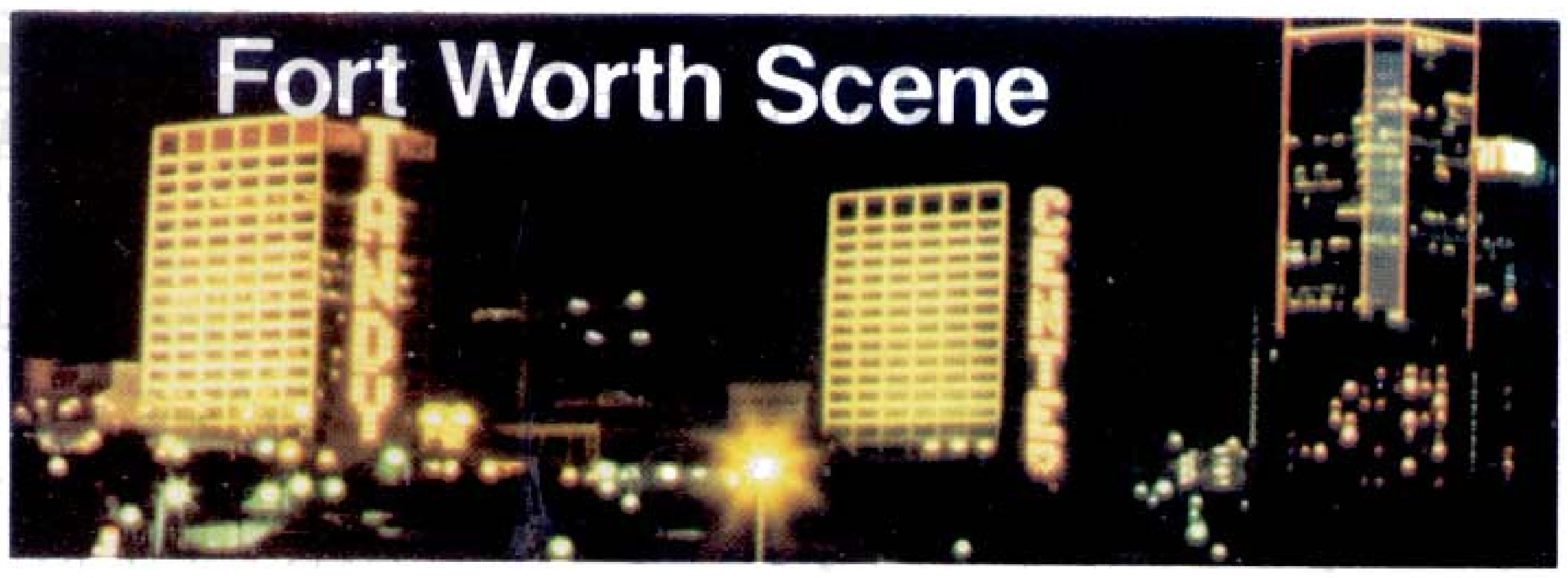
TRS-80® Microcomputer News

Volume 4, Issue 9 October, 1982
Price $1.50
Information Published for TRS-80 Users.

Data Base Issue
Model I Double Density
Color Graphic Printer
Micro Research



Fort Worth Scene

In addition to the software and hardware necessary to make your TRS-80® computer operate like a terminal, Radio Shack now offers three separate terminals which are available at Radio Shack Computer Centers or the expanded computer departments of selected Radio Shack stores and dealers.

VIDEOTEX

The Videotex terminal requires the addition of a phone line and television to enable you to access national information networks like Dow Jones and CompuServe. It is the least expensive of the three Radio Shack terminals (16K for $499.00).

PT-210 PORTABLE DATA TERMINAL

The new TRS-80 PT-210 Portable Data Terminal (76-1001, $995.00) incorporates a full "typewriter" keyboard, a quiet thermal printer and a 110/300 baud (Bell 103A compatible) acoustic telephone coupler in a compact, attractive, portable package.

It provides exceptional value for applications where hard copy of in-computer information or information provided by Videotex or other on-line services is needed. An optional add-on RS-232-C Interface Module (76-1002) for $69.95 can be user (plug-in) installed. This facilitates use of the terminal as a local (to a computer or selected peripheral) "front end" dumb terminal and/or printer. RS-232-C is also a convenient way to connect the terminal to a direct connect modem.

The PT-210 features a full-size ASCII keyboard, generating a total of 99 codes. 110 baud or 300 baud operation may be switch selected, as may half-duplex or full-duplex operation and odd-parity/even-parity/no-parity modes.

