niangua, MO sent this version:

```
For X=1 TO number of copies wanted:
CSAVE"filename":NEXT X
```

Deltasin (Page 10)

Mr. David Sligar, author of Deltasin wrote (among others) to point out three errors in Deltasin. Please change these lines:

```
20 PI=3.14159
41 IF Q>=1 THEN R=R+Q
70 PSET ((X+140),(80+Y),1)
```

Gary Nance of Vallejo, California became so frustrated trying to correct Deltasin that he submitted his own version:

```
10 PMODE4,1:PCLS:SCREEN1,1
20 FOR S=0 TO 12.56 STEP .07
30 CIRCLE(S*20,96-SIN(S)*85),4,5,2,.13,.90
40 NEXT
50 GOTO50
```

Relocating Machine Language Program (Page 11)

We received the following from Jerel Peterson:

Since I had just bought a disk-based system to replace my cassette drive Model III, I was happy to see an article on transferring machine language programs from tape to disk. I had tape versions of Scripsit and Eliza, as well as several other non-Radio Shack machine language tapes that I wanted on disk. When I saw the article, I figured on getting Scripsit and Eliza transferred, but not the others.

Following the DEBUG routine was easier said than done for someone who has trouble following directions. After many trials and errors, I finally got Scripsit to transfer to disk, but I was always getting an error message when I tried to do the same with Eliza. Finally, I gave up.

Later, as I was paging through the TRSDOS manual, I came across the TAPE command, which allows for direct transfer of machine language tape programs to disk. This looked much easier than going through the long DEBUG routine listed in Microcomputer News, so I decided to try it.

I wasn't really expecting it to work (if it would, why bother with a DEBUG routine) but it did! The program loaded right on to the disk as soon as the cassette stopped.

Getting the program to run caused a few problems. In BASIC, when I typed RUN"ELIZA", I got a "file not found" message. In TRSDOS, I also got an error message. Finally, I simply typed the filename, ELIZA, at TRSDOS Ready, and the program loaded up.

All of my non-Radio Shack machine language programs also loaded up and ran using this procedure, something that could not have been done by using published

procedure because the hexadecimal values were not available.

I also found that I could transfer Eliza and Scripsit (which were recorded at low baud rate) from disk to tape at the high baud rate. This saves quite a bit of time when loading by cassette.

I now have all my machine language tapes recorded on one disk. Rather than switching tapes and waiting several minutes for some of these programs to load up, I can switch from one program to another in about ten seconds.

CLEAR (Page 13)

Mr. Sheldon J. Kramer of Schaumburg, IL sent us the following:

The article on "CLEAR" in the March 1982 issue did not mention that when CLEAR is executed, it also does a RESTORE on both Models I and III. I have not seen this function of CLEAR described previously.

This feature was very useful when I combined several small programs into a larger menu driven program so that each section could sequentially process the same data. Since each section required different dimension (DIM) statements, the use of one CLEAR statement at the beginning of the menu routine eliminated the need for any RESTORE statements.

The following short program demonstrates this function:

```
10 CLEAR
20 READ A$
30 IF A$="END" THEN GOTO 10
40 PRINT A$
50 GOTO 20
60 DATA ONE, TWO, END
```

When RUN, this program generates an endless series of ONE, TWO, ONE, TWO . . . on the screen. However, if in line 30, the transfer is changed to line 20 instead of line 10, then an out of data (OD) error occurs after the first TWO is displayed.

Model II Bugs, Errors and Fixes (Page 35)

Under the PATCHes for TRSDOS 1.2a, an O was inadvertently inserted into the second IODVRS/SYS patch. The O should have been a zero (0).

Pocket Computer Programs (Page 36)

Fred Nachbaur sent the following corrections to programs we ran in March:

TECH-3

```
12 F=1/(2pi*SQR(LC)): B$=" HZ"
14 IF F>Exp3 LET F=F/Exp3: LET B$=" MHZ"
22 L=1/((2piF)^2*C): LET B$=" H"
30 INPUT "F=" ;F
52 F=1/(2piRC): B$=" HZ"
```

TRF-1

```
10 "A" PAUSE "TRF-1 PENTODE": D=60: J=1
30 INPUT "R(P)=";O: O=OJJ
140 PRINT "Q=";K: END
150 "B" J=1: D=70: PAUSE "PENTODE BW": GOTO 20
180 S=1/(1/O+1/N): IF D=160 LET
R=1/(1/S+1/JJQWL): ...
190 R=1/(1/S+1/(L/(CP)))
195 B=JJ/(2piRC): ...
```

Sub Destroyer (Page 47)

We had numerous people point out an error in line 290 of Sub Destroyer. Line 290 should read:

```
290 IF A$="H" THEN K=K+2
```

Our thanks to the many people who wrote.

Additional Use of "@LOOKUP" in VisiCalc

Larry Scanlon
230 Lorraine Drive
Berkeley Heights, N.J.

I read with great interest the article in the March issue of MICROCOMPUTER NEWS entitled "New Commands and Functions for VisiCalc." Normally, when I read about enhancements to software that I already own I get very frustrated, as I wish I had these enhancements. I was not as frustrated as usual this time as I had already figured a way to incorporate "if" and "and/or" statements into my VisiCalc worksheets. I have version 1.20Z of VisiCalc, and run it on a Model I. This version contains the "@LOOKUP" function, and I have always been able to construct a relatively simple decision table to be used in combination with arithmetic logic that I build into the formulae to simulate IF, AND and OR statements that I wish to use. I have also always used tables to simulate the functions that "@CHOOSE" would allow. Obviously, these methods are a little cumbersome to set up, and cause the worksheet to grow (which can sometimes be a real problem), but they do give a great deal of flexibility to us users of older versions of VisiCalc.

COMMISSION

To illustrate the decision table and arithmetic logic I utilize, the following is how to use them in the same examples you gave in your article on the new functions: "A salesman's commission is 10% of sales; but if one month he sells over \$10,000, he gets a bonus of \$200.

	A	B	C
1) Sales		(Your Input)	
2) Commission		(B1 * .1)	
3) Bonus		@LOOKUP(B1-10000,A8..C8)*200	
4) Income		+ B2 + B3	
5)			
6)			
7)			
8) -10000	0		1
9) 0	0		1

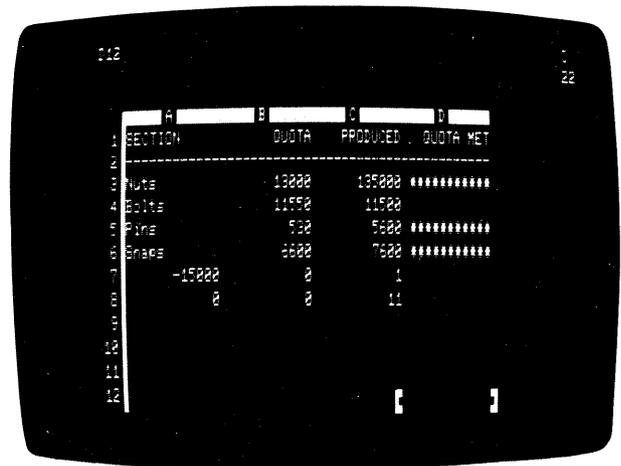
Thus if Sales is \$10,000 or less the Bonus will be (0*200) or 0. If Sales is \$10,001 or more the Bonus will be (1*200) or 200. It should also be noted that the value of A8 should be smaller than the lowest possible value of B1-10000 or it is possible for "@LOOKUP" to return a value of "NA."

QUOTA

The example of each department exceeding its production quota is a little trickier. I use the "/F*" format command to print a row of asterisks when the statement is true rather than a number vs. 0 to indicate if the decision is true:

	A	B	C	D
1) Section	Quota	Produced	Quota Met	
2)				
3) Nuts	13000	135000	/F* @LOOKUP(C3-B3,A7..C7)	
4) Bolts	11550	11500	/F* @LOOKUP(C4-B4,A7..C7)	
5) Pins	530	5600	/F* @LOOKUP(C5-B5,A7..C7)	
6) Snaps	6600	7600	/F* @LOOKUP(C6-B6,A7..C7)	
7) -1500	0		1	
8) 0	0		11	

Note that the number 11 at C8 will cause 11 asterisks to be printed if quota is exceeded. Use of number 1 at C8 would result in one asterisk; 2 would yield two asterisks; etc. Simulation of "@OR" and "@AND" requires some slightly different arithmetic logic. Again, I use a field of asterisks to indicate a positive result.



BONUS

A salesman gets a bonus if his sales are over \$10,000 or if the number of units he sells is over 35 and his total sales are over \$6,000.

	A	B	C
1) Sales		(Your Input)	
2) Units		(Your Input)	
3) Bonus		/F* 15*(A6 + (A7*A8)	
4) -10000	0		1
5) 0	0		1
6) @LOOKUP(B1-10000,A4..C4)			
7) @LOOKUP(B1-6000,A4..C4)			
8) @LOOKUP(B2-35,A4..C4)			

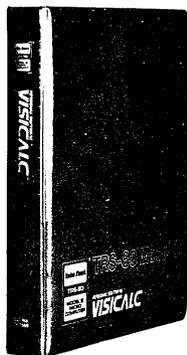
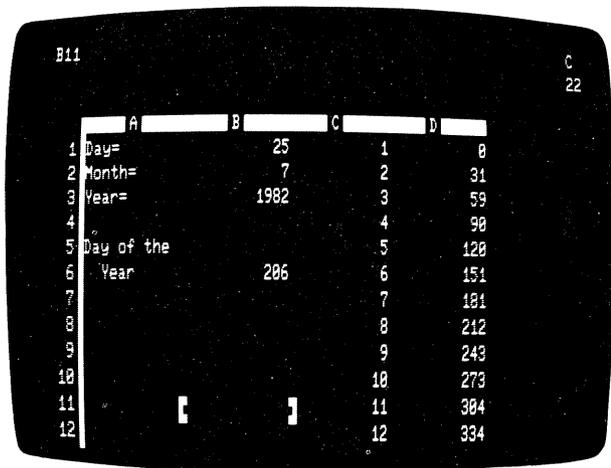
Note that B3 is multiplied by 15 in order to assure that the column is all asterisks when the answer is true.

JULIAN

The example giving the calculation of a Julian Date was complicated by the function for leap years; however, it was still relatively easy to do:

	A	B	C	D
1) Day =		(Your Input)	1	0
2) Month =		(Your Input)	2	31
3) Year =		(Your Input)	3	59
4)			4	90
5) Day of the			5	120
6) Year		(A18*A20) + A17	6	151
7)			7	181
8)			8	212
9)			9	243
10)			10	273
11)			11	304
12)			12	334
13) 0		1	3	
14) 0		0	1	
15) -5		0		
16) 0		1		
17) @LOOKUP(B2,C1...C12)+B1				
18) @LOOKUP(B2,A13...C13)				
19) (@INT(B3/4)*4)-B3				
20) @LOOKUP(A19,A15...B15)				

When this is calculated, it will return the Julian Date at B6. Note the placement of the first table in columns C and D rather than in rows as in the other examples. This allows the use of a lengthy decision table without causing the worksheet to use as much memory as if it were set up as two rows. Thus, a judiciously placed decision table and some arithmetic logic can simulate the more powerful of the new functions in VisiCalc without causing the worksheet to expand beyond the capacity of the system.



Digitizer (Continued from Page 18)

```
8090 CLS
      : PRINT@180, USING "LOCATION:
      (###.##,###.##)"; X;Y
8100 RETURN
```

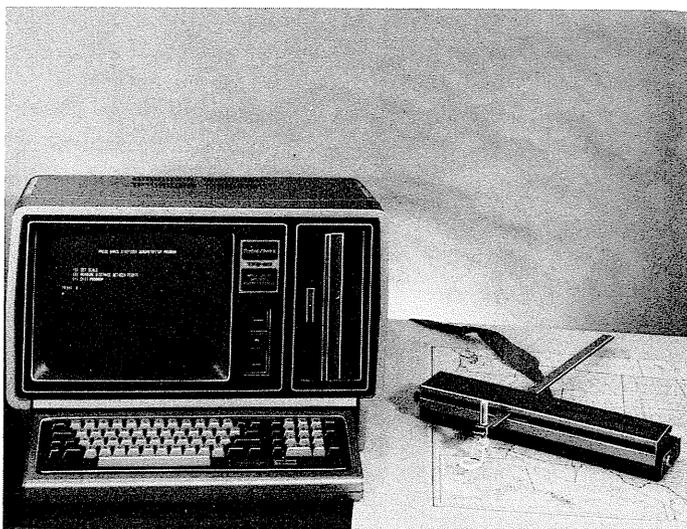
CORRECTIONS FOR THE MODEL II DIGITIZER PROGRAM

Some Model II program lines in the Digitizer manual need clarification. In the introductory paragraphs for Appendix A in the Digitizer manual, it is indicated that the shaded areas of the sample programs include the driver and initialization routines and would therefore have to be included in any program written to use the Digitizer. In the Model II program, line 150 and lines 360-370 should be included in the shaded area. In order to get a proper value back into the `USR`, the `DEFINT I-N` in line 150 has to be included and lines 360 and 370 are `DATA` lines which contain the code for the Machine Language routine that allows the data to be read from the Digitizer.

Additionally, there is an error in line 7060. The right arrow should be replaced with a `^` symbol for exponentiation. Line 7060 should read:

```
7060 DE=DE+J*16^(3-I)
```

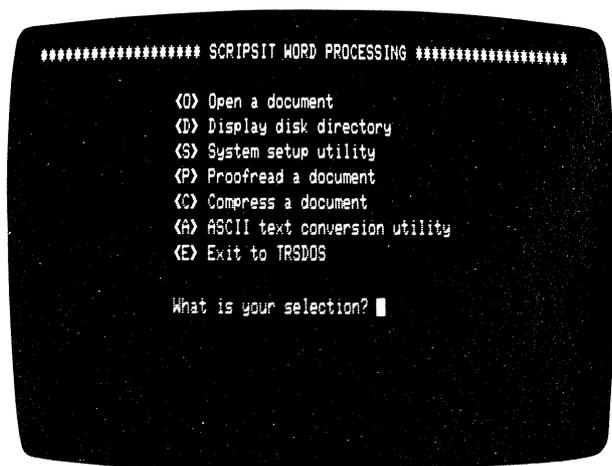
Because in TRSDOS 2.0a `SETCOM` is a library command, after line 200 or 4030 (`SYSTEM"SETCOM". . .`) is executed you will not return to BASIC to continue execution of the program but will remain in TRSDOS. By deleting lines 200 and 4030 from the BASIC program, initializing the RS-232 port by doing the `SETCOM` while still in TRSDOS, and then running the BASIC program, this problem should be eliminated. The error trapping routine to check RS-232 initialization in lines 190 and 9000-9030 will never be utilized because as soon as line 200 is executed you will be left in TRSDOS.



SuperSCRIPSIT and SCRIPSIT 2.0 — A Comparison

SuperSCRIPSIT for the Models I/III and SCRIPSIT 2.0 for the Model II are powerful word processing programs. There are many similarities in the two programs and some notable differences.

For Openers . . .



Screen 1 — An example of a Model III SuperSCRIPSIT main menu.

The first option (O) is selected to open an existing document or to create a new document. To get a disk directory select (D). The System setup utility will be accessed by pressing (S). System defaults for other menus (e.g., Printer options) can be changed by choosing one of the options provided through the system setup utility. (P)roof reading is used to check the spelling of a document. (C)ompress rewrites a document to a file using the least possible space. ASCII text conversion (converting an ASCII file to SuperSCRIPSIT or a SuperSCRIPSIT document to ASCII format) is done through option (A). (E) is self explanatory. (R) will return the user to the current document at the current page/cursor position.

Screen 2 shows the combined disk directory and main menu of SCRIPSIT 2.0. A lot of information about the document files including date created and last worked on, pages in length is displayed in this directory. At the bottom of the screen additional operations are indicated that can be initiated from the directory such as Open (open a file), Copy (copy a document to another file), Print (print a document), F1 Create (create a new document).

WRITING AND EDITING FEATURES

Cursor Movement

Both programs make it possible to move the cursor freely through text. Pressing any of the four arrow keys will



Screen 2 — An example of a SCRIPSIT 2.0 directory or main menu

move the cursor in the direction of the arrow. The arrow keys are repeating in SuperSCRIPSIT while in SCRIPSIT 2.0 holding down the (REPEAT) key with the arrow key will cause the cursor to continue moving through text in the direction of the arrow until it reaches the beginning or end of the document.

SuperSCRIPSIT has the ability to define cursor movement by the next or previous word, paragraph, page, or occurrence of a search string. Cursor movement can be used to access the header (O or E) or footer (O or E) pages.

SCRIPSIT 2.0 uses (CTRL) (N) for next page, (CTRL) (P) to get the previous page, and the global utility to search for a word or string. Any page can be "gotten" or displayed on the screen by doing a (CTRL) (G) and entering the number of the desired page.

