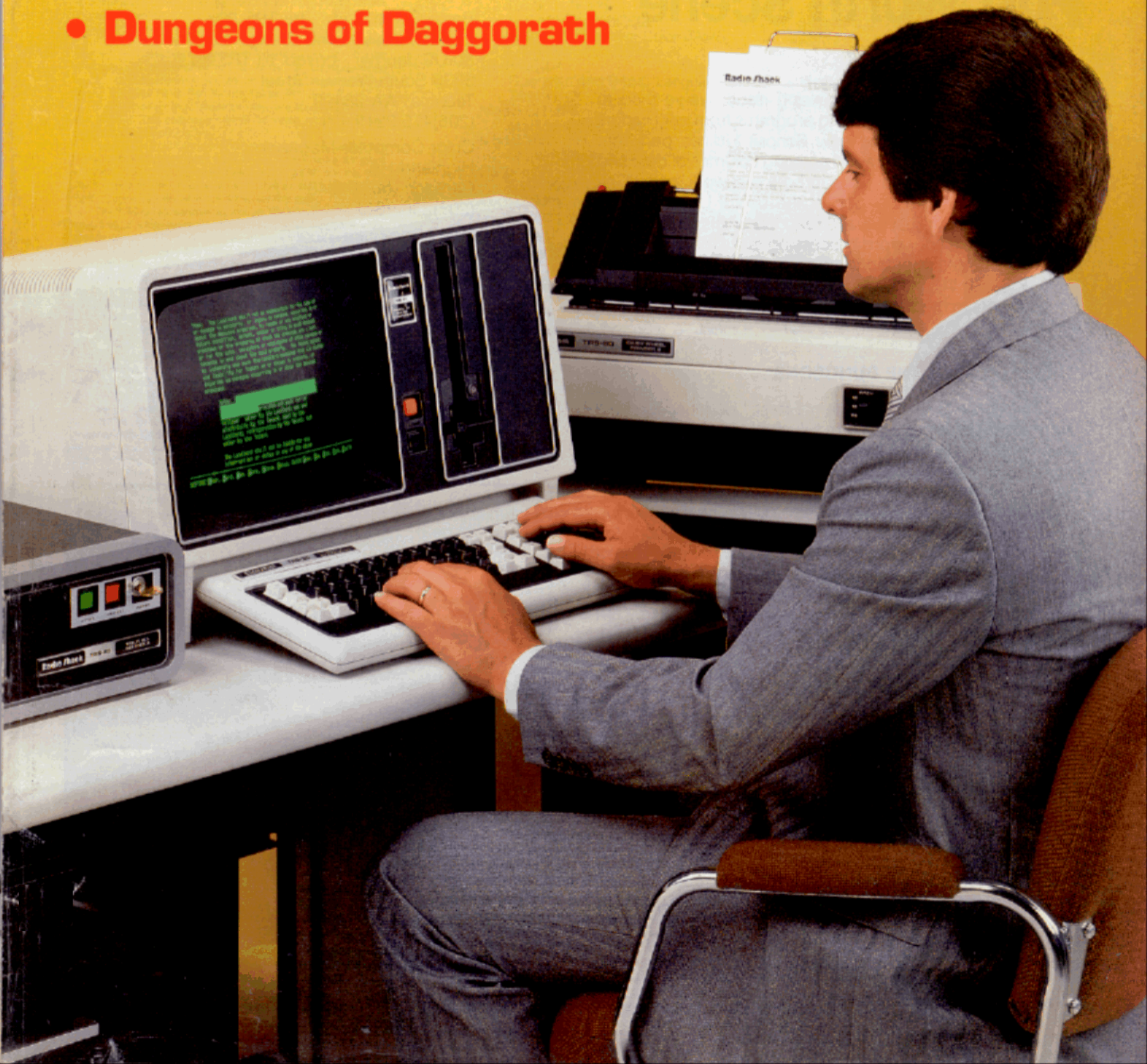
Volume 5, Issue 8 August, 1983
$1.50

TRS-80® Microcomputer News

- Word Processing Issue
- Pocket Computer 3
- Dungeons of Daggorath



Fort Worth Scene

WORD PROCESSING

In this annual Word Processing issue, we've tried to include several word processing programs from our readers along with articles on the new Scripsit Utilities package. Those of us who have dreaded entering columns of data and numbers will find this package to be a real work saver. Column manipulation is just one of the useful Utilities in this package, but you'll just have to read the article to learn how else it can make your word processing tasks more pleasant.

Implementation of the Merge utility of Scripsit 2.1 is another article in this issue. Merge has been difficult for some Scripsit users to master so we've tried to make the process a little clearer in the article.

DARE YE ENTER THE DUNGEONS . . .

This game is "the absolute greatest." Because Mr. Shirley reads these pages, too, I'm reluctant to tell you just how many hours have been spent "researching" the most exciting adventure game yet—The Dungeons of Daggorath. This newest Color Computer game has everything: graphics, sound, intense excitement, and hair raising challenges.

It's no secret around here that for days my most consuming ambition was to defeat the Smiling Blob. The euphoria of success soon evaporated when I found myself face to face with even more death defying creatures. Still there are creatures that I have yet to see! Do try the game; it is great fun, but I feel that it's only fair to warn you that it is addictive.

ANOTHER COMPUTER?

Of course! The Pocket Computer 3 makes it debut in this issue. Surely some of you must have suspected something when we announced the PC-4 back in March. Don't miss reading about this compact, 4K beauty.

ARCNET AND XENIX INSTALLED

We have recently been in the midst of having XENIX and ARCNET installed in the offices of the *Microcomputer News*. There were technicians in and out, holes drilled in the floor, cables appearing, hardware modified, equipment tested, and so on. As quickly as they came, the technicians were gone, the dust cleared and except for the file processor in the storeroom and a couple of innocuous looking cables, everything looks essentially the same as it did before. But, of course, beneath the calm exterior waits all that remarkable technology ready to spring into action at a moments notice.

The only thing left undone is the process of self education which we must all undergo to take advantage of all the available power of both ARCNET and XENIX.

MAGAZINES

Below are five magazines of speical interest to TRS-80 owners that we believe have editorial content of high quality and will be of use to our customers.

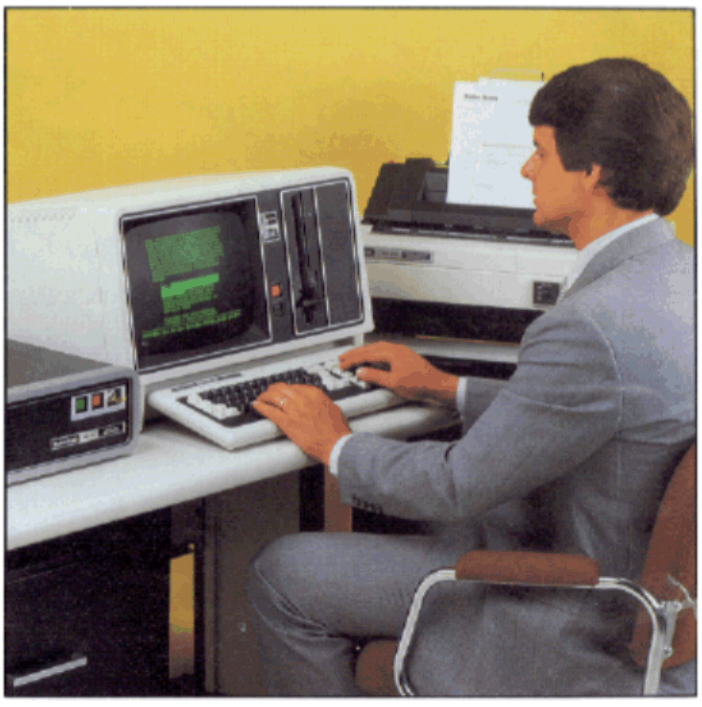
Basic Computing - The TRS-80
User Journal (Name change for
80-US Journal—covers all TRS-80's)
3838 South Warner Street
Tacoma, WA 98409
(206)475-2219

Color Computer Magazine
Highland Hill
Camden, ME 04843
(207)236-9621

Color Computer Weekly
P.O. Box 1355
Boston, MA 02205

Rainbow (Covers the TRS-80 Color Computer)
5803 Timber Ridge Dr.
Prospect KY 40059
(502)228-4492

two/sixteen magazine
P.O. Box 1216
Lancaster, PA 17603
(717)397-3364



TRS-80 Microcomputer News
Volume 5 Issue 8
AUGUST 1983

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 full size TRS-80 Microcomputer systems with addresses in the United States, Puerto Rico, Canada and APO or FPO addresses. Certain smaller TRS-80 Microcomputers will not include this free subscription. 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 available as stock number 26-2240 (Suggested Retail Price $9.95 for the set).

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

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

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

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

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

Trademark Credits

CompuServe™	CompuServe, Inc.
CP/M®	Digital Research
Dow Jones™ NEWS/RETRIEVAL Service®	Dow Jones & Co., Inc.
LDOS™	Logical Systems, Inc.
VisiCalc®	VisiCorp, Inc.
XENIX™	Microsoft
Program Pak™	Tandy Corporation
SCRIPSIT™	Tandy Corporation
TRSDOS™	Tandy Corporation
TRS-80®	Tandy Corporation

TRS-80® Microcomputer News

Contents:

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

SCRIPSIT Utilities—A Review

One of the most useful software products to come along in quite a while is the SCRIPSIT Utilities package (Catalog number 26-4532, Suggested Retail Price $129.00). This package adds to SCRIPSIT the functions of column manipulation, boilerplate, and document to document communications. These utilities will work with all of the current Model II, 12 or 16 SCRIPSIT packages, including the recently released SCRIPSIT for Thinline Floppy and Hard Disk Use (Catalog number 26-4835). Once the SCRIPSIT Utilities are installed, the Utilities disk must be placed in drive 1.

THE COLUMN MANIPULATION UTILITY

Anyone who prints financial reports will find the column manipulation function quite useful. This utility allows the user to move entire columns as easily as moving text. Columns can also be subtotaled or totaled, alleviating the need for a desk-side calculator. Let's see how we can apply the column manipulation utility.

Column manipulation works in much the same way as the SCRIPSIT "define text block" (CTRL D) utility. Column manipulation is used to move already created text while eliminating the drudgery of setting up the format line with elaborate tab settings. First we will create some text with which to work. Open a sample document and create the following text:

```
■
                Proposed Budget for Fiscal 1984 ■

■
■
Department ■
■
Accounting ■
Data Proc. ■
Manufacturing ■
Research ■
Shipping ■
Gen. Admin. ■
■
■
Budget ■
Requested ■
■
137,000.00 ■
250,000.00 ■
134,000.00 ■
275,000.00 ■
127,000.00 ■
110,000.00 ■
■
■
Budget ■
Approved ■
■
127,000.00 ■
200,000.00 ■
124,000.00 ■
250,000.00 ■
117,000.00 ■
100,000.00 ■
■
```

To start the column manipulation utility, place the cursor on a single-line, non-centered paragraph. (This is essentially a single line of text with a hard carriage return.) In this example, place the cursor on the "D" of Department and press (ESC) (C). You will see the following message on the bottom line:

Crs:ll,ccc DEFN Elmnt Col Above Below; Inquire Recall Subtot Tot Mean Format

At this point we will select the Elmnt option to define the first element "Department." If you had selected Col, you would have defined the entire first column, including the numbers below. Select the rest of the department headings using the Elmnt option. We have now defined everything from Department to Gen. Admin., including the blank line below Department. In this example we want the departments to be indented. We can either choose the Move option or the Shift option. Since we only need to reposition the text on the same lines, we can use the Shift option. Select the "Col" option and using the left and right arrow keys, position the column where you want it. In our example we indented the column five spaces. Press the (BREAK) key until you return to normal SCRIPSIT operation. Your text should now look like this:

```
■
                Proposed Budget for Fiscal 1984 ■
■
■
\ Department ■
\            ■
\ Accounting ■
\ Data Proc. ■
\ Manufacturing ■
\ Research ■
\ Shipping ■
\ Gen. Admin. ■
■
■
Budget ■
Requested ■
■
137,000.00 ■
250,000.00 ■
134,000.00 ■
275,000.00 ■
```

127,000.00 ■
110,000.00 ■
■
■
Budget ■
Approved ■
■
127,000.00 ■
200,000.00 ■
124,000.00 ■
250,000.00 ■
117,000.00 ■
100,000.00 ■
■

Now define the next group of elements and select the "Move" option. On the line above the "Department" heading, move the cursor five spaces to the right of "Department." Press (R) to recall the column. Then repeat the process with the remaining column, placing the cursor ten spaces to the right of the "Budget" heading. Press the (BREAK) key to exit the column manipulation operation.

Now your screen should look like this:

■
Proposed Budget for Fiscal 1984■
■

\ \	Budget \	Budget ■
\ Department \	Requested \	Approved ■
\ \ \		■
\ Accounting \	137,000.00 \	127,000.00 ■
\ Data Proc. \	250,000.00 \	200,000.00 ■
\ Manufacturing \	134,000.00 \	124,000.00 ■
\ Research \	275,000.00 \	250,000.00 ■
\ Shipping \	127,000.00 \	117,000.00 ■
\ Gen. Admin. \	110,000.00 \	100,000.00 ■
\	1,033,000.00 \	918,000.00 ■

■

Notice that the columns have been totaled. This was accomplished by defining the elements and choosing the "Total" option, moving the cursor onto the blank line under the decimal point, and pressing the "T" for "Total" again. If you get the message, "Tab type doesn't match for column to be inserted," make sure you have at least one hard carriage return below the line where you want the totals to appear. Again press the (BREAK) key to return to normal SCRIPSIT operation. As an added touch we centered the heading (CTRL)(L).

This has been a simple example of column manipulation, but it should give you an idea as to the usefulness of this utility.

THE BOILERPLATE UTILITY

How many times have you found yourself copying a paragraph from an existing document to a new one? If you are in a business that uses standard forms such as insurance contracts or real estate contracts, you will certainly be able to use this utility. "Boilerplate" is the ability to create documents with standard paragraphs and then easily insert them into your new document.

The first step in using the Boilerplate Utility is to create a boilerplate document. This is the document that will contain all of your standard paragraphs or pages. A boilerplate document differs from a regular document in that the first page contains a "table of contents" (only) and the boilerplates are on the remaining pages, one to a page. The maximum number of boilerplates in a document is 220. However, you can create as many boilerplate documents as you need. Each one must have a different name as in any other SCRIPSIT document.

To begin, create a new document using the (F1) key and give the document the name *Boilerplate-V*. The "V" represents vertical format. (*Boilerplate-V* and *Boilerplate-H* are the default document names the Boilerplate utility looks for when a boilerplate is requested).

On the first line of the first page, type the following **EXACTLY** as shown below:

Boilerplate (ENTER)

The remainder of the page will contain the table of contents. Each entry in the table of contents must follow this format:

boilerplate name (TAB) page number termination character

The termination character for the table of contents entry can be either a (TAB), (ENTER) or a forced end of page, (ESC) (?). To maximize the number of boilerplates in a document, group three entries on each line in the vertical format and five entries on each line when using the horizontal format. Here is a sample table of contents.

Boilerplate■
Letter heading \ 2 \ Std remarks \ 3■
Disclaimer \ 4 \ Warranty \ 5■
Conclusion \ 6■

Note that the table of contents does not have a fancy format. Do not put anything between the word "Boilerplate" and the boilerplate titles, not even an additional (ENTER). If there is an additional (ENTER), the Boilerplate Utility will respond with an error message and not find the requested boilerplate.

The text for page two might look like this:

■
Letter heading \ Standard letter heading ■
■
\ \ \ \ {D}■
■
■
AnyCorp Credit Dept.■
3400 Any Tower Suite 100■
Anytown, Your State 92333■

Note the boilerplate name at the beginning of the page. This is an optional comment line that can be used to designate the boilerplate. Any initial comment **must** begin with the boilerplate name, followed by a (TAB), followed by the comment. The suggested format for boilerplate pages is:

an initial comment, text, and end of page marker.

When the Boilerplate Utility looks for a boilerplate, it will verify the boilerplate name if it is present in an initial comment. Any number of comment lines may be used within a boilerplate. These comment lines may consist of an (ESC) (,), the text, and an (ESC) (,) (ENTER).

The useful part of the Boilerplate Utility is when it is put to work. Create a new document such as a business letter and activate the Boilerplate Utility by pressing (ESC) (B). The bottom line will display:

Boilerplate name? _____

If you used the default names of *Boilerplate-V* or *Boilerplate-H* then you need only enter the individual boilerplate name. If you create other boilerplate documents under other names, then you must answer the prompt with: diskname: document name boilerplate name .password

If the diskname is SCRIPSIT then diskname may be omitted.

After entering the boilerplate name, press (ENTER) and in a few seconds the requested text is inserted into the new document. The new document may now be edited, moved, reformatted, or rearranged as required by the new document.

THE COMMUNICATIONS UTILITY

You've been asked to prepare a special report needed today by the home office in another city. There isn't enough time to send it by courier. How will you get it there? You could dictate it over the phone but, there is a better way. Now with the SCRIPSIT Communications Utility you can easily transmit your report directly into SCRIPSIT on another similarly equipped Radio Shack computer on the other end. The report is printed, and you've made your deadline.

The Communications Utility may be used on any combination of two Model II, 12, or 16 computers and any mixture of SCRIPSIT versions from SCRIPSIT 2.1, SCRIPSIT HD, or SCRIPSIT for Thinline Floppy. Of course, suitable communications hardware must be used on each end of the communications link, and the SCRIPSIT Utilities must be installed on each end. Both of the computers must also be using the same communications protocol (baud rate, parity, etc.).

With the Communications Utility you have the option of using "manual," "terminal," or "automatic" connections. Using manual connections, you are responsible for establishing contact with the other computer at the other end. Terminal connections allows the use of non-Radio Shack devices to establish connections. Using two Modem IIs (Catalog number 26-1173), the Utility allows you to automatically call up the other computer at a specified time, transmit the document or documents, and disconnect, all from within SCRIPSIT.

To use the Communications Utility, at the SCRIPSIT directory select the (U) option for the Utilities Menu. At the Utilities Menu select (T) for the Communications Utility. Next you will see a screen that requests answers to four questions. First you will be asked if you will be transmitting or receiving. Next set the baud rate. This must be the same for both computers in the system. Next you must decide if you wish to disconnect after your transmission is complete. The default is "disconnect." The last item on the screen is the connection type: manual, terminal, or automatic. Select the connection type and press (ENTER) or (ESC).

If you are receiving a document, the Utility requests a destination diskname on which to put the documents. If you are transmitting documents, the Utility then asks for a document name. At the prompt, type in the document name in the following format:

diskname:document name.password (ENTER)

The diskname is optional as the utility will search the current disk for the document. After you have pressed (ENTER), the screen will be cleared and you will be asked for another document name. Up to ten documents may be selected. After the last document is selected, press (ENTER) on the blank prompt line.

Next you are requested to make your communications connections. In the manual mode this is true for both receive and transmit. When connections are made, press any key to continue. At this point both computers will display the message "Connecting," and the transmitting computer will send the requested document(s). The Automatic option works slightly differently in that once you have completed the setup, the Utility will display "Connecting" until the transmitting computer calls the receiving computer and transmits the requested documents. Of course both computers must be ready to communicate at the time specified in the Automatic setup. At the end of the transmission the computer will display the names of the documents transmitted or received and any errors that may have occurred. Pressing any key returns you to the Utilities Menu.

SUMMARY

SCRIPSIT Utilities brings a new dimension to your word processing repertoire. Now you can easily format your columnar text into professional looking reports using the Column Manipulation Utility without having to set multiple tabs and special format lines. The Boilerplate Utility eliminates the tedious task of either retyping standard paragraphs or copying them from existing documents. The ultimate utility, the Communications Utility, lets you transmit from computer to computer from within SCRIPSIT without having to go through the time consuming steps of creating an ASCII text file and then transmitting it and converting the text back into SCRIPSIT on the other end. Now it is as simple as ABC.

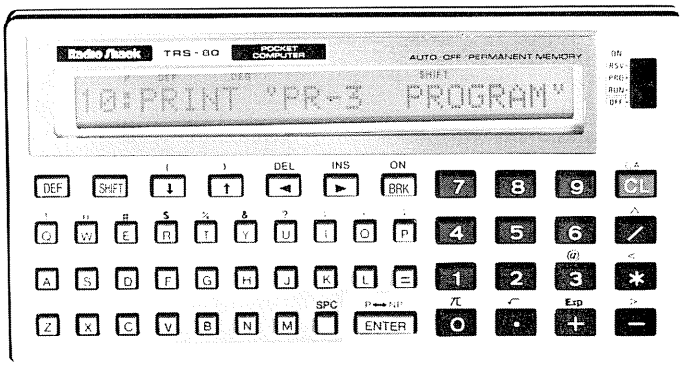
Computer Customer Service Address and Phone Numbers

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

Model I/III/4 Business Group	(817) 390-3939
Model II/12/16 Business Group	(817) 390-3935
Languages and Compilers	(817) 390-3946
Color/Model 100/Pocket Computer Group	(817) 390-3944
Hardware and Communications Group	(817) 390-2140
Educational Software	(817) 390-3302
Games, Books, and New Products	(817) 390-2133
Newsletter Subscription Problems	(817) 870-0407

The New PC-3!

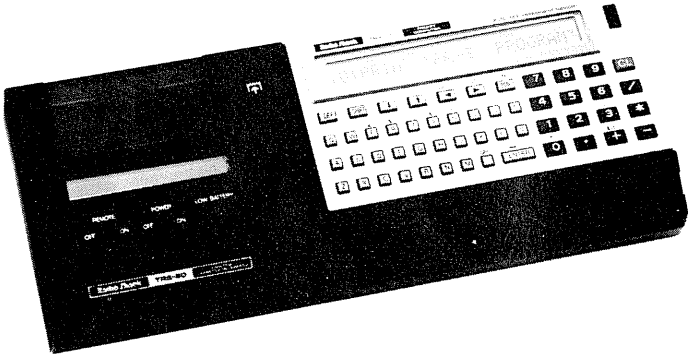
by Annette Zamberlin-Main



Just when you thought Radio Shack had introduced all its new models for 1983, yet another appears—the Pocket Computer 3. And what a charmer this newest addition to the PC family is!

The PC-3 (Cat. No. 26-3590) has a 52-character keyboard, a 24-character display, a powerful BASIC in 24K of ROM, and 4K of RAM. One feature I like is the PASSWORD protection you can apply to stored programs.

The PC-3 is $5^{5}/_{16}$"(W) × $2^{3}/_{4}$"(D) × $^{3}/_{8}$"(H) and weighs approximately .25 lbs. It is available for $99.95 (suggested retail price) at Radio Shack Computer Centers and participating Radio Shack stores and dealers. Its Printer Cassette Interface (Cat. No. 26-3591) is available separately for $119.95 (suggested retail price.)



Three of the Radio Shack pocket computers, the PC-1, the new PC-3, and the PC-2 have many features in common, but there are some significant differences. Occasionally the same features are present, but are somewhat different in presentation or execution. In order to better facilitate the use of the various programs for the different models, the following comparative charts have been prepared for our readers.

VERBS AND COMMANDS

In the following chart the symbol:
M indicates that the feature can only be used in manual execution, i.e., as a command:
P indicates that the feature can only be used within a program;
B indicates that the feature can be used in both contexts.
When no symbol is shown, the feature is not available on that machine.

	PC-1	PC-3	PC-2	Comments
AREAD	P	P	B	NOTE 1
ARUN			P	
BEEP	P	B	B	NOTE 2
CHAIN	P	P	P	
CLEAR	B	B	B	
CLOAD	M	M	M	
CLOAD?	M	M	M	
CLS			B	
COLOR			B	
CONT	M	M	M	
CSAVE	M	B	B	
CSIZE			B	
CURSOR			B	
DEGREE	B	B	B	
DATA		P	P	
DEBUG	M			
DIM		B	B	
END	P	P	P	
FOR . . . TO . . . STEP	P	P	P	
GOSUB	P	P	P	
GOTO	P	B	B	
GCURSOR			B	
GPRINT			B	
GRAD	B	B	B	
GRAPH			B	
IF . . . THEN	P	P	P	
INPUT	P	P	P	
INPUT#	B	B	B	
LET	P	P	P	
LF			B	
LINE			B	
LIST	M	M	M	
LLIST		M	B	NOTE 3
LOCK			B	
LPRINT		P	B	NOTE 4
MERGE	M	M	M	
NEW	M	M	M	
NEXT	P	P	P	
ON . . . ERROR			P	
ON . . . GOSUB		P	P	
ON . . . GOTO		P	P	
PAUSE	P	P	B	
PASS		M		

PRINT	P	P	B	NOTE 4
PRINT #	B	B	B	
RADIAN	B	B	B	
RANDOM		B	B	
READ		P	P	
REM	P	P	P	
RESTORE		P	P	
RETURN	P	P	P	
RLINE			B	
RMTOFF			B	
RMTON			B	
ROTATE			B	
RUN	M	M	M	
SORGN			B	
STOP	P	P	P	
TAB			B	
TEST			B	
TEXT			B	
TROFF		B	B	
TRON		B	B	
UNLOCK			B	
USING	P	B	B	NOTE 5
WAIT		B	B	

NOTE 1: There are some minor differences between the PC-3 and the PC-1 in behavior of AREAD following PRINT, but these are unlikely to cause problems in ordinary usage.
NOTE 2: The PC-3 has tone and duration.
NOTE 3: The PC-1 can emulate with LIST.
NOTE 4: With the PC-1 all PRINT statements go to the printer, and not to the display. Because the PC-3 is a bit more versatile than the PC-1, an LPRINT statement will go to the printer, while a PRINT statement directs to the display.
NOTE 5: On the PC-1 the USING format applies to all displays on the line in which the USING clause appears, even if the variable precedes the verb. On the other PCs, the USING format applies only to displays which follow the verb and remains in effect until cancelled by another USING verb.
Example:

```
10 A=-123.456
20 PAUSE USING"####.##";A
30 PAUSE A, USING"####";A
```

When executed, this program displays the following:

PC-1		-123.45
	-123	-123
PC-3		-123.45
	-123.45	-123

As compared with the PC-1, the PC-3 is faster in processing speed for calculations. Therefore, when game programs written for the PC-1 are used with the PC-3, adjust the game speed, etc.

VARIABLES

In this and the following charts the features are simply marked with a 'Y' when the machine has the feature.

	PC-1	PC-3	PC-2	Comments
INKEY$		Y	Y	
MEM	Y	Y	Y	
PI or π	Y	Y	Y	PC-1 has only π
TIME			Y	

NUMERIC FUNCTIONS

	PC-1	PC-3	PC-2	Comments
ABS	Y	Y	Y	
ACS	Y	Y	Y	
ASN	Y	Y	Y	
ATN	Y	Y	Y	
COS	Y	Y	Y	
DEG	Y	Y	Y	
DMS	Y	Y	Y	
EXP	Y	Y	Y	
INT	Y	Y	Y	
LOG	Y	Y	Y	
LN	Y	Y	Y	
NOT		Y	Y	
POINT			Y	
RND	Y	Y	Y	
SGN	Y	Y	Y	
SIN	Y	Y	Y	
SQR or √	Y	Y	Y	PC-1 has only √
STATUS			Y	
TAN	Y	Y	Y	

STRING FUNCTIONS

	PC-1	PC-3	PC-2	Comments
ASC		Y	Y	
CHR$		Y	Y	
LEFT$		Y	Y	
LEN		Y	Y	
MID$		Y	Y	
RIGHT$		Y	Y	
STR$		Y	Y	
VAL$		Y	Y	

OPERATORS

	PC-1	PC-3	PC-2	Comments
^	Y	Y	Y	NOTE 1
*,/,+,–	Y	Y	Y	
>,>=,=,<>, <=,<	Y	Y	Y	
AND, OR		Y	Y	
&		Y	Y	

NOTE 1: Raising a negative number to a power with the ^ operator can result in incorrect signs on all three pocket computers.

PC-1 program tapes can be used with the PC-3. When entering the PC-1 programs into this unit from its keyboard, however, the following precautions should be observed:

For example, the following are keyed-in for program entry:

10 IF N = LPRINT A [ENTER]

With the PC-1, this results in a command for "If N = L, display A" (IF N = L PRINT A). With the PC-3, however, it becomes a command for "If N =, print A" (IF N = LPRINT A), causing a syntax error (ERROR 1) to occur when executed. This is because the PC-3 has an LPRINT command not available with the PC-1.

Therefore, an IF statement should be keyed-in as:

10 IF N = L THEN PRINT A

Thus, a character-string for "variable with command" with the PC-1 may be regarded as "a command" in the PC-3.

Merge for Scripsit 2.1

By Linda Miller

Merge is one of the most useful utilities of Scripsit 2.1.0. Merge is frequently used in form letters where the same basic information is sent to several people but each letter is personalized to include the recipient's name, address, or other pertinent information.

THE SYSTEM SETUP

To set up a floppy disk system, insert Scripsit in drive 0 and a TRSDOS formatted diskette in drive 1, 2, or 3. The TRSDOS diskette is used to store the variable or Merge document (the document containing the information that changes) in ASCII format. Because the Scripsit program preallocates most of the space on the Scripsit disk for its own use, a TRSDOS formatted diskette is used to store the ASCII formatted document.