Its quiet non-impact thermal printer uses a 35-element (5 × 7) matrix and offers variable contrast control. 71 characters are printable, with lower case letters automatically printed as their upper case equivalents. Each 8-inch line can include up to 80 characters (10 characters per inch), and carriage return is automatic at the 81st column on any line. Printing speed is 50 characters per second with 6 lines per vertical inch on 100-foot rolls of 8½-inch-wide thermal paper.

The PT-210 is housed in a sleek silver-grey case measuring 14½ × 14½ × 5 inches and weighing 15 pounds with paper installed. It is FCC registered and UL-listed for 120 VAC 60 Hertz operation. The line cord is detachable for easy portability.

DATA TERMINAL 1

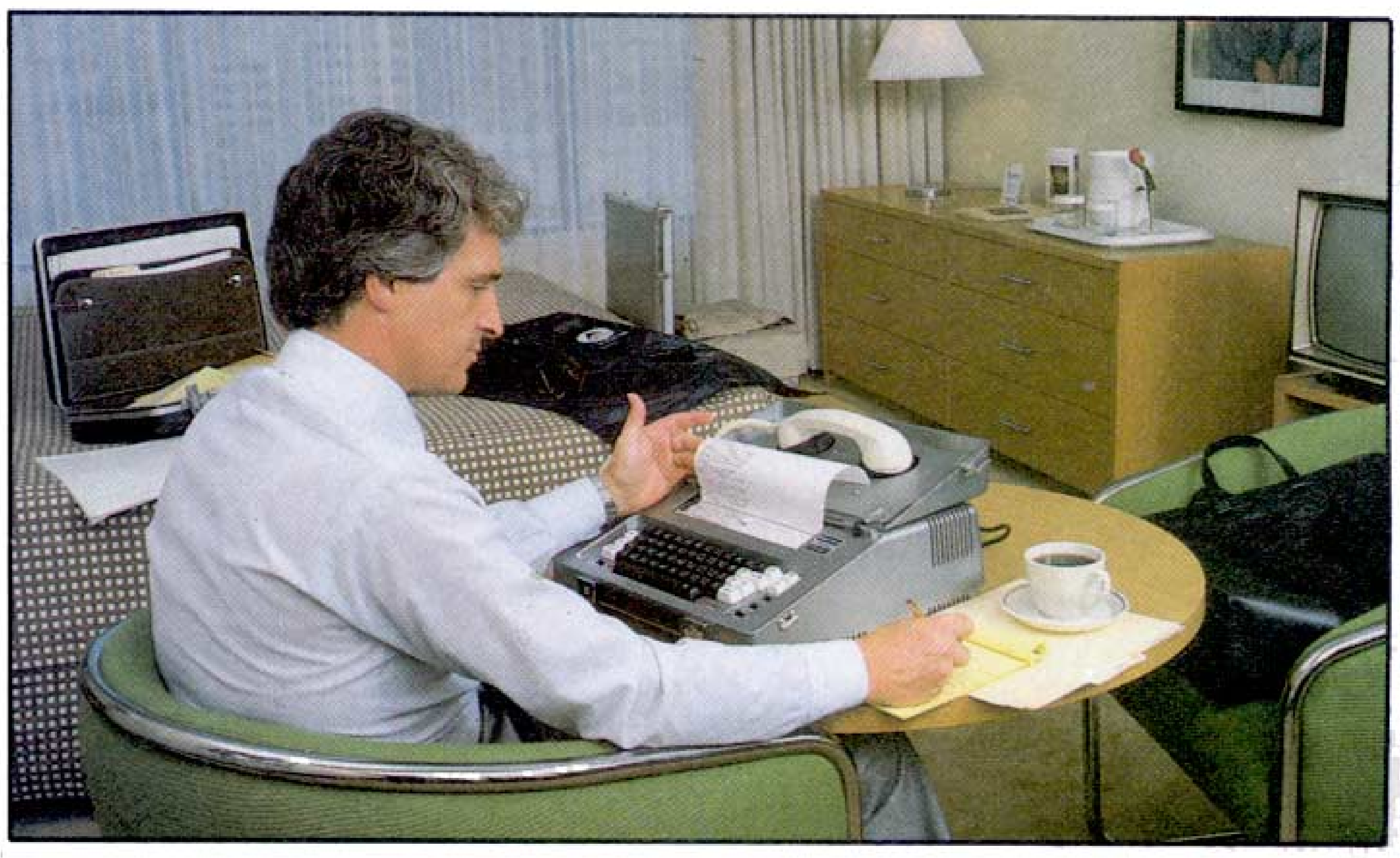
Radio Shack also recently announced TRS-80 DT-1 (26-6050, $699.00) which is an advanced video data terminal. The DT-1 can be configured to emulate any of four popular terminals: the Televideo 910, Lear-Siegler ADM-5, ADDS 25 or Hazeltine 1410. Its configuration is keyboard selectable (other multiple-protocol terminals use tiny DIP-switches for this selection which is often an awkward procedure). The selected protocol is maintained independently of power through the use of a breakthrough in memory technology: the EEPROM (Electrically Erasable Programmable Read-Only Memory).