Editing Existing Text and Text Block Manipulation

Characters can be typed over, and text can be opened to allow insertions of characters, words, sentences, or blocks of text.

Both SuperSCRIPSIT and SCRIPSIT 2.0 include the ability to define a single character, a word, a sentence, a paragraph, or define text below the cursor as a block of text. In SCRIPSIT 2.0 a block of text can be defined above the cursor. In SuperSCRIPSIT a block can be defined across pages of text (more than one page of text at a time in a single block) while SCRIPSIT 2.0 allows a maximum of one page to be defined in a block of text.

Once the block of text is defined it is possible to:

- Reformat the block
- Move the block

- Delete the block
- Hyphenate the block
- Copy (duplicate) the block
- Lock the block (SCRIPSIT 2.0)
- Freeze the block (SuperSCRIPSIT)

By reformatting a block, the margins, indent tab, line spacing, tabs, etc. can all be changed. Moving the block means that a defined block can be moved from where it is to another document position either in the same or a different document. The copy block command is useful for duplicating text.

When a block of text is locked or frozen any reformatting that is done does not affect that block until it is unlocked or unfrozen. In SCRIPSIT 2.0 the block lock feature is used primarily to prevent a block of text, such as a table or chart, from being split up during a repaginate. If the block of text containing the table or chart is locked, then it will be printed in a block of text on one page. In SuperSCRIPSIT the block is literally frozen so that it cannot be edited. Printing will break at the usual page ending.

SuperSCRIPSIT has the option of defining a single block of text within a document and printing it without printing the rest of the document. In SCRIPSIT 2.0 a range of pages to be printed may be specified. In a 10-page document, page five only can be printed by entering the number five as the beginning and ending page number.

Both SuperSCRIPSIT and SCRIPSIT 2.0 allow one to delete a page, reorder or move pages. SCRIPSIT 2.0 has a utility to renumber pages. SuperSCRIPSIT is document oriented, which means that it looks at the document as one continuous block, while SCRIPSIT 2.0 is page oriented and looks at the document a page at a time.

Line Spacing

In SuperSCRIPSIT line spacing may be defined when the document is opened if it is a new document (one not already created), or if it is an existing document, then line spacing is changed by defining a block and changing the line spacing of the block. SCRIPSIT 2.0 document line spacing may be changed during a repaginate or in a defined block.

The line spacings available in the two programs are illustrated by the following chart.

Line spacing	SuperSCRIPSIT	SCRIPSIT 2.0	Blank lines b/t text
Single space	1	1	No blank lines
Double space	2	2	One blank line
Triple space	3	3	Two blank lines
Fourth line	*NA	4	Three blank lines
1½	1+	6	Half line
2½	2+	7	1½ lines
3½	3+	8	2½ lines
No space	NA	5	No space at all between lines

*NA—Not available in this program

Global Commands

Globally finding, deleting, or replacing a string of characters can be done very quickly in both programs. (Globally refers to every occurrence of the string in the document.) In SCRIPSIT 2.0 the global find, delete, and replace instructions are executed in a forward direction only. This means that in a global search for a string, the search begins at the current cursor position and searches

through to the end of the document for the string. SuperSCRIPSIT can also perform these functions in a reverse direction which means that the search can begin at the current cursor position and go back through text to the beginning of the document.

Paragraph Centering, Cascading Margins, and a Margin Command

SCRIPSIT 2.0 will center one line paragraphs where SuperSCRIPSIT will center each line in a multi-line paragraph. Cascading margins is a handy feature of SCRIPSIT 2.0 for entering text in an outline form. The margin command in SuperSCRIPSIT provides the ability to instantly change the indent tab, left or right margin of a paragraph.

Repagination

SuperSCRIPSIT automatically repaginates. The program keeps track of the line spacing and lines per page that are set in the open document options by either the default value or a value entered when the menu is on the screen. As soon as the last line of the page is typed in, the program begins a new page. In the status line at the bottom of the screen, the program advances the page number indicator by 1 and resets the line number indicator to 1. In SCRIPSIT 2.0 repaginate is a utility.

Format (Tab) Line

In SCRIPSIT 2.0 twelve format or tab lines (1-11 plus the default format line) can be pre-defined. Ten tab lines plus a default or system tab line can be pre-defined in SuperSCRIPSIT. There is a major difference between these two programs regarding the format line that is displayed when moving through a document. In SCRIPSIT 2.0 the last format line recalled or stored with the document is the one displayed. In SuperSCRIPSIT the format line for the current paragraph is displayed. Therefore, in SuperSCRIPSIT if the paragraph format changes from one paragraph to another then the format line displayed changes accordingly. The format line can be edited and tabs are set in the format line. The tabs remain active in the line for as long as it exists.

Align Tab in the Format (Tab) Line

An align tab is used to align data by a defined character. The decimal point is normally used as the align character. This means that a list of decimal numbers (or any other characters) can be aligned by a decimal point as they are entered. Consider the three numbers below for an example.

```

12.44
3000.11
      .12

```

The default align tab character is a decimal point (period) in both programs, but can be changed to another character (e.g., a space).

User-defined Keys

SuperSCRIPSIT allows up to 10 user-defined keys with up to 127 key strokes stored in each key. User-defined keys are very useful for storing a sequence of characters and/or instructions that are used frequently. Refer to View from the 7th Floor page 4 for an example of a typical letter signature block that might be stored in a user key. SCRIPSIT 2.0 allows up to 20 user-defined keys each containing

up to 256 key strokes. User-defined keys can be edited easily in SuperSCRIPSIT or SCRIPSIT 2.0.

Scrolling

Text can be scrolled horizontally and vertically. SCRIPSIT 2.0 scrolls horizontally in one character increments while SuperSCRIPSIT scrolls horizontally in eight character increments.

Automatic Word Wraparound

Both programs support automatic wraparound of text. If the word being typed exceeds the amount of space at the end of the line then the whole word is moved down to the next line, hence the term wraparound. This keeps the word intact and eliminates the worry of running out of space at the end of a line.

Forced End of Page

The end of a page of text can be forced in either program by inserting the proper symbol. When the symbol is encountered during printing the program advances the paper to top of form so that it's ready print at the top of the next page.

PRINTING FEATURES

Headers and Footers

Multiple headers and footers can be used in SCRIPSIT 2.0 while SuperSCRIPSIT allows one header and/or footer for even numbered pages and another for odd numbered pages or one header or footer for both even and odd numbered pages.

Background Printing

Background printing in SCRIPSIT 2.0 means that while one document is printing, the operator can work on a different document. This feature is not available in SuperSCRIPSIT.

Form Letters

A form letter is one that is printed repeatedly, and each time that it is printed, new values (e.g. name, address) are inserted in preset spaces.

SuperSCRIPSIT and SCRIPSIT 2.0 have the capability of creating a form letter (file) and a separate variables file (one which might contain the names and addresses) and then merging the two files together to print personalized letters.

User Print Codes

User print codes are used to embed code which contains instructions to the printer in a document. When the embedded code is encountered in the printing process, the printer executes the instructions stored in that code. This feature can be used to create special characters, issue back spaces, line feeds, etc. As an example, a user print code might be used to print a trade mark (™) symbol which cannot be entered directly from the keyboard.

Twenty-six SCRIPSIT 2.0 user print codes (A-Z) (up to six two-character hex codes) can be defined.

SuperSCRIPSIT provides for twenty user print codes, BUT each code can have up to eleven instructions stored in it. With the ability to store so many instructions, the user print keys can be quite versatile. Any of the numeric keys and shifted numeric keys (!, ", #, \$, %, &, ', (,), @, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0) can be assigned a print code.

Justification

SuperSCRIPSIT offers two methods of justification — Proportional and Monospaced. In proportional justification, the program inserts partial spaces between the words to fill out the line and even up the right margin. SuperSCRIPSIT supports true justification in the proportional print mode. Monospaced justification inserts whole spaces between words to fill out the line to even up the right margin.

SCRIPSIT 2.0 allows justification by three different methods — Space Insertion, Word Increments, and Character Increments. The space insertion method inserts full spaces between words to spread out the line. In order to spread out the line when using word increments to justify, the system adds a fraction of a space to the space between words. When using character increments a fraction of a space is added between characters to spread out the line. SCRIPSIT 2.0 does not right justify text when using proportional characters.

Drivers, Sheet Feeder, and Envelope Feeder

Both of these word processors make provision for the user to write a custom serial printer driver, and information is provided to assist in doing this. The sheet and envelope feeders for the Daisy Wheel II can be used with SuperSCRIPSIT and SCRIPSIT 2.0.

Pause Printing

A pause printing code within a SuperSCRIPSIT or SCRIPSIT 2.0 document halts printing of that document until the computer is instructed to resume printing. Pause printing makes it possible to change the ribbon, print wheel, etc., whenever desired in a document.

Screen Print

SCRIPSIT 2.0 also has a screen print function built into it. From anywhere in the program you can press <CTRL> and without releasing <CTRL> pressing <.> to print the screen contents.

More print features

Feature	SCRIPSIT 2.0	SuperSCRIPSIT
Superscripts	Yes	Yes
Subscripts	Yes	Yes
Underline	Yes	Yes
Double underline	Yes	Yes
Half line spacing	Yes	Yes
Headers/Footers	Yes	Yes
Multi-column print	Yes	Yes*

*While both programs support multiple column print, SuperSCRIPSIT handles a header differently than SCRIPSIT 2.0. On a SuperSCRIPSIT multi-column page the header is printed only once. SCRIPSIT 2.0 will print the header each time that a column of text is printed.

SPECIAL FEATURES

Double-Space Display Character

SuperSCRIPSIT displays a Δ (delta) character on the screen to indicate two spaces. Many typists type two spaces after the end of a sentence. When a sentence ends at the end of a video line, SuperSCRIPSIT uses delta to avoid beginning the next line with a space. The program

program also uses the delta to assure the best possible interline spacing for justified text. The delta can be overridden in order to type two or more spaces in a row.

Utilities

Utility	SCRIPSIT 2.0	SuperSCRIPSIT
Append (assemble) documents	Program feature	TRSDOS utility
Copy	Program feature	TRSDOS library command
Disk format/backup	Program feature	TRSDOS utility
Delete document	Program feature	TRSDOS utility
List directory to screen	Program feature	Names only †
List directory to printer	Program feature	TRSDOS library command
Clock display	Program feature	NA

†A directory containing file size and status, may be obtained through TRSDOS.

ASCII File Conversion

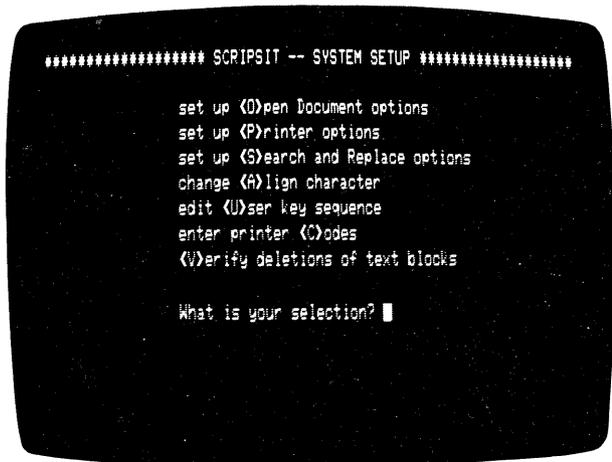
Conversion of a Scripsit file to an ASCII file or an ASCII file to a Scripsit file is available through either program. An ASCII formatted document can be transmitted out an RS-232 port for communication purposes. Another use of the ASCII conversion is to save a BASIC program in ASCII format, load it into Scripsit, edit it, convert it back to an ASCII file, and load and run it from BASIC.

Dictionary Programs

The capacities of the two dictionaries are different but both SuperSCRIPSIT and SCRIPSIT 2.0 can use a spelling dictionary program to proofread a document for misspelled words.

Model II Spelling and Hyphenation Dictionary contains 100,000 words and allows the user to add an additional 2,000 words to the dictionary. Hyphenation can be done automatically using the Spelling and Hyphenation Dictionary program.

The SuperSCRIPSIT dictionary for Model III contains over 73,000 commonly used words, and the Model I version contains 34,000. Both the Model I and Model III dictionaries allow the addition of up to 2,000 words to the existing dictionary. There is no automatic hyphenation feature in this program as there is in SCRIPSIT 2.0.



Screen 3— SuperSCRIPSIT System Setup Menu

Changing the System Defaults

The system defaults can be changed in either SuperSCRIPSIT or SCRIPSIT 2.0. What are system defaults? Certain values are predefined in both SuperSCRIPSIT and SCRIPSIT 2.0 which are used by the system when performing tasks. There is a default format line, page length, printer type, align tab character, etc. It is possible to change these defaults to the values that suit the individual's need. For example, in SuperSCRIPSIT the default for the type of printer is Daisy Wheel II. If the user has a Line Printer VIII, then the DW2 default would need to be changed to LP8.

HELP Menus and Comment Lines

A summary of program instructions can be viewed by the Help menus which are available in either program. There are seven Help screens in SuperSCRIPSIT and three in SCRIPSIT 2.0. SCRIPSIT 2.0 lets you enter a comment line in the text. Comment lines are not printed.

Disk Write Feature

In SuperSCRIPSIT **(CONTROL) (W)** writes the buffer contents to disk and empties the buffer. The buffer can hold up to 11,821 characters. This feature can be helpful where electric current is variable or unreliable and in documents that are unusually long. SCRIPSIT 2.0 writes the current page of text to disk each time another page is accessed.

Both Are Great Programs!

SuperSCRIPSIT and SCRIPSIT 2.0 are both great programs. Word processing has come a long way in the past few years and these two programs reflect the high standards and capabilities which should be present in the sophisticated word processors of today.

Comparison Chart of SCRIPSIT 2.0 and SuperSCRIPSIT

Print Features	SCRIPSIT 2.0	SuperSCRIPSIT
Background print	Yes	No
Print	Yes	Yes
User Print Codes	26 x 6	20 x 11
Merge	Yes	Yes
Justification	3 types (Character Incremental, Word Incremental, Space Insertion)	2 types (Monospaced, Proportional)
Superscript	Yes	Yes
Subscript	Yes	Yes
Boldface	Yes	Yes
Underline	Yes	Yes
Double underline	Yes	Yes
Multiple column print	Yes	Yes
Half-line spacing	Yes	Yes
Pause printing	Yes	Yes
Comment lines	Yes	No
Header/footer	Yes	Yes
Page numbers	Yes	Yes
Date/time	Yes	No
User Printer Driver	Yes	Yes
Sheet feeder	Yes	Yes
Print screen contents	Yes	No
Print single block	No	Yes
Print single page	Yes	Yes
Proportional printing with right justification	No	Yes
Double-space display char	No	Yes

Edit Features

	SCRIPSIT 2.0	SuperSCRIPSIT
Access Commands:		
Get page/block	Yes	Yes
Insert page/block	Yes	Yes
Delete page/block	Yes	Yes
Reorder page/block	Yes	Yes
Renumber page	Yes	No
Global find, delete, replace		
Reverse direction	No	Yes
Text block manipulation:		
Define character	Yes	Yes
Define word	Yes	Yes
Define sentence	Yes	Yes
Define paragraph	Yes	Yes
Define above cursor	Yes	No
Define below cursor	Yes	Yes
Define across pages	No	Yes
Reformat block	Yes	Yes
Move block	Yes	Yes
Delete block	Yes	Yes
Hyphenate block	Yes	Yes
Lock/Freeze block	Yes	Yes
Character insert/delete	Yes	Yes
Repagination	Utility	Automatic
User-defined keys	28 x 256	10 x 127
User-defined keys editor	Yes	Yes
Paragraph Format	Yes	Yes
Format line display	Last line displayed	Line for current paragraph displayed
Unique file format		
ASCII file conversion	Yes†	Yes†
Decimal aligned tabs	Yes	Yes
Outline mode (indentation)		
Word wrap	Yes	Yes
Horizontal scrolling	One character increments	Eight character increments
Vertical scrolling	Yes	Yes
Full video mode	Yes	Yes
Full cursor control	Yes	Yes
Center/uncenter paragraph		
Help	Yes	Yes
Format line editing	Yes	Yes
Forced end of page	Yes	Yes
Recall text block	Yes	Yes
Local (quick) search	Yes	Yes
Cascading margins	Yes	No
Proportional spacing with right justification	No	Yes
Utilities		
Spelling Dictionary	Yes	Yes
Automatic Hyphenation*	Yes	No
Append (assemble) documents		
Copy	Yes	No†
Disk format/backup	Yes	No†
Delete document	Yes	No†
Change system defaults		
List directory to screen	Yes	Yes
Compress document	No	Yes
Clock display	Yes	No
Return to current document at current page/cursor position		
	No	Yes

*This feature is available in the SCRIPSIT 2.0 Hyphenation and Spelling Dictionary.