THE HYPOTHETICAL DATA

The president of the small, newly organized Cowtown Computer Club wants to send a letter to all the club members informing them of the subject, time, and location of the next club meeting. All three of the other members—I did say small—need to have the information, and the club president, V.T. Humble, is anxious to show off the capability of his Scripsit word processor. The club members are:

Clay T. Clayton
100 Main Street
Fort Worth, TX 76102
555-1111

Mary S. Maryland
200 South Street
Fort Worth, TX 76102
555-1112

John W. Johnson
300 Center Street
Fort Worth, TX 76102
555-1113

Figure 1.

THE PROCEDURE

Merging is basically a three step operation.

1. A Base document is created. A Base document contains the information that will be identical for each club member.
2. A Merge document is created. The Merge document contains the variables or information that changes each time the final product is printed.
3. The Base and Merge documents are merged and printed.

THE BASE DOCUMENT

The Base Document consists of two parts: the information that will be identical for each person on the mailing list and a set of code names. Code names are enclosed in braces ({}) as shown in the sample Base Document in Figure 2.

```
Cowtown Computer Club■
500 West Street■
Fort Worth, TX 76016■
■
{FIRST NAME} {LAST NAME}■
{STREET ADDRESS}■
{CITY, STATE ZIP}■
Dear {FIRST NAME},■
   So you won't forget! Here's the information regarding
our next meeting.■
   ■
   Date: September 7, 1983■
   Time: 7:30 P.M.■
   Place: 500 West Street■
   Theme: Your TRS-80 as a Word Processor■
   ■
   Be sure to come and bring your friends.■
      ■
      Sincerely,■
      V.T.Humble■
```

Figure 2.

THE MERGE DOCUMENT

The Merge document also consists of two parts: the same code names as the Base document (and possibly some code names not shown in the Base document) and the actual data or variables to be merged with the base letter. Both the code names and the variables are enclosed in braces. Figure 3 is the Merge file list of code names and sets or groups of variables. Mary S. Maryland, her address, city, and phone number constitute what is for convenience referred to as a "set of variables."

```
{FIRST NAME}■
{LAST NAME}■
{STREET ADDRESS}■          'Code names
{CITY, STATE ZIP}■
{PHONE NUMBER}■
■
{Clay T.}■
{Clayton}■
{100 Main Street}■         'Sets of variables list
{Fort Worth, TX 76102}■
{555-1111}■
■
{Mary S.}■
{Maryland}■
{200 South Street}■
{Fort Worth, TX 76102}■
{555-1112}■
■
{John W.}
{Johnson}■
{300 Center Street}■
{Fort Worth, TX 76102}■
{555-1113}■
■
■
```

Figure 3.

The left and right braces must surround each code name and each variable. One missing brace will guarantee an error.

(Setting up a user key to enter the braces simplifies the process and decreases the chance for error—provided the user key is set up accurately.) If the right brace on {John W. Johnson} were missing so it appeared as

{John W. Johnson■

an error would occur.

Noice that there is an equal number of elements in the list of code names as there is in each set of variables. (See Figure 4.)

{FIRST NAME}■	{Clay T.}■
{LAST NAME}■	{Clayton}■
{STREET ADDRESS}■	{100 Main Street}■
{CITY, STATE ZIP}■	{Fort Worth, TX 76102}■
{PHONE NUMBER}■	{555-1111}■
■	■

Figure 4.

If there are five code names, then there must be five elements in every set of variables. If one group of variables is missing an element such as a phone number, the braces must still be in the position where the missing element would have appeared as in Figure 5.

{Clay T.}■
{Clayton}■
{100 Main Street}■
{Fort Worth, TX 76102}■
{}■
■

Figure 5.

Each set of variables must be separated by two enters (shown as the graphic character ■ in the examples). Since this point has been confusing, look at the sets of variables in Figure 3. The set of variables for Mary S. Maryland and the set of variables for John W. Johnson must be separated by two enters. The first enter is the one that follows Mary S. Maryland's phone number. The second enter appears between her phone number and John Johnson's name. Also, the Merge document must be ended by pressing ENTER three times. In Figure 6 John W. Johnson's set of variables is the last group in the list, and there are three enters that follow his phone number.

{John W. Johnson}■	
{300 Center Street}■	
{Fort Worth, TX 76102}■	
{555-1113}■	'1st ENTER
■	'2nd ENTER
■	'3rd ENTER

Figure 6.

The first of the three enters follows his phone number and then there are two enters immediately below where no text appears. The last three enter blocks terminate the list of variables.

The wrong number of enters and an occasional missing brace are small errors, but they will defeat you every time if they are not corrected.

MATCHING CODE NAMES

Code names in the Base document and the Merge document must match exactly. A code name {FIRST NAME} in the Base document and the code name {FIRSTNAME} in the corresponding Merge document would cause an error because the first code name {FIRST NAME} includes a space between FIRST and NAME, but FIRST NAME does not. Therefore, the two code names are not identical.

CONVERTING THE VARIABLES INTO AN ASCII FILE

The Merge document is converted to an ASCII file by typing CTRL U M which brings up the Merge Documents prompt. Change the B for Base document to a C for Convert. Next enter a TRSDOS filename and drive number 1, 2, or 3 depending on which floppy drive contains the TRSDOS formatted diskette. The filename for the Cowtown Computer Club's list of addresses is ADDRESS:1. Press ENTER. The Merge document is converted to ASCII format and stored on the diskette in drive 1 as indicated in the TRSDOS filename ADDRESS: 1.

IT'S TIME TO MERGE AND PRINT

Now that there is a Base document and the Merge document has been converted to an ASCII file, it is time to actually merge and print the final product (letters in this example) providing the printer is ready. Open the Base document. Type CTRL U M ENTER and the TRSDOS file name for the Merge Document (ADDRESS:1). The Merge Documents Menu is displayed so printing parameters (pause between pages, number of copies, type of printer, etc.) can be changed. Finally, press ESC to begin printing the merged documents.

VARIATIONS ON A THEME

While it is necessary to have a code name in the Base document that corresponds to a code name in the Merge document, it is not necessary to have every code name that appears in the Merge Document appear in the Base Document. The telephone numbers that appeared in the Merge Document in Figure 3 were not used in the Base Document in the example, but that Merge document could be used with a different Base document to create totally different output.

IN SUMMARY

The Base or the Merge document can be created first. In this example, the Base document was created first for illustration purposes.

In capsule form, here are the steps in the Scripsit 2.1 Merge.

1. Insert Scripsit in drive 0 and a TRSDOS formatted diskette in drive 1, 2, or 3. Verify that the printer is ready.
2. Create the Base document which includes the text that is to be duplicated for each document and code names surrounded by braces.
3. Create the Merge document with code names that are identical to those in the Base document. Every variable and code name is surrounded by braces. Verify that there are two enters separating each set of variables and three enters at the end of the variable list. (See Figure 5.)
4. Convert the Merge document to an ASCII file storing the file on a TRSDOS disk in drive 1, 2, or 3 if using floppy disk.
5. Open the Base document. Type CTRL U M ENTER and the ASCII file name followed by an ENTER.
6. Change any printing parameters that need to be changed.
7. Print the documents.

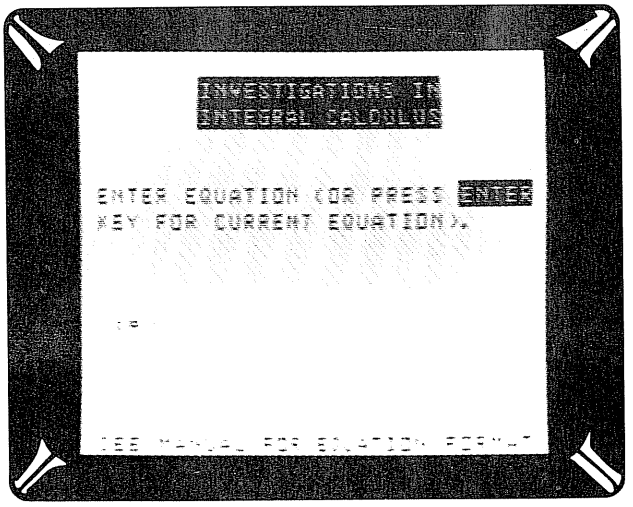
Once understood, Merge is easy to use and its versatility hasn't even been tapped. Additional helpful information can be found in the Scripsit Reference Manual.

Color Calculus—Secondary Math on the Color Computer

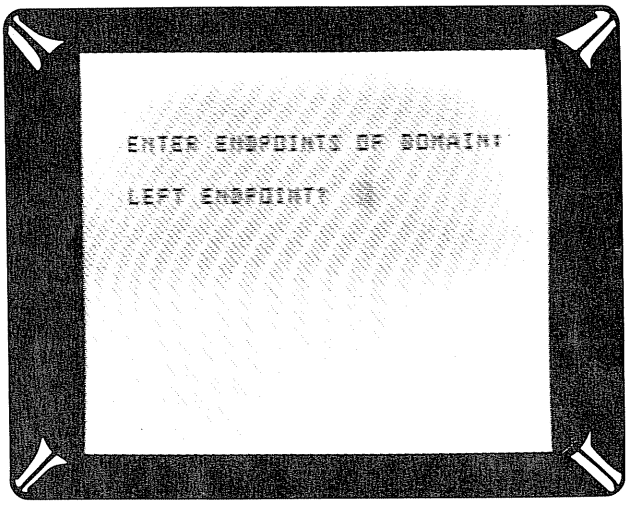
The first of a series of Radio Shack® secondary math programs for the TRS-80® Color Computer is now available from your local Radio Shack store, Radio Shack Computer Center, or Regional Educational Coordinator. Investigations in Integral Calculus for the TRS-80 Color Computer (Cat. No. 26-2641) is the Color Computer version of Radio Shack's Investigations in Integral Calculus Program for the Model III, Model 4, and Model I (Cat. No. 26-2600). The program is designed to help students gain an understanding of calculus concepts while avoiding the complex theoretical approach traditionally used in calculus textbooks. The program graphs and computes the areas of functions, using equations entered by the student, or using a "default" equation that is built into the program.

RUNNING THE PROGRAM

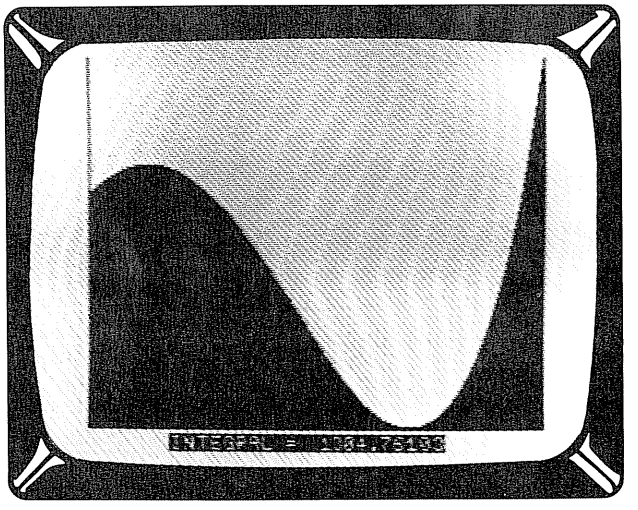
Once the program is loaded and the student has pressed [ENTER] to continue past the title screen, the student can enter the equation that he or she wants to see graphed. (The teacher's manual includes a concise explanation of the rules for entering equations using computer BASIC expressions.) If no equation is entered, the computer uses the "default" equation, $y = x^4 - 2x^3 - 43x^2 + 80x + 300$.



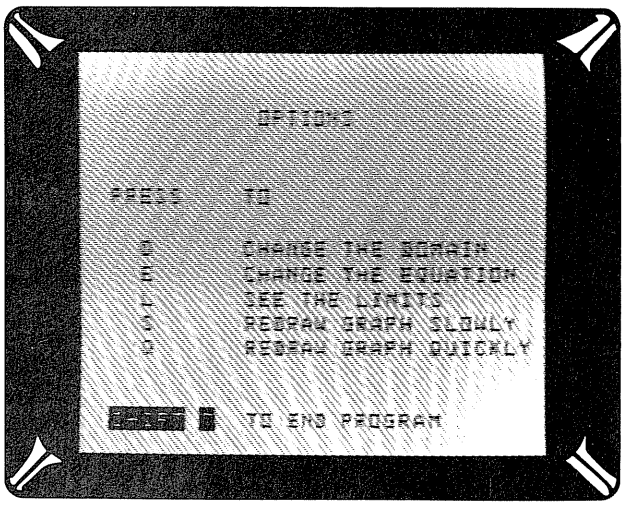
Next, the student enters the desired domain. For example, to determine the integral of the current equation from 0 to 7 by dividing the region into 100 trapezoids, the student would type [0] and press [ENTER] at the message "LEFT ENDPOINT?", and then enter [7] for the "RIGHT ENDPOINT". At "NUMBER OF INTERVALS?", the student would type [1][0][0] and press [ENTER].



The computer makes calculations. Then the graph is drawn, the area or integral that has been calculated is shaded in, and the integral is displayed at the bottom of the screen.



After a pause, a menu of options appears. From this screen, the student can change the domain of the integral to be calculated, change the equation to be integrated, redraw the graph slowly or quickly, see the limits of the current graph, or end the program.



SELECTED INVESTIGATIONS

The teacher's manual for the color calculus program includes a Selected Investigations section, which the teacher can reproduce and distribute to the students. Selected Investigations is a series of fourteen investigations that cover the following integral calculus topics:

The trapezoidal rule
Integral notation
The integral of a constant function
Negative integrals
The integral of x^n
The integral of $1/x$
The integral of the sine and cosine functions
The integral of the exponential function
The integral of a constant times a function
The integral of the sum of two functions.

An Answer Key, also in the teacher's manual, provides the teacher with answers and explanations of the problems in the Selected Investigations. Graphs and areas for the Selected Investigations are also given in the Answer Key.

HOW TO USE THE PROGRAM

A few of the ways that Investigations in Integral Calculus can be used in the classroom are:

- As a one- to two-week unit in integral calculus near the end of an Algebra II course
- As a short unit in integral calculus in a survey course in mathematics at the senior level
- As an introduction to integral calculus at the beginning of a course in calculus
- As an independent study unit in calculus for the interested student.

Investigations in Integral Calculus for the TRS-80 Color Computer (26-2641) can be used with a TRS-80 16K Color Computer tape system (with Extended Color BASIC) or a TRS-80 32K Color Computer disk system (with Disk Extended Color BASIC). It can also be used with a Radio Shack Network 2 System with a Color Computer host and Color Computer student stations with at least 16K of memory.

Radio Shack's original version of Investigations in Integral Calculus (26-2600) can be used with a TRS-80 Model III, Model 4, or Model I, and is compatible with Network 2 Systems. Minimum memory requirements are 16K for tape and 32K for disk. 26-2600 can also be used with a Network 3 system; this application requires RS-232C and an additional 16K in the student stations.

COLOR PROGRAMS TO FOLLOW

A series of secondary math and science programs for use with the TRS-80 Color Computer is currently under development at the Radio Shack Education Division. A Color Computer version of Radio Shack's Vector Addition program will be available soon. (Model III, 4, and I version currently available, Cat. No. 26-1720.)

Computers and Reading/Learning Difficulties Conference

Between 700 and 1,000 are expected at the Computers and Reading/Learning Difficulties Conference scheduled for October 14 and 15 at the Sheraton-Dallas Hotel in Dallas, Texas.

This is the largest conference specifically targeted for computer users in reading, language arts, and learning disabilities.

Classroom teachers; reading specialists; learning disability specialists; university professors of education, reading, and special education; speech and language clinicians; administrators, and parents could all benefit from attending this conference.

The conference is sponsored by Computers, Reading and Language Arts (CRLA, a professional journal) and provides learning experiences for the novice as well as the experienced computer user in the area of:

Utilizing Microcomputers in reading clinics, learning disability classrooms, regular classrooms

Evaluating software

Authoring systems

Microcomputers as motivators

Implementing programs.

For a brochure with full details on the program, faculty and registration, write:

Frost Conference Management
Department I
1070 Crows Nest Way
Richmond, CA 94803

Vocational Computing as a Start-Up Alternative

by John McGinty
Assistant Principal for Instruction
Grand Prairie (Texas) High School

The curricular area of vocational computing is reasonably well defined in terms of need. The aim of public school vocational computing is to prepare practically trained, vocationally oriented graduates with solid applications skills. These students will enter positions subject to rapid technological change, where continual learning is the expected pattern of employment. Their training must address job attitudes as well as job skills.

A good model for a program of vocationally directed school computing may be found at Grand Prairie High School, a school of 1700 students located in the Dallas/Fort Worth metroplex. The school's vocational department trains students for entry-level positions offered by a number of local employers. Employment opportunities are currently good for word processing operators, small-business bookkeepers, and computer electronics technicians. The instructional program is initially focusing on job training in these areas but is designed to be responsive to employer needs.

The vocational computing program at Grand Prairie is laboratory based and provides extensive hands-on practice. The oldest and best-developed element of the program is word processing. In their classroom, junior and senior students develop their skills using ten TRS-80 Model III microcomputers and Radio Shack's SuperSCRIPSIT software. Output is directed to one of four Radio Shack DMP-200 printers for draft or practice documents, or to a Radio Shack Daisy Wheel II printer for letter-quality copies of finished work.

The training in word processing is directed by an experienced Vocational Office Education teacher. After thorough grounding in SuperSCRIPSIT, students gain valuable "real world" experience by applying their skills to many routine tasks required in school operation. In addition to standard correspondence jobs, word processing students handle such tasks as preparation of the school newspaper and provide assistance with the academic handbook. New word processing applications are often suggested by visitors who come to observe the class.

Word processing was the initial element in Grand Prairie's vocational computing endeavor. Other instructional areas soon followed, to expand the offerings into a comprehensive office automation program. Radio Shack's PROFILE software opened the area of data management to students. The powerful data-handling capabilities of PROFILE quickly generated numerous ideas for useful applications around the school.

Each of these, in turn, provided additional practical experience for students in the program. The most recent addition to the office automation program is spreadsheet analysis. The class is learning to use the popular VISICALC program to perform common business calculations and to plan and forecast.

The impact of Grand Prairie High School's vocational computing program has far exceeded the original goal of providing job training to students. A variety of spin-offs have been observed. The visibility, to members of the faculty, of early word-processing products (school newspaper, handbooks, correspondence, homework assignments) helped to stimulate curiosity and teacher interest in computing. This teacher interest led to the establishment of an after-school computer instruction program for school staff members. So far, eighty employees of the Grand Prairie School District (half from Grand Prairie High School) have completed this eighteen-hour introductory course. Today, almost one fifth of the GPHS faculty owns a home computer, and ninety percent wish to incorporate computing into their teaching within the next five years.

New vocationally directed school computing programs have followed the pioneering lead of the office automation program at Grand Prairie High School. The most ambitious of these is a two-year vocational training program in Computer Electronics and Robotics, to address the rapidly growing need for technicians to service computers and automated manufacturing equipment. In addition, the business department of the high school is finalizing plans to provide comprehensive training in automated business applications for general accounting, banking and finance, marketing and sales, insurance, real estate, and taxes.

The experiences of Grand Prairie High School can provide a model for other schools. Educators who are currently planning school computing programs for their districts should seriously consider an initial focus on the development of a program of vocational computing. There are a number of advantages to the approach.

Though an instructor of vocational computing requires more specialized technical training than a classroom teacher using computers to support instruction in a non-computing subject, a significant educational impact can be made by training only a few vocational teachers to use computers. In addition, start-up hardware costs are often less. Excellent software—the same used in business applications—is available now. Additionally, the needs of local employers are the

best indicators of the direction that a vocational computing program should take, and these employers will often provide hardware, software, and/or teacher training in addition to valuable advice.

Compared to a microcomputer used for instructional support, a microcomputer devoted to vocational training is almost always operated for more hours per week. This increased usage and the sophisticated applications software tend to make a vocational computing center highly visible. This increases the likelihood that the experience of Grand Prairie will be repeated—the vocational computing laboratory will stimulate further teacher interest in computing and serve as a center for training teachers in computing fundamentals.

All three forms of educational computing—administrative, instructional support, and vocational—are necessary to a comprehensive program of school computing. It is not the intent of this article to minimize the value of computing in support of the existing instructional program. The greatest impact of computing on education will come in that sector. But to get to that point, an administrator's wisest move at this time may be to concentrate initial efforts on development of a sound program of vocational computing.

SuperSCRIPSIT Super Script

William O'Connell
41 Marland Road
Worcester, MA 01606

You have typed out your long article or report using SuperSCRIPSIT and are standing over your daisy wheel printer watching it do its magic. Suddenly two blank pages go zipping by. What happened? You hit the BREAK key and throw the document into view mode. Then by advancing the cursor to the suspect point, you find the reason—two accidental force-end-of-page symbols. A quick deletion of the unwanted symbols and then what? How do you get the machine to start on page twelve of the report without needless duplication of the first eleven pages? Following is a method of forcing SuperSCRIPSIT to skip the first pages and begin printing on the page you wish.

1. Position the cursor so that it appears on the first character of the page where you wish to resume the printing. Back up one character and enter an end of block symbol (@ E).

2. Use SHIFT ↑ to position the cursor on the first character of the first page of the document and press @ S for a start of block symbol.

3. Now go to Block action with a @ B and answer its prompt with M for Move. The disk drive will now store all the text from the beginning of the document to the point at which you wish to resume printing, and the text will vanish from the screen.

4. Select the print option by using @ P and answer the options very carefully. On the third line below the Document name of the Print Menu where the prompt "Begin numbering as page" appears, the number should be changed from the current value to the number of the page you want to begin printing on. After changing the page number option, press ENTER and print to the end of the document.

5. When printing is completed and your document reappears on the screen, use SHIFT ↑ to position the cursor at the beginning of your now headless document. Now type @ R to recall the text which was temporarily moved to the disk.

From this point onward, you may do whatever you want with the text as an entire unit or in newly defined blocks. By using the SuperSCRIPSIT "Begin numbering as page" print option, the writer is encouraged to break up long documents into smaller chapters which may be stored as individual documents. This allows for much easier insertions and deletions of blocks of text and may be found to be a great convenience feature of SuperSCRIPSIT.

Editor's note: Here is another way to print a portion of a document using SuperSCRIPSIT.

1. Surround only the section of the document that you want to print with block markers. Do a @ S at the beginning and a @ E at the end of the section.

2. Then press @ B for Block action command and P to print the block. When the print menu appears, enter the correct page number by the "Begin numbering as page prompt:" and print the document. Only the section enclosed within the block markers will be printed.

Merging SuperSCRIPSIT Files

Bradley Murray, S.J.
Loyola College
4501 N. Charles St.
Baltimore, MD 21210

One of the features of Scripsit sorely missed in the vastly improved SuperSCRIPSIT is the L,C (load, chain) command, which allows several Scripsit files to be merged into one. For example, a lawyer preparing a will could merge together many standard legal paragraphs, make necessary changes, and quickly produce a will tailored to an individual client's needs. At first blush, SuperSCRIPSIT does not seem to have this ability. Read on.

Each paragraph to be merged is prepared as follows. ⟨O⟩pen the document, calling it, for instance, PAR1/SCR. Write the text, and save it in the ordinary way with @ Q. This is a SuperSCRIPSIT file. It must now be converted to an ASCII file, by choosing ⟨A⟩ for the ASCII text conversion utility from the main menu. The following prompt appears:

FROM which format do you wish to convert (Scripsit/ASCII)

Type S in order to convert FROM your SuperSCRIPSIT file to an ASCII file. You will then be asked:

Name of Scripsit File:
Respond with PAR1/SCR.

Name of ASCII file:
Respond with PAR1/ASC.

Your SuperSCRIPSIT file will be converted to an ASCII file and saved under the name PAR1/ASC. SuperSCRIPSIT will return you to your text, from which a @ Q will return you to the main menu. You will then open the file PAR2/SCR; write your text; save it, and return to the main menu with a @ Q. Make a conversion to an ASCII file, PAR2/ASC, just as you did with your first file, PAR1/SCR. The procedure is repeated for each standard paragraph you wish to save for later merging.

Unless you think they will need future modification, you no longer need the original SuperSCRIPSIT files. They can be KILLed.

The stage is now set for the merge. From the main menu, ⟨O⟩pen the master file to which you will append PAR1/ASC, PAR2/ASC, etc. Write the text (it might only be a few words) and save it in the ordinary way with @ Q. Assume you saved it as MAIN/SCR. You will now respond to the main menu with ⟨A⟩ and answer the prompts,

FROM which format do you wish to convert (Scripsit/ASCII)

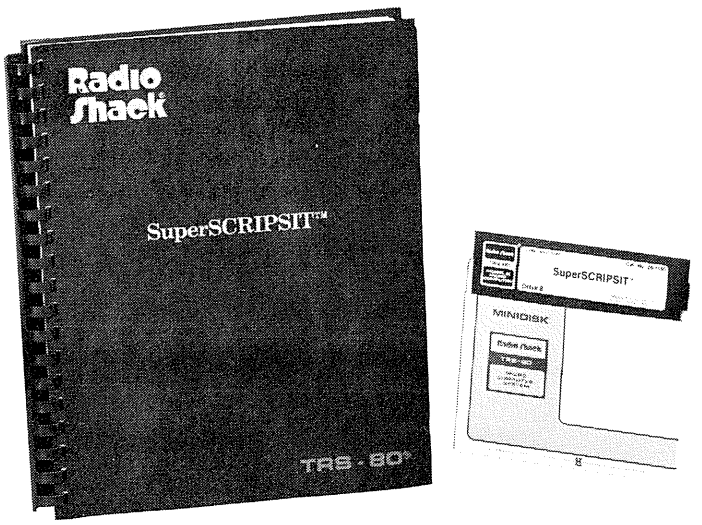
will be answered with ⟨A⟩ for ASCII. The prompt

Name of Scripsit file

will be answered with the name of your main file, in this case MAIN/SCR while

Name of ASCII FILE

will be answered with PAR1/ASC. This will re-convert the ASCII file PAR1/ASC back to a SuperSCRIPSIT file and at the same time will append it to MAIN/SCR. You must now save this version with a @ Q, which will also bring you back to the main menu. From here, you can repeat the procedure with PAR2/ASC and append it to MAIN/ASC, which will now consist of the original MAIN plus PAR1 plus PAR2. Simply repeat the procedure for as many files as you wish to merge in MAIN/SCR. If you load one or more in the wrong order, you can, of course, use SuperSCRIPSIT's block move command.



Computer Clubs

ACSCO-SWAP
Box 28606
Columbus, OH 43228-0606

Broome CoCo Club
c/o Bucky Helmer, President
57 Front Street
Binghamton, NY 13015
723-8223/724-5726

Central Alabama Microcomputer Society
Jon Dawkins, President
P.O. Box 17021
Montgomery, AL 36117
(205) 281-8220 (President)
(205) 272-5069 (Forum 80)

Color Computer Club
P.O. Box 337
Gardendale, AL 35071

Dal-Clif Computer Club
Kimball H.S.
David Gattis
3606 S. Westmoreland at Kiest
Dallas, TX
243-6764

DALTRUG TRS-80 Users Group
UTD-Richardson, Jonnson Hall
2601 N. Floyd Rd.
Richardson, TX
(214) 370-2432

Fort Worth TRS-80 Users Group
3006 SE Loop 820
Fort Worth, TX
(817) 429-7055

Iowa City Color Computer Users Group
c/o Susan P. Chapler
R.R. #6 The Woods
Iowa City, IA 52240

K.C. South Computer Club
Bill Baker, Secretary
2419 Queen Ridge Dr.
Independence, MO 64055
(816) 252-3616

Mid Cities TRS-80 Users Group
UTA Arlington, Davis Hall, Room B87
800 South Cooper
Arlington, TX
(817) 637-4510

Penn-Jersey Color Computer Club
1231 Walnut St.
Allentown, PA 18102
(215) 253-5733 or (215) 434-6387

TRS-80 Users Group of Central Connecticut
P.O. Box 1575
Hartford, CT 06144

Alternate Cursor for the Color Computer

This program for changing the cursor on the Color Computer comes from Computer Customer Services with the following disclaimer.