The custom-programmed, high speed 8-bit microprocessor inside the DT-1 is the basis of its high performance, flexibility, and low price.

The legibility and flexibility of the DT-1 display are remarkable. The 12-inch (diagonal measure) CRT provides a full 80 character by 24 line (80 × 24) video display. Upper- (shift key) and lower-case characters are available, and control characters can be displayed without using special symbols. A variety of video attributes are available, including normal, reverse, invisible, blinking, underlined, and half-intensity video. And four types of cursors may be selected from the keyboard—steady or blinking, block or underline. The DT-1's 70-key keyboard has been designed for smooth, convenient operation with full, ergonomic keyboard "feel". It includes the full set of special keys used by the TRS-80 Model II and Model 16 computers.

In addition to a standard typewriter-style keyboard area, the DT-1 has a 12-key data pad which functions in three ways. It can be used calculator-style for quick entry of numbers and the ENTER command; it can be used with the SHIFT key for cursor up, cursor down, cursor left, cursor right and home commands, plus special characters; and it can be used with the CONTROL key for control code commands. Any keystroke can be repeated automatically by holding down the desired key; after the first several repeats, the DT-1 will automatically repeat at a faster rate.

The TRS-80 DT-1 is equipped with both a built-in RS-232-C serial interface and a Radio Shack printer-compatible parallel interface, either of which may be used as a printer port for hard copy output. A local monitor mode can be used to enable printer ports and to position the video cursor using Local Control and Escape modes. The CARRIAGE RETURN key can be programmed to operate with or without sending a Line feed command. CONTROL key termination codes are also user-programmable. Ten keyboard selectable communications rates from 75 to 19,200 baud are available. All keyboard selectable options are stored in the EEPROM, and they may be changed or temporarily overridden from the keyboard at any time. An electronic bell is standard. The DT-1 is U.L. listed for 120 VAC 60 Hz operation; a 50 Hz option is also available.



The very portable Radio Shack PT-210—the newest in our line of data terminals—includes many outstanding features.

Volume 4, Issue 9 **October, 1982**

Price $1 50

TRS-80® Microcomputer News

Information Published for TRS-80 Users.

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

The subscription rate for renewals and other interested persons with U S. APO or FPO addresses is twelve dollars ($12 00) per year, check or money order. Single copies of the Microcomputer News may be purchased from Radio Shack Computer Centers or Computer Departments for $1 50 suggested retail each The subscription rate for renewals and other interested persons with Canadian addresses is Fifteen dollars ($15 00) per year, check or money order in U.S funds. All correspondence related to subscriptions should be sent to: Microcomputer News, P.O Box 2910, Fort Worth, Texas 76113-2910.

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

Back issues of Microcomputer News prior to January, 1981 are available through your local Radio Shack store as stock number 26-2115 (Suggested Retail Price $4 95 for the set) Back issues of 1981 copies are not available.

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

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

Material may be submitted by mail to P O. Box 2910, Fort Worth, Texas 76113-2910, or through CompuServe. The Microcomputer News' CompuServe user ID number is 70007.535.

Notes to Program Users:

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

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

Trademark Credits

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

Contents:

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

Notes on Previous Newsletters

JUNE 1981
SAFE

In the SAFE program for the instructions indicate that the Left joystick will be used when the right joystick is assigned in line 110 A=JOYSTK(0). Change the word left to right in line 50.

DECEMBER 1981
PATCH and SPELLING CHECKER

Gilbert & Kathy Keith
Route 1, Box 285
Rineyville, KY 40162

It is with reading your article on page 31, PATCH, that really gave me the idea that possibly I should do some kidding about your article on page 30. On page 31, THE PATCH states that a vocabulary has surfaced that discovers the more serious types of bugs, such words as: bomb, blow up, silent death, hang, crash, limbo. I would like to add a few more of my own to this most serious type of problem, such as: little goof, goof, big goof, and big splash.