†The document file formats of SCRIPSIT 2.0 and SuperSCRIPSIT are not compatible.

†These are available as TRSDOS utilities and can be done from TRSDOS Ready

Model I/III Bugs, Errors, and Fixes

Note to Users:

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

Changes to BASIC programs

There are general procedures that need to be followed when any corrections are made.

1. Make a backup of the tape or disk that contains the program to be corrected. Changes should be made on the backup copy.

2. Load the program to be changed by typing CLOAD"filename" (tape) or LOAD"filename" (disk) where filename is the name of the program to be modified.

3. Make the line changes indicated in the fix. For existing line numbers, edit or retype the line to match the one in the fix. Enter new lines.

4. Save the corrected program (the one now in memory). Type CSAVE"filename" <ENTER> (tape) or SAVE"filename" <ENTER> (disk) where filename is the name of the program that has been modified.

5. Now make a backup of the corrected tape or diskette.

PATCHES

PATCHes are entered from TRSDOS READY and are used to make corrections to files stored on the disk.

1. Before making a PATCH, back up the diskette that requires modification and make the PATCHES to the backup copy of the diskette.

2. Apply PATCHES according to the information given in your TRSDOS manual.

CASSETTE PORTFOLIO (26-1506)

In Model III Version 3.0, 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 sell are not exactly equal to zero).

The problem is corrected by making the following corrections to the program TTRAN.

```
123 ...IFD!>S(J,0)+.005THEN116
126 ...IFS(J,0)<.005THENGOSUB132
```

The ellipses (. . .) indicate that the rest of the line up to that point remains as is.

STOCKPAK (26-1507 Version 3.0)

The Model III program demonstrates erratic performance on printed reports (syntax errors, reboot, etc.). The problem is corrected by making the following change to the program PTREPORT.

```
350 IFPEEK(16425)>50 THENPRINTCHR$(12);
:POKE 16425,1:Q%=Q%+1ELSERETURN
```

Add the following line:

```
7 POKE 16412,255
```

Add the following lines to the program REPRUN/BAS.

```
5 POKE 16412,255
1503 POKE 16425,1
```

ACCOUNTS PAYABLE (26-1554)

Following are two sets of corrections for Model I Accounts Payable (26-1554). The first corrections concerns adding vendors to APS, and the second set of procedures for Model I users concerns the ability to print out a preview of checks.

1 — When vendors are deleted in 3.0 and prior versions of AP, the system does not deduct the deleted vendors from the total number of vendors so more vendors may be added.

Make changes to the appropriate version of the program APS.

MODEL I VERSION 3.0

```
161 P1=0:J=VN:IFPV=VN-1-VD THENPL=704:W1$=
"VENDOR FILE IS FULL"+STRING$(24," ")
:GOSUB121:GOTO157ELSETN=0:FORW9=1TO7
:V$(W9)="":NEXT
```

MODEL I VERSIONS PRIOR TO 3.0:

```
510 P1=0:J=VN:IFPV=VN-VDTHENPL=704:W1$="VENDOR
FILE IS FULL":GOSUB430:GOTO500ELSEVV=0
:FORW9=1TO7:V$(W9)="":NEXT
```

2 — In the Model I versions, if you cannot print out a preview of checks it is because you have not selected an invoice to print a check for. (All Versions)

Symptom = If the CHECKS MENU (P) option is used and nothing happens, then execute the following steps.

1. Press the @ key — this returns program to MAIN MENU.
2. Press the (I) key — This loads the invoice selection program once MENU appears.
3. Press the (S) key — to select invoices.
4. Invoices must be posted, then one of the options: (A) option for ALL INVOICE or (V) option for VENDOR/INVOICE or (D) option for DISCOUNT/DUE DATE must be used to select invoices.

If any invoices have been put on hold and you wish to include them in the selection process, they must be released first.

If DISCOUNT DATE is chosen and the checks printed are different from that date, it is caused by a invoices being entered with a discount date of 00/00/00.

ACCOUNTS RECEIVABLE (26-1555)

Status Code 4 (over 90 days overdue) does not return to Status Code 1 (current balance due) when previous balance is paid. (Model III Version 3.0 and prior)

If you have Model III versions 3.0 or prior, please contact your local Radio Shack Computer center or Store. For version 3.1 order stock #700-2217 (proof of purchase required).

DISK PAYROLL (26-1556 Version 2.0)

In Model III Version 2.0, you cannot exempt employer from FICA on tips. The program will appear to accept the exemption, but will double employee amount when showing FICA payable on Journal.

If you are running Model III Payroll with a version number prior to 3.0, contact your local Radio Shack Computer Center or Store for Version 3.0 (stock #700-2221, proof of purchase required).

MANUFACTURING INVENTORY CONTROL (26-1559)

1 — In Model I MIC, it has been found that a problem can occur in Finish Good Maintenance which causes the machine to lock up when updating the files if more than 300 raw materials are in the finished good. (All versions prior to 3.0)

Version 3.0 of Manufacturing Inventory Control is available through your local Radio Shack Computer Center or Store (stock #700-2223, proof of purchase required).

2 — In Model I version 1.2, if Drive 4 (Data Disk) is not ready on entry, the Raw Material Use Report will be inaccurate.

Version 3.0 of Manufacturing Inventory Control is available through your local Radio Shack Computer Center or Store (stock #700-2223, proof of purchase required).

DISK SCRIPSIT (26-1563)

The following PATCH to Model III Disk Scripsit, Version 3.2 on TRSDOS 1.3 will allow documents created under Tape Scripsit at 1500 baud to load.

```
PATCH SCRIPSIT/CMD (ADD=530A,FIND=1142,CHG=0000)
```

MEDICAL OFFICE SYSTEMS (26-1568)

Below are corrections for two problems with Medical Office System.

PROBLEM 1 — Medical Office System contains a bug which may cause problems if more than three patient data files are used. (Model I/III, Version 1.0)

Changes to Medical Office Systems are somewhat critical. Make any changes carefully.

Step 1 — Make the following changes to the program ALPHASOR.

```
1240 FZ=INT((AC-1)/Z1):IFFZ<>FOTHENFO=FZ:CLOSE3
:FM$="PDATA"+CHR$(48+FO)+"":3":NU=3:GOSUB400
:CLS
1250 LR=AC-FO*Z1:PR=INT((LR-1)/2)+1:SR=LR-2*
INT((LR-1)/2)-1
```

Step 2 —

```
SAVE "ALPHASOR"
```

Step 3 — Next add the following lines to the program named MENU.

```
120 FZ=INT((AC-1)/Z1):IFFZ<>FOTHENFO=FZ:CLOSE3:
FM$="PDATA"+CHR$(48+FO)+"":3":NU=3:GOSUB400
140 LR=AC-FO*Z1:PR=INT((LR-1)/2)+1:SR=LR-2*
INT((LR-1)/2)-1
1400 DEFFNA(X)=PEEK(X)+PEEK(X+1)*256:E=
FNA(&H40A4)
```

```
1410 S=FNA(E):IF FNA(S+2)<1000 THEN E=S:GOTO1410
1420 PRINT"TP=TP+S-FNA(&H40A4)"IN LINE 1110":END
```

Step 4 — Next type:

```
GOTO 1400 <ENTER>
```

(The computer will pause and print TP = TP + NNNN in line 1110. The NNNN will be a number.)

Step 5 — Type:

```
LIST 1110 <ENTER>
```

(If the number after TP = TP + ... does not match the computer generated number from Step 4, line 1110 must be edited to contain the new value.)

Step 6 — Type:

```
EDIT 1170 <ENTER>
```

Line 1170 should look like this:

```
1170 DATAMONSORT,PRINT STATEMENTS,BLLPRT,SET UP
SYSTEM DISKS,INIT,16922,16923,16924,1390,
4170,3,16452,16453,16454,660,3960,6
```

If your line 1170 does not match the above line 1170, then edit your line 1170 to match.

Step 7 — Type:

```
SAVE"MENU" <ENTER>
```

Step 8 — Format a new data disk for each existing patient data disk plus any new patient data disks that you might want to add.

Step 9 — Type in the PROGRAM below:

```
10 CLEAR2000:CLS:AC=1:FO=-1:B2=-1
15 IFPEEK(293)=73 THEN Z1=1390 ELSE Z1=660
20 D$(1)="OLD":D$(2)="NEW"
30 PRINT@320,"WORKING ON RECORD #";
40 IFFO<>INT(AC/(Z1+1)) THEN CLOSE1:FO=
INT(AC/(Z1+1)):D=1:GOSUB110:OPEN"R",1,"PDATA"+
HR$(FO+48)+"":1"
50 PRINT@339,AC
60 LR=AC-FO*Z1+1:PR=INT((LR-1)/2)+1:SR=LR-2
*INT((LR-1)/2)-1
70 FIELD1,SR*127ASAJ$,127ASAS$:GET1,PR
80 B1=INT((AC-1)/Z1):IFB1<>B2 THEN B2=B1:
CLOSE2:FM$="PDATA"+CHR$(48+B1)+"":2":D=2:GOSUB11
0:OPEN"R",2,"PDATA"+CHR$(48+B1)+"":2"
90 B=AC-B1*Z1:B3=INT((B-1)/2)+1:B4=B-2*
INT((B-1)/2)-1
100 FIELD2,B4*127ASB1$,127ASB$:LSEIB$=A$:
PUT2,B3:AC=AC+1:GOTO40
110 IFAC>1 THEN MS$="NEXT"
120 PRINT@1,"PUT "D$(D)" PATIENT DATA
DISK"FO+1"IN DRIVE"D
130 PRINT:PRINT"PRESS <ENTER> WHEN READY, OR <@>
TO END"
140 W$=INKEY$:IFW$="" THEN 140
150 IFW$=CHR$(13) THEN RETURN ELSE IFW$="@ " THEN
CLOSE:ENDELSE 140
```

Step 10 — Type:

```
RUN <ENTER>.
```

Follow the prompts, inserting old and new disks as requested. These new disks will be used with the modified MOS disk. DO NOT attempt to use the old patient data disks with the modified program.

PROBLEM 2 — In Model III Version 1.0 if you're using a password of fewer than eight characters, there is a possibility that files may be improperly opened on drive 0.

The problem is corrected by following the steps listed below.

Make the following corrections to the program named MENU.

```
590 GET1,3:NF=CVI(NF$):DT$=DA$:PW$=MID$(PZ$,1,
INSTR(PZ$+" ","")-1):Z1=CVI(Z1$)
:Z2=CVI(Z2$):Z3=CVI(Z3$):DP$=Z4$:BP$=Z5$
:YN$=Z6$:CLOSE1:RETURN
```

Examine lines 1350-1360. They should look like this:

```
1350 DEFFNA(X)=PEEK(X)+PEEK(X+1)*256:E=FNA
(&H40A4)
1360 S=FNA(E):IF FNA(S+2)<1000 THEN E=S:GOTO1360 ELSE
PRINT"TP=TP+S-FNA(&H40A4)":END
```

If they do not look like this, add them as shown.

Type:

```
GOTO 1350 <ENTER>
```

The computer will respond with TP = TP + NNNN (where 'NNNN' is a number).

Edit Line 1110 so that the statement TP = TP + NNNN equals the number you just got from the computer. (Line 1110 looks like this before the change.)

```
1110 CLS:PRINT@467,"** LOADING PROGRAM
**":GOSUB570:GOSUB580:TP=PEEK(16548)+PEEK
(16549)*256:TP=TP+3836:POKE16424,
PEEK(16548):POKE16425,PEEK(16549)
:POKE16549,CINT(TP/256):POKE16548
TP-256*CINT(TP/256):RUNPG$(PR)
```

Finally,

```
SAVE"MENU"
```

with all the changes.

ADVANCED STATISTICAL ANALYSIS (26-1705)

Rounding errors appear in the standard deviation output in the Chi Square Analysis program. (Model I/III Version 3.0 & Prior)

The problem is corrected by following the steps listed below.

CLOAD the Chi Square Analysis Tape.

Make the following change:

```
280 FORI=1TONR:FORJ=1TONC:CS=CS+(ABS(O(I,J)
-E(I,J))-CC)^2/E(I,J)
```

Type:

```
CSAVE"A"
```

to save the changes in the program.

NOTE: The exponentiation operator "" is made by typing the up-arrow, but on the Model III, it will appear on the screen as a left bracket rather than an up arrow.

COBOL (26-2203)

When using the TAB option with the ACCEPT statement in an indexed file, improper program halts occur, contrary to the usage described in pages 139-144 of "RSCOBOL" in manual 26-2203. (Model I, Version 1.3b)

To correct the above problem, APPLY the following patch by typing the following line.

```
PATCH RUNCOBOL/CMD (ADD=AE61,FIND=C3F6A9,
CHG=AFC900)
```

NETWORK 3—New Capabilities for the Classroom

Radio Shack's new Network 3 Controller gives non-disk student stations most of the capabilities of disk-equipped systems. Using the Controller, a teacher can create a classroom computing network by connecting as many as 16 TRS-80 Model III non-disk systems to one Model III disk system. The disk system acts as a "host" computer and the non-disk systems act as "student stations." This new Network 3 system gives the non-disk student stations the following special disk-related capabilities:

1. Stations can load and save programs at the student station keyboard, using the disk at the "host" system.
2. Stations can use a line printer attached to the "host" without interrupting other students or the teacher.
3. When running student management programs, stations can automatically send information such as student scores to the host disk. This allows class records to be kept in one location.
4. Student stations can let computer science students learn and use Disk BASIC on non-disk Model III systems.



THE DIFFERENCE BETWEEN NETWORK 3 AND NETWORK 2

The Network 3 Controller differs in important ways from Radio Shack's current Network 2 Controller and the earlier Network 1 model. Network 2 and Network 1 systems are cassette-port based. Active commands come from the "host" computer. The teacher loads a program into the central disk-based host and then sends the program through the network to the non-disk stations. One program can be loaded into all of the student stations at once, or different programs can be loaded into different stations. Once a program is loaded into a student station, the student can work independently with the program at his or her own pace.

The major advantage of the Network 2 Controller is its ability to save time for the teacher and help organize classroom use of computer resources. Network 2 centralizes

software management, keeps control of loading programs in the hands of the teacher, and gives the classroom the increased efficiency of the disk system at low cost.

The Network 3 system, in contrast, is an RS-232 based network. Most active commands come from the student stations. The teacher determines what choices (if any) will be available to the student by putting the proper program diskettes into the host's disk system. Then the student loads the program into his or her station by entering the necessary commands at that station's keyboard. Students can save and load programs directly to and from the disk, open and close data files, and print out program listings and other information. Student management data can be sent to the host disk for certain programs.

If you want your network system to be used by programming students or other students who will need to print out information, by students who are working with a student management program, or by a group of students who will be working independently with a wide variety of disk or non-disk programs, the Network 3 Controller will provide the flexibility and classroom efficiency you are looking for.

The Network 3 Controller performs more advanced networking functions than the Network 2 Controller but does not replace Network 2. The two controllers do different things. Your own school's individual needs will determine whether you'd benefit most from a Network 2 Controller, a Network 3 Controller, or a configuration that would allow you to use both as needed.

Both Network Controllers allow each participating TRS-80 student station to be used as an independent, cassette-based, stand-alone unit in addition to acting as part of the network. This flexibility helps make the network an ideal configuration for the classroom.

Now let's look a little more closely at the student station capabilities available through Network 3.

WORKING WITH DATA FILES AT NETWORK STUDENT STATIONS

With the Network 3 system and Network 3 operating software, students can open data files from student stations, put data into the files and get data from them, and close the files. Different stations can work with different data files simultaneously.

PRINTING FROM A STUDENT STATION AND "SPOOLING"

Students working at student stations can send printing requests to the host computer. From there, the material to be printed is sent to the printer attached to the host.

If more than one student station sends a printing request at the same time, a function called "spool" will store the material to be printed until the printer can get to it. Since the spool accepts information faster than the printer can print it, the student station can return to its program without delay.

If students will be printing out small amounts of material, you may want to keep the printer off during the class period and allow the "spool" storage space to fill up as students place their printing requests. Near the end of the class period, you can turn on the printer and print out all of the material at once. This technique will keep printer noise from interrupting other activities.

STUDENT MANAGEMENT

With Network 3, it is now possible to use Radio Shack's K-8 Math With Student Management Program in a network configuration. If you use 32K student stations (required by the K-8 With Student Management Program), you can load and run the program normally on a student station just as if you were running the program on a disk-based computer. At the end of each session, the system transfers student scores from the student station to the host computer, where they are stored on the K-8 Math Program diskette.

To view the scores stored on the diskette, the teacher types a special code at any student station which is running the K-8 Math Program. Student scores will appear on the screen at that station.

PROGRAMMING IN DISK BASIC AT THE STUDENT STATION

Numerous Disk BASIC commands are available at the student stations. The Network 3 Operating Software User's Manual contains a "quick reference" section that lists and describes the available commands.