Radio Shack does not support the use of the following program. Use of this program is left strictly to the user's discretion, and Radio Shack is not responsible for any results that the program may produce.

```
10Ø '****************************
*   ALTERNATE CURSOR FOR   *
*    THE COLOR COMPUTER    *
*       VERSION  2.Ø       *
****************************
* 1) PROTECT MEMORY FOR    *
*    MACHINE LANGUAGE      *
11Ø '*    PROGRAM (THE CURSOR   *
*    ROUTINE IS 152 BYTES  *
*    LONG)                 *
* 2) ENTER AND RUN PRO-    *
*    GRAM AS WRITTEN       *
* 3) RESPOND TO THE        *
*    "START ADDRESS ?"     *
12Ø '*    PROMPT BY ENTERING    *
*    THE ADDRESS OF THE    *
*    FIRST BYTE IN MEM     *
*    ORY AT WHICH YOU      *
*    WISH TO LOAD THE      *
*    ROUTINE               *
* 4) IF THE PROGRAM RE-    *
13Ø '*    TURNS THE "DATA       *
*    ERROR" MESSAGE THEN   *
*    CHECK DATA STATE-     *
*    MENTS FOR ERRORS      *
* 5) AFTER A SUCCESSFUL    *
*    RUN TYPE  :           *
*    <N><E><W><ENTER>      *
14Ø '* 6) START ADDRESS + 99    *
*    CONTAINS THE ASCII    *
*    CODE TO BE USE FOR    *
*    THE NORMAL CURSOR     *
*    WILL RETURN IN THE    *
*    EDIT MODE; BACKARROW  *
15Ø '*    WILL RETURN ALTER-    *
*    NATE CURSOR           *
****************************
16Ø CLS
17Ø INPUT "START ADDRESS";START
18Ø FOR ADDRESS = START TO START + 151
19Ø READ CODE
2ØØ CHECKSUM = CHECKSUM + CODE
21Ø POKE ADDRESS , CODE
22Ø NEXT ADDRESS
23Ø IF CHECKSUM = 12660 THEN 27Ø
24Ø PRINT "DATA ERROR"
25Ø PRINT "CHECK DATA LINES"
26Ø END
27Ø DEFUSRØ = START
28Ø A = USRØ (Ø)
29Ø END
3ØØ DATA Ø26 , Ø8Ø , 182 , ØØ1
31Ø DATA 13Ø , 167 , 141 , ØØØ
32Ø DATA Ø94 , 19Ø , ØØ1 , 131
33Ø DATA 175 , 141 , ØØØ , Ø88
34Ø DATA 182 , ØØ1 , 1Ø3 , 167
35Ø DATA 141 , ØØØ , 126 , 19Ø
36Ø DATA ØØ1 , 1Ø4 , 175 , 141
37Ø DATA ØØØ , 12Ø , 182 , ØØ1
38Ø DATA Ø12 , 167 , 141 , ØØØ
39Ø DATA Ø58 , 19Ø , ØØ1 , Ø13
4ØØ DATA 175 , 141 , ØØØ , Ø52
41Ø DATA 134 , 126 , 183 , ØØ1
42Ø DATA 1Ø3 , 183 , ØØ1 , 13Ø
43Ø DATA 183 , ØØ1 , Ø12 , Ø48
44Ø DATA 141 , ØØØ , Ø41 , 191
45Ø DATA ØØ1 , 131 , Ø48 , 141
46Ø DATA ØØØ , Ø4Ø , 191 , ØØ1
47Ø DATA 1Ø4 , Ø48 , 141 , ØØØ
48Ø DATA ØØ6 , 191 , ØØ1 , Ø13
49Ø DATA Ø28 , 175 , Ø57 , 1Ø9
5ØØ DATA 141 , ØØØ , Ø15 , Ø39
51Ø DATA Ø1Ø , 166 , 141 , ØØØ
52Ø DATA Ø1Ø , 167 , 159 , ØØØ
53Ø DATA 136 , Ø15 , 148 , Ø18
54Ø DATA Ø18 , Ø18 , ØØØ , 175
55Ø DATA 1Ø8 , 14Ø , 251 , Ø18
56Ø DATA Ø18 , Ø18 , 129 , Ø13
57Ø DATA Ø38 , ØØ2 , Ø32 , Ø34
58Ø DATA 129 , ØØ8 , Ø38 , Ø26
59Ø DATA 111 , 14Ø , 235 , Ø93
6ØØ DATA Ø39 , Ø14 , Ø52 , Ø16
61Ø DATA 158 , 136 , 134 , Ø96
62Ø DATA 167 , 132 , Ø48 , Ø31
63Ø DATA 159 , 136 , Ø53 , Ø16
64Ø DATA Ø79 , 1Ø8 , 14Ø , 214
65Ø DATA Ø32 , ØØ7 , 129 , Ø21
66Ø DATA Ø38 , ØØ3 , 111 , 14Ø
67Ø DATA 2Ø5 , Ø18 , Ø18 , Ø18
```

Poor Man's Editor II and Mailing Labels

Darrin Smith
Box 221
Fort Ashby, WV 26719

These two programs were written for a 48K Model III.

POOR MAN'S EDITOR

```
2 ' BY DALLAS L. POWELL 11/27/81
4 ' REFINED BY  DARRIN P. SMITH  Ø8/23/82
6 ' FOR TRS-8Ø MODEL III & LINE PRINTER VI
1Ø X = 1
2Ø CLS
3Ø PRINT TAB(15),"***  E D I T O R  II  ***"
4Ø FOR Q = 1 TO 4ØØ
    : NEXT Q
5Ø ' *** EDITOR ***
6Ø CLEAR 8ØØ
    : POKE 16427, 62
7Ø DIM A$(1Ø)
8Ø A$(1) = CHR$(8)
    : A$(2) = CHR$(9)
    : A$(3) = CHR$(1Ø)
    : A$(4) = CHR$(13)
    : A$(5) = CHR$(32)
    : A$(6) = CHR$(64)
    : A$(7) = STRING$(8,A$(5))
    : A$(8) = CHR$(95)
9Ø X = 1
1ØØ CLS
    : INPUT "NEW PRINT <Y> OR <N> "; Y$
11Ø IF Y$ = "Y" THEN B$(X) = ""
    : ELSE 33Ø
12Ø CLS
13Ø PRINT@9ØØ, "Type less than 241 characters: To end
    type @"
14Ø FOR I = 1 TO 23
    : SET (1,I)
    : SET (127,I)
    : NEXT I
```

```
150 '
160 I$ = INKEY$
170 IF I$ = CHR$(9) THEN 160
180 FOR I = 1 TO 6
     : IF A$(I) = I$ THEN 210
190 NEXT I
     : GOSUB 380
     : GOTO 160
200 '
210 ON I GOTO 230, 240, 250, 250, 310, 330
220 '         BK-SP  TAB DN-AW  ENT SP   "@"
230 ON ERROR GOTO 460
     : Z1 = LEN(B$(X))
     : Z2 = Z1 - 1
     : C$ = LEFT$(B$(X),Z2)
     : B$(X) = C$
     : GOSUB 390
     : GOTO 160
240 B$(X) = B$(X) + A$(7)
     : GOSUB 390
     : GOTO 160
250 Z1 = LEN(B$(X))
260 IF Z1 < 64 THEN Z3 = 64-Z1
     : GOTO 300
270 IF Z1 < 128 THEN Z3 = 128-Z1
     : GOTO 300
280 IF Z1 < 192 THEN Z3 = 192-Z1
     : GOTO 300
290 IF Z1 < 241 THEN Z3 = 241-Z1
     : ELSE 420
300 Z4$ = STRING$(Z3,A$(5))
     : B$(X) = B$(X) + Z4$
     : GOSUB 390
     : GOTO 160
310 B$(X) = B$(X) + A$(5)
     : GOSUB 390
     : GOTO 160
320 '
330 CLS
     : PRINT B$(1)
     : PRINT B$(2)
     : INPUT "HARD COPY <Y> OR <N> ";Y$
     : IF Y$ = "Y" THEN INPUT "Large or small (L) or
    (S)"; L$
     : ELSE 100
340 IF L$ = "L" THEN LPRINT CHR$(31)
     : ELSE LPRINT CHR$(30)
350 LPRINT B$(1)
360 LPRINT B$(2)
     : GOTO 100
370 '
380 B$(X) = B$(X)+I$
390 Z1 = LEN(B$(X))
     : Z2 = Z1+64
     : IF Z1 = 241 THEN 420
400 IF Z1 > 240 THEN 420
410 PRINT@64, B$(X)
     : PRINT@Z2, A$(8)
     : RETURN
420 IF LEN(B$(1)) > 240 THEN 430
430 IF X = 2 THEN 450
440 X = X + 1
450 IF LEN (B$(2)) > 240 THEN 330 ELSE 120
460 RESUME
480 END
```

MAILING LABELS

```
4 CLEAR 1000
2 ' ***************************
4 ' *     BY DARRIN SMITH     *
6 ' * FOR MICRO COMPUTER NEWS *
8 ' *     CREATED ON        *
10 '*    A 48K MODEL III     *
12 '* REFINEMENTS ARE WELCOME *
14 '***************************
16 CLS
18 PRINT "FOR LINE PRINTER VI "
     : PRINT
     : PRINT
20 LINE INPUT "NAME OR BUSINESS..."; N$
30 LINE INPUT "ADDRESS "; B$
40 LINE INPUT "CITY..."; C$
50 LINE INPUT "STATE..."; S$
60 LINE INPUT "ZIP ..."; Z$
70 INPUT "HOW MANY (2 TO 100) EVEN NUMBER PLEASE...";
    M
80 INPUT "IF EVERY THING IS SPACED CORRECT THEN PRESS
    <Y> ELSE PRESS <N> HIT
<ENTER>..."; Y$
90 IF Y$ = "N" THEN 20
100 M = M / 2
110 FOR Q = 1 TO M
120 LPRINT N$; TAB(34); N$
130 LPRINT B$; TAB(35); B$
140 LPRINT C$; " "; S$;
150 LPRINT TAB(35); C$ " "; S$
160 LPRINT Z$; TAB(35); Z$
170 LPRINT
     : LPRINT
180 NEXT Q
```

Label Printing Utility

David D. Bailey
3962 Petite Dr. W.
Jacksonville Beach, FL 32250

LABEL/BAS is a program for use with SCRIPSIT or SuperSCRIPSIT word processing software.

I have found it inconvenient to write labels by hand or block the address section of a letter to make a label. The program requires a disk with BASIC. If you are using SCRIPSIT, it can be on the SCRIPSIT disk. If you are using SuperSCRIPSIT, it must be on a utility disk since BASIC is not included with SuperSCRIPSIT.

The program should be self-explanatory except for lines involving "Company." If you do not want a company name, press ENTER at the prompt. The printed label will not have an extra line when company is not used.

I would also suggest pressing the reset button when changing from SuperSCRIPSIT to a disk with BASIC. Sometimes BASIC will not load properly and "hangs up," if loaded directly after using SuperSCRIPSIT.

```
10 'LABEL/BAS
20 'LABEL PRINTING UTILITY
30 'BY DAVID D. BAILEY
40 CLEAR200
50 POKE16409,0
60 X$="LABEL PRINTING UTILITY"
     : SG$=STRING$(63,136)
70 CLS
     : PRINT TAB(20)X$
     : PRINT
     : LINE INPUT "NAME ? "; G$
80 LINE INPUT "COMPANY ? "; L$
90 LINE INPUT "ADDRESS ? "; B$
100 LINE INPUT "CITY ? "; C$
110 E$=" "
120 LINE INPUT "STATE ZIP ? "; D$
130 CLS
     : PRINT TAB(20)X$
     : PRINT
     : PRINT
     : PRINT "LABEL TO PRINT:"
     : PRINT
140 PRINTG$
150 IF L$="" THEN 170
```

```
160 PRINT L$
    : PRINT B$
    : PRINT C$; ", ";
    : PRINT D$
    : GOTO180
170 PRINT B$
    : PRINT C$;", ";
    : PRINT D$
    : GOTO180
180 PRINT
    : PRINT SG$
    : PRINT
    : PRINT TAB(10) "SELECT <P> TO PRINT <E> TO
    EXIT"
190 A$=INKEY$
200 IFA$="E" THEN 40
210 IFA$="e" THEN 40
220 IFA$="P" THEN 250
230 IFA$="p" THEN 250
240 GOTO190
250 LPRINT G$
260 IF L$="" THEN 280
270 LPRINT L$
280 LPRINT B$
290 LPRINT C$; ","; E$D$
300 LPRINT
    : LPRINT
310 CLS
    : PRINT TAB(20)X$
    : PRINT
    : PRINT
    : PRINT TAB(20)"<R> REPEAT"
320 PRINT TAB(20)"<N> NEW NAME"
330 PRINT TAB(20)"<E> EXIT TO BASIC"
340 PRINT TAB(20)"<T> EXIT TO TRSDOS Ready"
350 PRINT
    : PRINT SG$
    : PRINT
    : PRINT TAB(25)"SELECT"
360 A$=INKEY$
370 IF A$="R" THEN 250
380 IF A$="r" THEN 250
390 IF A$="N" THEN 40
400 IF A$="n" THEN 40
410 IF A$="E" THEN POKE16409,1
    : CLS
    : END
420 IF A$="e" THEN POKE16409,1
    : CLS
    : END
430 IF A$="T" THEN POKE16409,1
    : CLS
    : CMD"S"
440 IF A$="t" THEN POKE16409,1
    : CLS
    : CMD"S"
450 GOTO360
```

Little Ol' Label Maker

Paul Martino
Bob Rich Schroeder Trucking Inc.
61 Napoleon St.
San Francisco, CA 94124

Necessity being the mother of invention, here is another label maker. After running a 2,000 name list with Profile 3+, we found some 60 errors that we made in names or addresses. Unfortunately, I couldn't think of an easy way to print individual labels with Profile 3+ without the program reading the whole file to print each label and then re-reading again for the next label.

My program gives you four options for salutations to eliminate some repetitive typing, an option for the number of labels you want of each, and an option for using your local city, state and zip code or to insert a different one.

It runs fast enough the way it is, so deleting the REMS and compressing won't make much difference. You could alter lines 80-100 to allow for printing of more labels. You could also eliminate lines 370, 380, and 430 to eliminate using the person's record number if it's not important to your files. If you do so, add one LPRINT statement to force a line feed to keep your label assignment correct.

When I first got my computer and didn't understand programming I depended on your newsletter for programs and hints. I hope this little program will help some other novice gain some insights into the functions and abilities of the microcomputer.

The program runs on Model III disk BASIC but can easily be altered for non disk users by eliminating the line input statements and using simple inputs instead.

Congratulations on Profile 3+. I can now sort 1,800 records of 80 characters each in less than three minutes. It used to take me 45 minutes to insert 125 records into a 800 record base with the old version of Profile.

```
10 REM**** LITTLE OL' LABEL MAKER
20 REM**** BY Paul 1. Martino,   S.S.F.  CA.
30 REM**** NO RIGHTS RESERVED, USE PROGRAM FREELY
40 REM**** ALIGN YOUR LABELS IN PRINTER BEFORE
    RUNNING PROGRAM
50 CLEAR 1000
60 CLS
    : GOSUB 530
70 CLS
80 PRINT@450,"    HOW MANY LABELS OF EACH DO YOU WANT
    ? <1-9>
90 Q$=INKEY$
    : IF Q$="" THEN 90
100 N=VAL(Q$)
    : IF N<1  THEN GOTO 90
110 CLS
120 PRINT"           SALUTATION SELECTION
130 PRINT
    : PRINT"            1) MR.& MRS.
140 PRINT"           2) MRS.
150 PRINT"           3) MR.
160 PRINT"           4) MS.
170 PRINT"           5) END PROGRAM
180 PRINT
    : PRINT"         < ENTER YOUR SELECTION 1-4 >
190 K$=INKEY$
    : IF K$="" THEN 190  ELSE K=VAL(K$)
200 IF K=1 THEN A$="MR. & MRS.
210 IF K=2 THEN A$="MRS.
220 IF K=3 THEN A$="MR.
230 IF K=4 THEN A$="MS.
240 ID K=5 THEN CLS
    : PRINT@470,"PROCESSING TERMINATED"
    : PRINT@536,"REMOVE DISKETTES"
    : END
250 CLS
260 PRINT A$
270 LINEINPUT"NAME ";B$
280 CLS
    : PRINT A$+" "+B$
290 LINEINPUT"ADDRESS ";C$
300 CLS
    : PRINT A$+" "B$
    : PRINT "  "+C$
310 PRINT"IS CITY SAN FRANCISCO, CA. (Y OR N)
320 K$=INKEY$
    : IF K$="" THEN 320
```

```
330 IF K$="Y" THEN D$="SAN FRANCISCO,  CA.   94080"
    : GOTO360
340 PRINT
    : LINE INPUT "WHAT IS THE CITY AND ZIP CODE
    ";DD$
350 D$=DD$+"   "
360 CLS
    : PRINT A$+" "+B$
    : PRINT"  "+C$
    : PRINT"  "+D$
370 LINE INPUT" WHAT IS REC # ";FF$
380 F$=FF$
390 CLS
400 PRINT A$+" "+B$
410 PRINT"  "+C$
420 PRINT"   "+D$
430 PRINT
    : PRINT"("+F$+")"
440 PRINT @470,"IS ENTRY OK <Y OR N>
450 K$=INKEY$
    : IF K$="" THEN 450
460 IF K$="N" THEN 110
470 FOR X=1 TO N
480 LPRINT A$;B$
    : LPRINT C$
    : LPRINT D$;E$
    : LPRINT
    : LPRINT F$
490 LPRINT
500 NEXT X
510 GOTO 110
520 END
530 PRINT@64,STRING$(64,188);
540 PRINT@896,STRING$(64,188);
550 PRINT@467,"LITTLE OL' LABEL MAKER
560 PRINT@536,"PAUL MARTINO
570 PRINT@602,"01/20/83
580 FOR I=128 TO 896 STEP 64
590 PRINT@I,CHR$(191);
600 PRINT@I+63,CHR$(191);
610 NEXT I
620 FOR T=1 TO 1200
    : NEXT T
    : RETURN
```

Zip Code Files

Mark C. Smith
2225 Shattuck Rd.
Saginaw, MI 48603

Most programmers eventually find a use for data files in which a name, street address, city, state and zip code are all stored on disk or cassette. This program is a demonstration one in which the last three fields (city, state, zip) are simply abbreviated. Only the zip code is entered and stored for each record. A master file of zip codes along with the corresponding city and state is made. This file is called "ZIPFILE." In even modest applications this can save entering time and, obviously, magnetic storage space. I plan to use this technique in every data file application I can.

The program uses a variable length file so be sure to answer the BASIC "How Many Files?" prompt correctly with the number of files followed by a V. I wrote this program on a 48K Model III with dual floppy disk drives. It should be applicable to other models and many new or existing programs. I am 16 years old and have been programming for around 2 years now. Thank you for a well written publication and good luck to all who try the idea!

```
100 'DEMO OF ZIPFILE TECHNIQUE
110 'IN MODEL III DISK BASIC 1.3
120 'BY MARK C. SMITH  10 MAY,1982
130 CLEAR 3000
140 DEFINT A-Z
    : DIM N$(20), AD$(20), CS$(20), Z$(40), CT$(40)
150 CLS
160 GOSUB 1000
    : GOSUB 5000
165 'INPUT NAMES
170 INPUT "HOW MANY NAMES"; A
180 IF A>20 THEN 170
190 FOR B=1 TO A
200 CLS
    : PRINT TAB(23); "Z I P  D E M"
    : PRINT @125, B;
    : PRINT @64, "";
210 LINE INPUT "NAME:"; N$(B)
220 LINE INPUT "STREET ADDRESS:"; AD$(B)
230 INPUT "ZIPCODE"; IZ$
240 IF VAL(IZ$)>99999 THEN 230
250 GOSUB 2000
260 NEXT B
265 'LIST NAMES
270 CLS
    : PRINT "HERE IS THE LIST: "
280 FOR B=1 TO A
290 PRINT
    : PRINT B;"--"
    : PRINT N$(B)
    : PRINT AD$(B)
    : PRINT CS$(B)
300 IF B/2=INT(B/2) AND B<>A THEN PRINT
    : PRINT
    : PRINT "ANY KEY..."
    : GOSUB 4510
    : CLS
310 NEXT B
320 GOSUB 3000
    : CLOSE
    : PRINT @920, "** ANY KEY **"
    : GOSUB 4500
    : FOR X=1 TO 15
    : PRINT
    : NEXT
    : PRINT TAB(25); "PROGRAM TERMINATED"
    : END
999 'GET ZIPCODE FILE
1000 GOSUB 3500
    : FIELD 1, 2 AS NB$
    : GOSUB 4000
    : GET 1,1
    : D=CVI(NB$)
    : FIELD 1, 33 AS CY$, 5 AS ZP$
1010 IF D=1 THEN RETURN
    : ELSE FOR B=2 TO D
    : GET 1, B
    : C1$=CY$
    : GOSUB 1500
    : CT$(B-1)=C1$
    : Z$(B-1)=ZP$
    : NEXT
1020 RETURN
1499 'REMOVE SPACES FROM C1$
    : ELSE C1$=LEFT$(C1$,INSTR(C1$,"  "))
    : RETURN
1500 IF INSTR(C1$,"  ")=0 THEN RETURN
1999 'CHECK FOR ZIP
2000 FG=0
    : IF D<2 THEN 2040
    : ELSE FOR E=1 TO D-1
    : IF FG>0 THEN 2020
2010 IF Z$(E)=IZ$ THEN FG=E
2020 NEXT E
2030 IF FG>0 THEN GOSUB 2500
    : RETURN
```

```
2039 'ENTER AND SAVE NEW ZIP INFO
2040 PRINT "ZIP NOT ON FILE..."
2050 LINE INPUT "PLEASE ENTER CITY,STATE (EX.
     SAGINAW, MI.)
     : "; I$
2060 FG=D
     : D=D+1
     : LSET ZP$=IZ$
     : LSET CY$=I$
     : PUT 1, D
     : C1$=CY$
     : GOSUB 1500
     : CT$(D-1)=C1$
     : Z$(D-1)=IZ$
     : FG=D-1
     : GOSUB 2500
     : RETURN
2499 'ASSIGN CITY,STATE & ZIP TO VARIABLE FOR LISTING
2500 CS$(B)=CT$(FG)+"  "+IZ$
     : RETURN
2999 'PUT NUMBER OF ZIPS ONTO DISK
3000 FIELD 1, 2 AS NB$
     : LSET NB$=MKI$(D)
     : PUT 1, 1
     : RETURN
3499 'OPEN ZIPFILE
3500 OPEN "R", 1, "ZIPFILE", 38
     : RETURN
3999 'CHECK FOR PREVIOUSLY CREATED FILE
4000 IF LOF (1)=0 THEN LSET NB$=MKI$(1)
     : PUT 1, 1
4010 RETURN
4499 'RETURN ON ANY KEY
4500 K$=""
4510 K$=INKEY$
     : IF K$="" THEN 4510
     : ELSE RETURN
5000 A$=STRING$(63,191)
     : B$=CHR$(236)+"Z I P  D E M"
     : V=896
5010 FOR X=1 TO 7
     : PRINT @V, CHR$(255)+CHR$(255);
     : FOR Y=1 TO 80
     : NEXT
     : PRINT @V, A$; B$;
     : FOR Y=1 TO 80
     : NEXT
     : NEXT
5015 PRINT @V, CHR$(255);
     : B$="Z I P  D E M"
     : V=V + 38
5020 FOR X=1 TO 15
     : PRINT @(V-64*(X-1)-X-1), CHR$(255);
     : PRINT @(V-64*(X-1)-X),B$
     : FOR Y=1 TO 30
     : NEXT
     : NEXT
     : PRINT @64, "";
     : RETURN
```

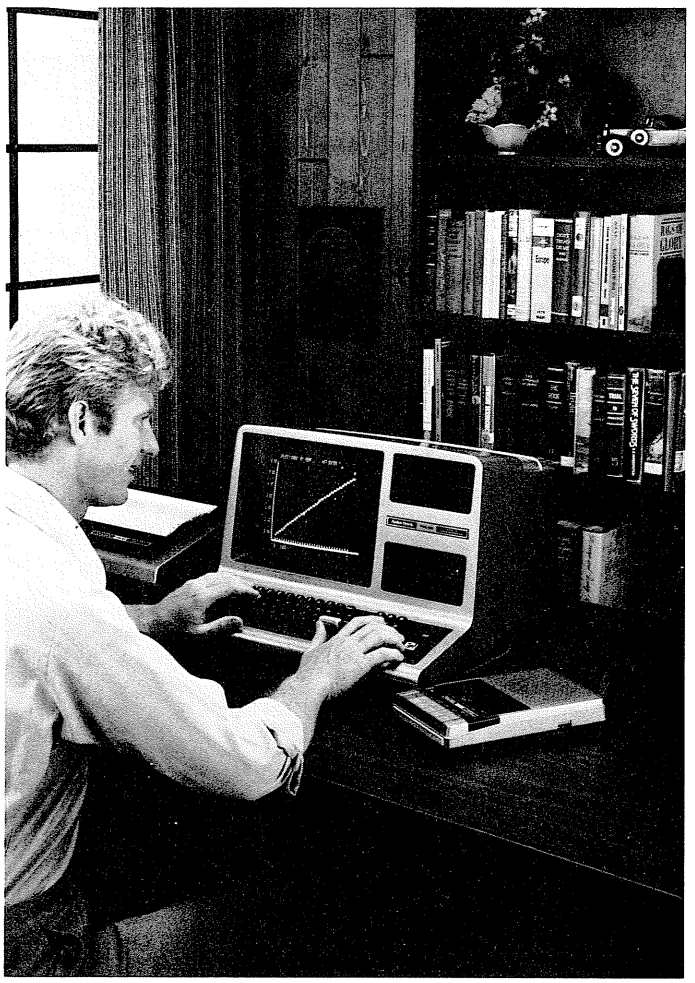
Better Cursor Response for the Directory Menu

Henry P. Stevens
1024 Coyote Rd.
San Jose, CA 95111

Recently while looking through back issues of the *TRS-80 Microcomputer News*, searching for programs that I could use with my newly installed disk drives, I spotted Mr. Ditucci's Directory/Menu program in the November 1981 issue. I started using it right away, and for the same reasons as those expressed by Mr. Ditucci, I find the routine tremendously helpful. But I do have one complaint. The cursor does not always respond promptly to the single tap of a key. To correct the problem, I have installed a machine language "Keyboard scan and cursor blink" subroutine. The fully relocatable code is packed into string K$ by lines 130 and 140. You will find your cursor much more responsive with this modified version of the routine.

```
10 REM***       9 FEB 83 NO RIGHTS RESERVED      ***
20 REM***      MOD III DIRECTORY/MENU PROGRAM     ***
30 REM***  ORIGINAL PROGRAM WRITTEN BY: JAMES C.
     DITUCCI ***
40 REM***     NOV 81 ISSUE TRS-80 NEWSLETTER PAGE 16
     ***
60 REM***      MODIFIED BY: HENRY P. STEVENS     ***
70 REM***              1024 COYOTE RD      ***
80 REM***              SAN JOSE,CA.95111     ***
90 CLEAR 1000
100 ON ERROR GOTO90
105 CMD "B","OFF"
110 K$=STRING$(191,"Z")
120 X1#=VARPTR(K$)
     : X#=PEEK(X1#+2)*256+PEEK(X1#+1)
     : IF X#>32767 THEN X#=-1*(65536-X#)
130 A=X#
     : FOR A=ATOA+191
140 READ Y
     : POKEA,Y
150 NEXT
     : RESTORE
160 CMD"D
     : 0"
     : POKE15367,49
170 PRINT CHR$(15)
     : PRINT@960,"USE ARROWS TO POSITION CURSOR-PRESS
     <ENTER> TO SELECT PROGRAM";
180 D = PEEK(16419)
     : POKE16419,143
     : P$=""
190 DEFUSR0=X#
200 V=USR(0)
210 REM ***EXECUTE PROGRAM SELECTED ***
220 FOR X=0 TO 11
     : C=PEEK(V+X)
     : IF C<32 THEN C=(C+64)
230 IF C=32 THEN NEXT ELSEP$=P$+CHR$(C)
     : NEXT
240 LP$=LEFT$(P$,7)
     : IF LP$="DiskDri" OR LP$="Ready"THEN RUN
250 IF LP$="DRIVE
     : 1 "THEN CMD"D
     : 1"
     : POKE15367,48
     : GOTO170
260 IF LP$="DRIVE
     : 0" THEN RUN ELSE POKE16419,D
270 L = LEN(P$)
     : IF RIGHT$(P$,4)="/CMD" P$=LEFT$(P$,L-4)
     : GOTO300
280 IF RIGHT$(P$,4)="/BLD" THEN P$="DO "+P$
     : GOTO300
290 GOSUB310
     : RUN P$
300 GOSUB310
     : CMD"I",P$
310 ZB$="NOW LOADING "+CHR$(244)+CHR$(245)+CHR$(246)
     +" "+P$
320 ZN$=STRING$(63,32)
     : Z=960+(64-LEN(ZB$))/2
330 PRINT CHR$(21);
     : FOR X=1TO5
     : PRINT @960,ZN$;
     : FOR T=1TO30
```

```
     : NEXT T
340 PRINT @Z,ZB$;
     : FOR T=1TO90
     : NEXT T
     : NEXT X
     : PRINT CHR$(21);
     : CMD"B","ON"
     : RETURN
350 DATA 33, 1, 60, 43, 126, 245, 1, 32, 255, 17,
     255, 255, 19, 58, 255
360 DATA 56, 187, 32, 2, 24, 26, 241, 245, 119, 58,
     64, 56, 254, 1, 40
370 DATA 49, 254, 8, 40, 49, 254, 16, 40, 78, 254,
     32, 40, 96, 254, 64
380 DATA 40, 125, 16, 216, 13, 40, 2, 24, 249, 30,
     255, 28, 58, 35, 64
390 DATA 190, 32, 5, 241, 245, 119, 24, 194, 58, 35,
     64, 119, 62, 255, 60
400 DATA 187, 40, 184, 24, 27, 241, 195, 154, 10, 17,
     64, 255, 20, 183, 237
410 DATA 82, 241, 126, 245, 62, 60, 188, 56, 225, 40,
     223, 25, 241, 126, 245
420 DATA 24, 196, 1, 128, 255, 16, 254, 13, 32, 251,
     24, 145, 17, 64, 255
430 DATA 20, 25, 241, 126, 245, 62, 32, 190, 32, 194,
     241, 33, 1, 60, 43
440 DATA 126, 245, 24, 185, 17, 15, 255, 20, 62, 255,
     60, 189, 40, 175, 62
450 DATA 64, 189, 40, 170, 62, 128, 189, 40, 165, 62,
     192, 189, 40, 160, 183
460 DATA 237, 82, 241, 126, 245, 24, 152, 17, 15,
     255, 20, 25, 241, 126, 245
470 DATA 62, 32, 190, 32, 139, 17, 4, 255, 20, 25,
     24, 186
```



Supershopper for the PC-1

John Graham
2650 N. W. 63rd Street
Seattle, WA 98107

This program may amuse those who shop calculator-in-hand. You may enter thirty items. Since the string variable can only hold seven letters, abbreviate longer items or expect to lose the tail end of a long name. You may enter the number of each item wanted. The program will keep a running total of how much you have spent.