EQUIPMENT REQUIREMENTS

The minimum "host" computer for the Network 3 system is a TRS-80 Model III with 32K of memory, one disk drive, and an RS-232C interface. A "student station" needs a minimum of 16K of memory, Model III BASIC, and an RS-232C interface. The Network 3 Controller requires one cable for each student station (10 to 100 feet in length) and one cable for the host system. The current operating software (NETWORK 3 OPERATING SOFTWARE, Cat. No. 26-2775) is designed for Model III only. Future versions will permit TRS-80 Model I, Model III, or Color Computer systems to be used as student stations and will permit a Model II (with hard disks if desired) or a Model III to be used as the host computer.

When deciding how much memory you'll need for your student stations, it is important to consider what kinds of programs you'll be running. For computer science applications, where students are writing their own programs, 16K may be sufficient. However, most packaged courseware programs will require a larger memory size. The reason for this is that the Network's operating software takes up 9.5K of memory in each student station, leaving only 6.5K of available memory in a 16K machine. Radio Shack's educational software packages will thus require 32K student stations.

LIST OF RADIO SHACK COURSEWARE FULLY COMPATIBLE WITH THE NETWORK 3

Many Radio Shack software packages are fully compatible with the Network 3 (that is, no modifications to the program are necessary). The packages which are fully compatible include:

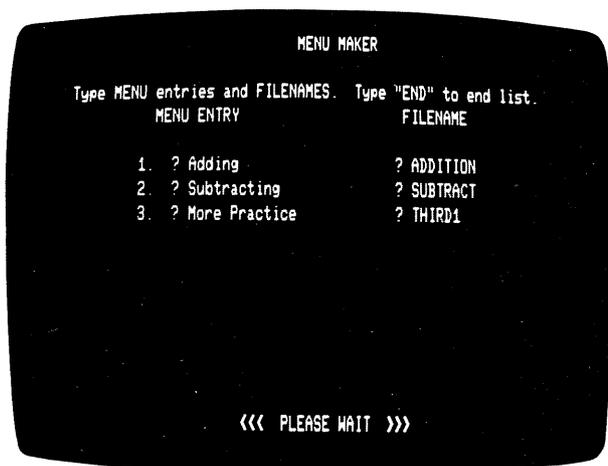
- K-8 Math Program (Cat. No. 26-1715)
- K-8 Math with Student Management Program (26-1725)
- Essential Math Programs, Volumes One and Two (26-1716 and 26-1719)
- Euclid Geometry Tutor (26-1724)
- Advanced Graphics (26-1714)
- Vector Addition (26-1720)
- Graphical Analysis of Experimental Data (26-1722)
- Interpreting Graphs in Physics (26-1721)
- Investigations in Integral Calculus (26-2600)
- Plane Analytic Geometry (26-2602)
- AlphaKey (26-1718)

Again, please note that 32K of memory in the student stations is required to run these programs in the Network 3 configuration.

SPECIAL FEATURES OF THE NETWORK 3 OPERATING SOFTWARE

Included on the Network 3 Operating Software diskette is a BASIC program called "MENU." You can use this program to set up one or more menus that allow students to choose what programs they'll run. Options in the menus correspond to the filenames of BASIC programs that are stored on a lesson diskette at the host computer. When a student selects one of the options you've provided, the menu program automatically runs the chosen program at that student's station.

A demonstration program called "MAILBOX" is also furnished on the Network 3 diskette. It allows students to send messages to one another from the student stations.



These messages are stored on the Network 3 diskette at the host computer. The teacher (working at a student station) can post a "bulletin board" message, or a message that will be seen at all student stations. These bulletin board messages cannot be erased by the students.

HOW YOU CAN USE THE NETWORK 3

Network 3 hardware and operating system software is available through your local Radio Shack store or Computer Center. The listed price of the Network 3 Operating System software (Cat. No. 26-2775) is \$149.00. (Prices, however, may vary at individual stores and dealers.) For more information on Network 3 operation and educational applications, contact your local Radio Shack retailer, or contact the Radio Shack Regional Educational Coordinator in your area.

Compress Graphics Using Model III DEBUG

Dennis F. Tanner
Ft. Worth, Texas

For the May, 1981 Microcomputer News I wrote an article on using the VARPTR instruction to compress graphics on the Model I and Model III disk and non-disk systems. DEBUG, a utility included in Model III TRSDOS, provides an easier way.

Let's take a moment to review. Graphics may be produced on the TRS-80 Models I and III in a variety of ways: using the SET and RESET statements, POKEing character codes into the video RAM, building them using character codes, and using the VARPTR to compress the graphics. The result of using Model III DEBUG is similar to the result of using VARPTR, but the process is much simpler.

DEBUG is a powerful TRSDOS utility which lets you examine, edit, and (if you don't use it properly) really mess up disk files. For this reason, I recommend that you use a BACKUP copy of a TRSDOS disk to learn this process.

On your Model III disk system, go into BASIC and type in the following program:

```
10 A$=" " ' 8 SPACES
20 PRINT"LENGTH=" LEN(A$)
30 PRINT"A$=" A$
```

Type:

```
RUN <ENTER>
```

to run the program once to make sure you have 8 spaces inside the quotes in line 10. Then save it in ASCII format with this instruction:

```
SAVE"DEMO",A <ENTER>
```

Return to TRSDOS with this instruction:

```
CMD"S" <ENTER>.
```

At "TRSDOS Ready" go into DEBUG by typing the instruction:

```
DEBUG <ENTER>.
```

You will see the contents of some memory locations, the contents of the locations following the program counter, and the contents of the various registers. The cursor will be blinking in the lower left-hand corner. Type:

```
F
```

The "Filespec:" prompt will appear. Type:

```
DEMO <ENTER>.
```

You are now looking at the contents of the disk file. Since the file was saved in ASCII format, the program is legible on the right side of the display. The center area shows the hex (base 16) value of each byte in the program. "31" represents 1 in the line number "10," "30" represents 0, "20" represents the space, and so on.

You can now modify this disk file in DEBUG. Type:

```
M
```

A graphics block now covers the "31" in the upper left hand corner. To compress some graphics, we want to modify the contents of the quotation marks in program line 10. Using the arrow keys, position the cursor over the first "20" in the center of the top line of the screen. We will change the string of 8 spaces to a string of 8 graphic characters. Let's use graphics character 191. This is represented as "BF" in hex numbers. To replace the spaces, type:

```
BFBFBFBFBFBFBFBFB
```

Notice that the display on the right side now shows the graphics inside the quotation marks. To make this change on the disk, press:

```
<ENTER>
```

When the disk drive stops, press:

```
<BREAK> <BREAK>
```

to return to TRSDOS. Go into BASIC and type:

```
LOAD"DEMO" <ENTER>
```

When you list the program, you will see that the graphics are indeed stored in the program. Run that program to demonstrate that this process works.

Let's try some fancier graphics. Type in line 10 like this:

```
10 A$=" " ' 27 SPACES
   <ENTER>
```

Run the program once to make sure the length is 27. Then type:

```
SAVE "DEMO1",A <ENTER>
CMD"S" <ENTER>
DEBUG <ENTER>
F
DEMO1 <ENTER>
```

We are going to make a graphic box with character 191 (BF) on the sides, character 131 (83) on the top row, and character 176 (B0) on the bottom row. The tricky part of this is, some special characters are needed to align the characters properly. Let's begin. Type:

M

Position the cursor over the first "20" in the center of the top line. Type:

BF838383BF

to make the top row. Character 26 (1A) moves the graphics position to the next row down in the finished graphic. Type:

1A

to position it properly. (A period appears on the right side of the screen because this hex code has no ASCII representation.)

Now the graphic position must be moved back five spaces with the backspace character 8 (08) to align it under the beginning of the previous line in the finished graphic. Type:

0808080808

The middle row of this 3-line graphic will have character 191 (BF) on each end; in the middle it will have spaces, character 32 (20). Type:

BF202020BF

To align the third row under the second row, type:

1A0808080808

Don't worry if the graphics on the right side of the screen look funny. If you have typed the codes correctly, the graphics will turn out right.

To create the bottom row, use character 191 (BF) and character 176 (B0). Type:

BFB0B0B0BF

To save these changes, press:

<ENTER>

When the drive goes off, type:

<BREAK>

<BREAK>

BASIC <ENTER>

<ENTER>

<ENTER>

LOAD "DEMO1" <ENTER>

When you list it, the result may surprise you. The graphics box appears in the listing. Run the program to see again what is stored in A\$. If you want to insert a number in the box, you can do that in BASIC. Type:

EDIT 10 <ENTER>

17 <SPACE>

C5 <ENTER>

Have fun with DEBUG and graphics!

Poor Man's Editor

Dallas L. Powell
1804 Skyland Drive
Tallahassee, FL 32303

This routine for a 16K Model III and a Line Printer VI was used to write this letter. I call it my Poor Man's Editor.

It needs refinements, but I am a novice with BASIC. (I received the system in October.)

```

100 '          *** E D I T O R ***          11/27/81  REV 22
110 CLEAR 800
      : CLS
      : POKE 16427,62
120 DIM A$(10)
130 A$(1)= CHR$(8)
      : A$(2)=CHR$(9)
      : A$(3)= CHR$(10)
      : A$(4)=CHR$(13)
      : A$(5)=CHR$(32)
      : A$(6)=CHR$(64)
      : A$(7)=STRING$(8,A$(5))
      : A$(8)=CHR$(95)
140 CLS
      : INPUT "NEW PRINT <Y> OR <N> "; Y$
150 IF Y$="Y" THEN B$=""
      : ELSE 370
160 CLS
170 PRINT@ 900,"Type less than 241 characters: To end
      type @"
180 FOR I=1 TO 23
      : SET( 83,I)
      : SET(127,I)
      : NEXT I
190 '
200 I$=INKEY$
210 FOR I=1 TO 6
      : IF A$(I)= I$ THEN 240
220 NEXT I
      : GOSUB 410
      : GOTO 200
230 '
240 ON I GOTO 270, 280, 290, 290, 350, 370
250 '          BK-SP TAB DN-AW ENT SP  "@
260 '
270 Z1= LEN(B$)
      : Z2=Z1-1
      : C$=LEFT$( B$,Z2)
      : B$=C$
      : GOSUB 420
      : GOTO 200
280 B$=B$+A$(7)
      : GOSUB 420
      : GOTO 200
290 Z1=LEN(B$)
300 IF Z1 < 64 THEN Z3=64-Z1
      : GOTO 340
310 IF Z1 < 128 THEN Z3=128-Z1
      : GOTO 340
320 IF Z1 < 192 THEN Z3=192-Z1
      : GOTO 340
330 IF Z1 < 241 THEN Z3=241-Z1
      : ELSE 370
340 Z4$=STRING$(Z3,A$(5))
      : B$=B$+Z4$
      : GOSUB 420
      : GOTO 200
350 B$=B$+A$(5)
      : GOSUB 420
      : GOTO 200
360 '
370 CLS
      : PRINT B$
      : INPUT"HARD COPY <Y> OR <N> ";Y$
      : IF Y$="Y" THEN INPUT"Large or small (L) or
      (S)";L$
      : ELSE 140
380 IF L$="L" THEN LPRINT CHR$(31)
      : ELSE LPRINT CHR$(30)
390 LPRINT B$
      : GOTO 140
400 '
410 B$=B$+I$
420 Z1=LEN(B$)
      : Z2=Z1+64
      : IF Z1=241 THEN 370
430 IF Z1 > 240 THEN 370
440 PRINT@ 64, B$
      : PRINT@ 22, A$(8)
      : RETURN
450 END

```

Scripts Ideas

XERCH

Peter D. Beeson
250 Central Building
Seattle, WA 98104

I use the Model II SCRIPSIT 2.0 Convert Utility with the Dow Jones News Service with the intention of receiving text data over the wire and then editing the data with the SCRIPSIT program. When I first tried to do this, during the process of having the SCRIPSIT load the ASCII file created by the TERMINAL Utility of TRSDOS, I kept receiving an error message before the entire ASCII file was loaded into SCRIPSIT.

I found that the error message resulted from embedded 1E's and 1F's (Hex Notation) that were transmitted along with the text from the Dow Jones News Service. Attached is a short BASIC program that corrects the problem. This program can be executed directly from the "S" option on the TERMINAL menu by typing "BASIC XERCH -F:1". After the corrections have been made, the program immediately loads SCRIPSIT. The SCRIPSIT disk must be in drive 0 and the TRSDOS disk must be in drive 1.

XERCH

```

100 '
110 '
120 ' *** XERCH ***
130 '
140 '
150 ' THE FOLLOWING PROGRAM SCANS AN ASCII FILE FOR
160 ' 1E's AND 1F's (HEX NOTATION) AND REPLACES THEM
170 ' WITH A CARRIAGE RETURN SO THAT THE ASCII FILE
180 ' CAN BE LOADED INTO SCRIPSIT 2.0
190 '
200 '
210 CLS
    : PRINT
    : PRINT
    : PRINT
    : PRINT
215 PRINT@ (10,10), "TYPE IN THE NAME OF THE ASCII
    FILE THAT"
216 PRINT@ (11,10), "IS TO HAVE THE 1E's AND THE 1F's
    REPLACED:"
217 PRINT
    : PRINT
220 LINE INPUT WDS
230 PRINT@ (17,10), "WE ARE AMENDING "; WDS
    : PRINT
    : PRINT
    : PRINT
240 OPEN "D", 1, WDS, 1
250 FIELD 1, 1 AS A$
260 FOR X = 1 TO LOF(1)
270 GET 1, X
280 IF HEX$(ASC(A$))="1F" THEN LSET A$=""
    : PUT 1,X
    : PRINT "1F:"; X
290 IF HEX$(ASC(A$))="1E" THEN LSET A$=""
    : PUT 1, X
    : PRINT "1E:"; X
300 NEXT
310 CLOSE
320 PRINT
    : PRINT
    : PRINT
330 PRINT "THE REPLACEMENT PROCESS IS COMPLETE"
340 SYSTEM "STARTUP"
350 END

```

Underscore with SCRIPSIT 2.0

J. G. Lowder
Communication Architects
P.O. Box 300
Lynnwood, WA 98036

One contribution and one request:

SCRIPSIT 2.0 FAST UNDERSCORE

I very much appreciate the new SCRIPSIT 2.0 software, but I found all the keystrokes required to get words underscored to be cumbersome. In 1.0 Scripsit it was much easier. However, thanks to the User Sequence Editing feature I came up with a solution which others might find helpful, also. Now when I want to underscore a single word or several words, it takes only about half the keystrokes.

Here is the User Sequence, which I set up on User key 7:

F1 ESC ESC ESC _ ESC ESC ESC _ F2 LT F1^Z

The procedure for using the sequence is:

1. PRESS ESC ESC 7
2. TYPE the words to be underscored
3. PRESS F2
4. PRESS →

This sequence is much easier than having to interrupt text development, both at the opening and at the ending of the string to be underscored. It's also less likely that you might end up forgetting to end the underscore and find yourself with a whole page of underscored material.

NOW THE REQUEST: Please publish a list of all the PATCHES for MODEL II. When I call your customer service line, they tell me it is the store's responsibility to advise owners of PATCHES and when I contact the store, they try to help, but they tell me that the owner registration is sent to Fort Worth and it is Fort Worth's responsibility. In any case, I'm sure there must be many PATCHES which are optional or lost between the cracks somewhere. I recently lost an entire disk full of information because I had not been notified of a series of PATCHES which had been put out two months ago. They still haven't appeared in Microcomputer News. HELP!!

Editor's comments:

First, with Scripsit 2.0 all that is needed to begin or end an underscore is **(ESC)(_)**.

The procedure for underscoring a word or phrase becomes:

1. PRESS ESC _
2. TYPE the words to be underscored
3. PRESS ESC _

(Continued on Page 39)

Medical Office System

Here is some information which may be of interest to users of the Model II Medical Office System.

1. If you expand the transaction file past the limits specified in the manual and receive a "Disk Space Full" error message, you should return to a Backup copy and expand the file with a smaller number of records. If you fail to do this, you will receive an error 28 — attempt to read past end of file, during End of The Month Purge processing, and all data may be lost.

2. The Daily Input program is set up to detect an overflow during math operations (posting charges, payments and adjustments). If an overflow occurs, an error message will flash on the bottom of the screen. This error will occur if:

The total due on an invoice exceeds \$9999.99 or is less than \$ - 999.99.

The total due for the patient exceeds \$99999.99 or is less than \$ - 9999.99.

The month-to-date or year-to-date fields on the specified procedure exceed \$99999.99. This can occur when using the technique of entering previous balances with a single procedure called PREVIOUS BALANCE DUE, if the A/R exceeds \$99999.99. Once this occurs, the user can delete the procedure and reenter it to clear the month- and year-to-date fields.