When asked "NEW, OLD OR ADD?," enter NEW to make a list (use NEW at no other time), ADD to add to a list, and OLD to review or enter prices to a list. You may review, BREAK and then ADD, but you cannot ADD once you begin entering prices unless you play with variable G.

The program in OLD/E (entry) mode will ask for an item. If you enter a name that is on the list, the computer will display the item name and ask for a price. Enter 0 for a subtotal. If you enter any positive number, the program will multiply by the number of items and add this to the total. It will then scratch the item off the list. Reviewing an empty list yields the grand total.

If you fetch an item from the end of a long list, expect a frustratingly long wait. The program will move items forward into "blank" spaces. This is some help but readers may see an improved approach. With demands on flexible memory, this program occupies all but 27 steps, so don't add much without picking up memory somewhere.

```
10 PAUSE "   SUPPERSHOPPER"
20 INPUT " NEW, OLD OR ADD?", A$
     : IF A$ = "ADD" THEN 265
30 IF A$ = "NEW" THEN 260
40 PAUSE "YOUR SHOPPING LIST"
     : D = 0
50 FOR B = 8 TO 38
     : E = B - 7
     : C = B + 31
60 IF A(C) < 0 NEXT B
70 IF A(C) > 0 LET D = 1
80 IF A$(B) = "" THEN 100
90 PAUSE E; "   ";A(C); A(B); "(S)"
     : NEXT
100 B = 38
     : NEXT B
     : IF D > 0 THEN 120
110 PAUSE "LIST IS EMPTY"
     : GOTO 250
120 INPUT "REVIEW OR ENTRY? R OR E", A$
130 IF A$ = "R" THEN 40
140 INPUT "ITEM TO ENTER?" A$
150 FOR B = 8 TO 38
     : C = B + 31
160 IF A$(B) = "" THEN 200
170 IF A(C) > 0 THEN 180
175 D = B + 1
     : A$(B) = A$(D)
     : D = C + 1
     : A(C) = A(D)
     : A(D) = -1
     : IF A(C) < 0  NEXT B
180 IF A$ = A$(B) THEN 210
190 NEXT B
200 PAUSE "NAME NOT ON LIST"
     : GOTO 100
```

```
21Ø D = B + 62
     : PAUSE "ENTER PRICE OF "; A$(B)
     : INPUT " $ ?", A(D)
22Ø E = A(C) * A(D)
     : IF A(D) = Ø THEN 3ØØ
23Ø PRINT "  "; A(C); A$(B); " FOR $"; E
24Ø F = F + E
     : A(C) = A(C) * -1
     : GOTO 1ØØ
25Ø PRINT "YOUR TOTAL $"; F
     : PRINT "GOODBYE"
     : END
26Ø PAUSE "ENTER 3Ø ITEMS"
     : PAUSE " 7 LETTERS/ITEM"
     : CLEAR
     : G = 8
265 IF G > 36 THEN 29Ø
27Ø FOR G = G TO 37
     : INPUT "ENTER ITEM", A$(G)
275 PAUSE ""
     : IF A$(G) = "" THEN 29Ø
28Ø C = G + 31
     : INPUT "HOW MANY?", A(C)
     : NEXT G
29Ø PRINT "END LIST"
     : END
3ØØ PRINT "SUBTOTAL $"; F
     : GOTO 21Ø
```

Shopper and Code for the PC-1

Richard O. Sabine
113 Greenpark Dr.
Mobile, AL 36609

The first program Code is a lot of fun and has found its way into several subroutines. It is self documenting.

```
1Ø "CODE"
2Ø "C" CLEAR
     : PRINT"    THIS PROGRAM WILL OUTPUT ALL
     POSSIBLE ELEMENT COMBINATIONS OF"
21 PRINT "ELEMENTS ENTERED     AN ELEMENT CANNOT
     EXCEED 7 CHARACTERS."
22 PRINT "  YOU CAN SET AN ELEMENT TO EQUAL LETTERS,
     WORDS, OR NUMBERS."
23 PRINT "ENTER INDIVIDUAL ELEMENTS ONE AT A TIME.
     ENTER (.) TO INDICATE STOP"
24 PRINT "9 ELEMENTS MAX."
25 Y = 1
     : X = 1
29 FOR Z = 11 TO 19
3Ø INPUT "ELEMENT = "; A$(Z)
31 IF A$(Z) = "." THEN 5Ø
35 Y = Y + 1
4Ø NEXT Z
5Ø GOSUB "COUNTER"
7Ø IF A(Y) = 1 END
9Ø FOR Z = 1 TO Y
1ØØ IF A(Y) = 1 PRINT A$(Z + 1Ø)
11Ø NEXT Z
13Ø PRINT "------"
14Ø GOTO 5Ø
2ØØ "COUNTER"
21Ø X = 1
22Ø FOR Z = 1 TO (Y + 1)
23Ø W = (A(Z) <> X)
     : X = X * A(Z)
24Ø A(Z) = W
25Ø IF X = Ø LET Z = (Y + 1)
26Ø NEXT Z
27Ø RETURN
```

Shopper is a handy little thing to carry grocery shopping. To speed up entries it is designed so no decimal point should be entered. Before using this program, you should always type [C][L][E][A][R] [ENTER]. I found it best not to include the CLEAR statement. If in the program the automatic off feature of the PC-1 shuts off the machine, just run the program again; no values will be lost. If you are like me, you will want to know when you have reached your budget—thus the purpose of the dollar limit. Remember the decimal point is assumed. In line one the value of T can be altered to describe the tax in your area.

```
1 "G"
     : PRINT "CASHIER PROGRAM"
     : PRINT "****************"
     : T =.Ø6
2 INPUT "DOLLAR LIMIT = ?"; R
     : R = R / 1ØØ
3 INPUT "PRICE = ?"; A
     : A = A / 1ØØ
4 IF A = Ø.Ø GOTO 8
5 B = B + A
     : PRINT USING "###.##"; A
6 IF R <= ((B + (B * T)) BEEP 5
7 GOTO 3
8 C = INT ((B +(B*T)) * 1ØØ + .5)
     : C = C / 1ØØ
9 PRINT USING ; "TOTAL = $"; C
     : GOTO 3
1Ø END
```

A Modified Bubble Sort for the PC-1

Edward M. Cohen
Keene, NH 03431

The following program is a modified bubble sort which I call bubble sort. This program sorts 150 numbers. Enter 999 to terminate input.

```
1 "A" CLEAR
     : PRINT "MOD BUBBLE SORT"
     : N=27
2 INPUT " #?"; A(N)
     : PRINT A(N)
3 IF A(N) = 999 GOTO 18
4 N = N + 1
     : GOTO 2
18 N = N - 1
     : B = N
     : REM SORT
21 IF B = 26 THEN 33
22 J = 26
23 FOR I = 27 TO B - 1
     : IF A(I) <= A(I+1) THEN 29
25 T = A(I)
     : A(I) = A(I+1)
     : A(I+1) = T
     : J = I
29 NEXT I
     : B = J
     : GOTO 21
33 FOR X = 27 TO N
     : PRINT A(X)
     : NEXT X
     : END
```

CoCo Business Programs

Marty Ivins
RR 6, Box 88B
Lincoln, NE 68502

Here are six Color Computer programs I have been using for some time with some success in my business. The first three are to compile/revise, alphabetize, and print out an address list. They are developed along the lines of the guides in the color disk manual. They should be quite easily upgraded and expanded to be more useful in anybody's computer operation. The last three programs are developed to provide agricultural sprayer calibration information. They are very broad and may be used to calibrate and find useful rates, and plan application mixtures for many types of agri-application information. They are now used to calibrate everything from field sprayers to orchard and vegetable sprayers.

ALPHA

```
10 CLEAR 4000
20 DIM ARRAY$(300)
30 OPEN"D", #1, "MAIL/DAT", 69
40 FIELD #1, 69 AS INFO$
50 IF LOF(1)=0 THEN PRINT"END"
     : END
60 FOR I=1 TO LOF(1)
70 GET #1, I
75 ARRAY$(I)=INFO$
80 NEXT I
100 FOR J=1 TO I
110 FOR K=J TO I
120 IF ARRAY$(J) < ARRAY$(K) THEN 160
130 TEMP$=ARRAY$(J)
140 ARRAY$(J)=ARRAY$(K)
150 ARRAY$(K)=TEMP$
160 NEXT K
170 NEXT J
180 FOR N=1 TO I
190 LSET INFO$=ARRAY$(N)
200 PUT#1, N
210 NEXT N
220 IF EOF(1)=-1 THEN CLOSE #1
230 GOTO 90
240 CLOSE#1
250 END
```

COMPREV

```
10 CLS
20 PRINT @100, "THIS IS A PROGRAM OF A
     MAILING LIST COMPILER           AND A REVISION
     SEGMENT"
30 PRINT @164, "DO YOU WISH TO COMPILE          A
     MAILING LIST? OR TO           REVISE YOUR LIST?"
40 PRINT @260, "TO COMPILE PRESS 'C'"
50 PRINT @292, "TO REVISE PRESS 'R'"
60 AN$=INKEY$
70 IF AN$="C" THEN 100
80 IF AN$="R" THEN 390
90 GOTO 60
100 CLEAR 2000
110 OPEN"D", 1, "MAIL/DAT", 69
120 FIELD#1, 15 AS LAST$, 10 AS FIRST$, 15 AS
     ADDRESS$, 10 AS CITY$, 2AS STATE$, 5AS ZIP$, 12
     AS PHONE$
130 IF LOF(1)=0 THEN R=1
     : GOTO 190
140 FOR R=1 TO LOF(1)
150 GET #1, R
160 NEXT R
170 FIELD#1, 15 AS LAST$, 10 AS FIRST$, 15 AS
     ADDRESS$, 10 AS CITY$, 2 AS STATE$, 5 AS ZIP$,
     12 AS PHONE$
180 R=R+1
190 CLS
200 PRINT @64
205 INPUT "LAST NAME";L$
210 INPUT "FIRST NAME";F$
220 INPUT "ADDRESS";A$
230 INPUT "CITY";C$
240 INPUT "STATE";S$
250 INPUT "ZIP CODE";Z$
260 INPUT "PHONE NO.";P$
270 LSET LAST$=L$
280 LSET FIRST$=F$
290 LSET ADDRESS$=A$
300 LSET CITY$=C$
310 LSET STATE$=S$
320 LSET ZIP$=Z$
330 LSET PHONE$=P$
340 PUT #1, R
350 PRINT
360 PRINT "MORE DATA(Y/N)"
361 AN$=INKEY$
370 IF AN$= "Y" THEN 180
375 IF AN$="N" THEN 380
376 GOTO 361
380 CLOSE #1
390 CLS
400 PRINT @258, "DO YOU WISH TO REVISE? (Y/N)"
410 AN$=INKEY$
420 IF AN$="Y" THEN 450
430 IF AN$="N" THEN 1510
440 GOTO 400
450 CLS
460 PRINT @106, "SELECTIONS:"
470 PRINT @168, "1)EDIT RECORD"
480 PRINT @200, "2)DELETE RECORD"
490 PRINT @232, "3)END JOB"
500 PRINT @298, "1, 2,  OR 3"
510 AN$=INKEY$
520 IF AN$="" THEN 510
530 ON VAL(AN$) GOSUB 550, 1190, 1500
540 GOTO 450
550 OPEN "D", #1, "MAIL/DAT", 69
560 FIELD #1, 15 AS LAST$, 10 AS FIRST$, 15 AS
     ADDRESS$, 10 AS CITY$, 2 AS STATE$, 5 AS ZIP$,
     12 AS PHONE$
570 FOR R=1 TO LOF(1)
580 GET #1, R
590 CLS
600 PRINT @68, "RECORD NUMBER:";R
610 PRINT @100, LAST$FIRST$
620 PRINT @132, ADDRESS$
630 PRINT @164, CITY$", "STATE$
640 PRINT @196, ZIP$
650 PRINT @260, ;PHONE$
660 PRINT @292, "EDIT  THIS RECORD?(Y/N)"
670 AN$=INKEY$
680 IF AN$="Y"THEN 710
690 IF AN$="N" THEN 1170
700 GOTO 670
710 PRINT @324, "EDIT NAME$ (Y/N)"
720 AN$=INKEY$
730 IF AN$="Y" THEN 760
740 IF AN$="N" THEN L$=LAST$
     : GOTO 770
750 GOTO 720
760 INPUT "NEW LAST NAME";L$
770 GOSUB 2000
790 PRINT @ 324,  "EDIT STREET ADD.? (Y/N)"
800 AN$=INKEY$
810 IF AN$="Y" THEN 840
820 IF AN$="N" THEN A$=ADDRESS$
     :  GOTO 850
830 GOTO 800
840 INPUT "  NEW STREET ADD.";A$
850 PRINT @324, "EDIT CITY? (Y/N)                "
```

```
86Ø AN$=INKEY$
87Ø IF AN$="Y" THEN 9ØØ
88Ø IF AN$="N" THEN C$=CITY$
     : GOTO 91Ø
89Ø GOTO 85Ø
9ØØ INPUT "NEW CITY";C$
91Ø PRINT @324, "EDIT STATE? (Y/N)              "
92Ø AN$=INKEY$
93Ø IF AN$="Y" THEN 96Ø
94Ø IF AN$="N" THEN S$=STATE$
     : GOTO 97Ø
95Ø GOTO 91Ø
96Ø INPUT "NEW STATE";S$
97Ø PRINT @324, "EDIT ZIP (Y/N)"
98Ø AN$=INKEY$
99Ø IF AN$="Y" THEN 1Ø2Ø
1ØØØ IF AN$="N"THEN Z$=ZIP$
     : GOTO 1Ø3Ø
1Ø1Ø GOTO 97Ø
1Ø2Ø INPUT "NEW ZIP";Z$
1Ø3Ø PRINT @324,  "EDIT PHONE NUMBER? (Y/N)       "
1Ø4Ø AN$= INKEY$
1Ø5Ø IF AN$="Y" THEN 1Ø8Ø
1Ø6Ø IF AN$="N" THEN P$= PHONE$
     :  GOTO 1Ø9Ø
1Ø7Ø GOTO 1Ø4Ø
1Ø8Ø INPUT "  NEW PHONE NUMBER";P$
1Ø9Ø LSET LAST$=L$
11ØØ LSET FIRST$=F$
111Ø LSET ADDRESS$=A$
112Ø LSET CITY$=C$
113Ø LSET STATE$=S$
114Ø LSET ZIP$=Z$
115Ø LSET PHONE$=P$
116Ø PUT #1, R
117Ø NEXT R
118Ø RETURN
119Ø OPEN "D", #1, "MAIL/DAT", 69
12ØØ FIELD #1, 15 AS LAST$, 1Ø AS FIRST$, 15 AS
     ADDRESS$, 1Ø AS CITY$, 2AS STATE$, 5AS ZIP$,
     12AS PHONE$
121Ø OPEN "D", #2, "TEMP/FIL", 69
122Ø FIELD #2, 15 AS TLAST$, 1ØAS TFIRST$, 15AS
     TADDRESS$, 1ØAS TCITY$, 2AS TSTATE$, 5AS TZIP$,
     12AS TPHONE$
123Ø FOR R=1 TO LOF(1)
124Ø GET #1, R
125Ø CLS
126Ø PRINT @ 68, "RECORD#";R
127Ø PRINT @ 1ØØ, LAST$FIRST$
128Ø PRINT @ 132, ADDRESS$
129Ø PRINT @164, CITY$", "STATE$" "ZIP$
13ØØ PRINT @ 228, "PHONE NUMBER:";PHONE$
131Ø PRINT @ 26Ø, "DELETE THIS RECORD? (Y/N)"
132Ø AN$= INKEY$
133Ø IF AN$="Y" THEN 145Ø
134Ø IF AN$= "N" THEN 136Ø
135Ø GOTO 132Ø
136Ø LSET TLAST$=LAST$
137Ø LSET TFIRST$=FIRST$
138Ø LSET TADDRESS$=ADDRESS$
139Ø LSET TCITY$=CITY$
14ØØ LSET TSTATE$=STATE$
141Ø LSET TZIP$=ZIP$
142Ø LSET TPHONE$=PHONE$
143Ø J=J+1
144Ø PUT #2, J
145Ø NEXT R
146Ø CLOSE
147Ø KILL "MAIL/DAT"
148Ø RENAME "TEMP/FIL" TO "MAIL/DAT"
149Ø RETURN
15ØØ END
151Ø CLS
152Ø PRINT @261, "PROGRAM IS CONCLUDED"
153Ø END
2ØØØ PRINT @324, "EDIT FIRST NAME? (Y/N)"
2Ø1Ø AN$=INKEY$
2Ø2Ø IF AN$="Y" THEN 2Ø5Ø
2Ø3Ø IF AN$="N" THEN F$=FIRST$
     : GOTO 2Ø6Ø
2Ø4Ø GOTO 2ØØØ
2Ø5Ø INPUT "NEW FIRST NAME";F$
2Ø6Ø RETURN
```

ADD

```
1ØØ 'THIS PROGRAM NAME IS ?ENV/BAS
11Ø CLEAR 4ØØØ
12Ø OPEN "D", #1, "MAIL/DAT", 69
13Ø FIELD #1,  15 AS LAST$,  1Ø AS FIRST$,  15 AS
     ADDRESS$, 1Ø AS CITY$,  2 AS STATE$,  5 AS ZIP$,
     12 AS PHONE$
14Ø FOR A= 1 TO LOF(1)
15Ø CLS
16Ø GET #1, A
17Ø PRINT @ 64
18Ø PRINT FIRST$ + LAST$
19Ø PRINT ADDRESS$
2ØØ PRINT CITY$ "," STATE$
21Ø PRINT ZIP$
215 PRINT @232, "PRINT THIS ADDRESS?(Y/N)"
216 Q$=INKEY$
217 IF Q$="Y" GOSUB 4ØØ
218 IF Q$="N" GOTO 22Ø
219 GOTO 215
22Ø IF A=LOF(1) THEN 31Ø
23Ø PRINT
24Ø PRINT "DO YOU WANT ANOTHER NAME? (Y/N)"
25Ø AN$=INKEY$
26Ø IF AN$="Y"GOTO 29Ø
27Ø IF AN$="N" THEN CLOSE#1
     :  GOTO 33Ø
28Ø GOTO 25Ø
29Ø NEXT A
3ØØ GOTO 15Ø
31Ø PRINT @324, "END OF FILE"
32Ø CLOSE #1
33Ø END
4ØØ PRINT
41Ø PRINT #-2, FIRST$LAST$
42Ø PRINT #-2, ADDRESS$
43Ø PRINT #-2, CITY$STATE$
44Ø PRINT #-2, ZIP$
45Ø RETURN
```

APPLICAT

```
1ØØØ CLS
1Ø1Ø 'THIS PROGRAM IS TO COMPUTE  THE  CALIBRATION OF
     ANY SPRAYER'
1Ø2Ø 'THE REQUIRED DATA IS SELF-EXPLANATORY'
1Ø3Ø PRINT "ENTER THE WIDTH OF PATTERN OF   YOUR
     SPRAYER IN FEET."
1Ø4Ø INPUT A
1Ø5Ø PRINT "ENTER THE GROUND SPEED THAT YOU INTEND TO
     USE IN MPH."
1Ø6Ø INPUT B
1Ø7Ø PRINT "ENTER THE AMOUNT OF HERBICIDE OR
     PESTICIDE YOU INTEND TO APPLY   PER ACRE IN
     PINTS OR POUNDS."
1Ø8Ø INPUT C
1Ø9Ø PRINT "ENTER THE NUMBER OF GALLONS PER ACRE OF
     WATER YOU WISH TO APPLY."
11ØØ INPUT D
111Ø PRINT "HOW LARGE IS YOUR SPRAYER TANK  IN
     GALLONS?"
112Ø INPUT E
113Ø PRINT "HOW MANY NOZZLES DOES YOUR      SPRAYER
     HAVE?"
114Ø INPUT F
115Ø CLS
116Ø PRINT "WIDTH-  "A"FT"
117Ø PRINT "SPEED-  "B"MPH"
118Ø PRINT "PRODUCT-"C"PTS OR LB/ACRE"
119Ø PRINT "WATER-  "D"GAL/ACRE"
12ØØ PRINT "TANK-   "E"TOTAL GALS"
```

```
121Ø PRINT "NOZZLES-"F"USED"
122Ø FOR Z=1 TO 1ØØØ
     : NEXT Z
123Ø AB=(A*528Ø*B)/4356Ø
124Ø A$="###.##-ACRES/HOUR COVERAGE"
125Ø PRINT USING A$;AB
126Ø PRINT "MIX"C*E"PTS OR LBS. OF PRODUCT
     IN"E"GALLONS."
127Ø E$="ONE TANK COVERS ###.## ACRES"
128Ø PRINT USING E$;E/D
129Ø B$="#####.## GALS/HR-APPL RATE"
13ØØ PRINT USING B$;AB*D
131Ø FOR ZZ=1 TO 2ØØØ
     :  NEXT ZZ
132Ø C$="PRESSURE TIP COMBINATION NEEDED
     : ###.## GALS/HOUR"
133Ø PRINT USING C$;(AB*D)/F
134Ø D$="###.## GALS/MIN."
135Ø PRINT USING D$;((AB*D)/F)/6Ø
136Ø PRINT "DO YOU WANT ANOTHER RATE?"
     : INPUT Y$
137Ø IF Y$="YES" THEN 1ØØØ
     : IF "NO" THEN END
138Ø PRINT "THIS CONCLUDES THIS PROGRAM      THANK
     YOU!!!!!"
```

APLICAT

```
1ØØØ 'REM  FEED PAPER TO 132 COLUMNS
1Ø1Ø 'REM  THIS PROGRAM PRINTS OUT RATES OF
1Ø2Ø 'REM  SPRAYERS TO FACILITATE UNUSUAL
1Ø3Ø 'REM  RATES AND APPLICATIONS IN GAL/MN
1Ø4Ø PRINT #-2, "THE BELOW CHART GIVES THE RATE AND
     NOZZLE AND PRESSURE COMBINATION NEEDED TO SPRAY
     AT THIS RATE."
1Ø5Ø PRINT #-2, "REMEMBER, THE RATE IN GALS/MIN IS
     FOR ONE NOZZLE ONLY"
1Ø6Ø PRINT #-2, "YOU MUST FIND A NOZZLE PRESSURE
     COMBINATION FOR THIS OUTPUT IN GALLONS PER
     MINUTE"
1Ø7Ø PRINT #-2, ""
1Ø8Ø PRINT #-2, ""
1Ø9Ø CLS
11ØØ 'THIS PROGRAM IS TO COMPUTE  THE  CALIBRATION OF
     ANY SPRAYER'
111Ø 'THE REQUIRED DATA IS SELF-EXPLANATORY'
112Ø PRINT "ENTER THE WIDTH OF PATTERN OF   YOUR
     SPRAYER IN FEET."
113Ø INPUT A
114Ø PRINT "ENTER THE GROUND SPEED THAT YOU INTEND TO
     USE IN MPH."
115Ø INPUT B
116Ø PRINT "ENTER THE AMOUNT OF HERBICIDE OR
     PESTICIDE YOU INTEND TO APPLY   PER ACRE IN
     PINTS OR POUNDS."
117Ø INPUT C
118Ø PRINT "ENTER THE NUMBER OF GALLONS PER ACRE OF
     WATER YOU WISH TO APPLY."
119Ø INPUT D
12ØØ PRINT "HOW LARGE IS YOUR SPRAYER TANK  IN
     GALLONS?"
121Ø INPUT E
122Ø PRINT "HOW MANY NOZZLES DOES YOUR      SPRAYER
     HAVE?"
123Ø INPUT F
124Ø CLS
125Ø PRINT "WIDTH-  "A"FT"
     : PRINT #-2, "SWATH WIDTH-"A"FEET"
126Ø PRINT "SPEED-  "B"MPH"
     : PRINT #-2, "DELIVERY SPEED-"B"MPH"
127Ø PRINT "PRODUCT-"C"PTS OR LB/ACRE"
128Ø PRINT "WATER-  "D"GAL/ACRE"
     : PRINT #-2, "WATER RATE-"D"GAL/ACRE"
129Ø PRINT "TANK-   "E"TOTAL GALS"
     : PRINT #-2, "TANK SIZE-"E"TOTAL GALS"
13ØØ PRINT "NOZZLES-"F"USED"
     : PRINT #-2, "NUMBER TOTAL NOZZLES-"F"USED"
131Ø FOR Z=1 TO 1ØØØ
     : NEXT Z
132Ø AB=(A*528Ø*B)/4356Ø
133Ø A$="###.##-ACRES/HOUR COVERAGE"
134Ø PRINT USING A$;AB
     : PRINT #-2, USINGA$;AB
135Ø PRINT "MIX"C*E"PTS OR LBS. OF PRODUCT
     IN"E"GALLONS."
136Ø E$="ONE TANK COVERS ###.## ACRES"
137Ø PRINT USING E$;E/D
     : PRINT #-2, USING E$;E/D
138Ø B$="#####.## GALS/HR-APPL RATE"
139Ø PRINT USING B$;AB*D
     : PRINT #-2, USINGB$;AB*D
14ØØ FOR ZZ=1 TO 2ØØØ
     :  NEXT ZZ
141Ø C$="PRESSURE TIP COMBINATION NEEDED
     : ###.## GALS/HOUR"
142Ø PRINT USING C$;(AB*D)/F
     : PRINT #-2, USINGC$;(AB*D)/F
143Ø D$="###.## GALS/MIN."
144Ø PRINT USING D$;((AB*D)/F)/6Ø
     : PRINT #-2, USING D$;((AB*D)/F)/6Ø
145Ø PRINT "DO YOU WANT ANOTHER RATE?"
     : INPUT Y$
146Ø IF Y$="YES" THEN 1Ø9Ø
     : IF "NO" THEN END
147Ø PRINT "THIS CONCLUDES THIS PROGRAM      THANK
     YOU!!!!!"
```

RATE CHART

```
1ØØØ 'THIS PROGRAM PRINTS A CHART OF
1Ø1Ø 'SPRAYER OUTPUT IN GALS/MIN. FOR THE TOTAL
     NOZZLES OF A SPRAYER
1Ø2Ø 'YOU MAY CHANGE THE MEAN AVERAGE GROUND SPEED
     FROM 5 MPH IF YOU DESIRE
1Ø3Ø 'THE SELECTED MEAN SPEED WILL PRINT RATES FOR 9
     RATES ON EACH SIDE OF THE MEAN
1Ø4Ø CLS
1Ø5Ø PRINT "ENTER THE WIDTH OF SWATH FOR    CHART YOU
     DESIRE"
1Ø6Ø INPUT "SWATH IN FEET";W
1Ø7Ø Y=1
     : PRINT ""
1Ø8Ø PRINT "THE PROGRAM PRINTS GALS PER ACRE FROM .5
     TO 1Ø GALS/ACRE"
1Ø9Ø PRINT "IF YOU WISH TO CHANGE THIS INPUT THE
     MULTIPLIER OF THE STD RATE  (EXAMPLE, INPUT OF
     1Ø YIELDS A  CHART WITH RATES OF 5 TO 1ØØGALS"
11ØØ PRINT "IF THE STD RATE IS OKAY ENTER 1"
111Ø INPUT Y
112Ø GA=(W/1ØØ*.2Ø)*Y
113Ø S=1 'GROUND SPEED VAR.
114Ø PRINT "DO YOU WISH TO CHANGE THE GROUND SPEED
     MEAN FROM 5 MPH? (Y/N)"
115Ø A$=INKEY$
116Ø IF A$="N" GOTO 125Ø
117Ø IF A$="Y" GOTO 119Ø
118Ø GOTO 115Ø
119Ø PRINT "WHAT MEAN SPEED DO WANT IN THE CENTER OF
     THE GROUND SPEEDS?"
12ØØ PRINT ""
121Ø PRINT "ENTER THE MEAN IN MPH"
122Ø INPUT B
123Ø S=B/5
124Ø PRINTS
125Ø PRINT #-2, "                        SPRAYER VOLUME
     OUTPUT REQUIRED AT  "W" FEET SWATH WIDTH
     COVERAGE"
126Ø
     LL$="-----------------------------------------
     -----------------------------------------
     --------------------"
127Ø PRINT #-2, "                        OUTPUTS ARE IN
     GALLONS PER MINUTE FOR ALL NOZZLES USED
     TOTALLED"
128Ø PRINT #-2, LL$
129Ø PRINT #-2, "GROUND     GALLONS PER ACRE"
```

(Continued page 28)

Writer for the Model II

Robert M. Stoltz
RR #1 Box 43 B
Edwardsport, IN 47528

After seeing the many new articles on "easy to use" text editors, I thought someone might be interested in my own version that will save the text to disk for later printing or revision. This program is used in our business daily for writing correspondence letters and creating any type of business form where we need several copies. The only disadvantage to the program is that it will only let you revise a whole line at a time instead of character by character.

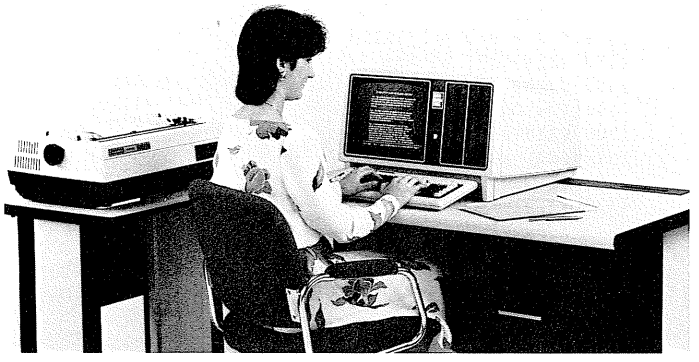
The program was saved to disk using the name WRITER. Then I created a DO file at TRSDOS named LETTER so at power-up it could be loaded by a simple command - DO LETTER. The DO file was created by BASIC WRITER -F:1. If you do not want to use the DO files, type in BASIC WRITER -F:1 at power-up.

The text is saved to disk using any name that you choose (eight characters max) and automatically adds the extension "/TXT" behind the name of the file. The program was written that way so as when looking at the DIRectory of the diskette I could tell programs from actual text saved.

```
1Ø REMARK  **   FOR EASY START-UP  CREATE A DO FILE
     FROM DOS                **
2Ø REMARK  **   OR FROM DOS  TYPE   BASIC -F:1  TO BE
     ABLE TO OPEN FILES    **
3Ø REMARK  ** ' THE PROGRAM WAS SAVED USING THE NAME
     WRITER              **
4Ø REMARK  **   THEN CREATED A  DO  FILE  AND NAMED
     IT     LETTER          **
5Ø REMARK  **   AFTER POWER - UP  ALL I HAVE TO DO IS
     TYPE --  DO LETTER    **
6Ø REMARK  **   C IS FOR LINE TEXT COUNTER    L IS
     FOR LEFT MARGIN         **
7Ø REMARK  **   M IS FOR PAGE LENGTH          R IS
     FOR RIGHT MARGIN        **
8Ø REMARK  **   P IS FOR EDIT TEXT LINES      Z IS
     ACTUAL TEXT LENGTH      **
9Ø REMARK  **   Q IS JUST COUNTER LOOP        U IS
     TIMER LOOP              **
1ØØ REMARK  **   WRITTEN BY :  ROBERT M. STOLTZ
     9th  VERSION           **
11Ø REMARK  **                  R.R. # 1 BOX 43 B
     **
12Ø REMARK  **                  EDWARDSPORT , INDIANA
     47528                  **
13Ø REMARK  **   SEPTEMBER 16, 1982
     **
14Ø CLS
15Ø REMARK  **   CLEAR & DIM L$ MORE IF TEXT IS
     GREATER THAN 12Ø LINES       **
16Ø CLEAR (1ØØØØ)
17Ø DIM L$(14Ø)
18Ø DEFINT C, L, M, P, Q, R, Z
19Ø REMARK  **   WRITTEN FOR TRS-8Ø MODEL II - SHOULD
     WORK ON MODEL I & III  **
2ØØ REMARK  **   K IS MAX. WIDTH OF LINE       CHANGE
     TO 63 FOR MOD I & III  **
21Ø REMARK  **   REMEMBER LINE COUNT STARTS  WITH   Ø
     NOT  1               **
22Ø K = 79
23Ø CLS
24Ø ON ERROR GOTO 19ØØ
25Ø PRINT@25, "  **    MENU   **"
26Ø PRINT
     : PRINT
     : PRINT
27Ø PRINT "           < A >      ADD to page
     already in memory"
28Ø PRINT "           < C >      CREATE new page"
29Ø PRINT "           < E >      EDIT - LOOK AT
     page in memory"
3ØØ PRINT "           < L >      LOAD page from
     disk"
31Ø PRINT "           < H >      HARDCOPY of page"
32Ø PRINT "           < S >      SAVE page to
     disk"
33Ø PRINT "           < X >      EXIT program"
34Ø PRINT
     : PRINT
     : PRINT
35Ø PRINT "NOTE :SAVE  PAGE if you have made any
     changes since last saved."
36Ø PRINT
     : PRINT
     : PRINT
     : PRINT "ENTER MENU CHOICE"
37Ø M$ = INKEY$
38Ø IF M$ = "" THEN 37Ø
39Ø IF M$ = "A" OR M$ = "a" THEN 64Ø
4ØØ IF M$ = "C" OR M$ = "c" THEN 47Ø
41Ø IF M$ = "E" OR M$ = "e" THEN 76Ø
42Ø IF M$ = "L" OR M$ = "l" THEN 158Ø
43Ø IF M$ = "H" OR M$ = "h" THEN 1Ø4Ø
44Ø IF M$ = "S" OR M$ = "s" THEN 136Ø
45Ø IF M$ = "X" OR M$ = "x" THEN 133Ø
46Ø GOTO 37Ø
47Ø IF C > Ø THEN 14Ø ELSE 48Ø
48Ø CLS
     : PRINT@25,"  **  START-UP ROUTINE  **"
     : PRINT
     : PRINT
     : PRINT
     : PRINT
49Ø PRINT "When entering TEXT the"
     : PRINT
     : PRINT "exit CODE is XXX OR xxx"
     : PRINT
5ØØ PRINT "this CODE can be changed within the
     program to suit your needs"
     : PRINT
51Ø REMARK  **   TO INCREASE OR DECREASE AMOUNT OF
     TEXT TO SEE AT ONE TIME  **
52Ø P = Ø
53Ø INPUT "enter LEFT  MARGIN "; L
     : L = L - 1
54Ø IF L < Ø THEN L = Ø
55Ø PRINT
     : PRINT "RIGHT margin must be = to or less than
     ";K + L + 2
     : PRINT
56Ø INPUT "enter RIGHT MARGIN "; R
57Ø IF R < L THEN 56Ø ELSE 58Ø
58Ø IF R > (K + L + 2) THEN 61Ø ELSE 59Ø
59Ø R = R - 1
     : M = 6Ø
     : GOTO 62Ø
6ØØ REMARK  ** THIS IS MAX. PAGE LENGTH   CHANGER AS
     NEEDED  **
61Ø R = Ø
     : PRINT "RIGHT MARGIN must be = to or less than
     "; K + L + 2
     : PRINT
     : GOTO 56Ø
62Ø Z = R - L
63Ø CLS
     : GOTO 65Ø
64Ø IF C = Ø THEN 23Ø ELSE 65Ø
65Ø C = C + 1
66Ø IF C => M THEN 176Ø
67Ø REM **  !  SHOWS TABS &  END OF LINE STOP POINTS
     . COULD BE DELETED   **
68Ø PRINT TAB(4) "!"; TAB(9); "!"; TAB(14); "!";
     TAB(19); "!"; TAB(24); "!"; TAB(Z-1) "!"
```

```
69Ø LINE INPUT L$(C)
7ØØ REMARK  **   NEXT LINE CONTAINS   EXIT  CODE
     CHANGE TO FIT YOUR NEEDS  **
71Ø IF L$(C) = "XXX" OR L$(C) = "xxx" THEN C = C - 1
     :  GOTO 23Ø
72Ø IF LEN(L$(C)) > Z THEN 73Ø ELSE 74Ø
73Ø L$(C) = ""
     : PRINT "TOO LONG try again "
     : GOTO 68Ø
74Ø IF LEN(L$(C)) = Ø THEN L$(C) = " "
75Ø GOTO 65Ø
76Ø IF C = Ø THEN 23Ø ELSE 77Ø
77Ø CLS
     : PRINT@25, "  **    EDIT ROUTINE    **"
     : PRINT
     : PRINT
     : PRINT
78Ø Q = Ø
     : P = Ø
79Ø Q = Q + 1
8ØØ P = P + 1
81Ø PRINT Q; TAB(4); L$(Q)
82Ø IF Q = C THEN 88Ø ELSE 86Ø
83Ø REMARK  **   LOOKS AT 15 LINES OF TEXT AT ONE
     TIME  CHANGE VALUE OF P   **
84Ø REMARK  **   TO INCREASE OR DECREASE AMOUNT OF
     TEXT TO SEE AT ONE TIME  **
85Ø REMARK  **    CHANGE THE IF P ='S STATEMENT IN
     NEXT LINE    **
86Ø IF P = 15 THEN 88Ø
87Ø GOTO 79Ø
88Ø P = Ø
89Ø PRINT
     : PRINT
9ØØ YY = Ø
     : INPUT"enter LINE # to CORRECT on this list or
     ENTER to continue";YY
91Ø IF YY > C THEN 9ØØ ELSE 92Ø
92Ø IF YY <= Ø THEN 93Ø ELSE 96Ø
93Ø IF Q = C AND P = Ø THEN 23Ø ELSE 94Ø
94Ø CLS
95Ø GOTO 87Ø
96Ø PRINT
     : PRINT
97Ø EE = Ø
     : EE = YY
     : YY = Ø
98Ø PRINT
     : PRINT "OLD line "
     : PRINT L$(EE)
     : L$(EE) = ""
99Ø REMARK  **  THE NEXT LINE IN THIS PROGRAM COULD
     BE LEFT OUT   **
1ØØØ PRINT
     : PRINT
     : PRINT TAB(Z-1) "!"
1Ø1Ø PRINT
     : PRINT "NEW line "
     : LINE INPUT L$(EE)
1Ø2Ø IF LEN(L$(EE)) > Z THEN PRINT "TOO LONG try
     again"
     : L$(EE) = ""
     : GOTO 1Ø1Ø
1Ø3Ø GOTO 77Ø
1Ø4Ø IF C = Ø THEN 23Ø ELSE 1Ø5Ø
1Ø5Ø CLS
     : PRINT@25, "  **   HARDCOPY ROUTINE   **"
1Ø6Ø PRINT
     : PRINT
     : PRINT
     : PRINT "DO YOU NEED TOP OF FORMS  ( Y or hit
     any key )"
1Ø7Ø Y$ = INKEY$
1Ø8Ø IF Y$ = "" THEN 1Ø7Ø ELSE 1Ø9Ø
1Ø9Ø IF Y$ = "Y" OR Y$ = "y" THEN 112Ø ELSE 113Ø
11ØØ REMARK  **  NEXT LINE IS FOR PRINTER TOP OF
     FORMS COMMAND     **
111Ø REMARK  **  CHANGE  CHR $ ( XX ) IF NOT SAME AS
     YOUR PRINTER  **
112Ø LPRINT CHR$(12)
113Ø CLS
114Ø CP = Ø
     : PRINT
     : PRINT "ENTER  Ø   TO RETURN TO MENU "
     : PRINT
115Ø PRINT
     : INPUT "HOW MANY COPIES DO YOU NEED"; CP
116Ø IF CP = Ø THEN 23Ø ELSE 117Ø
117Ø PRINT
     : PRINT "Press any key to continue "
     : PRINT
     : PRINT
118Ø Y$ = INKEY$
119Ø IF Y$ = "" THEN 118Ø ELSE 12ØØ
12ØØ FOR ZZ = 1 TO CP
121Ø FOR E = 1 TO C
122Ø LPRINT TAB(L) L$(E)
123Ø NEXT E
124Ø PRINT
     : PRINT
125Ø REMARK  **  NEXT LINE IS FOR PRINTER TOP OF
     FORMS COMMAND     **
126Ø REMARK  **  CHANGE  CHR $ ( XX ) IF NOT SAME AS
     YOUR PRINTER  **
127Ø LPRINT CHR$(12)
128Ø E = Ø
129Ø NEXT ZZ
13ØØ E = Ø
131Ø GOTO 23Ø
132Ø REMARK  **            EXIT ROUTINE        **
133Ø CLOSE
134Ø PRINT
     : PRINT
     : PRINT "type GOTO 4Ø if you want to continue
     same letter."
135Ø PRINT
     : PRINT
     : END
136Ø IF C = Ø THEN 23Ø ELSE 137Ø
137Ø CLS
     : PRINT
     : PRINT
     : PRINT@25, "  **     DISK SAVE ROUTINE    **"
138Ø PRINT
     : PRINT
139Ø PRINT "hit enter to return to MENU "
14ØØ PRINT
     : PRINT "NAME MUST BE BETWEEN    1 AND 8
     CHARACTERS LONG ONLY"
     : PRINT
141Ø NE$ = ""
     : INPUT"enter NAME you want to SAVE it under ";
     NE$
142Ø IF LEN(NE$) = Ø THEN 23Ø
143Ø IF LEN(NE$) > 8 THEN 138Ø ELSE 144Ø
144Ø PRINT
     : PRINT"       SAVING to  DISK at this time
     using the NAME     "; NE$
145Ø PRINT
     : PRINT
146Ø NZ$ = NE$ + "/TXT"
147Ø OPEN "O", 1, NZ$
148Ø PRINT#1, C, L, R, Z, M
149Ø FOR RR = 1 TO C
15ØØ PRINT#1, L$(RR)
151Ø NEXT RR
152Ø CLOSE
153Ø PRINT
     : PRINT
154Ø PRINT
     : PRINT
     : PRINT "hit any key to continue "
155Ø Y$ = INKEY$
156Ø IF Y$ = "" THEN 155Ø ELSE 157Ø
```

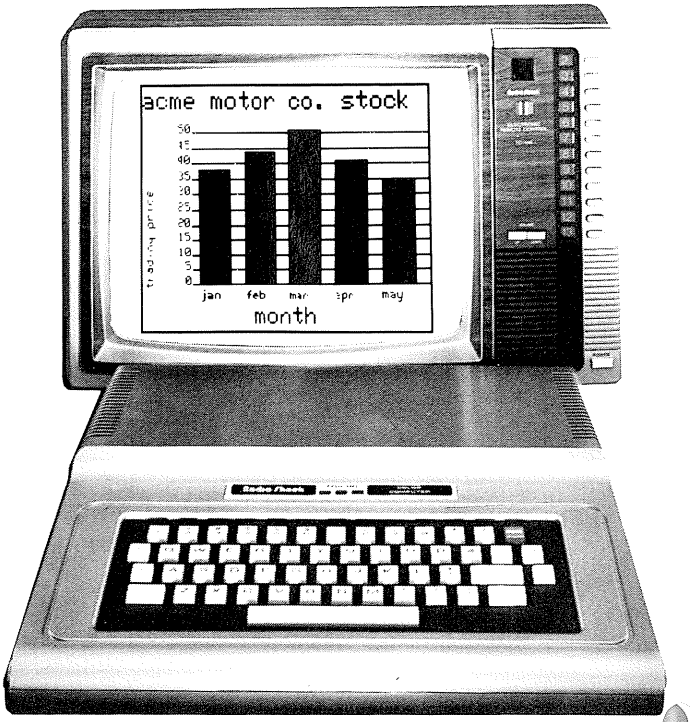
```
157Ø Y$ = ""
     : GOTO 23Ø
158Ø IF C > Ø THEN 14Ø ELSE 159Ø
159Ø CLS
     : PRINT@25, "  **    LOAD ROUTINE    **"
     : PRINT
     : PRINT
16ØØ PRINT "press ENTER to return to MENU "
     : PRINT
     : PRINT
161Ø NE$ = ""
     : INPUT "enter NAME of FILE "; NE$
162Ø IF LEN(NE$) = Ø THEN 23Ø ELSE 163Ø
163Ø IF LEN(NE$) > 8 THEN 161Ø ELSE 164Ø
164Ø NZ$ = NE$ + "/TXT"
165Ø OPEN "I", 1, NZ$
166Ø INPUT#1, C, L, R, Z, M
167Ø PRINT
     : PRINT
     : PRINT "left margin set at "; L + 1
168Ø PRINT "right margin set at "; R + 1
169Ø PRINT "page length set at "; M
17ØØ FOR RR = 1 TO C
171Ø IF EOF(1) THEN 23Ø
172Ø LINE INPUT#1, L$(RR)
173Ø NEXT RR
174Ø CLOSE
175Ø GOTO 23Ø
176Ø CLS
177Ø PRINT "YOU have gone to the LIMIT of your PAGE
     LENGTH "
178Ø PRINT
     : PRINT"Do you want to increase the length  ( Y
     or press any key )
179Ø Y$ = INKEY$
18ØØ IF Y$ = "" THEN 179Ø ELSE 181Ø
181Ø IF Y$ = "Y" OR Y$ = "y" THEN 182Ø ELSE 188Ø
182Ø PRINT
     : PRINT
     : MM = Ø
     : INPUT "enter new Page LENGTH "; MM
     : PRINT
     : PRINT
183Ø IF MM < M THEN 182Ø ELSE 184Ø
184Ø M = MM
     : MM = Ø
185Ø PRINT
     : PRINT
     : PRINT
     : PRINT
     " Press any key to continue"
186Ø Y$ = INKEY$
187Ø IF Y$ = "" THEN 186Ø ELSE 188Ø
188Ø GOTO 23Ø
189Ø REMARK  **       ERROR ROUTINE       **
19ØØ PRINT
     : PRINT "AN ERROR HAS OCCURRED   *****    PLEASE
     STAND-BY   *********"
191Ø CLOSE
192Ø FOR U = 1 TO 2ØØØ
     : NEXT U
193Ø GOTO 23Ø
```



CoCo Business

(From page 25)

```
13ØØ PRINT #-2, "SPEED   ";
     : PRINT #-2, USING"##.##   ";.5*Y, 1*Y, 1.5*Y,
     2*Y, 2.5*Y, 3*Y, 4*Y, 5*Y, 6*Y, 7*Y, 8*Y, 9*Y,
     1Ø*Y
131Ø PRINT #-2, LL$
132Ø PRINT #-2, "IN MPH       OUTPUT IN GALLONS PER
     MINUTE""
133Ø FOR N=1 TO 19
134Ø AO=.5*GA*S
     : BO=GA*S
135Ø CO=1.5*GA*S
     : DO=2.Ø*GA*S
136Ø EO=2.5*GA*S
     : FO=3*GA*S
137Ø OO=4*GA*S
     : HO=5*GA*S
138Ø IO=6*GA*S
     : JO=7*GA*S
139Ø KO=8*GA*S
     : LO=9*GA*S
14ØØ MO=1Ø*GA*S
141Ø PRINT Y
142Ø PRINT GA
143Ø R$="##.##   "
144Ø PRINT #-2, USING R$;S, AO, BO, CO, DO, EO, FO,
     OO, HO, IO, JO, KO, LO, MO
145Ø S=S+.5
146Ø IF N=19THEN GOTO 147Ø ELSE NEXT N
147Ø PRINT #-2, ""
148Ø PRINT #-2, "REFER TO CHARTS FOR THE TIPS YOU USE
     TO DETERMINE PRESSURE NEEDED FOR YOUR RATE"
149Ø CLS
     : PRINT "WOULD YOU LIKE TO PRINT A CHART FOR A
     DIFFERENT SWATH?(Y/N)"
15ØØ INPUT M$
151Ø IF M$="Y" GOTO 1Ø4Ø
152Ø CLS
     : PRINT "PROGRAM IS CONCLUDED"
```



Restructuring Profile Data Bases

The small Computer Company
P.O. Box 2910
Fort Worth, TX 76113-2910
By Ivan Sygoda, Director, Pentacle

Do you ever wish that you'd set up your Profile data base differently? Perhaps you've found you neglected to include an important sort field when you originally defined your fields—area code, for example. Or maybe you've discovered you allowed too little room for the address field. Or maybe you defined your fields before learning the whole repertory of Profile features, and now you want to take advantage of associated fields, for instance, or the date-of-last-update field.

More likely, however, you and your work have simply changed over the years. Your business has grown. Your needs are different. Now you have to deal with new types of clients, new information, new products or new services. To do this, you need new or altered fields.

A BIG INVESTMENT

On the other hand, you've already made a big investment in time and effort building up your data base. Somebody (probably you) spent dozens of hours keying in names, addresses, and so forth. As the Profile manual says, you can lose a lot of data if you change fields after the fact. All those ASCII characters—the building blocks of your data base—may still be there on your disks, but if you juggle field definitions recklessly, Profile will no longer be able to find them when needed.

BASIC TO THE RESCUE

As you know by now, I never bring up a problem unless there's a solution. If you're willing to write a program in BASIC, you can restructure your Profile data base any way you want, without losing any of the data you've already typed in. An extension of the techniques discussed in last month's column is all that's involved.

As an example, I'll show you how I restructured my SPONSTAT data base, which I use at Pentacle to keep track of the presenters who are interested in hiring my dance companies. I needed to restructure SPONSTAT because I did something that you might do some day as your own business grows: I bought a Model 12. (Actually, I went to Radio Shack to buy another Model III, but I came home with a Model 12 through some combination of reason and lust.)

At Pentacle during the next year, we plan to upgrade our Model 12 to a 16-bit Model 16, add the hard disk and TRS-XENIX and, finally, two DT-1s for multi-user capability. Here's the important point—my Profile data bases can follow me all the way up: from Model III to Model 12 (in Model II mode) to Model 16 to TRS-XENIX to multi-user! We'll take it one step at a time both at Pentacle and in these articles. In this article, step one: restructuring the Profile III Plus data base on the Model III.

THE KEY TO SUCCESS

Profile for the Model II is actually a bit rougher in some respects than the Model III version because the former was written first. The most important difference is that in Model II Profile (Profile II and Profile Plus), the key segment (userfile/KEY) is always exactly 85 bytes long, and three such key records are "blocked" into each 256-byte sector on the disk. In Profile III Plus, the key segment can be any length up to 255 bytes. My own SPONSTAT/KEY segment has 200 bytes.

Table 1 shows the complete list of fields in my Profile III Plus data base plus the buffer names I assigned to them in the BASIC program that follows. For simplicity's sake, I call Table 1 the "source." Table 2 shows the "destination" data base—the way I want my restructured file to look when transferred to Profile II Plus on my Model 12. I had to do some juggling to reduce my key segment from 200 to 85 bytes. Half the job was done when I transferred my associated fields into segment 2. (These fields keep track of the booking history of each presenter and of his or her current interests.)

While I was at it, I made some other changes. I combined associated field group A (the three-character code for companies previously booked by the presenter) and group B (the two-digit year of the booking) into a new five-character group A. As explained in my April 1983 article (page 17), combining these groups adds flexibility while conserving fields, thanks to Profile's "wild card" (=) feature. I also lengthened group B, (C in the destination data base) so as to indicate in which year the presenter wants one of my companies. (The larger presenters plan two or more years in advance. I hadn't been able to cope with this before.)

Plus, I increased the number of fields in each group by adding fields to describe a second performance facility ("F2") and adding a one-byte "Y/N" field to indicate whether the presenter was active in the summer.

HAVE YOUR CAKE AND EAT IT, TOO

There's a reason I didn't worry too much about transferring fields out of my key segment into segment 2: I have Profile Prosort for the Model II (26-4558). This valuable enhancement allows me to sort on fields in any segment, store up to five such indexes for printing, and even make indexes of indexes, which means I can sort across segments. Without Prosort, I would have had to agonize a great deal before abandoning key fields.

Also, except for the 85-byte limitation on the length of the key field, I could afford to be generous regarding the length of

Table 1: SPONSTAT ("SOURCE") Fields

Field No.	Heading	Length	Buffer Name	Length
(Segment 1: SPONSTAT/KEY)				
1	Zip	10		
2	Inst	25		
3	Last name	15		
4	1st name	12	TA$	62
5	Title	10		
6	Dept	20	TB$	30
7	City	20	TC$	20
8	State	2		
9	A/C	3	TD$	5
10	A)Bkd	3	A1$	3
11	A)	3	A2$	3
12	A)	3	A3$	3
13	A)	3	A4$	3
14	A)	3	A5$	3
15	A)	3	A6$	3
16	B)Yr	2		
17	B)	2	B2$	2
18	B)	2	B3$	2
19	B)	2	B4$	2
20	B)	2	B5$	2
21	B)	2	B6$	2
22	Last contact	8	TE$	8
23	Mode	8	TF$	8
24	C)Int	3	C1$	3
25	C)	3	C2$	3
26	C)	3	C3$	3
27	C)	3	C4$	3
28	C)	3	C5$	3
29	C)	3	C6$	3
30	C)	3	C7$	3
31	C)	3	C8$	3
32	Last update	8	TG$	8
33	Source	3	TH$	3
34	Marker	2	TI$	2
(Segment 2: SPONSTAT/DAT)				
35	Honorific	4		
36	Address	25	TJ$	9
37	Ofc. phone no.	8	TK$	8
38	Home phone A/C	3		
39	Home phone no	8	TL$	1
40	Facility	20	TM$	0
41	Dim (Width)	2	TN$	2
42	Dim (Depth)	2		
43	Seating cap	4	TP$	6
44	Budget	5	TQ$	5
45	Salutation	18	TR$	8
(Segment 3: SPONSTAT/DA2)				
46	Status	247	TS$	7

Table 2: SPONSTAX ("DESTINATION") Fields

Field No.	Heading	Length
(Segment 1: SPONSTAX/KEY)		
1	Zip code/Country	10
2	Institution/Organization	25
3	Contact last name	15
4	Contact first name	12
5	State code	2
6	Office phone A/C	3
7	Mode of last contact	8
8	Last update	8
9	Marker	2
(Segment 2: SPONSTAX/DAT)		
10	City	20
11	Lst ct dt	8
12	Title	10
13	Dept	20
14	HomeA/C	3
15	Home ph	8
16	Ofc ph	8
17	Srce	3
18	Bdgt	5
19	Sum?	1
20	A)Bkd/Yr	5
21	A)	5
22	A)	5
23	A)	5
24	A)	5
25	A)	5
26	A)	5
27	A)	5
28	A)	5
29	B)Int/Yr	5
30	B)	5
31	B)	5
32	B)	5
33	B)	5
34	B)	5
35	B)	5
36	B)	5
37	F1	8
38	Dim1/D	2
39	Cap1	4
40	F2	8
41	Dim2/D	2
42	Cap2	4
43	Hon	4
44	Addr	25
(Segment 3: SPONSTAX/DA2)		
45	Salutation (Dear)	18
46	Current status (Note)	232
47	Dim 1 (Width)	2
48	Width)	2

my other segments. My two Model 12 double-sided, double-density Thinline drives hold 2.5 megabytes, compared to the half-megabyte capacity of my three Model III drives. I had absolutely no problems FCOPYing Model II Profile, which is delivered on TRSDOS 2.0a, onto my Thinline drives under TRSDOS 4.2.0.

THE HEART OF THE MATTER

Listing 1 shows the BASIC program that restructures my data base. It uses SPONSTAT/IX1, my zip-ordered user index, to GET each record. It displays certain information from the record on the screen, LSETs restructured records into the

segments of another Profile III Plus data base, temporarily called SPONSTAX, and then PUTs these records onto the disk in drive 0. Be sure to initialize BASIC for the appropriate number of variable length files: one for each segment in the source data base, one for the source index file (if you want to use it to order records), and one for each segment in the destination data base.

The restructured data base will be written to drive 0 in batches because the entire data base won't fit on one diskette. However, none of the Profile III Plus creation or runtime programs are needed for the operation. The disk in drive 0 only needs to contain as much space as possible for the restructured file, and TRSDOS, BASIC, and the BASIC restructuring program, which I call SPONSHUF/BAS. My source data segments were already on non-system disks that go in drives 1 and 2. If you have intermixed data segments and formatting programs and if space is a problem, you'll have to COPY your data segments onto a data disk or set of disks.

I decided to restructure about 150 records per batch. Lines 70-80 let me specify the number of records per batch and then compute whether the 166 free granules (grans) on my system disk will hold that many restructured records. The total length of all three segments in my destination data base is 547 bytes. (The number of bytes in a gran is 768.)

Important note: Because I planned to restructure my multi-segment files record by record, I had to CREATE the destination segments on drive 0 in advance. This is required because TRSDOS allocates disk space as needed for each file as it grows. Unless I pre-CREATE, since my program accesses data segments in round-robin style, the various segment files will "leap-frog" over each other on the disk until an Error 30 is generated, "No More Extents." (The limit, by the way, is 13 extents per file on the Model III.)

To CREATE the first destination segment, I typed "CREATE SPONSTAX/KEY (LRL = 85, REC = 150)" at TRSDOS. ("REC = 150" means that I'm reserving space for 150 records.) I followed the same procedure for SPONSTAX/DAT,and the other data files,checking the directory carefully for the logical record length (LRL) of each file.

If my Model III had four disk drives, I could restructure all my records in one operation, rather than in batches. Also I needed to restructure using all segments, since I wanted to transfer data across segments. Otherwise, I could have done it segment by segment.

Lines 60-65 are very important. They let you process the file in several parts and specify with which record to begin (line 70 lets you specify with which record to end). When the prompt, "Begin shuffle at Logical Record no." appears, the program will not accept a number less than "4." As I mentioned last month, the first three records of a Profile index file are reserved for "housekeeping." My file begins with the fourth physical record.