3. The Aged A/R report will drop the most significant digit if the balance due for insurance or patient exceeds \$99999.99. The grand total will be correct up to \$999999.99.

4. If records have been added or changed by Daily Input, it is important that the user do a Daily Recap before running either the Aged A/R Report or EOM Purge.

5. The Insurance Print and Statement Print programs are written in BASIC. Their names are GENERAL/BAS and STATEMENT/BAS. On exceedingly large invoices, either an OS or BS error can occur. You may change the CLEAR and DIM statements as required to eliminate the OS or BS error. Don't forget to resave the program under the appropriate filename. Note—these errors have only been known to occur on invoices with more than 38 line items. Caution: Do not attempt to run either BASIC program directly, even for testing. You must enter these programs through the main menu to ensure that high memory is properly protected.

6. The system will work fine on the Hard Disk system under TRSDOS-HD 4.0. If the files are allowed to grow too large, the in-memory sort of the Aged A/R and Account Summary reports may not be able to sort all records. If this happens, simply narrow the scope of the report by selecting half the patients at a time, or a single doctor at a time.

7. YOU MUST NOT CHANGE THE FILE FORMATS USING PROFILE!!! You may create additional reports using the existing file.

8. To totally reset a medical system, you should use a backup copy of the Master Disk. If this is not available, the following steps must be used:

Pay off ALL open invoices with a WO (write off) payment code.

Run Daily Recap.

Run EOM Purge using a date of 12/99 (this should purge all invoices).

Change the system date to 01/01/1982 and run EOY Reset (EOY Reset won't run except in January.) Reply 1981 to the year that just ended question.

Change the system back to the real date.

If required, Run Daily Input and request the patient deletion option (to delete all patients).

9. You should be aware that the month-to-date and year-to-date figures on the procedure file are not adjusted when an existing charge is adjusted.

More Scripsit Ideas

C. M. Mendenhall
3905 Dogwood Drive
Greensboro, NC 27410

I read the "Scripsit Ideas" article for Model I and III in the April 1982 Microcomputer News with much interest and am sure that there are many ideas TRS-80 owners have which could be of much interest to other owners. I have become so obsessed with the SCRIPSIT program that I have retired my typewriter and do all my correspondence on the TRS-80. I have developed a couple of things in using SCRIPSIT that make it so much more useful that, even though they are pretty simple, enough of your users may not be aware of them that they may be worth printing.

I am using the SCRIPSIT with a Line Printer IV and, like many other printers, it provides several different type sizes and modes. To remind me to set the printer to the mode desired, I use the TRSDOS AUTO command to load it, but I have inserted a PAUSE in the auto routine which requires that I insert either "DO COMPTYPE," "DO PROPTYPE," or "DO NORMAL." It then shifts to BASIC and sends the proper command to the printer for the type required, and then shifts back to TRSDOS and loads SCRIPSIT.

As I use several sizes of stationary, I use several formats for the various types of stationary and type sizes. I have saved on my correspondence file disks all of the formats which I use and named them FORMAT1, FORMAT2, etc. I then set up a file which I call FORMINDX which contains a listing of the various formats with their names and a description of what they do. I can then check the FORMINDX and find the format which I wish to call for a particular document.

I also use another file on each correspondence file disk called CORRINDX which has a listing of the letters contained on that file. When I need to recall a document which has been filed previously, I need only to check the CORRINDX to find the name of the document. When a new document is saved on the disk, its name is added to the CORRINDX, to keep things current.

Model II Bugs, Errors and Fixes

Note to Users:

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

Changes to BASIC programs

There are general procedures that need to be followed when any corrections are made.

1. Make a backup of the disk that contains the program to be corrected. Changes should be made on the backup copy.
2. Load the program to be changed by typing LOAD"filename" where filename is the name of the program to be modified.
3. Make the line changes indicated in the fix. For existing line numbers, edit or retype the line to match the one in the fix. New lines should be entered.
4. Save the corrected program (the one now in memory). Type SAVE"filename" <ENTER> where filename is the name of the program that has been modified.
5. Now make a backup of the corrected diskette.

PATCHES

PATCHes are entered from TRSDOS READY and are used to make corrections to files stored on the disk.

1. Before making a PATCH, back up the diskette that requires modification and make the PATCHES to the backup copy of the diskette.
2. Apply PATCHES according to the information given in your TRSDOS manual.

TRSDOS-HD (26-4150 Version 4.0)

When a fixed length file with a record length of 128, 64, 32, 16, 8, 4, 2, or 1 is closed, with the last record of the file being the last record of a physical sector, an Error 50 (Invalid Space Descriptor) will be generated. This error appears in BASIC as "IE" Error; if the user does "PRINT ERRS\$;" BASIC will show that the error was TRSDOS Error 50. Apply the following patches to correct the above problem.

```
PATCH SYSTEM/SYS R=85 B=131 F=DD8610 C=C3AF1F
PATCH SYSTEM/SYS R=88 B=18 F=0202001E C=010E7420
PATCH SYSTEM/SYS R=88 B=22 F=000000000000
C=DD8610CA6C1E
PATCH SYSTEM/SYS R=88 B=28 F=000000000000
C=D26C1EC36B1E
PATCH SYSTEM/SYS R=88 B=34 F=00000000 C=0202001E
```

When 4800 BAUD operation is specified using either the SETCOM command or the RS232 Initialization Supervisor Call (55), the actual BAUD rate is being set to 9600

BAUD. The following patch is for the TRSDOS-HD 4.0 only and will correct the problem described above.

```
PATCH SYSTEM/SYS R=68 B=139 F=0D C=1A
```

Inventory Control (26-4502 Version 1.1)

INV allows user to change a stock number when quantity on hand is not equal to zero. The problem is corrected by making the following corrections to IMS/BAS:

DELETE LINES 238, 240, 242

```
Line: 234 GOSUB900:PUT1,I(I,0):GOSUB184:
IFKF<>0THENKF=0:N1=I:N2=I(I,0):
I(I,0)=-I(I,0):VN=I(I,1):ND=ND+1:
GOSUB292:NI=NI-1:RETURN
Line: 236 NI=NI-1:GOSUB318:RETURN
```

ADD THESE LINES:

```
199 C1=CVS(F1$(11)):C2=CVS(F1$(12))
900 GET2,3:LSETG1$(0)=MKD$(CVD(G1$(0))-PR!*C1):
LSETG1$(1)=MKD$(CVD(G1$(1))-PR!*C2)
910 LSETG1$(2)=MKD$(CVD(G1$(2))-CS*C1):
LSETG1$(3)=MKD$(CVD(G1$(3))-CS*C2)
920 LSETG1$(6)=MKD$(CVD(G1$(6))-QT*CS):
LSETG1$(7)=MKD$(CVD(G1$(7))-QT*PR!):
PUT2,3:RETURN
```

Type SAVE"IMS/BAS" to save the changes in the program.

Accounts Payable (26-4505 Version 1.0)

When selecting invoices, an unexpected error code 61 in line 1620 can occur. The problem is corrected by making the following corrections to "APS/BAS":

```
Line: 1615 IFP(N,AM)<1THENN=N+1:GOTO1610
Line: 3415 J=P(N,AM):IFJ<1THEN3410ELSEGET2,J:
GOSUB900:GOSUB920:GOTO1800
```

Type SAVE"APS/BAS" to save the changes in the program.

Accounts Payable (26-4505 Version 1.0/2.0)

When using the Print Checks option, the user cannot disable the printing of check numbers on the checks. The problem is corrected by changing "APCHECKS/BAS":

TO ELIMINATE CHECK NUMBER FROM PRINTING ON STUB:

```
Line: 2410 LPRINTL$:LPRINI;NAS:LPRINTL$:
LPRINTL$:RETURN
```

TO ELIMINATE CHECK NUMBER FROM PRINTING ON CHECK:

```
Line: 2515 LPRINTL$:LPRINTL$:LPRINTL$:
LPRINTL$:LPRINTL$
```

Type SAVE"APCHECKS/BAS" to save the changes in the program.

Scriptit 2.0 (26-4531)

If you attempt to create a document on an almost-full SCRIPSIT diskette, no check is made to be sure that there is enough free space to create both the document and its first page. This results in "garbage" being displayed as the first page. To fix the above problem, apply the following patches:

```
PATCH SCRIPSIT A=D68E F=0000000000000000
C=2A0A811128003E96
PATCH SCRIPSIT A=D68E F=0000000000000000
C=B7ED52D8116D4A
PATCH SCRIPSIT A=D695 F=0000000000000000 C=DDE5E1AFC9
PATCH SCRIPSIT/SYS R=15 B=87 F=116D4ADDE5E1
C=C8D6D6C2A93D
```

These patches will prevent SCRIPSIT from creating a document on any diskette that has less than 40 sectors free. After applying the above patches, apply the following patch to change the date in the initialization screen (it should show up as 01/08/1982 before you apply these patches; if not, obtain and apply all preceding patches first):

```
PAICH STARTUP A=E40E F=B0B1AFB0B8 C=B0B2AFB2B3
```

When this patch has been made, the SCRIPSIT initialization screen will show the date 02/23/1982 at the bottom.

Statistical Analysis (26-4540 Version 1.0)

Rounding errors appear in the standard deviation output in the Chi Square Analysis program. The problem is corrected by making the following changes to "CHISQ/BAS":

```
Line: 280 FORI=1TONR:FORJ=1TONC:
      CS=CS+(ABS(O(I,J)-E(I,J))-CC)^2/E(I,J)
```

Type SAVE"CHISQ/BAS" to save the changes in the program.

Statistical Analysis (26-4540 Version 2.0)

In Multiple Regression, the variable names are not being placed in the requested position. The problem is corrected by changing "REGAN/BAS":

```
Line: 120 PRINT:LL%=MEM/8-1000:DIMX$(LL%),
      B$(300),V$(12),X$(12),VN$(12),NN$(12)
Line: 2060 ONERRGOTO0:NR%=CVI(NR$):
      NV%=CVI(NV$):FORJ=1TO12:
      VN$(J)=V$(J):NN$(J)=V$(J):NEXTJ
```

ADD this New Line:

```
2140 FOR DD=1 TO NV:VN$(DD)=NN$(VW$(DD)):NEXT DD
```

TRSDOS (26-4910 Version 1.2)

The following patches (for QUME printers and TRSDOS 1.2a only) will correct LPRINT CHR\$(138) statements so that they cause a space character, carriage return to be issued to the printer. This is compatible with LP III mode.

After the patches are successfully applied, you must re-boot to use the printer. The patches are as follows:

```
PATCH IODVRS/SYS A=0056 F=FFFFFFFFFFFFFFFFFFFF
      C=213F0F3E8AB8C00620C9
PATCH IODVRS/SYS A=0F47 F=213F0F C=CD5600
```

TRSDOS (26-4910 Version 2.0)

Pressing any key during a boot-up sequence initiated by the "RESET" command causes unpredictable results if done prior to the date & time prompts.

The following patches will correct the problem

```
PATCH SYSIEM/SYS R=6 B=133 F=3E01D3F9C30000
      C=3E03D3F3C3C326
PATCH SYSIEM/SYS R=6 B=196 F=E5E5E5E5E5E5E5E5
      C=F33A5500CBADF3FF
PATCH SYSIEM/SYS R=6 B=204 F=E5E5E5E5E5E5E5E5
      C=3E01D3F9C30000
```

Scripsit (From Page 36)

It is not necessary to press (ESC) until you see the function you wish to use. As long as you know what you



want to do, simply press (ESC) once, then the appropriate key for the function you desire. If you forget what to press for a particular function you can repeatedly press (ESC) to see the various functions which are available.

Model II PATCHes: Sorry, but it would be impractical for Microcomputer News to publish a single list containing all the patches and corrections which are available for Model II and Model II software (It is a rather long list!). Also, it really isn't necessary.

In your local area you should have one or more Radio Shack Computer Centers or Expanded Computer Departments. Either of these locations will have at least one individual who is knowledgeable about program fixes and corrections, and he/she has immediate access to a manual which contains all current and past corrections to programs.

This manual is updated constantly with the most current information possible. In fact, the Bugs, Errors, and Fixes which are published in the Microcomputer News appear in this bulletin FIRST, and they are there TWO or THREE MONTHS before you get them here (we don't put them into an issue until they appear in bulletin form, and then it takes about two months before you get the information at home).

Late Note For 26-4604

The COBOL RUNTIME module of 26-4604 Accounts Receivable is version 1.3a. The current version of RUN-COBOL is 1.3b. If you have purchased 26-4604, you should go to your nearest Radio Shack Computer Center or Radio Shack Store with a Computer Department, and request up-grade 700-2023 immediately. This is to your benefit, as version 1.3a is no longer supported and 1.3b allows for error trapping capability that is otherwise unavailable.

You Want to Calculate What?

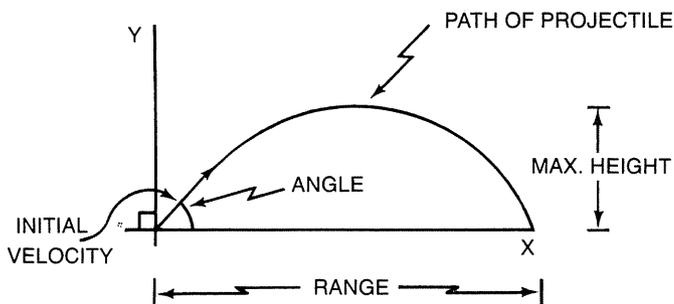
Projectile Trajectory

Larry Mermelstein
17 Bobolink Lane
Northport, NY 11768

I recently purchased a TRS-80 Pocket Computer. Being a high school senior, I felt this would be a wise investment for college. I have also found the PC great for high school work. For all those students that are having trouble with Physics, here is a program that computes the trajectory of a projectile ejected at an angle "B" with an initial velocity "A."

```

5:  "A"
   : DEGREE
10: PAUSE "PROJECTILE TRAJECTORY"
15: INPUT "ENGLISH OR METRIC?";Z$
20: IF Z$="METRIC" LET L=4.9
   : GOTO 30
25: L=16
30: BEEP (1)
   : INPUT "INITIAL VELOCITY?";A
35: BEEP (1)
   : INPUT "ANGLE?";B
40: X=(COS B)*A
   : V=(SIN B)*A
45: C=V/L
   : D=X*(V/L)
   : H=V^2/(L*4)
50: PRINT "AIR TIME=";C
55: PRINT "RANGE=";D
65: PRINT "MAX. HEIGHT=";H
70: END
    
```



Above is a diagrammatic representation of what the program calculates. Note: The "Air Time" is the amount of time the projectile stays in the air. The velocity is to be entered in feet or meters per second. This will yield the time in seconds and the range and height in feet or meters.

Measurement Conversion

Frederick E. Bishop, R.A.
P.O. Box 648
Sharon, PA 16146

I am an architect and have a couple of programs of interest to those who work with dimensions.

Program 1 — Addition and Subtraction of Feet/Inches

The first one allows you to add and subtract feet and inches. Note: When subtracting enter the number of feet to be subtracted as a negative number. To complete a calculation when adding or subtracting press **(ENTER)** twice.

```

99:  "A"
   : CLEAR
100: INPUT "FT ";A,"IN ";B,"NUM";C,"/";D
   : GOSUB 510
   : GOTO 100
150: PRINT USING "####";G;"FT ",F;" ";E;"/16 IN"
155: GOTO 99
160: END
510: IF D=2 LET C=C*8
   : GOTO 517
513: IF D=4 LET C=C*4
   : GOTO 517
516: IF D=8 LET C=C*2
517: IF A<0 GOTO 550
518: REM ADD
520: E=E+C
   : IF E>16 LET F=F+1
   : E=E-16
530: F=F+B
   : IF F>12 LET G=G+1
   : F=F-12
531: G=G+A
540: RETURN
545: REM SUBTRACT
550: E=E-C
   : IF E<0 LET F=F-1
   : E=E+16
555: F=F-B
   : IF F<0 LET G=G-1
   : F=F+12
560: G=G+A
570: RETURN
    
```

Program 2 — Shift Contents of Display to Memory

The second program is a very handy one for me. It allows storage of a displayed value for use as the variable "X." It can either be used for storage or, by pressing "SHIFT =" can be used for the initial calculation.

```

10: "="
   : AREAD X
   : PRINT X
   : END
    
```

Program 3 — Convert Decimal to Feet-Inches

The third program, which converts feet to feet and inches, is useful when working with surveyor's materials which are always in decimal.

```

20: "B"
   : CLEAR
22: INPUT"DECIMAL=";A
24: B=INT(A)

26: C=A-B
   : D=C*12
   : E=INI(D)
28: F=D-E
   : G=INI(F*16)
30: PRINT B;" FT ";E;" ";G;"/16 IN" "
32: END

```

If 21.673 is entered as the decimal value in this program then the result would be 21' 8¹/₁₆."

Note that in programs 1 and 3 the results are displayed in some fractional sixteenth inches. In order to convert back to halves, quarters, etc. would take another twenty or so steps, and was just not worth it to me.

Factorials

Thomas Cox
Drawer 2327
Greenville, SC 29602

Factorials are not a built-in function of the Pocket Computer, but they can be easily calculated as illustrated in Program 1.

Program 1. Calculation of Factorials

```

10: REM CALCULATE N FACTORIAL
20: INPUT "N?";N
30: IF N > 69 PAUSE "N MUST BE LESS THAN 70"
   : GOTO 20
40: Q=1
   : FOR W=1 TO N
   : Q=Q*W
   : NEXT W
50: PRINT N;"!=";Q
   : END

```

Unfortunately, Program 1 is very slow when a program must calculate a multitude of large factorials. Speed can be greatly enhanced if a look-up table approach is used. A disadvantage is that 70 memories are required to store the factorials of 0 through 69. An example of using a look-up table for factorials is given in Program 2.

Program 2. Look-up Table Factorials

```

10: REM LOOK-UP TABLE FACTORIALS
20: A(27)=1
   : FOR W=1 TO 69
   : A(27+W)=A(26+W)*W
   : NEXT W
30: INPUT "N?";N
40: IF N>69 PAUSE "N MUST BE < 70"
   : GOTO 30
50: M=27 + N
   : PRINT N;"!=";A(M)
60: GOTO 30

```

Cost Comparison

Vernon T. Scally
P.O. Box 447
Edwards, CA 93523

With the economy in the state it is, more people are becoming cost conscious. Unit cost comparison is one

method to help save money; however, when shopping in a busy supermarket, trying to figure what the unit cost is for items not marked can get very confusing. The pocket computer can be carried to the supermarket with you and this program can help you by doing the comparing for you with little effort on your part.

Enter the cost of one item and its weight in ounces or grams, then enter the cost and weight of a second item and the computer will display the unit cost for both items at the same time.

```

10: "A"
   : CLEAR
20: INPUT"OUNCES(O) OR GRAMS(G)?";Z$
30: INPUT"A PRICE =.";A
40: INPUT"A UNIT =.";B
50: INPUT"B PRICE =.";C
60: INPUT"B UNIT =.";D
70: IF Z$="G" GOTO 110
80: G=INT B
   : G=16G
   : H=(B-INT B)*100
   : I=G+H
90: J=INT D
   : J=16J
   : K=(D-INT D)*100
   : L=J+K
100: GOTO 120
110: I=B
   : L=D
120: E=A/I
   : F=C/L
130: E=INT(100E+.5)/100
   : F=INT(100F+.5)/100
140: IF Z$="G" GOTO 170
150: PRINT"A= $";E;" /OZ B= $";F;" /OZ"
160: GOTO 10
170: PRINT"A= $";E;" /GM B= $";F;" /GM"
180: GOTO 10

```

Note 1 lb. 5 oz. will be entered as 1.05
12 oz. will be entered as .12
120 grams will be entered as 120

Resistor Values

Robert F. Edwards
Steed Rd. Box 581-A
Chenango Forks, NY 13746

To calculate resistor values first enter the resistor colors in the RESERVE mode as follows:

A:	BLA	'Black
S:	BRO	'Brown
D:	RED	'Red
F:	ORA	'Orange
G:	YEL	'Yellow
H:	GRE	'Green
J:	BLU	'Blue
K:	VIO	'Violet
L:	GRA	'Gray
=:	WHI	'White

Run the following program in the RUN mode and enter the resistor color by using (SHIFT) plus the color key.

```

4: BEEP 1
   : PAUSE "RESISTOR CALCULATOR"
10: INPUT"1ST BAND";A$
   : GOSUB 200
20: E=D*10
30: INPUT"2ND BAND";A$
   : GOSUB 200

```

```

40: F=D
   : T=E+F
50: INPUT "3RD BAND";A$
   : GOSUB 200
60: IF D=0 LET G=1
70: IF D=1 LET G=10
80: IF D=2 LET G=100
90: IF D=3 LET G=1E3
100: IF D=4 LET G=1E4
110: IF D=5 LET G=1E5
120: IF D=6 LET G=1E6
130: IF D=7 LET G=1E7
140: IF D=8 LET G=1E8
150: IF D=9 LET G=1E9
160: H=T*G
170: IF H>10000 LET H=H/10000
   : GOTO 190
180 PRINT H;" OHMS "
   : GOTO 10
190: PRINT H;" K OHMS "
   : GOTO 10
200: IF A$="BLA" LET D=0
210: IF A$="BRO" LET D=1
220: IF A$="RED" LET D=2
230: IF A$="ORA" LET D=3
240: IF A$="YEL" LET D=4
250: IF A$="GRE" LET D=5
260: IF A$="BLU" LET D=6
270: IF A$="VIO" LET D=7
280: IF A$="GRA" LET D=8
290: IF A$="WHI" LET D=9
300: RETURN

```

```

60: PRINT USING"###.#";"MILEAGE =";W
70: W=(A(V)-A(1))/Z
   : X=A(V)-A(1)
   : PRINT USING"#####.#";"TOT. TRIP MILES =";X
80: PRINT USING"###.#";"AV. TRIP MILEAGE =";W
90: END
100: CLEAR
   : INPUT"ENTER STARTING MILES ";A
110: END

```

MPG on the PC

Bruce K. Taub
 75 Captains Road
 North Woodmere, NY 11581

After using the PC for sometime in my home and wood-working shop, I decided to take the PC onto the open road to test the MPG that my car gets.

I do a lot of traveling, and I prefer to know the actual MPG of my cars rather than the factory estimated MPG. By using this program I've found the conditions under which my car operated the best. I've also found the actual difference between city and country driving.

The program is very simple and easy to use. The PC takes the miles traveled, the amount of gas used, and finds the MPG. It also takes the past few MPG readings and averages them together for an overall average.

Program Description:

Press **(SHIFT) (A)** and the PC asks the TIME#:

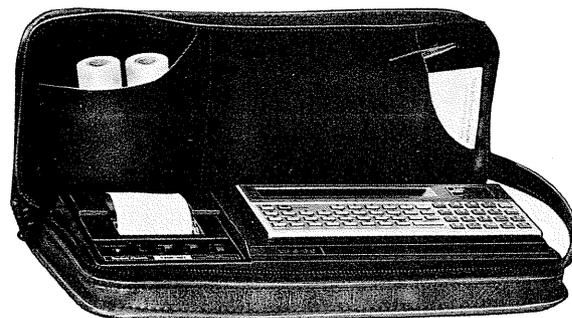
TIME#: — means how many times have you used the program. If it is your first time enter 1, not 0.

The first time, the computer sets the odometer. Be sure that you enter the odometer reading just after you filled the gas tank.

```

5: REM BY BRUCE K. TAUB APRIL 82
10: "A"
   : INPUT "TIME #:";T
20: INPUT "ODOMETER READING:";A(T)
25: IF T=1 THEN "MSG"
30: INPUT "AMT. LITERS:";X
   : Z=.26417X
60: LET V=(A(T)-A(T-1))/Z:PRINT USING "###.###";V;"
   MILES/GALLON"
70: LET S=(A(T)-A(T-1))/X
   : PRINT S;" MILES/LITER"
   : USING
80: LET R=T+5
90: A(R)=V
100: IF T>2 THEN 120
110: END
120: Y=0
130: FOR W=2 TO T
140: Y=Y+A(W+5)
150: NEXT W
160: U=Y/(T-1)
   : PRINT USING "###.###";U;" AVERAGE M/G"
170: END
180: "MSG"PAUSE"RESULTS NEXT TIME."
   : END

```



Mileage Calculations

Alan J. Leiwant
 301 Livingston Ave.
 Livingston, NJ 07039

On a recent vacation trip, since I had some time on my hands and my pocket computer with me, I worked out a small program to calculate total miles traveled, mileage per leg, trip miles traveled and average trip mileage.

The way the program is written, it could provide for an unlimited number of legs within a trip or within the program usage. However, we've set 21 as the number of storage registers required to store each trip leg. The remaining five registers are used to control the loop and to maintain the mileage calculations.

The program is entered by selecting the DEFINE mode and pressing **(SHIFT) (M)**. It then looks to see whether storage register A(1) is equal to 0. If it is, it clears all data registers and requests starting mileage. The next time through, the program automatically by-passes the clear position and requests current mileage and fuel used. It will then print out on the printer the miles traveled that leg, mileage, total trip miles, and average trip miles.

```

10: "M"
   : REM GAS MILEAGE ROUTINE
20: FOR V=1 TO 21
   : IF A(V)=0 GOTO 30
25: NEXT V
26: A(2)=A(V)
27: FOR V=3 TO 21
   : A(V)=0
28: NEXT V
29: V=3
30: IF V=1 GOTO 100
35: INPUT"ENTER CURRENT MILES ";A(V)
40: INPUT"ENTER FUEL USED ";Y
   : Z=Z+Y
50: W=(A(V)-A(V-1))/Y
   : X=A(V)-A(V-1)
   : PRINT USING"#####.#";"MILES TRAVELED = ";X

```

Color Editors

Text Editor 2.2

Brian Lieallen
9714 NW 20 Avenue
Vancouver, WA 98665

This is a text editing program for a Color Computer with Extended Color BASIC. In this program there are three basic modes: compose, edit, and list to printer.

In compose, you enter text for the first time. In the edit mode you can move the cursor up, down, left or right. You can also insert, delete or change characters, or you can extend your text.

Here is a list of the control keys in the different modes:

I. COMPOSE

<Left Arrow> = Backspace and erase
<Up Arrow> = Exit to menu
<ENTER> = Carriage return when sent to printer

II. EDIT

<Up Arrow> = Move cursor up one line
<Down Arrow> = Move cursor down one line
<Left Arrow> = Backspace
<Space Bar> = Advance one space
<1> = Delete one character
<2> = Change one character
<3> = Insert until <Up Arrow>
<4> = Extend until <Up Arrow>
<ENTER> = Return to Menu

```

1  '*** TEXT EDITOR 2.2 ***
2  '*** BY BRIAN LIEUALLEN ***
3  '*** NO RIGHTS RESERVED ***
10 CLS
   : CLEAR 200, 16311
   : GOTO 5000
20 INPUT "LINE WIDTH"; LW
30 DEFUSR0=16370
   : DEFUSR1=16342
   : DEFUSR2=16313
40 CLS
50 PRINT "'1' COMPOSE"
60 PRINT "'2' EDIT"
70 PRINT "'3' LIST TO PRINTER"
80 INPUT A
   : ON A GOTO 100, 210, 550
90 GOTO 40
95 ' *** COMPOSE ***
100 POKE 65479, 0
   : POKE 65481, 0
110 X=USR0(0)
120 A=1536
130 GOSUB 3000
140 A$=INKEY$
150 POKE A, 207
   : POKE A, 143
160 IF A$="" THEN 140
170 IF A$="^" THEN POKE A, 255
   : GOTO 40
180 IF A$=CHR$(8) THEN A=A-1
   : GOTO 140
190 GOSUB 2000

```

(Continued on Page 44)

SUPERTEXT

Matthew L. McGovern
RFD#1 Box 1330
Jay, Maine 04239

This letter was written using my program which I have dubbed SUPERTEXT.

I developed this program so that the Line Printer VII and Extended Color BASIC could be used to create paragraph and/or manuscript type papers. To the best of my knowledge, the program can perform each of the following: Title Positioning and underlining (7BS or 8BS, the 8BS requires TRS-80 PTFX16 for Extended Color BASIC computer), form new paragraphs, automatically indent paragraphs, and use the computer's upper/lower case mode (<SHIFT> and <O> key).

The program can only be appreciated by giving some examples of its use. A most notable example is the fact that I used it to write a term paper for school (I'm a 17 year old Junior at Jay High School) and it was gladly accepted by my teacher. And, while it does not produce a quality high enough to send to publishers, I find it quite useful in writing rough drafts of my material.

The program works by using several Line Printer commands. These commands are outlined in the booklet supplied with each LINE PRINTER VII, so I feel it is not that necessary to go in depth about their technical workings. However, CHR\$(26) was quite helpful in intercepting the computer's automatic message to the printer to advance a line, and CHR\$(18) and CHR\$(30) in switching from graphics to character printing modes (The graphic mode is used with 8BS underlining).

The program responds to very simple commands — <Y>=YES, <N>=NO. This is the major understanding necessary for immediate operation and execution of the program. Yet, a few aspects of it must be noted. First of all, an input must be entered to the prompt of Y=8BS/N=7BS or an ERROR message (one written into the program) will appear in the extreme upper left hand corner of the screen. If a lengthy paragraph is being entered, and you come to the end of the input line allowance (225 characters), simply enter the existing material and start typing where you left off on your next TEXT...? prompt. And, as in typing, remember to space twice after each sentence and at the end of each TEXT...? input to assure correct sentence spacing. Good luck with the program!

P.S. I have thoroughly enjoyed the new TRS-80 Screen Print Program for the Extended BASIC Computer. I hope more material of this sort is in the planning process.

```

10 ' *** SUPER TEXT ***
20 ' *** BY MATT MCGOVERN ***

```

(Continued on Page 45)

Text Editor 2.2 (From Page 43)

```

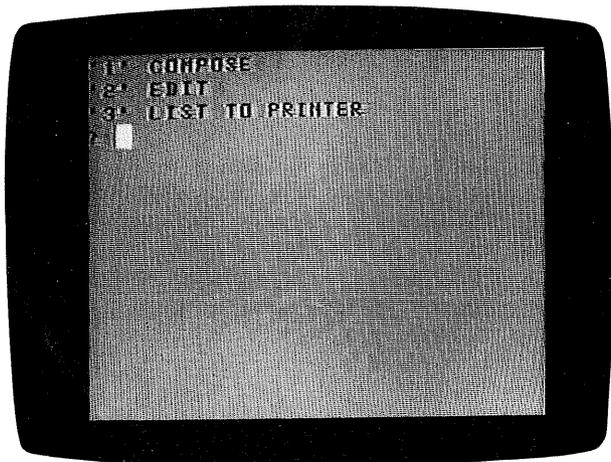
200 POKE A, ASC(A$)
   : A=A+1
   : GOTO 130
205 ' *** EDII ***
210 POKE 65479, 0
   : POKE 65481, 0
220 A=1536
230 P=PEEK(A)
   : POKE A, 175
240 A$=INKEY$
   : IF A$="" THEN 240
250 POKE A, P
260 IF A$=CHR$(32) THEN A=A+1
270 IF A$=CHR$(8) THEN A=A-1
   : IF A=1535 THEN A=1536
280 IF A$=CHR$(10) THEN A=A+32
290 IF A$="" THEN A=A-32
   : IF A<1536 THEN A=1536
300 GOSUB 3000
310 IF A$=CHR$(13) THEN 40
320 ON VAL(A$) GOTO 340, 380, 410, 490
330 GOTO 230
335 ' ***DELETE***
340 F1=(A+1)/256
   : F2=(F1-INT(F1))*256
350 POKE 16343, F1
   : POKE 16344, F2
360 X=USR1(0)
   : GOTO 230
375 ' *** CHANGE ***
380 A$=INKEY$
   : IF A$="" THEN 380
390 GOSUB 2000
400 POKE A, ASC(A$)
   : A=A+1
   : GOTO 230
405 ' *** INSERT ***
410 A$=INKEY$
   : IF A$="" THEN 410
420 GOSUB 2000
430 IF A$="" THEN 230
440 F1=(A/256)
   : F2=(F1-INT(F1))*256
450 POKE 16314, F1
   : POKE 16315, F2
460 X=USR2(0)
470 POKE A, ASC(A$)
   : A=A+1
480 GOTO 410
485 ' *** INSERT ***
490 A$=INKEY$
   : POKE A, 207
   : POKE A, 143
   : IF A$="" THEN 490
500 A1=ASC(A$)
510 IF A1=8 THEN A=A-1
   : GOTO 490
520 GOSUB 2000