Line 130 sets up and "scroll protects" a header that describes the fields displayed as each record is processed. The line that commands the displaying is line 330. None of this is necessary, but it's nice to have something to look at while the program does its work.

Each file is OPENed for random access and assigned a buffer in lines 140-200. Consult your owner's manual if you are unsure of the purpose or format of these lines.

Each of the above buffers is FIELDed in lines 210-270. I found it much easier to check my programming if I assigned buffer names consecutively (TA$, TB$, TC$, etc.). There was no reason to divide groups of fields which were to remain grouped in the destination data base. For instance, zip, institution and name fields were to remain together, in that order, so I moved them in one block. My source key segment had too many fields to FIELD in one program line (line 210), so the operation carries over into line 215. "TT$" is a "dummy" field that lets the program "skip over" the items already fielded.

Line 240 FIELDs the index file. "KY$" (zip) is the 10-character key field on which the file was sorted. "PR$" is the two-byte hexadecimal representation of the physical record desired. "XX$" is a housekeeping byte of no relevance here. In lines 250-270, "X1$," "X2$" and "X3$" are simply the as yet empty segments of SPONSTAX/KEY, -/DAT and -/DA2.

Line 290 checks for the end of the file and, if it finds it, jumps to an exit routine.

Line 300 GETs the next index record. Line 310 converts the physical record number, stored as a string, into the numeric form needed by BASIC's file handling utilities. Lines 320 and 330-370 do most of the work. Line 320 GETs all three data segments of the appropriate record. Line 340 LSETs into SPONSTAX/KEY the fields I wanted to place in it. Lines 350 and 360 do the same thing for SPONSTAX/DAT and -/DA2. Fields to be shortened can be manipulated using LEFT$, MID$ or RIGHT$ functions. Fields to be lengthened must be "padded" with blanks, either by adding " " for shorter spans, or "STRING$(n,32)" for greater ones. (The ASCII code for a blank space is 32.) The alternate format is "STRING$(n," ")." Line 370 PUTs the newly restructured segments into the files waiting for them on drive 0.

Finally, the alternate exit routines in lines 410 and 420 both CLOSE all open files.

Listing 1.

```
10 'SPONSHUF/BAS- OPEN BASIC FOR 7V FILES
20 CLS
   : CLEAR 5000
   : DEF INT A-Z
   : EF$ = STRING$(26,"=") + "END OF FILE" +
     STRING$(27,"=")
30 PRINT@128, "Shuffle SPONSTAT for transfer to
     Model 12"
   : PRINT STRING$(64,95);
   : PRINT
   : PRINT "Insert data disks in drives 1 and 2."
   : PRINT "Press <ENTER> when ready.";
40 K$ = INKEY$
   : IF K$ = "" GOTO 40
50 IF K$ <> CHR$(13) GOTO 40
60 PRINT
   : PRINT
   : INPUT"Begin shuffle at Logical Record no."; LR
65 IF LR < 4 PRINT "This number must be greater than
     3."
   : GOTO 60
70 INPUT "Number of Records to include in this
     batch"; NR
80 PRINT
   : PRINT "You will need approximately";
     INT(NR*587/768)+1; "grans for this shuffle."
   : PRINT "There are 166 grans free on the transfer
     disk."
   : IF INT(NR*587/768)+1 > 166 PRINT "Select fewer
     records."
   : IF INT(NR*547/768)+1 > 166 GOTO 70
90 PRINT "Press <ENTER> to proceed."
100 K$ = INKEY$
   : IF K$ = "" GOTO 100
110 IF K$ <> CHR$(13) GOTO 100
```

```
130 PRINT@0, "=N= =ZIP= =====INSTITUTION=====
    ========CONTACT NAME=======REC#";
  : POKE 16916,1
  : PRINT@64, CHR$(31);
140 OPEN "R", 1, "SPONSTAT/KEY:1", 200
150 OPEN "R", 2, "SPONSTAT/DAT:1", 100
160 OPEN "R", 3, "SPONSTAT/DA2:2", 247
170 OPEN "R", 4, "SPONSTAT/IX1:2", 13
180 OPEN "R", 5, "SPONSTAX/KEY:0", 85
190 OPEN "R", 6, "SPONSTAX/DAT:0", 248
200 OPEN "R", 7, "SPONSTAX/DA2:0", 254
210 FIELD 1, 62 AS TA$, 30 AS TB$, 20 AS TC$, 5 AS
    TD$, 3 AS A1$, 3 AS A2$, 3 AS A3$, 3 AS A4$, 3
    AS A5$, 3 AS A6$, 2 AS B1$, 2 AS B2$, 2 AS B3$,
    2 AS B4$, 2 AS B5$, 2 AS B6$, 8 AS TE$, 8 AS
    TF$, 3 AS C1$, 3 AS C2$, 3 AS C3$, 3 AS C4$, 3
    AS C5$, 3 AS C6$, 3 AS C7$, 3 AS C8$, 8 AS TG$
215 FIELD 1, 195 AS TT$, 3 AS TH$, 2 AS TI$
220 FIELD 2, 29 AS TJ$, 8 AS TK$, 11 AS TL$, 20 AS
    TM$, 2 AS TN$, 6 AS TP$, 5 AS TQ$, 18 AS TR$
230 FIELD 3, 247 AS TS$
240 FIELD 4, 10 AS KY$, 2 AS PR$, 1 AS XX$
250 FIELD 5, 85 AS X1$
260 FIELD 6, 248 AS X2$
270 FIELD 7, 254 AS X3$
280 FOR N = 1 TO NR
300 GET 4, LR
310 PR = CVI(PR$)
320 GET 1, PR
  : GET 2, PR
  : GET 3, PR
330 PRINT RIGHT$("00"+RIGHT$(STR$(N),
    LEN(STR$(N))-1),3) + " " + LEFT$(KY$,5) + "" +
    MID$(TA$,11,21) + " " + RIGHT$(TA$,27) + " " +
    RIGHT$("   " + STR$(LOC(1)),4);
340 LSET X1$ = TA$ + TD$ + TF$ + TG$ + TI$
350 LSET X2$ = TC$ + TE$ + TB$ + TL$ + TK$ + TH$ +
    TQ$ + "" + A1$ + B1$ + A2$ + B2$ + A3$ + B3$ +
    A4$ + B4$ + A5$ + B5$ + A6$ + B6$ +
    STRING$(15,32) + C1$+"  " + C2$ + " " + C3$ + "
    " + C4$ + "  " + C5$ + "  " + C6$ + "  " + C7$ +
    "  " + C8$ + "" + LEFT$(TM$,18) + TP$ +
    STRING$(24,32) + TJ$
360 LSET X3$ = TR$ + LEFT$(TN$,232) + TN$ + "  "
370 PUT 5
  : PUT 6
  : PUT 7
380 LR = LR + 1
390 IF LR > LOF(4) GOTO 420
400 NEXT N
410 CLOSE
  : PRINT
  : PRINT N-1; "records have been shuffled."
  : PRINT "Begin the next batch with Logical Record
    no. "; LR
  : GOTO 470
420 PRINT EF$;
  : CLOSE
  : PRINT
  : PRINT N; "records have been shuffled."
470 END
```

As always in such operations, you should only work with backups of your data disks. For extra protection, you can write protect the source data diskettes in drives 1 and 2, as they are not written to during restructuring.

WHAT NEXT?

Now that you've restructured your Profile data base, there are many things you can do with it. You can redefine fields, reformat your screens, labels and reports, and use your restructured data as before. Just be sure to rename either your restructured files or your newly defined data base so that both have the same file name. Otherwise Profile won't know what you are doing.

You can also transfer your data to another computer, be it a Model II, III, 4, 12, 16 or 100, either directly by cable, which is how I did it, or via a modem. This is done segment by segment, ASCII file by ASCII file. If you had to restructure in batches, you can APPEND one batch to the end of the previous batch, also file by file.

Things are moving so rapidly at Radio Shack that there are new subjects to write about that didn't even exist a month ago. For instance, I just bought a Model 100, which is an amazing and talented not-so-little computer. Then there's the marvelous hard disk version of Profile III Plus (26-1593), which incorporates the advanced sorting functions available on the II/12 as Prosort. Over the next few issues, we'll discuss these goodies and more, as we upgrade our system at Pentacle.

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

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

PROSORT II USERS

We've found a problem in Prosort for the Model II. The print, label and selector programs, when running off an index, will fail to print records properly when the records selected are located in an extended key segment.

Both the installation diskette and the diskette onto which Prosort has been installed can be easily patched. But be careful—there are two different sets of patches, one for the installation diskette and the other, if you've already installed Prosort, for the Profile Plus system diskette.

Patches for the Prosort installation diskette:

```
PATCH LBLIDX/EFC A=625C, F=20202020202020,
                         C=CD8995C33894
PATCH LBLIDX/EFC A=65E3, F=C33894,
                         C=C34A90
PATCH LBLIDX/EFC A=6647, F=C3,
                         C=C9
PATCH LBLIDX/EFC A=4927, F=33,
                         C=34
PATCH SELIDX/EFC A=284A, F=20202020202020,
                         C=CD7A2AC32929
PATCH SELIDX/EFC A=28C2, F=C32929,
                         C=C34A28
PATCH SELIDX/EFC A=2926, F=C3,
                         C=C9
PATCH SELIDX/EFC A=4CE5, F=33,
                         C=34
PATCH PRTIDX/EFC A=604A, F=20202020202020,
                         C=CDEA94C39694
PATCH PRTIDX/EFC A=63D6, F=C39694,
                         C=C34A90
PATCH PRTIDX/EFC A=6491, F=CDEA94,
                         C=000000
PATCH PRTIDX/EFC A=471D, F=33,
                         C=34
```

Patches for the Profile Plus system diskette on which Prosort is installed:

```
PATCH LABEL/EFC A=625C, F=2Ø2Ø2Ø2Ø2Ø2Ø,
                        C=CD8995C33894
PATCH LABEL/EFC A=65E3, F=C33894,
                        C=C34A9Ø
PATCH LABEL/EFC A=6647, F=C3,
                        C=C9
PATCH LABEL/EFC A=4927, F=33,
                        C=34
PATCH SELECTOR/EFC A=284A, F=2Ø2Ø2Ø2Ø2Ø2Ø,
                           C=CD7A2AC32929
PATCH SELECTOR/EFC A=28C2, F=C32929,
                           C=C34A28
PATCH SELECTOR/EFC A=2926, F=C3,
                           C=C9
PATCH SELECTOR/EFC A=4CE5, F=33,
                           C=34
PATCH PRINT/EFC A=6Ø4A, F=2Ø2Ø2Ø2Ø2Ø2Ø,
                        C=CDEA94C39694
PATCH PRINT/EFC A=63D6, F=C39694,
                        C=C34A9Ø
PATCH PRINT/EFC A=6491, F=CDEA94,
                        C=ØØØØØØ
PATCH PRINT/EFC A=471D, F=33,
                        C=34
```

TAB and USING on the CoCo

James R. Demers
377 Springfield Street
Chicopee, MA 01013

It has been brought to my attention that the CoCo is being used more and more for business. With this being the case, I thought it would be good to know how to use the TAB and USING at the same time. So I have put together a program using both. This is what the program will look like when it is sent to a printer.

```
**********************************************
```

SUMMARY OF INVENTORY LISTING.
02/18/83
SUMMARY OF OTHER BRANDS.

ARMSTRONG	$	123.55
SEMPERIT	$	1,256.25
OTHER	$	1,254.25
VALVES	$	12.25

TOTAL OTHER BRANDS	$	2,646.30
CARLISLE	$	12,365.25
MICHELIN	$	0.00
GILLETTE	$	12,354.25
GOODRICH	$	32,145.25
INDUSTRIAL	$	13.22
TUBES	$	1,225.25
TOTAL INVENTORY	$	60,749.52

```
**********************************************
```

```
1Ø ' PROGRAM WRITTEN ESPECIALLY
2Ø ' FOR THE COLOR COMPUTER TO
3Ø ' SHOW HOW TO USE BOTH THE
4Ø ' "TAB" AND "USING" FUNCTIONS
5Ø ' TO LINE UP DECIMAL POINTS.
6Ø ' WRITTEN BY
7Ø ' JAMES R. DEMERS
8Ø ' 377 SPRINGFIELD ST.
9Ø ' CHICOPEE, MA. Ø1Ø13
1ØØ ' INVENTORY PROGRAM
11Ø CLS
12Ø K$ = "$###,###.##"
13Ø PRINT#-2, STRING$(65,"*")
14Ø PRINT#-2
15Ø PRINT#-2, TAB(5) "SUMMARY OF INVENTORY LISTING."
16Ø PRINT "ENTER THE DATE-MM/DD/YY "
17Ø INPUT A$
18Ø PRINT#-2, TAB(5); A$
19Ø PRINT#-2
     : PRINT#-2
     : PRINT#-2
2ØØ PRINT#-2, TAB(1Ø) "SUMMARY OF OTHER BRANDS."
21Ø PRINT#-2
22Ø INPUT "ARMSTRONG"; A
     : INPUT "OTHER BRANDS"; O
     : INPUT "SEMPERIT"; S
23Ø INPUT "GILLETTE"; G
     : INPUT "CARLISLE"; C
     : INPUT "INDUSTRIL"; I
     : INPUT "MICHELIN"; M
24Ø INPUT "GOODRICH"; H
     : INPUT "TUBES"; T
     : INPUT "VALUES"; V
25Ø  PRINT#-2, TAB(15) "ARMSTRONG"; TAB(3Ø);
     : PRINT#-2, USING K$; A
26Ø PRINT#-2, TAB(15) "SEMPERIT"; TAB(3Ø);
     : PRINT#-2, USING K$; S
27Ø PRINT#-2, TAB(15) "OTHER"; TAB(3Ø);
     : PRINT#-2, USING K$; O
28Ø PRINT#-2, TAB(15) "VALVES"; TAB(3Ø);
     : PRINT#-2, USING K$; V
29Ø Y = A + S + O + V
3ØØ PRINT#-2
     : PRINT#-2
31Ø PRINT#-2, TAB(2Ø) "TOTAL OTHER BRANDS"; TAB(5Ø);
     : PRINT#-2, USING K$; Y
32Ø PRINT#-2, TAB(2Ø) "CARLISLE"; TAB(5Ø);
     : PRINT#-2, USING K$; C
33Ø PRINT#-2, TAB(2Ø) "MICHELIN"; TAB(5Ø);
     : PRINT#-2, USING K$; M
34Ø PRINT#-2, TAB(2Ø) "GILLETTE"; TAB(5Ø);
     : PRINT#-2, USING K$; G
35Ø PRINT#-2, TAB(2Ø) "GOODRICH"; TAB(5Ø);
     : PRINT#-2, USING K$; H
36Ø PRINT#-2, TAB(2Ø) "INDUSTRIAL"; TAB(5Ø);
     : PRINT#-2, USING K$; I
37Ø PRINT#-2, TAB(2Ø) "TUBES"; TAB(5Ø);
     : PRINT#-2, USING K$; T
38Ø Z = Y + C + M + G + T + H + I
39Ø PRINT#-2
4ØØ PRINT#-2, TAB(2Ø) "TOTAL INVENTORY"; TAB(5Ø);
     : PRINT#-2, USING K$; Z
41Ø PRINT#-2
42Ø PRINT#-2, STRING$(65,"*")
43Ø END
```

Speller

Robert K. Wilcox
111 1st Avenue S.E.
Cedar Rapids, IA 52401

This little program was inspired by my son's need for help with his spelling words at school. It was fun to create, and I hope it will be useful to other parents and kids.

To create a word list on tape, first 'RUN' the program then choose option one at the menu. You will be allowed to enter ten words of up to twelve characters each. You must have ten words, no less. The program will not allow words longer than twelve characters. When your list is complete, the computer will prompt you to place a data tape in the recorder. The computer will store the ten words on tape, then tell you to plug a microphone into the recorder. You have better results with a hand mike rather than the condenser mike built into the recorder. You should also disconnect the auxiliary input plug.

The computer will prompt you to pronounce each word three times, but it's best not to wait for the second and third prompts. For example, I will say the word, use it in a short sentence, and then pronounce it again—all without pauses. This is required because of recorder 'drift' during the on/off routines of playback. Here's a sample.

Right way: 'BOOK; JOHN IS READING A BOOK; BOOK'

Wrong way: 'BOOK . . . (PAUSE) . . . JOHN IS READING A BOOK . . . (PAUSE) . . . BOOK.'

After the last word is pronounced, the program will return to the menu. At this point you can rewind the tape to try out your word list. Choose number two at the main menu and follow the prompts. Be sure the volume control on your T.V. is turned up so you can hear the words pronounced. After each word, wait for the recorder to be turned off and for the cursor to return to the screen. That's all there is to it! I hope you have fun improving your spelling.

```
2 SAVE "SPELLER"
    : END
1Ø '              SPELLER
2Ø '              ROBERT K. WILCOX
3Ø '              APRIL 1982
4Ø '
5Ø CLS(Ø)
6Ø W = Ø
    : C1 = Ø
7Ø DIM WORD$(1Ø)
    : DIM P$(12)
8Ø GOSUB 182Ø
    : GOSUB 87Ø  'TITLE PAGE
9Ø 'SCREEN SETUP
1ØØ PRINT@1, "words";
    : PRINT@8, "right";
    : PRINT@14, "word";
    : PRINT@19, "words";
    : PRINT@25, "missed";
11Ø GOSUB 38Ø
    : GOSUB 42Ø
    : GOSUB 34Ø
    : GOSUB 36Ø
12Ø GOSUB 152Ø  'INPUT DATA TAPE
13Ø TC = Ø
14Ø FOR W = 1 TO 1Ø
15Ø GOSUB 44Ø
16Ø AUDIO ON
17Ø MOTOR ON
    : FOR X = 1 TO 3
    : GOSUB 164Ø
    : NEXT X
18Ø MOTOR OFF
19Ø PRINT@449, "HOW'S IT SPELLED? ";
    : GOSUB 186Ø
2ØØ PRINT@479, CHR$(128);
    : GOSUB 63Ø
    : GOSUB 73Ø
21Ø IF Q = 1 THEN 26Ø
22Ø GOSUB 33Ø
    : GOSUB 35Ø
    : PRINT@449, "SPELL IT AGAIN--> ";
    : GOSUB 186Ø
23Ø PRINT@479, CHR$(128);
24Ø IF A$ = "" THEN GOTO 22Ø
25Ø IF A$ <> LEFT$(WORD$(W), LEN(A$)) THEN GOTO 22Ø
26Ø GOSUB 33Ø
    : GOSUB 35Ø
27Ø NEXT W
28Ø AUDIO OFF
29Ø I$ = "PRESS ANY KEY FOR SCORE..."
    : GOSUB 165Ø
3ØØ GOSUB 168Ø 'TO SCOREBOARD
31Ø GOTO 31Ø
32Ø 'SUBROUTINES
33Ø 'BLANK MESSAGE LINE 1
34Ø PRINT@449,"                                ";
    : RETURN
35Ø 'BLANK MESSAGE LINE 2
36Ø PRINT@481,"                                ";
    : RETURN
37Ø 'SCREEN SETUP
38Ø FOR X = Ø TO 384 STEP 32
    : PRINT@X, CHR$(138+48);
    : NEXT X
39Ø FOR X = 13 TO 384 + 13 STEP 32
    : PRINT@X, CHR$(133+48);
    : NEXT X
4ØØ FOR X = 18 TO 384 + 18 STEP 32
    : PRINT@X, CHR$(138+48);
    : NEXT X
41Ø  FOR X = 31 TO 415 STEP 32
    : PRINT@X, CHR$(133+48);
    : NEXT X
    : RETURN
42Ø FOR X = 33 TO 32 + 12
    : PRINT@X, CHR$(14Ø+32);
    : NEXT X
43Ø FOR X = 32 + 19 TO 62
    : PRINT@X, CHR$(14Ø+32);
    : NEXT X
    : RETURN
44Ø 'GOLD BOX ROUTINE
45Ø P = 32 * W
    : IF W > 1 THEN GOSUB 59Ø
46Ø W$ = STR$(W)
47Ø PRINT@47 + P, RIGHT$(W$,2);
48Ø PRINT@14 + P, CHR$(129+16);
49Ø PRINT@15 + P, CHR$(131+16);
5ØØ PRINT@16 + P, CHR$(131+16);
51Ø PRINT@17 + P, CHR$(13Ø+16);
52Ø PRINT@46 + P, CHR$(133+16);
53Ø PRINT@49 + P, CHR$(138+16);
54Ø PRINT@78 + P, CHR$(132+16);
55Ø PRINT@79 + P, CHR$(14Ø+16);
56Ø PRINT@8Ø + P, CHR$(14Ø+16);
57Ø PRINT@81 + P, CHR$(136+16);
58Ø RETURN
59Ø P = P - 32
6ØØ FOR X = 14 TO 17
61Ø PRINT@X + P, CHR$(128);
62Ø NEXT X
    : P = P + 32
    : RETURN
63Ø 'COLOR BAR
64Ø FOR X = 1 TO 15
65Ø PLAY "O4;V5;T25"
66Ø C = RND(8) - 1
67Ø S = RND(12)
68Ø L = RND(32)
69Ø PRINT@415 + L, CHR$(14Ø+(C*16));
7ØØ PLAY "XP$(S);"
71Ø NEXT X
    : RETURN
72Ø 'WORD CHECK
73Ø P1 = 33
    : P3 = 51
74Ø IF A$ = WORD$(W) GOTO 8ØØ
75Ø Q = Ø
```

```
     : GOSUB 830
760 PRINT@(P3+(32*W)), WORD$(W);
770 FOR X = 1 TO 5
     : PRINT@449,"                    ";
     : FOR Y = 1 TO 100
     : NEXT Y
     : PRINT@449, "CHECK SPELLING-->";
     : FOR Y = 1 TO 100
     : NEXT Y
     : NEXT X
780 FOR X = 1 TO 5
     : PRINT@481,"                              ";
     : FOR Y = 1 TO 100
     : NEXT Y
     : PRINT@481, "SPELL IT RIGHT TO GO ON...";
     : FOR Y = 1 TO 100
     : NEXT Y
     : NEXT X
790 GOTO 820
800 TC = TC + 1
     : Q = 1
     : GOSUB 830
810 PRINT@(P1+(32*W)), WORD$(W);
820 RETURN
830 'LENGTH ROUTINE
840 IF LEN(WORD$(W)) = 12 THEN 860
850 WORD$(W) = WORD$(W) + " "
     : GOTO 840
860 RETURN
870 ' TITLE PAGE
880 READ LO, CO
890 IF LO = 99 GOTO 1080
900 POKE LO, CO
910 GOTO 880
920 'S
930 DATA 1064, 156, 1032, 147, 1031, 147, 1030, 147,
     1029, 147, 1028, 147
940 DATA  1060, 159, 1092, 159, 1124, 156, 1125, 156,
     1126, 156, 1127, 156
950 DATA  1128, 159, 1160, 159, 1192, 159, 1191, 147,
     1190, 147, 1189, 147, 1188, 159
960 'P
970 DATA  1131, 159, 1163, 159, 1195, 159, 1227, 159,
     1259, 159, 1291, 159, 1132, 156, 1133, 159,
     1165, 159, 1197, 159, 1196, 147
980 'E
990 DATA  1168, 156, 1169, 156, 1137, 159, 1136, 156,
     1135, 159, 1167, 159, 1199, 159, 1200, 147,
     1201, 147
1000 'L
1010 DATA  1043, 147, 1075, 159, 1107, 159, 1139,
     159, 1171, 159, 1203, 159
1020 DATA  1045, 147, 1077, 159, 1109, 159, 1141,
     159, 1173, 159, 1205, 159
1030 'E
1040 DATA  1176, 156, 1177, 156, 1145, 159, 1144,
     156, 1143, 159, 1175, 159, 1207, 159, 1208, 147,
     1209, 147
1050 'R
1060 DATA  1147, 159, 1179, 159, 1211, 159, 1148,
     156, 1149, 159
1070 DATA 99, 99
1080 PRINT@359, " ROBERT K. WILCOX ";
1090 PRINT@391, " ---------------- ";
1100 PRINT@423, "       1982       ";
1110 FOR Y = 1 TO 6
1120 C = RND(8) - 1
     : IF C = C1 THEN GOTO 1120
1130 FOR X = 1350 TO 1369
     : POKE X, 140 + (C*16)
     : NEXT X
1140 FOR X = 1370 TO 1498 STEP 32
     : POKE X, 143 + (C*16)
     : NEXT X
1150 FOR X = 1497 TO 1478 STEP - 1
     : POKE X, 131 + (C*16)
     : NEXT X
1160 FOR X = 1477 TO  1349 STEP - 32
     : POKE X, 143 + (C*16)
     : NEXT X
1170 C1 = C
1180 NEXT Y
1190 CLS(0)
     : PRINT@13, " MENU ";
1200 PRINT@66, "1.";
     : PRINT@69, "PREPARE NEW WORD LIST   ";
1210 PRINT@130, "2.";
     : PRINT@133, "RUN THE SPELLER PROGRAM ";
1220 P = 32 * 8
     : GOSUB 480
1230 PRINT@47 + P, "  ";
     : A$ = INKEY$
     : IF A$ = "" THEN 1230
1240 IF A$ < "1" OR A$ > "2" THEN 1230
1250 IF A$ = "1" THEN PRINT@48 + P, A$;
     : FOR X = 1 TO 460 * 1
     : NEXT X
     : GOTO 1270
1260 PRINT@48 + P, A$;
     : FOR X = 1 TO 460 * 1
     : NEXT X
     : CLS(0)
     : GOTO 90
1270 'WORD ENTRY
1280 CLS(0)
     : PRINT@10, "WORD ENTRY";
     : PRINT@35, "12 LETTER LIMIT PER WORD";
1290 FOR X = 1 TO 10
     : W$ = STR$(X)
     : PRINT@(71+(32*X)), RIGHT$(W$,2);
     : NEXT X
1300 P = 75
     : FOR X = 1 TO 10
1310 GOSUB 1620
     : LINE INPUT WORD$(X)
1320 FOR Y = P + 12 + (32*X) TO P + 25 + (32*X)
     : PRINT@Y, CHR$(128);
     : NEXT Y
1330 IF LEN(WORD$(X)) > 12 THEN GOTO 1310 ELSE NEXT X
1340 CLS(0)
     : PRINT "PREPARE TAPE FOR DATA RECORDING ";
1350 GOSUB 340
     : GOSUB 350
1360 I$ = "PRESS ANY KEY WHEN READY..."
     : GOSUB 1650
1370 OPEN "O", #-1, "WORDS"
1380 FOR X = 1 TO 10
     : PRINT@239, X;
     : PRINT#-1, WORD$(X)
     : NEXT X
1390 CLOSE#-1
     : MOTOR ON
     : GOSUB 1630
     : MOTOR OFF
1400 CLS(0)
     : PRINT "PREPARE RECORDER FOR VOICE
     RECORDING BY REMOVING THE       AUXILIARY INPUT
     PLUG.           ";
1410 PRINT@32 * 4, "YOU WILL BE PROMPTED TO
     PRONOUNCE EACH WORD.            ";
1420 GOSUB 330
     : GOSUB 350
1430 I$ = "PRESS ANY KEY TO BEGIN..."
     : GOSUB 1650
1440 MOTOR ON
     : FOR X = 1 TO 10
     : CLS(0)
1450 X$ = STR$(X)
1460 PRINT "WORD NO."; RIGHT$(X$,2); " IS ";
     WORD$(X); "..."
1470 GOSUB 1630
     : PRINT@14+64, WORD$(X); "..."
1480 GOSUB 1630
     : PRINT@14 + 128, WORD$(X); "."
1490 GOSUB 1630
```

```
     : NEXT X
1500 MOTOR OFF
     : GOSUB 1630
1510 GOTO 1190 'MENU
1520 'READ DATA TAPE
1530 PRINT@449, "PREPARE TO LOAD DATA TAPE.";
1540 I$ = "PRESS ANY KEY WHEN READY..."
     : GOSUB 1650
1550 GOSUB 330
     : GOSUB 350
1560 OPEN "I", #-1, "WORDS"
1570 FOR X = 1 TO 10
     : INPUT#-1, WORD$(X)
     : NEXT X
1580 IF EOF(-1) THEN CLOSE#-1 ELSE 1580
1590 MOTOR ON
     : FOR Y = 1 TO 460 * 3
     : NEXT Y
1600 MOTOR OFF
1610 AUDIO ON
     : RETURN
1620 PRINT@P + (32*X), "            ";
     : PRINT@P + (32*X),;
     : RETURN
1630 FOR Y = 1 TO 460 * 4
     : NEXT Y
     : RETURN
1640 FOR Y = 1 TO 440 * 4
     : NEXT Y
     : RETURN
1650 'INKEY$ ROUTINE
1660 A$ = INKEY$
     : PRINT@481, I$;
     : GOSUB 1670
     : GOSUB 350
     : GOSUB 1670
     : IF A$ = "" THEN 1660 ELSE RETURN
1670 FOR Z = 1 TO 60
     : NEXT Z
     : RETURN
1680 'SCOREBOARD
1690 W = 10
1700 CLS(0)
1710 PRINT@5, " SPELLING  SCOREBOARD ";
1720 TW = W - TC
     : TC$ = STR$(TC)
     : TW$ = STR$(TW)
1730 PRINT@32 * 4, " TOTAL WORDS SPELLED RIGHT ";
1740 P = 110
     : W$ = RIGHT$(TC$,2)
1750 GOSUB 470
1760 PRINT@32 * 8, " TOTAL WORDS MISSED        ";
1770 P = 110 + (32*4)
     : W$ = RIGHT$(TW$,2)
1780 GOSUB 470
1790 GOSUB 330
     : GOSUB 350
1800 I$ = "PRESS ANY KEY TO GO ON..."
     : GOSUB 1650
1810 RESTORE
     : CLS(0)
     : GOTO 80
1820 'TONE GENERATOR
1830 FOR X = 1 TO 12
1840 P$(X) = STR$(X)
1850 NEXT X
     : RETURN
1860 A$ = ""
     : 'INKEY SUBROUTINE
1870 PRINT@467, A$;
1880 C = LEN(A$)
     : IF C = 12 THEN 1970
1890 PRINT@467 + LEN(A$), CHR$(142);
     : GOSUB 1960
1900 PRINT@467 + LEN(A$), CHR$(143);
     : GOSUB 1960
1910 B$ = INKEY$
     : IF B$ = "" THEN 1890
1920 IF B$ = CHR$(8) THEN PRINT@467,"            ";
     : GOTO 1860
1930 IF B$ = CHR$(13) THEN 1970
1940 IF B$ < CHR$(65) OR B$ > CHR$(90) THEN 1900
1950 A$ = A$ + B$
     : GOTO 1870
1960 FOR X = 1 TO 30
     : NEXT X
     : RETURN
1970 RETURN
```

Scripsit 2.1 and the Model 12

In Scripsit 2.1 on the Model 12, the F1 and F2 function keys operate the same as on the Model II.

(F1) Insert Text
(F2) Delete Text

The F3 through F8 functions keys of the Model 12 perform the following operations in Scripsit 2.1:

Function Key No.	Scripsit Operation	Model II Equivalent
(F3)	Define Text Blocks	(CTRL) (D)
(F4)	Line Centering	(CTRL) (L)
(F5)	Utilities	(CTRL) (U)
(F6)	Get Previous Page	(CTRL) (P)
(F7)	Get Next Page	(CTRL) (N)
(F8)	Search Text	(CTRL) (S)

String Thing

David Lewis
Box 88
Shady, NY 12479

I stumbled across an interesting phenomenon that I think others would like. It allows for interesting effects at machine-language speed.

What I did was change the index of a dummy string so that it pointed to part of the memory reserved for the video screen. I then printed the string and found that its value changed while it was being printed! Listing 1 for the Model I/III shows this.

```
10 CLEAR 300
     : A$ = STRING$(255,32)   'SET UP DUMMY
20 V = VARPTR(A$)
     : POKE V+1,0
     : POKE V+2,60 'CHANGE TO 3C00H
30 CLS
     : PRINT "<YOUR NAME>"
     : PRINT "A$ = "A$
```

The value of the string is being printed over part of itself, and then that overlay is printed, and so on!

Here's another use.

```
10 CLEAR 300
     : A$ = STRING$(255,32)
20 V = VARPTR(A$)
     : POKE V+1, 0
     : POKE V+2, 60
30 CLS
     : PRINT "!*%#&";
     : PRINT A$
```

WORDPRO and PRINTING

Joseph A. Count
BVM Software
824 Sedgefield Lane
Sylacauga, AL 35150

The following programs are designed to operate as a pair. The first is a word processing program. The second is a printing program for files created with the word processing program. These programs will operate on a Model III with 48K and two disk drives.

The word processor is relatively simple and will create an ASCII file that is in the format of a BASIC program except that each line is a remark. This prevents the interpreter from converting the lower case letters to upper case. Obviously you can type in both upper and lower case text. The program allows you to type on a blank video screen just as you would on a typewriter. The VIDEO RAM is read and written to a file. This file can be edited by LOADING it while in BASIC. All of the standard editing features of the Model III can be used in this way. In addition, renumbering can be done to allow the insertion of whole paragraphs. Following editing, the normal SAVE command (SAVE "FID/TXT:1",A) is used. With each blank screen a period (.) is printed in the first column of the last video line. This is done to remind the user that (ENTER) should not be used at the end of this line or else the video display will scroll. To have the computer write the page to a file just key the (↓). If this is done on the last video line, a new blank page will be presented. To terminate input before the end of a video page just type (↓) at the beginning of the first blank line. This prevents all the blank lines from going to the file and also serves to terminate the program. Of course, during typing, the (←) can be used to backspace and erase as normal. The reading of the video ram takes a few seconds. You can add to the file at any later time as the program will automatically increment line numbers based on the highest in the existing file.

Printing is accomplished with the PRINTING program and can be on tractor forms or single sheets depending on your printer capabilities. What follows is a detailed discussion for programmers.

WORDPRO:

Line 110 - clears string space
120 - V$( ) will hold each video line
125 - NL is the starting line number for the data file
130 - beginning of video RAM
137 - appends /TXT to file name and specifies D:1. Can be D:0 if you only have one drive.
200 - beginning of Video RAM read. Based on routine in Cat. No. 26-2107 page 293.
290 - adds info to file
310 - writes a line number and makes it a REMARK
320 - increments line number by 5

The portion of the program from line 60000 is to save the program without having to remember the name or type SAVE.

PRINTING:

Line 20 - This is for a 48K system. This can be adjusted to suit your particular system.
120 - beginning of lines/page counting. This routine will by-pass intercepts in BASIC and form feed on printers.
151 - used to compensate for leading blank in line depending on how the file was SAVEd or edited

Lines 200-350 can be bypassed if you do not have a DMP-200 printer by deleting line 25.

WORDPRO

```
100 '---------- PLACES VIDEO DISPLAY MEMORY MAP IN
     STRING -----
110 CLEAR 16*64+255
120 DEFINT A-Z
     : DIMV$(15)
125 NL=1100
130 VM=15360
132 CLS
135 INPUT"Enter file name";F$
136 IF LEN(F$) > 8 THEN PRINT,"** TOO MANY LETTERS
     **"
     : GOTO 135
137 INPUT"Is this a new file  (Y/N)";G$
     : F$=F$+"/TXT:1"
138 IF G$ = "N" THEN GOSUB 500
139 IF G$="Y" THEN OPEN "O",1,F$
     : CLOSE 1
140 PRINT CHR$(14);
     : CLS
     : PRINT @ 960,".";
     : PRINT CHR$(28);
150 A$=INKEY$
     : IF A$ = "" THEN 150
160 IF A$ > CHR$(31) THEN PRINT A$;
     : GOTO 150
170 IF A$ = CHR$(8) OR A$ = CHR$(13) THEN PRINT A$;
     : GOTO150
180 IF A$ = CHR$(10) THEN 200    ' REMOVED THIS LINE
     TEMP 200
190 GOTO 150
200 PRINT CHR$(15);
210 FOR LI = 0 TO 15
220 START = LI*64+VM
230 FOR CL = 0 TO 63
240 AD = START + CL
250 V$(LI) = V$(LI) + CHR$(PEEK(AD))
260 NEXT CL
265 NEXT LI
270 CLS
     : PRINT"...Page is being stored"
280 '------------ WRITE INFO TO FILE -----------
290 OPEN "E",1,F$
300 FOR LI = 0 TO 15
305 IF LEFT$(V$(LI),1) = "!" THEN QI$="!"
     : GOTO 340
310 PRINT #1,NL;"'";V$(LI)
320 NL = NL + 5
330 NEXT LI
340 CLOSE 1
350 CLS
354 IF QI$="!" THEN CLS
     : END
356 LPRINT"** LAST VIDEO LINE RECORDED **"
     : LPRINT V$(LI-1)
     : LPRINT
360 FOR LI = 0 TO 15
     : V$(LI)=""
     : NEXT LI
370 GOTO 140
500 '---------- READS HIGHEST LINE NUMBER IN OLD FILE
     ------
```

```
51Ø OPEN "I",1,F$
52Ø IF EOF(1) THEN 545
53Ø LINE INPUT#1,QQ$
54Ø GOTO 52Ø
545 D=INSTR(QQ$,"'")
     : IF D = 6 THEN Q1$=LEFT$(QQ$,4)
     : GOTO 555
55Ø Q1$=MID$(QQ$,2,4)
555 Q2 = VAL(Q1$)
56Ø NL = Q2 + 5
565 CLOSE 1
57Ø RETURN
6ØØØØ '************* AUTO SAVE SEQUENCE
    ********************
6ØØ1Ø OO$="WORDPRO:Ø"
6ØØ3Ø PRINT
     : PRINT "File ";OO$;" will now be SAVED
     automatically"
     : PRINT
6ØØ4Ø SAVE OO$,A
```

PRINTING

```
1Ø 'PROGRAM TO OUTPUT A TEXT FILE TO PRINTER W/PAGE
    CONTROL
11 '.......... THIS ONE USES A WIDER BORDER
    .........
2Ø CLEAR 2ØØØØ
    : DIM II$(5ØØ)
25 GOSUB 2ØØ
26 GOSUB 5ØØ
27 CLS
3Ø INPUT"ENTER FILE NAME ";A$
4Ø F$ = A$ + "/TXT:1"
42 INPUT"Enter COLUMN # to start printing from left
    side (1Ø/inch)";CO
    : CO=CO-1
44 PRINT
    : INPUT"Enter number of SPACES between lines (1,
    2, or 3)";SP
46 IF SP < 1 OR SP> 3 THEN PRINT
    : PRINT"enter a number between 1 and 3..."
    : GOTO 44
48 PRINT
    : INPUT"Enter the # of LINES/PAGE (66 maximum :
    6/inch)";PG
49 IF PG > 66 THEN PRINT
    : PRINT"66 is the most lines you can have..."
    : PRINT
    : GOTO 48
5Ø '------------ FILE INPUT ---------------
6Ø OPEN "I",1,F$
7Ø FOR X = 1 TO 5ØØ
8Ø IF EOF(1) THEN 11Ø
9Ø LINEINPUT#1,II$(X)
1ØØ NEXT X
11Ø CLOSE 1
    : PRINT
    : PRINT"...File loaded - please index paper to
    top of page"
115 INPUT" ** Hit ENTER to continue ";Y$
116 CLS
12Ø POKE 16425,1
13Ø FOR X = 1 TO 5ØØ
135 IF II$(X) = "" THEN 18Ø
14Ø P = PEEK(16425)
15Ø IF P >= PG THEN GOSUB 4ØØ
151 D=INSTR(II$(X),"'")
    : IF D = 6THEN II$(X) = MID$(II$(X),7)
    : GOTO 153
152 II$(X)= MID$(II$(X),8)
153 IF LEN(II$(X)) = Ø THEN LPRINT
16Ø IF SP=1 THEN LPRINT TAB(CO);II$(X)
161 IF SP=2 THEN LPRINT TAB(CO);II$(X)
    : LPRINT
162 IF SP=3 THEN LPRINT TAB(CO);II$(X)
    : LPRINT
    : LPRINT
17Ø NEXT X
18Ø GOSUB 4ØØ
182 GOSUB 31Ø
185 END
2ØØ '----------------- FONT STYLE SELECTION
    ---------------
21Ø CLS
22Ø PRINT"CHOOSE ONE OF THE FOLLOWING TYPE STYLES:"
23Ø PRINT,"1. NORMAL (1Ø CPI - Data Processing)"
24Ø PRINT,"2. COMPRESSED (12 Characters Per Inch)"
25Ø PRINT,"3. CONDENSED (16.7 Characters Per Inch)"
26Ø PRINT,"4. Proportional"
27Ø PRINT,"5. Correspondence"
28Ø INPUT FZ
29Ø IF FZ >= 1 AND FZ <= 5 THEN 3ØØ ELSE PRINT,"**
    You MUST pick a number between 1 and 5"
    : GOTO 28Ø
3ØØ ON FZ GOSUB 31Ø,32Ø,33Ø,34Ø,35Ø
3Ø5 RETURN
31Ø LPRINT CHR$(27);CHR$(19)
    : RETURN
32Ø LPRINT CHR$(27);CHR$(21)
    : RETURN
33Ø LPRINT CHR$(27);CHR$(2Ø)
    : RETURN
34Ø LPRINT CHR$(27);CHR$(17)
    : RETURN
35Ø LPRINT CHR$(27);CHR$(18)
    : RETURN
4ØØ '------------ PAGE ADVANCE SUBR ---------
41Ø IF SI$="Y" THEN 45Ø
42Ø FOR R = P TO 65
    : LPRINT
    : NEXT R
    : POKE 16425,1
    : RETURN
45Ø CLS
    : POKE 16425,1
46Ø PRINT STRING$(63,"*")
47Ø PRINT
    : PRINT"PLEASE PLACE YOUR NEXT SHEET OF PAPER IN
    THE PRINTER"
48Ø PRINT
    : PRINT"Index the sheet so it is at the location
    you desire"
49Ø PRINT
    : INPUT"...Hit ENTER to continue";Y$
495 RETURN
5ØØ '--------- SUBROUTINE TO PAUSE & CHANGE SHEETS OF
    PAPER ---
51Ø CLS
52Ø PRINT
53Ø INPUT"Are you using single sheets of paper
    (Y/N)";SI$
54Ø IF SI$="Y" OR SI$="N" THEN RETURN
55Ø PRINT,"** INCORRECT RESPONSE **"
    : GOTO 52Ø
6ØØØØ '************* AUTO SAVE SEQUENCE
    ********************
6ØØ1Ø OO$="PRINTING:Ø"
6ØØ3Ø PRINT
    : PRINT "File ";OO$;" will now be SAVED
    automatically"
    : PRINT
6ØØ4Ø SAVE OO$,A
```

Communications Corner

by Al and Dru Simon

Welcome back to our corner. As promised we're going to spend some time this month talking about modems. The first thing we thought you'd like to know is what the word MODEM means. The word is an acronym for Modulator/Demodulator. This is simply a device which translates digital data into audio frequencies which can be transmitted over telephone lines; and then translates these frequencies back into digital data at the other end of the telephone connection.

There are various types of modems with various speeds and functions, and we will discuss some of these, breaking them into generalized categories. The type of modem you should choose for yourself depends very much on the type of use you will make of it. Certainly in the case of modems, the most expensive is not necessarily the best for each end user!

One thing you should be familiar with is the term BAUD RATE. This merely refers to the number of Bits Per Second (BPS) which the device is capable of transmitting or receiving. Much communicating is done at speeds of 300 and 1200 baud (or BPS), though higher speeds are certainly available. It is usually true, however, that the higher the baud rate, the more reliable the telephone connection must be to ensure accurate transmission of data; therefore most hobbyists prefer to use either 300 or 1200 baud modem or devices that allow them to choose between these two popular speeds, since transmission at these speeds is not usually affected by poor telephone lines.

Let's talk about the names of the different types of modems. There's the Acoustic Modem type of device, which is an instrument into which you insert a telephone handset; then there's the Direct Connect Modem, which you connect to your computer by inserting a plug into a jack, and control by hand. At the top of the modem hierarchy are the Smart, or Intelligent Modems, and for the purpose of this column we'll call an Intelligent Modem anything that has any function which does not need human intervention and is also a Direct Connect modem. We'll discuss what each of these modem types do and some of their applications.

Remember that the top of the hierarchy is not necessarily the best modem for your own specific needs. Let's talk about a traveling salesman, for a perfect example. Let's say that he has a Color Computer or some other small computer which, for business purposes, he must cart around with him (Let's say that he doesn't have a Model 100, because that happens to have communications built into it). He will have to interface with as many kinds of telephone connections as possible because he stays in different motels in different cities, and each may have a different type of phone connector. What happens when he has to hook up his Coco and call the home office? Can you see the poor fellow digging under a bureau in his motel room to find the telephone outlet, only to discover that it has the wrong type of connector? He'd have to unscrew the wall plate and attach a couple of alligator clips, and then run out to a jack that he's hastily put together so he could interface with his computer, and if the next motel has yet another type of jack he must start all over again! Whew! Well, there's something easier for him to do; he uses an ACOUSTIC MODEM, like the AC-3 (26-1174). This way he can just pick up the phone in his motel room and place the handset happily into the cradle of the modem.

Because of the vagueries of different telephone handsets, this type of modem is generally limited to a speed of up to 300 baud. There are a few 1200 baud acoustic modems available, but their reliability might be in question depending upon the type of microphone which is in the telephone being used with the modem. This reliability factor is probably why one would more than likely have difficulty finding an Acoustic Modem which operates at higher than 1200 baud. Also, constant use of carbon based microphones often results in a condition called CARBON PACKING in which the granules of carbon actually pack closely together and therefore do not transmit sounds properly, resulting in data dropouts (loss of pieces of data).

There are two ways to deal with Carbon Packing. The better of the two is to replace the carbon microphone in your telephone with an electronic microphone. The second method (which will hold you if this occurs in the middle of a transmission) is to strike the microphone end of the telephone with lateral force (perpendicular to the length) of about 10 to 20 foot-pounds, perhaps five or six times. This will loosen up the carbon sufficiently to restore communications, temporarily at least. A symptom of Carbon Packing is frequent parity errors when you know you've set your parity correctly. Another good indicator is the SPORADIC insertion of garbage characters in an otherwise normal line of data. Perhaps as much as 20% of the characters may be "scrambled." After you've terminated your connection, hold the handset with the microphone away from you facing upwards and strike it against the side of a desk (etc., etc. as above).

Generally speaking, the adaptability of the Acoustic Modem to almost any type of phone makes it ideal for a person who does a lot of traveling. Despite the speed limitation imposed on that particular type of device, it is very convenient for use with different telephones because all the user does is insert the handset into the modem.

The next type of modem we'll discuss is the DIRECT CONNECT modem like the Modem I (26-1172). This is very convenient for the person who does a lot of calling out but does not have a lot of money to invest in a "Smart Modem." This type of machine makes an excellent workhorse. They come in all speeds, featuring great ease of operation; you simply put in your software, initiate it and throw the switch.

The next step up the modem ladder is the SMART or INTELLIGENT MODEM. For the purpose of this article we'll use this term as a broad description. Let's define it as a

modem that has any sort of built-in function which does not require human interface; for example AUTOMATIC ANSWER or AUTOMATIC DIAL. There are some modems on the market which recognize what speed a caller is transmitting at (many BBS callers are familiar with this). There are Protocol Converter Modems and Short Haul modems, many varied and specialized modems, each made for a very specific purpose.

The simplest form of an Intelligent Modem would be a device with an Auto Answer feature. There are many Auto Answer/Manual Originate modems on the market. These are used in business quite frequently when one location is expecting many incoming calls but doesn't really make many outgoing ones.

Another frequently seen type is the Auto Answer/Auto Dial modem, such as the Modem II (26-1173). This type is often used by locations who expect to be doing a lot of outgoing dialing as well as receiving incoming connections. It is a convenient feature to have when one is in the position of having to make a lot of calls because while your modem is dialing your number for you, you are free to make any other preparations which you may require.

Another type of Intelligent Modem is the Programmable modem.This is a device that can perform special types of functions such as redialing a number, if it gets a busy signal, or adjusting its baud rate depending on the rate of the incoming call.

Another interesting modem type is the PROTOCOL CONVERTING MODEM. This is a product that has only hit the market recently, but is expected to spread like wildfire. It is an Intelligent Modem with a built-in translator which enables the user to input one protocol (for example Bit Serial Asynchronous) and output in a completely different format (for example CCITT X.25).

It seems likely that communication is going to become more and more of a moving force in our lives. Not long ago a Seattle, Washington radio station (KMPS) conducted an experiment which broadcast a computer signal over the radio. Listeners were to tape the signals and feed them via modem into their computers and thus receive a specific message. If this method of data disbursal proves successful, it opens almost limitless possibilities!

John Roach, CEO of Tandy Corporation, wrote recently that one day soon people will have pens in their pockets that are really telephones. Very possibly these little pocket devices will have data interfaces on them. Ma Bell is already bringing out telephone networks and central switches which have data interfaces at every station. How long can it be until that is made portable? The interface between telephones and computers is becoming more and more apparent every day.

Yet another area where computers are interfacing with other communications devices is in the realm of video. If you'd like to learn something about this topic, please write and let us know!

GETTING ON BOARD

We have received so many comments and questions concerning Bulletin Boards that we have begun this new feature, in which we hope to give you interesting information about Bulletin Boards around the country every month.

This month we'd like to feature a brand new newsletter called *P L U M B* which is put out by Riverside Data Inc. and edited by Ric Manning. This is the first newsletter of its kind; it is entirely dedicated to information and services for Bulletin Board callers. The newsletter highlights interesting Boards and events happening in the Bulletin Board world, such as the inception of employment notices being placed on certain boards and the closing of a famous (infamous?) "Pirate" board. It is chock-full cover to cover with information of interest to every bulletin board caller, and subscriptions to *P L U M B* are available through Riverside Data Inc at P.O. Box 300, Harrods Creek, KY 40027, 502-228-3820. Inquiries can also be made through their accounts on CompuServe #72715,210 or The Source, STQ007.

P L U M B fills a gap that has never before been dealt with. It is an exciting, fun, and informative publication dealing entirely and exclusively with Bulletin Board information. With Ric Manning's permission, you will find below just a taste of what is waiting for you between the covers of *P L U M B*.

"The ARK-NET PBBS in Little Rock, AR, has an extensive library of TRS-80 programs available for downloading to all callers. No passwords or access fees are necessary The board opens at midnight Monday-Thursday, with extended hours on weekends, at 501 372 1563."

"Three new computerized message systems now serve professional writers and photographers. One is Photonet, a New York-based network for advertising, publishing and photography, Photo-1 is a communications system for photojournalists and photo departments to help each share resources and avoid duplication The Notebook lists notices from writers and photographers who are available for freelance assignments, databases of magazines and private sections for agents and publishers. Here's where to call for more information: Photonet, 212 750 1386 (voice); Photo-1, 212 929 8030 (voice); The Notebook, 305 686 4862 (modem) and 305 648 8751 (voice)."

"The Dragon's Lair Adventure System in Southern California has 15 adventure and strategy games ready to play online. Enter the lair by calling 213 428 5206."

P L U M B not only spotlights interesting boards in each issue but has a list of other Bulletin Board phone numbers for you to try, along with a many helpful bits of information regarding each one. We thank Ric for his permission to reprint a small taste of his treasures here for you.

THE CORNER MAILBOX

Here's a question about available software to use with Radio Shack hardware: I own a Model II and a Modem II. I'm trying to find a communications package that will support auto-redial with this modem. I dial into a busy university time sharing system. I want the modem to keep dialing until it gets me a port and then log me on and run a program.

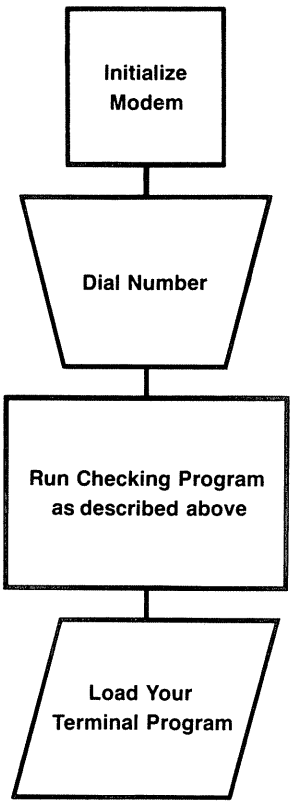
The documentation that comes with the Modem II suggests that this program would be relatively easy to write for an experienced programmer . . . which I'm not . . . programs that support this auto-redial feature abound under CP/M, but what about under TRSDOS for the Model II?

Warren S. Williams, Ed.D.
Eastern Michigan University

Dear Warren:

To our knowledge there is no such software package currently available, but you can accomplish your objective by creating a "DO" file. Embedded in this file (which I will de-

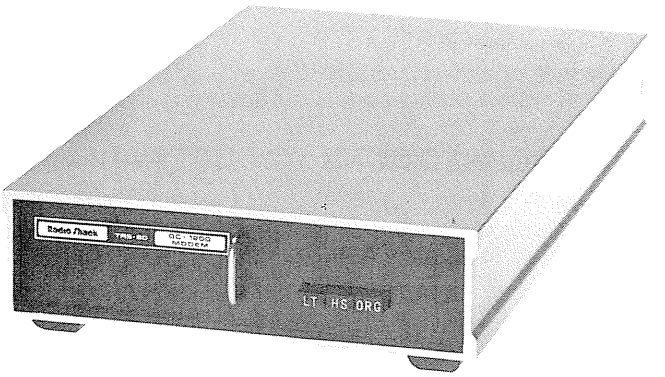
scribe below in block form) will be a program which you will have to write to check the carrier detect. If you refer to June's article it will explain how to address the port. In your particular case there is a service call available. Please refer to page 282 of your Model II Owner's Manual. There you will find in lines 980 through the end a carrier check routine. You must put this routine in a loop (You cannot use the BC register as a counter since the service routine uses it—I suggest you use the HL register to store the number to decrement.) long enough to allow for three rings of the telephone followed by about five seconds for the carrier to come up. If the carrier is still not present after that time, have the program disable interrupts and jump to location 0000. This will reboot your machine, so you must perform the following patch: PATCH SYSRES/SYS A=2367 F=73 C=CA. This will remove the date and time prompts from your system, thereby allowing you to do an "AUTO DO Filename." Therefore, if no carrier is present, you will go through a cold reboot. If you look at the block diagram beneath, you will see the the first thing to do in the DO file is to Initialize your Modem and dial.



When your menu comes up, you know you're connected, and you can use the O option (NOTE: you must first use the A option to build your Auto sign-on and then save the changed program as described on page 178 of your manual) to enter Terminal Mode with Auto Sign-on.

The above should solve your problem, however, if you have trouble don't despair; go to your nearest Radio Shack Computer Center and ask to speak to the Customer Service representative. If that does not prove satisfactory ask THEM to contact their NAR (National Accounts Representative), since the NARs deal with communications quite frequently. Good luck!

That will about round off our corner for this month. We hope you'll come back to visit with us again next month. Happy Communicating!



Changing Strings to Graphic Characters

Nicholas Leon
26 Tanners Drive
Wilton, CT 06897

I am a regular reader of your magazine, and I have noticed that you have been doing a lot of articles on changing strings into graphic. So I wrote the following program.

This program will run on any TRS-80 with Level II BASIC or Model III BASIC. It will not run on a disk system because BASIC takes up a different address. This program takes a string and replaces it with any ASCII character.

You can use this program to set up variables in another program. To use it correctly you MUST have both quotes, otherwise the program will crash. Also, it MUST be before the input statement or else that will be changed. Here is the list of variables.

C—The ASCII code of the character with which you want to replace the string.
S—The start of the string in the memory.
E—The end of the string in the memory.
A, A1, A2—Counters for memory addresses.

```
Ø CLS
1 'Put your string between lines 1 and 9
9 PRINT "ASCII CODE OF THE CHARACTER YOU WANT TO
    REPLACE THE STRING WITH"
    : INPUT C
1Ø FOR A = 17385 TO 32767
2Ø IF PEEK(A) = 34 THEN S = A + 1
    : GOTO 4Ø
3Ø NEXT A
4Ø FOR A1 = S TO 32767
5Ø IF PEEK(A1) = 34 THEN E = A1 - 1
    : GOTO 7Ø
6Ø NEXT A1
7Ø FOR A2 = S TO E
    : POKE A2,C
    : NEXT A2
8Ø LIST
```

Bugs, Errors, and Fixes

LETTER CORRECTIONS TO SOFTWARE

Following are brief descriptions of problems to be fixed in specific software packages and the dates of letters that were sent to all registered owners containing the corrections to the problem.

If you are a registered owner of a software package described below and have not received the letter detailing the software problem and its correction, then contact your nearest Radio Shack Computer Center or Computer Customer Services. If you have not registered and are a legal owner of the software, you need to register by sending in the card that came with the package.

Model I/III/4

ADVANCED STATISTICAL ANALYSIS (26-1705)

There is a correction in the Analysis of Variance program that allows the Standard Deviation to be calculated correctly when a limited number of data entries are made.

Letter dated: January 4, 1982

Model II, 12, 16

PAYROLL (26-4503)

These corrections allow you to enter wage amounts containing up to three decimal places and correctly calculate checks.

Letter dated: December 15, 1982

SCRIPSIT 2.0 (26-4531)

Due to on going improvement of this package and the release of several new Radio Shack printers, it has become necessary to modify your software. We have provided drivers for the DMP series of printers and the DWP-410 Daisy Wheel, and made corrections to the following Scripsit functions: formatting, text, tab setting, line spacing values, global search and replace, repagination, user-defined print codes, editing, underlining, printing multiple copies of a merge document and use of merge with a serial printer.

Letter dated: December 17, 1982

COBOL GENERATOR (26-4707)

Changes made to the RSCOBOL GENERATOR include cosmetic changes to error messages and prompts within the program, use of pause and clear with data entry screens, sub screen data entry, the use of 3 and 4 alternate keys on Indexed Sequential files, and corrections to error messages when defining the length of report lines.

Letter dated: February 24, 1983

REFORMATTER (26-4714)

This letter contains modifications to correct problems associated with record transfer using the command TRANSFER→IBM→TRSDOS.