```

```

530 IF A$="" THEN POKE A, 255
   : GOTO 230
540 POKE A, ASC(A$)
   : A=A+1
   : GOSUB 3000
   : GOTO 490
545 ' *** SEND TO PRINTER ***
550 FOR A=1536 TO 3071
560 A1=PEEK(A)
570 IF A1>0 AND A1<27 THEN A1=A1+96
580 IF A1=143 THEN A1=32
590 IF A1=255 THEN 640
600 IF A1=191 THEN 640
610 B$=B$+CHR$(A1)
620 IF LEN(B$)=LW THEN 640
630 NEXT A
640 PRINT #,-2, B$
   : B$=""
650 IF A1=255 THEN 40
660 IF A=3071 THEN 40
670 NEXT A
675 ' *** CHR CODE CONVERT ***
2000 A1=ASC(A$)
2010 IF A1>96 AND A1<127 THEN A$=CHR$(A1-96)
   : RETURN
2020 IF A1=32 THEN A$=CHR$(143)
   : RETURN
2030 IF A1=13 THEN A$=CHR$(191)
2040 RETURN
2045 ' *** SCREEN SELECT ***
3000 IF A>2559 THEN POKE 65479, 0
   : POKE 65480, 0
   : POKE 65483, 0
   : RETURN
3010 IF A>2047 THEN POKE 65483, 0
   : POKE 65478, 0
   : POKE 65480, 0
   : RETURN
3020 POKE 65479, 0
   : POKE 65481, 0
   : POKE 65482, 0
   : RETURN
3025 ' *** MACHINE LANGUAGE ***
5000 FOR A=16313 TO 16382
5010 READ B
   : POKE A, B
   : NEXT
5020 DATA 142, 6, 0, 166, 128, 230, 132, 167,
128, 30, 137, 129, 255, 39, 7, 140, 11, 255,
39, 2, 32, 239, 231, 132, 134, 255, 167,
132, 57
5030 DATA 142, 0, 0, 230, 132, 231, 130, 166,
129, 230, 132, 193, 255, 39, 10, 140, 11,
255, 39, 5, 231, 130, 126, 63, 221, 231,
130, 57
5040 DATA 198, 143, 142, 6, 0, 231, 128, 140, 11,
255, 38, 249, 57
5050 GOTO 20

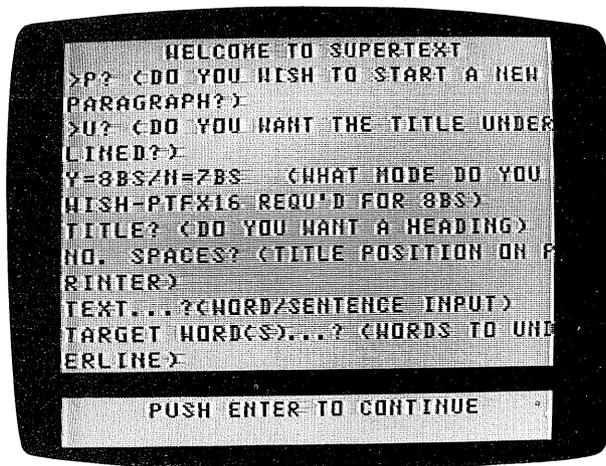
```



SUPERTEXT (From Page 43)

```
700 PRINT #-2, CHR$(30);
710 GOTO 300
```

```
30 ' *** JANUARY 4,1981 ***
40 ' *** RFD#1 BOX 1330 ***
50 ' ***JAY, MAINE 04239***
60 ' *** WORD PROCESSOR ***
70 ' ***UPPER/LOWER CASE***
80 ' ***CHANGE PARAGRAPH***
90 ' ***UNDERLINE TITLES***
100 ' ***7BS/8BS CAPACITY***
110 CLS
120 PRINT@ 416, STRING$(32, 128);
130 PRINT@ 6, "WELCOME TO SUPERTEXT"
140 PRINT ">P? (DO YOU WISH TO START A NEW
PARAGRAPH?)"
150 PRINT ">U? (DO YOU WANT THE TITLE
UNDERLINED?)"
160 PRINT "Y=8BS/N=7BS (WHAT MODE DO YOU
WISH-PTFX16 REQU'D FOR 8BS)"
170 PRINT "TITLE? (DO YOU WANT A HEADING)"
180 PRINT "NO. SPACES? (TITLE POSITION ON P
RINTER)"
190 PRINT "TEXT...?(WORD/SENTENCE INPUT)"
200 PRINT "TARGET WORD(S)...? (WORDS TO
UNDERLINE)"
210 PRINT@ 453, "PUSH ENTER TO CONTINUE"
220 Q$=INKEY$
: IF Q$=CHR$(13) THEN 230 ELSE 220
230 CLEAR 1000
240 CLS
250 INPUT "Y=8BS / N=7BS"; G$
260 IF G$="N" OR G$="Y" THEN 270 ELSE 610
270 CLS
: INPUT "TITLE"; F$
: IF F$="Y" THEN 280 ELSE 300
280 INPUT "NUMBER OF SPACES OVER"; A
290 L=L+A
300 CLS
310 IF F$="Y" THEN 330
320 LINE INPUT ">P?"; A$
: IF A$="Y" THEN A=3
: PRINT#-2, CHR$(10);
: L=0
: R$=""
: GOTO 270
330 LINE INPUT "TEXT...?"; S$
340 IF S$="" THEN 610
350 IF R$="STOP" THEN P$=""
: GOTO 380
360 IF F$="" OR F$="N" THEN 380
370 INPUT ">U"; P$
380 IF A>=80 THEN A=ABS(A-80)
390 IF A>=80 THEN A=ABS(A-80)
400 IF A>=80 THEN A=ABS(A-80)
410 FOR G=0 TO A
420 PRINT #-2, CHR$(32);
430 NEXT G
440 B=LEN(S$)
450 A=A+B
460 PRINT #-2, S$; CHR$(26);
470 IF P$="Y" THEN 500
480 L=L+A
490 GOTO 320
500 LINE INPUT "TARGET WORD(S)...?"; Y$
510 FOR G=0 TO INSTR(1, S$, Y$)+(L-1)
520 PRINT #-2, CHR$(32);
530 NEXT G
540 IF G$="Y" THEN 640
550 FOR T=0 TO (LEN(Y$))-1
560 PRINT #-2, CHR$(95);
570 NEXT T
580 PRINT #-2, CHR$(26);
590 R$="STOP"
600 GOTO 320
610 PRINT@ 0, "***ERROR***"
620 FOR P=1 TO 300
: NEXT P
630 GOTO 240
640 PRINT #-2, CHR$(18);
650 FOR T=0 TO (LEN(Y$)*6)
660 PRINT #-2, CHR$(192);
670 NEXT T
680 R$="STOP"
690 PRINT #-2, " "
```



Color Scripsit and the Line Printer VIII

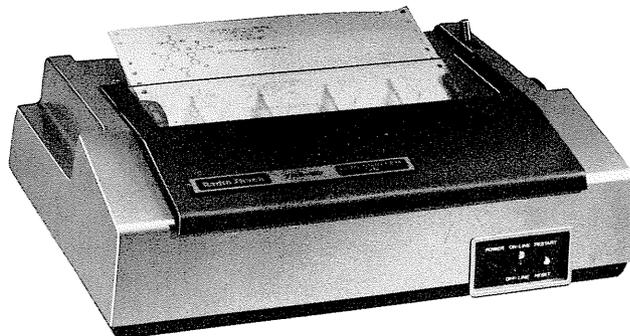
Garrett E. McGowan
3207 Evergreen Ave.
Hopewell, Virginia 23860

This correspondence was prepared on a Line Printer VIII, which will do 3 sizes of characters. It was also done by Color Scripsit on a 32K Color Computer. While the Color Scripsit was basically prepared to be used with a Line Printer VII, which I also have, it will do the job on a Line Printer VIII as well.

To do so the computer must be off, with the Program Pak out. Now follow these steps:

1. Turn on the printer
2. Turn on the computer (comes up in BASIC)
3. Initialize the printer to the appropriate type style 'PRINT#-2, CHR\$(27);CHR\$(n)'
4. Turn off the computer (leave the printer on)
5. Insert the Scripsit Program Pak
6. Turn the computer on
7. Choose option #6 (Change Standards)
8. Set Line Width and Margins as appropriate
9. Proceed as usual

Maybe this will assist those who have a Line Printer VIII. It certainly helps me.



Color Computer Bugs, Errors and Fixes

Notes to Users:

The following corrections and clarifications are provided for your information. If you feel that you do not fully understand this information, please contact your local Radio Shack Computer Center or Expanded Computer Department for assistance. If you do not have access to one of these stores, then you may want to call Computer Customer Services in Fort Worth for assistance.

Color Scripsit (26-3105)

Several problems have appeared in the Color Computer Scripsit Program Pak.[™]

First, at the main menu, do not press (7) for an option.

Pressing (7) will cause the program to lock-up. To recover you will have to reset the machine, which will result in the loss of any material which is currently in Scripsit.

Second, if you are INSERTing material into a text line, do NOT press (SHIFT) (RIGHT ARROW) (a tab) if you are inserting over a carriage return ((ENTER) key). If you are in the regular text mode, tabbing when no tab position has been defined will simply move the cursor to the end of the current line. Tabbing over a carriage return while in the INSERT mode, with no tab defined, will result in the computer apparently locking up, because the program is inserting spaces to the end of memory. No data will be lost but it will be a long wait before you regain control of the program.

Third, in a Global Replace, if you are trying to change all occurrences of a character (example: change all occurrences of X in the string XXX to Y), the program will only change every other occurrence on the first pass. Simply make a second pass to change the remaining characters. Example: The text had the string 'XXXXXX' next to the string 'YYYYYY.' In this phrase, we want (for what ever reason) to change every occurrence of the letter X to a W. (BREAK)(7) gets us into the Global Replace mode. We indicate X and W as the character to be changed and the character to change to respectively. The computer displays the text, and we indicate that all occurrences should be changed. The resulting sentence will be: The text had the string 'WXWXWX' next to the string 'YYYYYY.' To change the remaining Xs to Ws, simply repeat the process.

Color Spectaculator (26-3104)

An error can occur when a number less than one is subtracted from a larger number which is also less than one. For example, if you tried to subtract .08 from .1. This type of subtraction will result in a large negative (and incorrect) result.

If position C1 contains 0.1, and C2 contains 0.08, the result of C1-C2 will be -9999.98, instead of the correct answer of 0.02. Changing the formula to read: C1-(C2+0) will result in the correct answer.

To apply this fix generally, all columns or rows that might contain a value less than one (1) should be entered

as: (Column Designation+0) or (Row Designation+0). Also, constants less than 1 should be entered as (value+0).

Asteroids

Richard Zepp
1972 Battle Row
Augusta, GA 30904

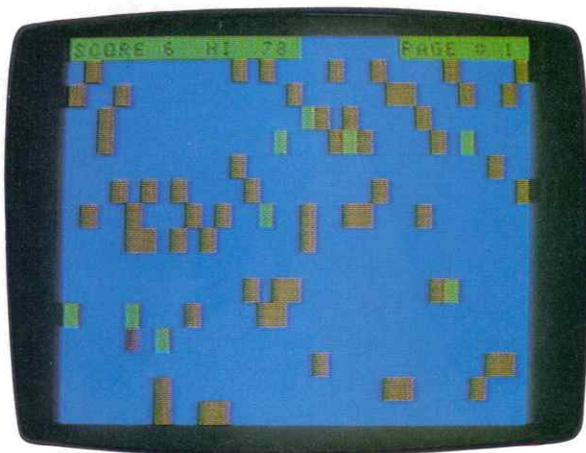
The program is for 4K Color Computers, high score to date 3349. I did not include the instructions because it is easy to figure out. HINT: use ← → for control, aim at the yellow ones.

```
1 CLS
      : PRINT "PRESS ANY KEY TO START"
2 A$=INKEY$
      : A=RND(0)
      : IF A$="" THEN 2
3 HS=0
94 C=1000
95 B=RND(64)
96 P=0
      : S=0
99 CLS(3)
100 FOR X=1 TO 63
110 Y=RND(31)
120 SET(X, Y, 8)
130 NEXT X
135 P=P+1
140 FOR X=26 TO 31
150 SET(B, X, 3)
160 NEXT X
170 SET(B, 31, 4)
171 FOR X=1 TO 10
172 A=RND(63)
      : W=RND(25)
      : SET(A, W, 2)
      : NEXT X
180 FOR X=1 TO 460
190 NEXT X
195 V=31
200 A$=INKEY$
205 IF A$="" THEN 210 ELSE SOUND 1, 1
210 SET(B, V, 3)
220 IF A$=CHR$(8) THEN B=B-1
      : IF B<1 THEN B=1
230 IF A$=CHR$(9) THEN B=B+1
      : IF B>63 THEN B=63
235 V=V-1
240 IF POINT(B, V)=8 THEN 1000
245 IF POINT(B, V)=2 THEN S=S+25
250 SOUND 200, 1
270 IF V=1 THEN 99
280 SET(B, V, 4)
290 IF B<>BC THEN S=S+1
300 BC=B
310 S=S+1
315 IF S=>C THEN 2000
320 PRINT@0, "SCORE" S " HI " HS;
330 PRINT@23, "PAGE #" P;
335 GOTO 200
1000 PRINT "YOU LOSE"
1010 PRINT
      : PRINT
      : PRINT
      : PRINT
      : PRINT "TO PLAY AGAIN PRESS <ENTER>"
1020 INPUT Q
1030 IF S>HS THEN HS=S
1040 GOTO 94
2000 C=C+1000
2010 CLS(3)
2050 Z=30
2055 Y=33
2060 FOR X=10 TO 31
2080 SET(Y, X, 2)
```

```

2090 SET(Z, X, 2)
2095 Z=Z-1
      : Y=Y+1
2100 NEXT X
2105 Z=33
      : Y=31
2106 Y=30
2110 FOR X=10 TO 1 STEP-1
2120 Z=Z+1
      : Y=Y-1
2130 FOR A = Y TO Z
2135 SET(A, X, 8)
2140 NEXT A
2150 SET(31, X, 1)
2160 SET(32, X, 1)
2170 NEXT X
2171 RL=RND(10)+25
2172 V=31
2200 A$=INKEY$
2201 SET(RL, V, 3)
2210 U=RND(3)
2211 IF U=1 THEN RL=RL+1
2212 IF U=3 THEN RL=RL-1
2220 IF A$=CHR$(8) THEN RL=RL-1
2230 IF A$=CHR$(9) THEN RL=RL+1
2240 IF POINT(RL, V)=2 THEN 99
2245 V=V-1
2259 IF POINT(RL, V)=1 THEN 2500
2270 SET(RL, V, 4)
2275 FOR X=1 TO 25
2276 NEXT X
2280 P=P+1
2300 GOTO 2200
2500 S=S+250
2501 GOTO 99

```



Fireworks

Franklin G. Pigounias
2226 Texas Avenue
Savannah, GA 31404

As of yet I have not approached the advanced stage in programming, but here is a program for Extended Color BASIC. I receive your Microcomputer News on a regular basis and enjoy the helpful hints and information made available to me through it and hope that my programs may help others through your newsletter as it has helped me in the past.

P.S. Hope to get a 32K added soon.

```

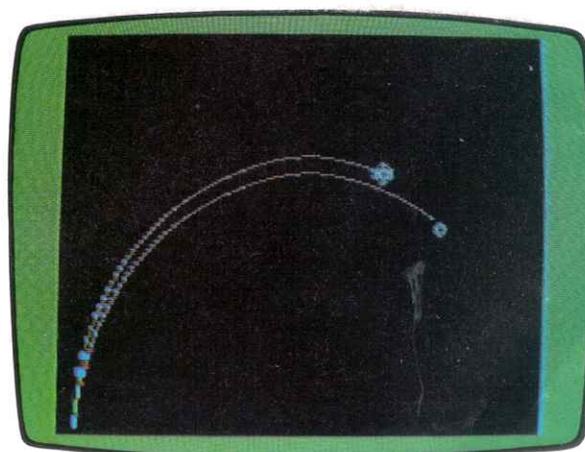
1 '-----
2 'FIREWORK'S DISPLAY
3 '-----
10 PMODE 4, 1

```

```

20 PCLS
30 SCREEN 1, 1
40 CIRCLE(128, 188), 110, , 1, .50, .75
50 CIRCLE(128, 79), 3
60 CIRCLE(128, 79), 5
70 CIRCLE(128, 79), 7
80 CIRCLE(128, 168), 110, 0, 1, .50, .75
90 CIRCLE(128, 79), 10
100 PCLS
110 LINE(256, 196)-(160, 100), PSET
120 CIRCLE(160, 100), 2
130 CIRCLE(160, 100), 3
140 LINE(256, 196)-(160, 100), PRESET
150 CIRCLE(160, 100), 5
160 CIRCLE(160, 100), 8
170 CIRCLE(160, 100), 10
180 CIRCLE(160, 100), 2, 0
190 CIRCLE(160, 100), 3, 0
200 CIRCLE(160, 100), 5, 0
210 CIRCLE(160, 100), 8, 0
220 CIRCLE(160, 100), 10, 0
230 CIRCLE(128, 188), 120, , 1, .50, .85
240 CIRCLE(128, 180), 120, , 1, .50, .80
250 CIRCLE(198, 94), 3
260 CIRCLE(168, 68), 3
270 CIRCLE(168, 68), 5
280 CIRCLE(128, 188), 120, 0, 1, .50, .85
290 CIRCLE(128, 180), 120, 0, 1, .50, .80
300 CIRCLE(168, 68), 8
310 CIRCLE(198, 94), 6
320 CIRCLE(198, 94), 10
330 PCLS
340 CIRCLE(52, 176), 3
350 CIRCLE(204, 176), 3
360 CIRCLE(128, 176), 3
370 CIRCLE(52, 176), 6
380 CIRCLE(204, 176), 6
390 CIRCLE(128, 176), 6
400 CIRCLE(52, 176), 3, 0
410 CIRCLE(204, 176), 3, 0
420 CIRCLE(128, 176), 3, 0
430 CIRCLE(52, 176), 8
440 CIRCLE(204, 176), 8
450 CIRCLE(128, 176), 8
460 CIRCLE(52, 176), 6, 0
470 CIRCLE(204, 176), 6, 0
480 CIRCLE(128, 176), 6, 0
490 PCLS
500 LINE(128, 192)-(128, 50), PSET
510 FOR D=1 TO 70
      : NEXT D
520 LINE(128, 192)-(128, 50), PRESET
530 CIRCLE(128, 50), 4
540 CIRCLE(128, 50), 4, 0
550 CIRCLE(128, 50), 6
560 CIRCLE(128, 50), 8
570 CIRCLE(128, 50), 6, 0
580 CIRCLE(128, 50), 10
590 CIRCLE(128, 50), 8, 0
600 CIRCLE(128, 50), 10, 0
610 GOTO 20