Letter dated: November 28, 1982

THINLINE TRSDOS 2.0 (26-6001, 26-6002, 26-4004, 26-4005)

With some thinline drives an ERROR 8 will occur when the drive is accessed when using THINLINE TRSDOS 2.0. TRSDOS does not allow enough time to come up to speed before assuming a valid error has occurred. There is also an erroneous operator prompting for DRIVE NOT READY when running FORMAT.

Letter dated: December 17, 1982 (26-6001, 26-6002)
February 10, 1983 (26-4004, 26-4005)

TRSDOS 4.1 (26-6001, 26-6002, 26-4150)

Recent modifications to the TRSDOS 4.1 operating system allow: speeding up the access of BASIC files, speeding up the memory test, proper handling of flawed diskettes with the SAVE option, corrections to the BACKUP utility, the clearing of records when the "I" command is used, and the correction of some misspellings.

Letter dated: December 17, 1982

OTHER CORRECTIONS

Model I/III/4

MODEL III/4 OPERATION AND BASIC LANGUAGE REFERENCE MANUAL (26-1067)

In the keyboard and video display control characters table on page 228, SHIFT ↓9 should read SHIFT ↓=.

TRENDEX (26-1509)

Short Term Market Trend program, Intermediate Term Market Trend program, and Long Term Market Trend program do not print a future trend for each date when a printout is requested at Computer Trend.

The following changes are optional.

Load the Short Term Market Trend program into the computer and add the following new program line:

```
4125 IF A$ = "" THEN IF ABS(T9) = 1 THEN A$ = "LEVEL"
     ELSE IF T9 = 2 THEN A$ = "UP" ELSE A$ = "DOWN"//@
```

Type CSAVE"S" to save the changes in the program.

Load the Long Term Market Trend program and add the following new line:

```
4125 IF A$ = "" THEN IF ABS(T9) = 1 THEN A$ = "LEVEL"
     ELSE IF T9 = 2 THEN A$ = "UP" ELSE A$ = "DOWN"
```

Type CSAVE"L" to save the program changes.

Load the Intermediate Term Market Trend program and change line 4405 to read:

```
44Ø5 TT = SGN(DD)+1:IF ABS(TT)= 1 THEN A$ = "LEVEL"
     ELSE IF TT = 2 THEN A$ = "UP" ELSE A$ = "DOWN"
```

Add the following new line.

```
4125 TT = SGN(DD)+1: IF ABS(TT) = 1 THEN A$ = "LEVEL"
     ELSE IF TT = 2 THEN A$ = "UP" ELSE A$ = "DOWN"
```

Type CSAVE"I" to save the changes in the program.

Pocket Computer

PROBLEM SOLVING WITH THE TRS-80 POCKET COMPUTER (62-2312)

Line 100 of the Cylinder Program located on page 36 should read as follows:

```
1ØØ "V":V= π * R^2*H
```

Color Computer

KLENDATHU (26-2567)

The first sentence on page 7 should begin:

"KLENDATHU is a video game for the TRS-80 Color computer 16K **with Extended BASIC** and cassette tape system."

COLOR COMPUTER OPERATION MANUAL (26-3001/3002)

Under Printer Software Requirements on page 26, 7 Data Bits (LSB first) should be changed to read 8 Data Bits (LSB first).

COLOR COMPUTER LEARNING LAB MANUAL (26-3153)

The program listing on pages 133-134 needs to be modified so that it matches the program on the tape. Make the following changes to the listing in the manual.

Lines 30 and 40 should read:

```
3Ø A=RND(S)
4Ø B=RND(S)
```

Line numbers 80 and 90 should be changed to lines 60 and 70 respectively

Line 730 should be changed to read:

```
73Ø PRINT "    YOU GOT";2Ø-E
```

GETTING STARTED WITH COLOR BASIC (26-3191) GOING AHEAD WITH EXTENDED COLOR BASIC (26-3192)

On page 270 of Getting Started with Color BASIC and page 208 of Going Ahead with Extended Color BASIC, the locations for the joysticks are incorrect. They should be changed to:

Left Joystick	Right Joystick
Up/Down 15D	Up/Down 15B
Right/Left 15C	Right/Left 15A

PERSONAFILE (26-3260)

On page 19 the sentence in the first paragraph that reads " . . . (Once a subject name has been changed, all records under the old subject name are permanently changed to the new subject name and must be accessed using the new name.)" should be moved and inserted between " . . . changed to the new name." and "The message:" in the fourth line up from the bottom of the page.

The following information should be added to the end of the paragraph that begins " . . . First, type the old name exactly . . . ":

(If you enter a subject name that is already stored on the disk, the message, INVALID NAME, will be displayed. Press [ENTER] and reenter a new subject name.)

On page 21 at the end of the paragraph that begins "First, type the old name exactly . . . ", enter the following sentence.

(If you enter a tag name that is already stored on the disk, the message, INVALID NAME will be displayed. Press [ENTER] and reenter a new tag name.)

Between " . . . to the new name." and "The message:" on the fourth line up from the bottom of page 21 insert the following information.

(Once a tag name has been changed, all records under the old tag name are permanently changed to the new tag name and must be accessed using the new name.

PYRAMID (26-3310)

The first paragraph under the heading "For the Truly Adventurous Only" on page 5 of the manual should be deleted. The magic word PLUGH is not operative in the Color Computer version of the game which makes this paragraph irrelevant.

EDTASM + (26-3250)

An explanation of how users with Color BASIC machines can use the BASIC CLOADM command to load their programs was omitted from the manual. Insert the following at the end of Step 3 on page 25 of the manual.

Note: If your Color Computer does not have Extended Color BASIC, you must perform the following steps:

(A) Write, Edit and Debug your machine language program using EDTASM+.
(B) Assemble your program onto cassette tape.
(C) Switch your Color Computer's power off then on.
(D) Enter ZBUG.
(E) Using ZBUG's "L" command, load your program into memory.
(F) Then save your program onto tape using ZBUG's "P" command.
(G) BASIC's CLOADM and EXEC commands will then operate normally.

Model 100

MODEL 100 OWNER'S MANUAL (26-3801)

Figure A-4 on page 194 is incorrect. The Model 100 should be connected to Port A on the Model II/16. The terminator plug would then be plugged into Port B.

On page 180 under the title SOUND, the sentence which reads " . . . pitch ranges from 0 to 16383, with the larger values corresponding to higher pitches" should be changed to read " . . . pitch ranges from 0 to 16383, with the larger values corresponding to lower pitches."

Under the title Printer I/O on page 122, the operation for keyword LPOS is incorrect. It should be changed to read:

Keyword	Operation
LPOS	Returns the column position of the print head.

Model II/12/16

MEDICAL OFFICE SYSTEM (26-4508)

The backup procedure on page 21 of the manual should include the following note.

Note: It is recommended that you make a backup copy of the program disk as well as your data disk daily. Refer to the backup instructions on how to backup your working master diskette.

Peripherals

DMP-200 OPERATION MANUAL (26-1254)

On page 9 under Single-Sheet Paper Loading number 6 should be changed to read:

6. Move the Platen Pressure Lever forward and align the paper. Push the Platen Pressure Lever back.

Number 7 should be changed to read:

7. Set the Print Head Control Lever to the appropriate position.

On page 24 in the second row of the middle column entitled Or Send a CHR$( ): the numbers should be 27 23 not 27 21 as shown.

Notes Previous

FEBRUARY 1982

Scientific Notation and PRINT USING on the Extended BASIC Color Computer

The following has been brought to our attention. When using scientific notation with the PRINT USING statement where the exponent is larger than nine it appears that the exponent is displayed in base 12 with : and ; being displayed for the digits 10 and 11. Thus "1.1E10" is displayed as "1.1E+0:". In response to this we would like to reiterate what appeared in the February 1982 issue of *TRS-80 Microcomputer News* on page 37.

"When using the PRINT USING function in Extended BASIC along with the exponential function, you are limited to exponents of nine or less. If you try to use an exponent of 10 or larger, the resulting answer will not be correct. My suggestion here is to not use PRINT USING when using exponents and if you must, limit them (exponents) to less than 10."

Advanced PRINT USING

Ben H. Nation
P.O. Box #391
Fairfield, IL 62837

My version of Advanced Print Using by Johnny Bond upgrades it from a birth date to an address-birth date-phone record.

```
10 ' ADVANCED PRINT USING by Johnny  Bond, P. 21, Feb
     82 MICROCOMPUTER NEWS as revised by Ben H.
     Nation, 26 April 83
20 CLS
30 S$="NAME: %                    %  STREET: %
     %  CITY: %                 % STATE: %
     % ZIP: %     %  BIRTHDAY:  ##/##/##  TELEPHONE:
     (###) ###/####"
40 INPUT "NAME "; N$
50 INPUT "ADDRESS"; A$
60 INPUT "CITY"; C$
70 INPUT "STATE (2 LETTERS)"; E$
80 INPUT "ZIP (5 DIGITS)" Z$
90 INPUT "YEAR OF BIRTH (2 DIGITS "; YB
100 INPUT "MONTH OF BIRTH (NUMBER)"; MB
110 INPUT "DAY OF BIRTH "; DB
120 INPUT "AREA CODE"; AC
130 INPUT "PHONE NUMBER (FORMAT #######) "; PN
140 T1 = INT(PN/10000)
150 T2 = PN-(T1*10000)
160 PRINT
170 PRINT USING S$; N$, A$, C$, E$, Z$, MB, DB, YB,
     AC, T1, T2
180 PRINT
190 END
```

Could someone please tell me how to print "on-screen" information from the Model III onto the DMP-200 without CMD"Z","ON"?

Editor's Note: Press (SHIFT) (1) (*) *simultaneously to do a screen print from the Model III.*

MAY 1982

Ultra Precision Multiplication

Ben H. Nation
P.O. Box 391
Fairfield, IL 62837

We changed line 50 to read:

```
40 CLEAR 1000: DIM T!, T1!, A$, OS!, DP!, B$, P!, H$,
     H2$, H1$, H3$, D1!, D2!, DD!, A!(600)
```

This decreased the computation time immensely! Try multiplying the numbers 1,000,000 × 9,999,999 before and after making the change and notice the tremendous increase in speed.

OCTOBER 1982

Mailing Labels for the CoCo

Larry Allen
207 E. Clark St.
Jonesville, IN 47665

It was with much interest that I read David Banaszak's article on CoCo mailing labels and the subsequent interchanges in *Microcomputer News*. Perhaps my experience with the problem and my solutions might be helpful.

In addition to my personal needs, I handle the mailing lists of several volunteer organizations. At first I wrote job-specific programs tailored to each need. This had the advantage of using little memory and would load quickly. If the parameters were changed, the program could be changed with little effort and even saved if necessary.

Soon, however, my desk was overflowing with program tapes (each slightly different) as well as data tapes. And the task of finding the correct program, loading it, checking it, then loading the correct data tape, updating (small organizations are notorious for monthly updates), printing, then saving new data (and if necessary, new program) led me to consider if there might not be a better way.

My first thought was Colorfile, a multi-talented ROM cartridge I have found very useful. With its lightning fast machine sorts, variable parameters, etc., I am finding new uses for it regularly. However, as I soon discovered there seems to be no way to change the parameters once you have built a file. And after trying for several months to figure a way to get it to print on 9-line labeled stock supplied by one of the organizations, I was ready for another idea.

The answer was not only simple, it added many capabilities—visual search, easy editing, etc., without tying up a lot of

memory with a custom program. I just use the Color Scripsit ROM cartridge. Data is entered so:

John Doe
2131 Amber Drive
Justso, MO 12345
Bill Smith
1212 Will Circle
Wilson, IA 54321

Then by setting the parameters (margins, page length, blank lines) I can use the same file for labels, fanfold postcards or anything else I can run through the printer. The parameters are saved with the data. So for labels it is a simple matter of load and print.

I still use Colorfile for many purposes, but for small multi-use mailing lists that get much editing, Scripsit is the easiest I have found.

Disk Directory Program

Rex Rivers
President, The Color Computer Club
1205 N. Mosley
Wichita, KS 67214

I have discovered a problem with the Disk Directory program. As software advisor for the Computer Club of Wichita, I have seen several people with the same problem. When you load a color disk program into a computer that does not have a disk, it cannot interpret the tokens for the disk commands. Therefore lines 110, 170, and 340 were not correctly listed. The exclamation point (!) in lines 110 and 170 should be replaced with DSKI$. The exclamation point in line 340 should be replaced with FREE.

Also, the entire program could be replaced by typing in the following line.

```
POKE 111, 254: DIR
```

IRA Account

Richard Ellers
426 Central Pky SE
Warren, Ohio 44483

Listing #1 is a rewriting to conform to Model III arithmetic operations of Harold R. Wright's Pocket Computer IRA calculation program on page 37.

This program produces a year-by-year balance of an IRA account, based on four variables: annual deposit, interest rate, compounding rate, and number of years.

The program ends with a return to start, since this is the kind of program that is run several times at a crack.

Listing #2 prints the final balance only and for the fun of it, I packed the program in a single line (eliminating CLS).

Listing #1.

```
10 CLS
     : CLEAR 10000
     : PRINT@20, "I.R.A. ACCOUNT"
20 INPUT "Annual deposit, interest rate, compounding
     rate, # years       "; X, Y, Z, N
40 PRINT "$" ;X; " deposited @"; Y; "%; compounded";
     Z; "times/year; "N" "years
50 PRINT "YEAR          BALANCE"
60 FOR U = 1 TO N
70 W = (1+Y/(100*Z))[Z*(W+X)
     : T=INT(100*W+.5)/100
80 PRINT U,
     : PRINT USING "$$####,.##          " ;T
90 NEXT U
100 GOTO 20
```

Listing #2

```
10 CLEAR:INPUT "ANNUAL DEPOSIT, INTEREST,COMPOUNDING,
     YEARS"; X, Y, Z, N: FOR U = 1 TO N: W = ((1 +
     Y/(100*Z))[Z)*(W+X):T=INT(100*W+.5)/100:NEXT
     U:PRINTUSING"$$##########,.##"; T: PRINT: GOTO
     10
```

C. Bruce Minturn Jr.
10047 Wimbledon Court
Cincinnati, OH 45242

While reading through the *TRS-80 Microcomputer News* May issue, I came across the article "IRA Account." The program interested me, but it was written for the pocket computer. My computing interest are with the Color Computer. So I converted the program written by Harold R. Wright to run on the CoCo.

The program is used for estimating the value of an I.R.A. account with equal yearly deposits. RUN the program and answer the questions on the display. Interest may be compounded from 1 to 365 times per year. The output shows the value each year up to the number of years requested.

```
100 REM IRA ACCOUNT PROGRAM
110 CLS
     : PRINT "I.R.A. ACCOUNT"
     : PRINT
120 A$ = "$$#,###.## %       %"
     : B$ = "##     $$##,###,##"
130 INPUT "DEPOSIT/YEAR $ "; X
     : INPUT "% INTEREST "; Y
140 INPUT "# COMPOUNDED/YEAR "; Z
     : INPUT "NO. OFYEARS "; V
150 CLS
     : PRINT "I.R.A. ACCOUNT"
160 PRINT USING A$; X; "DEPOSIT"; PRINT Y; "%
     INTEREST"
170 PRINT " COMPOUNDED"; Z; "TIMES PER YEAR"
     : PRINT " FOR"; V; "YEARS"
180 PRINT
     : PRINT "YEAR    BALANCE"
     : PRINT "----------------"
190 W = 0
     : I = Y/(100*Z)
200 FOR U = 1 TO V
210 W = W + X
220 FOR J = 1 TO Z
230 W = (1+I)*W
240 NEXT J
250 T = INT(100 * W + .5)/100
     : PRINT USING B$; U; T
260 NEXT U
270 END
```

MARCH 1983

Renumbering on the Models I and III

Matthew Belmonte
2120 Marlboro Drive
Alexandria, VA 22304

Renumbering on the Models I and III apparently does not run as stated. One Model III user employed the following change to make the program work when the new line numbers are below 256.

Delete NEXT Y in 63998. Above 255 the numbers have the correct increments and order but are too high. These changes may solve the problem.

In line 63996 change A = Y/255 to A = Y-256, and in line 63997 change IF . . . THEN Y = Y-255 to IF . . . THEN Y = Y-256.

APRIL 1983

New Disk Videotex for the Model I/III

Videotex Plus stock number should read Cat. No. 26-1588 instead of Cat. No. 26-2225.

The Dungeons of Daggorath

By Linda Miller

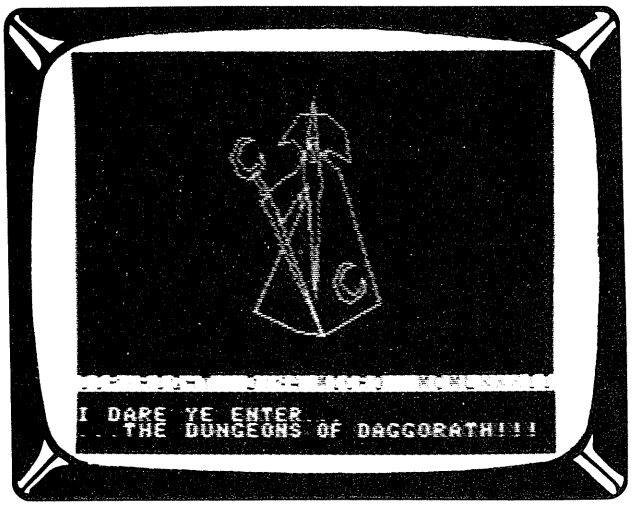
For many years the denizens of Rivenshire lived quietly in the protective shadow of the towering peaks of Daggorath. Then came the Year of Darkness when the crops failed, their cattle wasted away, and their precious children fell ill.

Perplexed by the misfortune that plagued them, the people cried out for an explanation. A wise man of the village came forward and told them of the curse on the land and the evil Wizard who had returned to live in the dungeons deep within the bowels of Daggorath.

Many years before the Wizard had been driven from the Dungeons of Daggorath by a brave adventurer, but even as he fled, the Wizard swore in vengeance that he would someday return with his legion of evil creatures to once more inhabit the Dungeons of Daggorath and wreak havoc on the villagers of Rivenshire. The wise man held up the Ancient Book left by the brave adventurer which explained many but not all the mysteries and dangers of the Dungeons. But where would one bold and wise enough to enter the Dungeons and defeat the Wizard be found?

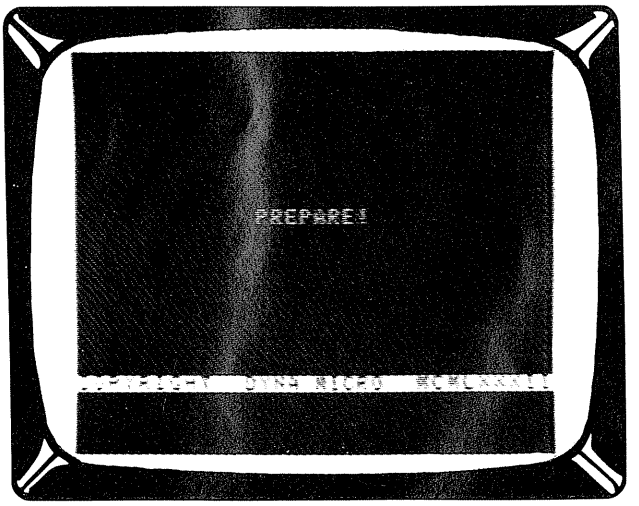
You, dear reader, are that noble adventurer. Only you can save the once peaceful hamlet of Rivenshire from the evil Wizard by entering the Dungeons of Daggorath armed only with a wooden sword, a torch, the knowledge gleaned from the Ancient Book, and your own skill and cunning. It will be your task to gain the necessary strength and wisdom to conquer the evil creatures and ultimately the Wizard who waits deep within.

Many will not return from the dungeons, but someone must persevere until the end or all will be lost and only evil will stalk the days of Rivenshire. As you prepare to enter the Dungeons of Daggorath a chilling view of the Wizard appears as he hurls his challenge "I DARE YE ENTER . . . THE DUNGEONS OF DAGGORATH."



In your possession is a backpack containing a pine torch and a wooden sword. As long as it lasts, the pine torch lights the way in the dungeon, but it will be up to you to find new sources of light. As the light burns low, it becomes more difficult to see and kill the creatures. The wooden sword will be your only weapon against the creatures unless you locate the other objects within the Dungeon to aid you. The creatures of the Dungeon move toward you relentlessly. If you stop to calm your beating heart, they continue onward with only one thought in mind—your total annihilation.

Once inside the Dungeon, there is no retreat. You must defeat all the creatures sent to destroy you in your search for the Wizard. Don't be deceived by feigned friendliness. Some creatures are more easily dispatched than others, but all are deadly. Given the opportunity each creature can and will destroy you!



At first it's quiet in the Dungeon and all you hear is the beating of your own heart. As you move faster your heart beat increases in speed. Take care that you don't expend your energies so quickly that you faint. The creatures of the Dungeon show no pity on unconscious intruders. Fainting can mean instant destruction.

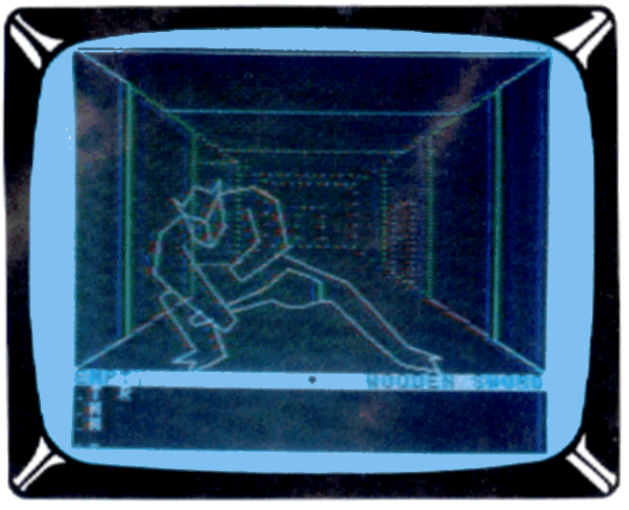
You soon hear other sounds. Vipers, Spiders, Evil Knights, Smiling Blobs, Stone Giants, Scorpions, Wraiths, and Galdrogs all inhabit the Dungeon. Each creature has a distinct sound which enables you to determine what approaches and whether you feel that you have gained sufficient strength to defeat the creature. Retreat is only a temporary solution because the creature will continue to track you until one of you is the victor.

There are weapons, magic rings, potions, and scrolls all hidden in the Dungeon. It will be up to you to find them and

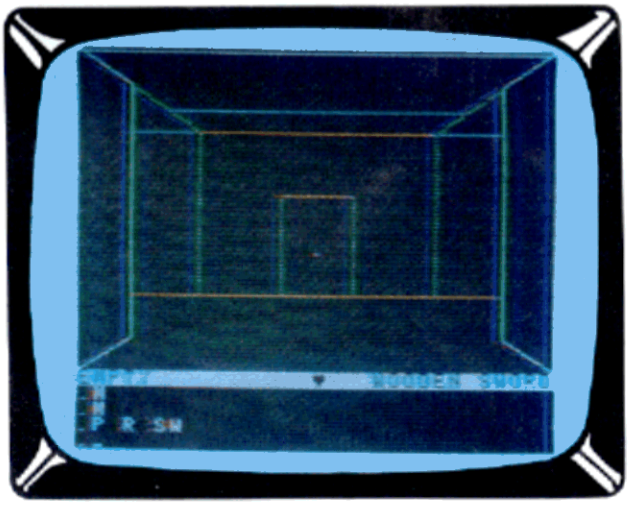
discover any secret powers they may possess. After you conquer the top three levels of the Dungeon, you will move into the lower level (levels?) where the Wizard resides. Each level increases in difficulty, and you will need the strength gained from the previous levels to defeat the increasingly treacherous creatures of the lower levels.

WHAT A THRILLER!

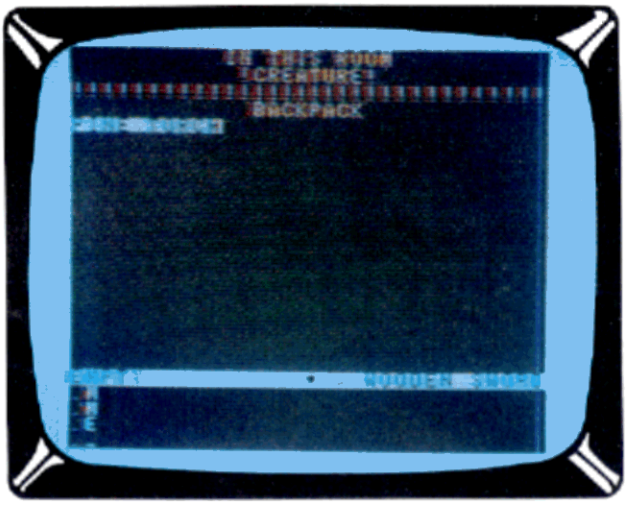
Dungeons of Daggorath challenges the skill of the most ardent adventure or action game enthusiast. It combines sound and graphics in a multi-level maze to provide a fascinating, action packed adventure game. The creatures move about the Dungeon tracking you in real time. Even if you stop, they don't. You can never be sure what order you will meet them from one game to the next.



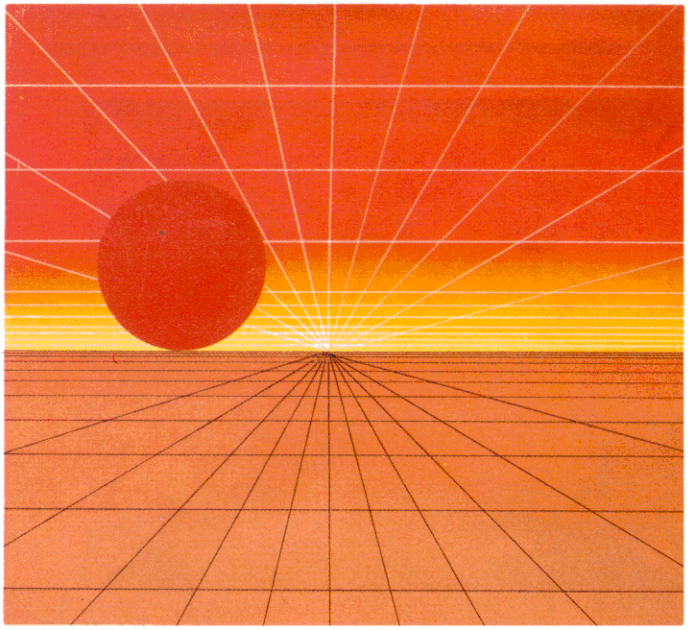
When they're getting close, you can also see the creatures that are advancing toward you as well as hear them. The objects that are in your left and right hand and a pulsing heart are displayed in the band across the lower portion of the screen. The faster you move and the more intense the conflict, the more rapidly the heart pulses on the screen.



Objects found in the Dungeon can be stowed in your backpack for later use. It is possible to look in your backpack to see what you're carrying.



Dungeons of Daggorath (Catalog no. 26-3093, Suggested retail price $29.95) is a Program Pak™ which requires a 16K TRS-80 Color Computer and color television to play. It comes with an instruction manual (also known as the Ancient Book) which gives you all the information necessary to play. It is a challenging game—one which should offer hours (days, weeks, years?) of exhilarating entertainment, but I feel that I must warn you that you can get hooked on the Dungeons of Daggorath.



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

BULK RATE
U.S. POSTAGE
PAID
Radio Shack
A Div. of Tandy Corp.

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

TRS-80® Computer Camp

This summer let Radio Shack's qualified instructors introduce your child to the world of computers at a TRS-80 BASIC or a TRS-80 LOGO Computer Camp. One TRS-80 Color Computer will be available for every two campers to give each participant the opportunity for hands on experience. No prior computer knowledge is required.

TRS-80 LOGO Computer Camp

For Ages 8-11

The Color LOGO Language

Color LOGO provides unique features that are not found in any other LOGO language. There's the "doodle" mode and the fascinating "hatch" command that creates multiple turtles. The result is a LOGO for the TRS-80 Color Computer that will excite young minds and help them welcome the challenge of learning.

Computer Concepts for Young People

Radio Shack's Color LOGO is a revolutionary computer programming language for children. It is designed to help young minds grasp graphic relationships and develop problem-solving skills. Through manipulation of "turtle" graphics, youngsters learn to write simple programs. By doing so, the computer becomes a friendly learning tool, and youngsters gain insight into advanced concepts.

Sessions are Self-Paced . . . And Fun

Each child will progress at his or her own pace under the guidance of an experienced, professional educator. The curriculum, designed by educators, employs a play-oriented approach to encourage children to learn and use the computer.

TRS-80 BASIC Computer Camp

For Ages 12-15

Computer Concepts for Young People

Extended Color BASIC is an exciting and powerful programming language that youngsters can readily appreciate. Dynamic graphics, music and sound effects, and progressive programming techniques let students write simple problem-solving programs or create animated games.

Sessions are Structured And Not Intimidating

The five-day camp will consist of professionally-designed activities to keep the pace interesting and challenging.

Extended Color BASIC Language

Each camper will learn to use such BASIC features as assignment statements, looping, branching, subroutines, READ/DATA statements, editing, string packing, and more to create their own programs.

Introduction to LOGO

As an added bonus, each attendee will be introduced to Radio Shack's Color LOGO programming language. This will give the campers insight into a second computer language.

Radio Shack COMPUTER CENTER

Enroll now at your nearby Radio Shack Computer Center.