```



Radio Shack Computer Center Addresses and Telephone Numbers

Radio Shack®

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

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

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

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

ARKANSAS
LITTLE ROCK Town & Country S/C, University & Asher, (501) 568-5634

CALIFORNIA
ANAHEIM 509 Katella, (714) 776-9540
BAKERSFIELD 2018 Chester Ave., (805) 323-8074
BERKELEY 1922 Grove St., (415) 848-9170
BEVERLY HILLS 8500 Wilshire Blvd., (213) 659-8870
CANOCA PARK 8371 Topanga Canyon, (213) 347-9600
CARMICHAEL 6305 Feroaks Blvd., (916) 484-6815
CHULA VISTA 1201 3rd Ave., (714) 420-3810
DOWNEY 8031 Florence Ave., (213) 927-7882
ESCONDIDO 347 W. Mission Ave., (714) 741-6032
FRESNO Princeton S/C, 2721 N. Blackstone Ave., (209) 225-5551

GARDEN GROVE 12821 Knott St., (714) 891-1160
GLENDALE 236 N. Brand Blvd., (213) 246-9310
HAYWARD 20942 Mission Blvd., (415) 278-2888
LAKEWOOD 5830 Lakewood Blvd., (213) 920-9671
LA MESA 5346 Jackson Dr., (714) 460-3610
LONG BEACH 2119 Bellflower Blvd., (213) 597-3377
LOS ANGELES 740 S. Olive St., (213) 529-2455
MONTCLAIR 5237 Arrow Hwy., (714) 625-1541
MONTEREY 484 Washington St., (408) 375-8430
MOUNTAIN VIEW 1933 El Camino Real W., (415) 961-0542

OAKLAND 1733 Broadway, (415) 763-3183
PASADENA 575 S. Lake Ave., (213) 449-5424
RIVERSIDE 3844 La Sierra Ave., (714) 689-0340
SACRAMENTO 4749 J. St., (916) 454-3287
SAN BERNARDINO 764 Inland Center Dr., (714) 884-6871
SAN DIEGO 3062 Clairemont Dr., (714) 276-6050, 3902 El Cajon Blvd., (714) 280-0227
SAN FRANCISCO One Market Place, (415) 777-9810
SAN JOSE 1228 S. Bascom Ave., (408) 297-2603
SAN MATEO 3180 Campus Dr., (415) 573-8607
SANTA BARBARA 4141 State St. A-1, (805) 967-4538
SANTA FE SPRINGS 14138 East Firestone Blvd., (213) 921-0702

STOCKTON College Sq. S/C, 963 West March Lane, (209) 957-3676
TARZANA 18545 Ventura Blvd., (213) 343-1696
TORRANCE 3840 Sepulveda at Hawthorne, (213) 373-0306
VENTURA 4005 E. Main St., (805) 654-0196
WEST COVINA 2516 E. Workman St., (213) 915-5791

COLORADO
BOULDER Arapahoe Plaza, 3550 Arapahoe, (303) 443-7142
COLORADO SPRINGS 4341 N. Academy, (303) 593-7500
DENVER 8000 E. Quincy, (303) 770-1362
LAKEWOOD 2099 Wadsworth Blvd., (303) 232-6277

CONNECTICUT
EAST HAVEN 51 Frontage Rd., (203) 467-8864
FAIRFIELD 1196 Kings Hwy. & Rt. 1, (203) 255-6099
HARTFORD The Richardson Bldg., 942 Main St., (203) 549-5717
MANCHESTER 228 Spencer St., (203) 649-8210
NORWALK Rt. 7-345 Main Ave., (203) 846-3418
ORANGE Woolco S/C, 538 Boston Post Rd., (203) 795-1291

WATERBURY 105 Bank St., (203) 573-8800
WATERFORD 122 Boston Post Rd., (203) 443-0716
WEST HARTFORD 39 S. Main St., (203) 523-4283

DELAWARE
WILMINGTON 3847 Kirkwood Hwy., (302) 999-0193

DISTRICT OF COLUMBIA
WASHINGTON 1800 M St. NW., (202) 822-3933

FLORIDA
ALTAMONTE SPRINGS 766 B. East Altamonte Dr., (305) 339-1313
CLEARWATER 2460-D US 19 North, (813) 797-3223
DAYTONA BEACH Volusia Plaza, (904) 255-5909
FT. LAUDERDALE 4368 N. Federal Hwy., (305) 772-6797
GAINESVILLE 3315 Archer Rd., (904) 375-7063
HOLLYWOOD 429 S. State Rd. #7, (305) 966-4382
JACKSONVILLE 8252 Arlington Expwy., (904) 725-2594
LAUDERDALE LAKES 4317-25 N. State Rd. 7, (305) 486-2240

MIAMI 9459 S. Dixie Hwy., (305) 667-2316, 1601 Biscayne Blvd., (305) 374-6433, 15 SE 2nd Ave., (305) 374-7310, 20761 S. Dixie Hwy., (305) 238-2518
N. MIAMI BEACH The Promenade, 1777 N.E. 163rd St., (305) 940-6887
ORLANDO 1238 E. Colonial Dr., (305) 894-0570
ST. PETERSBURG 3451 66th St. N., (813) 381-2366
SARASOTA 5251 S. Tamiami Tr. (Hwy. 41), (813) 923-4721

TALLAHASSEE 2529 S. Adams, (904) 222-4440
TAMPA 4555 W. Kennedy, (813) 879-7470, 1825 E. Fowler Ave., (813) 971-1130
W. PALM BEACH 2271-A Palm Beach Lakes Blvd., (305) 683-3100

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

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

ILLINOIS
AURORA 890 North Lake St., (312) 844-2224
CHICAGO 4355 S. Archer Ave., (312) 376-7617, CNA Plaza, 309 S. Wabash, (312) 922-0536
ELMWOOD PARK 7212 W. Grand Ave., (312) 452-7500
FAIRVIEW HEIGHTS #4 Market Place, (618) 398-6410
HOMewood/GLENWOOD 329 Glenwood Lansing, (312) 759-0440
LaGRANGE One S. LaGrange Rd., (312) 482-3484
LIBERTYVILLE 1350 S. Milwaukee Ave., (312) 367-8230
LOMBARD 4 Yorktown Center, (312) 629-5350
NILES 8349 Golf Rd., (312) 470-0670
OAK LAWN 4815 W. 95th St., (312) 425-9130
PEORIA 4125 N. Sheridan Rd., (309) 685-7056
ROCKFORD North Town S/C, 3600 N. Main St., (815) 282-1001
SCHAUMBURG 651 Mall Dr., (312) 884-8600
SPRINGFIELD Sherwood Plaza, 2478 Wabash, (217) 787-3066

INDIANA
EVANSVILLE 431 Diamond Ave., (812) 426-1715
FT. WAYNE 747 Northcrest S/C, (219) 482-9547
GRIFFITH 206 W. Ridge Rd., (219) 838-3000
INDIANAPOLIS 6242 E. 82nd St., Castleton Plz., (317) 489-6896, Speedway Plaza, 6129 B Crawfordsville, (317) 244-2221, 10013 E. Washington St., (317) 898-4887
TERRE HAUTE 3460 U.S. Hwy. 41 S., (812) 234-3212

IOWA
DAVENPORT 616 E. Kimberly Rd., (319) 386-3457
DES MOINES 7660 Hickman Rd., Sherwood Forest S/M, (515) 270-0193

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

KENTUCKY
FLORENCE 7727 Mall Rd., (606) 371-2811
LEXINGTON 2909 Richmond Rd., (606) 269-7321
LOUISVILLE 2900 Taylorsville Rd., (502) 459-9901

LOUISIANA
BATON ROUGE 7007 Florida Blvd., (504) 928-5260
METAIRIE 3750 Veterans Hwy., (504) 454-3681
NEW ORLEANS 327 St. Charles Ave., (504) 523-6408
SHREVEPORT 1545 Line Ave., (318) 221-5125

MAINE
BANGOR Maine Square, (207) 945-6491

MARYLAND
BALTIMORE 7942 Belair Rd., Putty Hill Plaza, (301) 882-9583, 115 N. Charles St. at Lexington, (301) 539-7251
CATONSVILLE One Mile West S/C, 6600 B Balt. Nat'l. Pike, (301) 788-3277
FREDERICK Shoppers World, Rt. 40W, (301) 695-8440
NEW CARROLLTON-LANHAM 7949 Annapolis Rd., (301) 459-8030
PASADENA 8120 Ritchie Hwy., (301) 544-2352
ROCKVILLE Congressional Plaza, 1673 Rockville Pike, (301) 984-0424
SALISBURY Shoppers World S/C, Rt. 50, (301) 546-9223

MASSACHUSETTS
BOSTON 730 Commonwealth Ave., (617) 739-1704, 111 Summer St., (617) 542-0361
BRAINTREE South Shore Plaza, 250 Granite St., (617) 848-9290
BROCKTON 675 Belmont, (617) 583-2270
BURLINGTON Crossroads Plaza, Rt. 3 S., (617) 229-2850
CAMBRIDGE Harvard Square, 28 Boylston St., (617) 354-7699
CHESTNUT HILL 200 Boylston St., (617) 969-2031
NATICK 1400 Worcester Rd., (617) 875-8721
SAUGUS 343 Broadway, (617) 233-4985
SPRINGFIELD 1985 Main St., Northgate Plz., (413) 732-4745
WORCESTER Lincoln Plaza, (617) 852-8844

MICHIGAN
BIRMINGHAM 3620 W. Maple Rd., (313) 647-2151
DETROIT DWTNT 1559 Woodward Ave., (313) 961-6855
FLINT 6329 Miller Rd., Yorkshire Plaza, (313) 732-2530
GRAND RAPIDS 3142 28th St. SE., (616) 957-2940
KALAMAZOO 25 Kalamazoo Center, (616) 343-0790
LANSING 2519 S. Cedar St., (517) 372-1120
LIVONIA 33470 W. 7 Mile Rd., (313) 476-6800
ROSEVILLE 31873 Gratiot Ave., (313) 296-6210
SOUTHFIELD 17651 West 12 Mile Rd., (313) 569-1027
TROY Oakland Plaza, 322 John R. Rd., (313) 589-9030

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

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

MISSOURI
FLORISSANT 47 Florissant Oaks S/C, (314) 921-7722
INDEPENDENCE 1325 S. Noland Rd., (816) 254-3701
KANSAS CITY 4025 N. Oak Trafficway, (816) 458-3381
ST. ANN 10472 St. Charles Rock Rd., (314) 429-1400
SPRINGFIELD 2684 S. Glenstone, (417) 883-4320

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

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

NEW HAMPSHIRE
MANCHESTER Hampshire Plaza, 1000 Elm St., (603) 625-4040

ADDRESS CHANGE
 Remove from list Change as shown
Please detach address label and mail to address shown above

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

NEW MEXICO
ALBUQUERQUE 2108 San Mateo NE., (505) 265-9687

NEW YORK
ALBANY Shoppers Pk., Wolf Rd., (518) 459-5527
BAYSHORE 1751 Sunrise Hwy., (516) 666-1800
BETHPAGE 422 N. Wantagh Ave., (516) 822-6403
BUFFALO 839 Niagara Falls Blvd., (716) 837-2590
BUFFALO 175 Rt. 17 S., (201) 262-1920
SPRINGFIELD Rt. #22 Center Isle, (201) 467-9827
VOORHEES 35 Eagle Plaza, (609) 346-0600

NORTH CAROLINA
CHARLOTTE 3732 Independence Blvd., (704) 535-6320
GREENSBORO 3718 High Point Rd., (919) 294-5529
RALEIGH Townside Sq., Hwy. 70 W., (919) 781-9380
WINSTON-SALEM 629 Peters Creek Pkwy., (919) 722-0039

OHIO
AKRON Fairlawn Plaza, 2727 W. Market St., (216) 836-9303
CANTON 5248 Dressler Rd. NW., (216) 494-7230, Mellet Plaza, 3826 W. Tuscarawas, (216) 478-1878
CENTERVILLE 2026 Miamisburg-Centerville Rd., (513) 435-5167
CINCINNATI 9725 Montgomery, (513) 793-8688
CLEVELAND 419 Euclid (Dwntwn), (216) 575-0800, 27561 Euclid Ave., (216) 289-6823
COLUMBUS 862 S. Hamilton, Great Eastern S/C, (614) 864-2806, The Patio Shop. Ctr., 4661 Karl Rd., (614) 436-4666, 400 N. High St., (614) 464-2781
DAYTON Northwest Plaza, 3279 West Siebenthaler, (513) 277-6500
NORTH OLMSTED Great Northern S/C, (216) 734-2255
PARMA 751 W. Ridgewood Dr., (216) 842-4030
TOLEDO 5844 W. Central Ave., (419) 531-5797
YOUNGSTOWN Union Square Plaza, 2543 Belmont Ave., (216) 744-4541

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

OREGON
EUGENE 390 Coburg Rd., (503) 687-0082
PORTLAND 7463 SW Barbur Blvd., (503) 246-1157, 9131 SE Powell, (503) 777-2223

PENNSYLVANIA
ALLENTOWN Crest Plaza S/C, Cedar Crest Blvd. US 22, (215) 939-7155
BALA CYNWYD 67 E. City Line Ave., (215) 668-9950
EIRY 5755 Peach St., (814) 868-5541
HARRISBURG Union Deposit Mall, Union Deposit Rd. #17, (717) 564-6753
LANCASTER Park City Plaza, US 30, (717) 393-5817
MONROEVILLE 3828 Wm. Penn. Hwy., (412) 823-3400
MONTGOMERYVILLE Airport Sq., Rt. 309, (215) 362-1200

PHILADELPHIA 7542 Castor Ave., (215) 342-2217, 1002 Chestnut St., (215) 923-3080
PITTSBURGH 5775 Baptist Rd., Hills Plaza, (412) 831-9694, 303 Smithfield St., (412) 391-3150
SCRANTON 206 Meadow Ave., (717) 348-1801

RHODE ISLAND
E. PROVIDENCE 850 Waterman Ave., (401) 438-2860
PROVIDENCE 177 Union St., (401) 272-0201

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

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

TEXAS
AMARILLO Wellington Sq. S/C, 1619 S. Kentucky, (806) 358-4567
AUSTIN 4005 E. Main St., (805) 654-0196
WEST COVINA 2516 E. Workman St., (213) 915-5791

UTAH
MURRAY 6051 S. State Ave., (801) 268-8978
SALT LAKE CITY 415 5th Ave., (801) 322-4893

VIRGINIA
ALEXANDRIA 3425 King St. at Quaker Ln., (703) 370-9000
FAIRFAX Westfair Center, 11027 Lee Hwy., (703) 273-6500
NEWPORT NEWS Newmarket South Shop. Ctr., (804) 245-8203
NORFOLK 5731 Poplar Hall Dr., (804) 461-0798
RICHMOND Willow Lawn S/C, 1617 Willow Lawn Dr., (804) 282-3453, 7728 Midlothian Turnpike, (804) 272-8603
ROANOKE Franklin Bldg., 3561 Franklin Rd. S.W., (703) 342-6335
ROSSLYN 1911 N. Ft. Myer Dr. at Rt. 29, (703) 243-5555

WASHINGTON
BELLEVUE Crossroads Mall, North East 8th & 156 St., (206) 644-1804
BELLINGHAM 1111 Cornwell Ave., Suite B & C, (206) 676-0811
FEDERAL WAY 33505 Pacific Hwy. South, (206) 838-6830
3rd Ave., (206) 447-1959, 5030 Roosevelt Way NE, (206) 527-1019

WEST VIRGINIA
HUNTINGTON 2701 1/2 5th Ave., (304) 523-3527

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