On page 30 SPELLING CHECKER, whoever wrote this article made a big splash since the dictionary for Model III is not out yet, I cannot speak for it, but I can speak for the dictionary for the Model II. The statement that was made in the article was "and it will be checked for spelling errors at about 2000 words per minute." The most that we have ever been able to check is about 800 to 900 words per minute. This is with SCRIPSIT, Version 2.0 and the new dictionary.

The reason for bringing this to your attention is that we are proud owners of both the Model II and the Model III computers, and we do not want erroneous information put out that would mislead the public.

We are your supporters, and we do not want you to take this as a personal affront. Our Radio Shack computers hold a warm spot in our heart, and we appreciate the good work that you are doing.

Editor's Note: I just spell checked a document containing 2757 words. There were 99 possible errors in the file, and dictionary took 1 minute 33 seconds to go through the file. That figures out to 1779 words/min. The speed depends on a lot of factors. I have had files checked at 2000+ word/minute, but the average is somewhat slower, as the Keiths point out.

JANUARY 1982
ONE-ARMED DIME BANDIT

Newell H. Claudy
1811 Filbert Way, Apt. A
Elkhart, IN 46514

This program suggests a clever application of the Pocket Computer. Unfortunately, the program won't run correctly the way it is listed.

Variables R,S, and T must be reset to zero each time before entering the FOR-NEXT loop. Otherwise, values accumulated in line 95 will be meaningless. The necessary correction to the program can conveniently be made by adding to line 50, resulting in:

```
5Ø M=M-.1
    : V=Ø
    : R=Ø
    : S=Ø
    : T=Ø
```

Delete the expressions R=0, S=0, and T=0 in line 30, where they function only the first time the program is run.

The display is esthetically more attractive if the value of A$(30) specified in line 20 is changed to have five dots, rather than three, corresponding to the five letter since the words "APPLE", "GRAPE", and "LEMON". This keeps spacing of the display the same while the "drums spin."

One final comment. To be fair to the player, the payback for a win should be 90 cents instead of 60 cents or M=M+.9 in line 130. The true probability of winning is 1/9, so out of nine tries (on the average) the player should spend 90 cents to play, getting back 90 cents on one "spin." The 60 cent payback yields a "house" percentage of 33.3% (the "house" keeps 30 cents out of each 90 cents the player spends). This one-armed bandit lives up to its name!

Donald Parson
Apt. 43-C
4475 N. Ocean Blvd.
Delray Beach, FL 33444

Mr. Bond is to be congratulated on his ingenious and elegant program.

I did however find that it was paying off when it shouldn't and vice versa.

I fiddled around for awhile and finally made the following changes:

```
95   Delete
125   Q=(H=J)+(H=K)+(J=K)
13Ø   If Q=3 etc.
```

Line 125 now tests directly for 'three-of-a-kind'.

SHOOT 'EM AGAIN

Levin J. Muth
700 First Avenue
Harvey, LA 70058

I am dropping this line to make minor corrections to J.W. Myers article.

Change

```
11Ø OT=T
```

Delete

```
112
```

Add

```
384 If W=A6 THEN 5ØØ
```

LINES

Mrs. Fern Christensen
Northwestern State University of Louisiana
1017 Oma Street
Natchitoches, LA 71457

Several have written about making lines, as teachers are so inclined to do for test making. My favorite is to have tabulation "t" set at appropriate spacing (i.e. such as 10 or 15 spaces). Use F1 SHIFT HYPHEN and tabulate or arrow to the place chosen to stop, SHIFT HYPHEN, and you will have your line drawn for a student to fill in "with the correct answer."

The 6809 Divides and Conquers

by William Barden, Jr.

One of the problems with assembly language is that it's even more basic than BASIC. In BASIC you can compute numbers with relative ease, spewing integers, fractions, and exponents all over your listings. The BASIC interpreter handles adds, subtracts, multiplies, and divides with ease, and one takes the computing power for granted.

THE 6809E MULTIPLY

Not so with assembly language. All arithmetic operations are done at rudimentary levels, and you're lucky to be able to handle adds and subtracts of even 16-bit integers. A case in point: The 6809E in the CoCo has several types of adds, several types of subtracts, and a multiply. Just having the multiply instruction alone is a terrific selling point for the 6809E microprocessor—its predecessor, the 6800, had no multiply, and neither did comparable microprocessors such as the 8080, Z-80 (Mods I, II, III), and the 6502 (Apple II, III, Vic 20).

Now that we have a built-in multiply in the 6809E, we do not have to write a "software" multiply. Software multiplies typically take 100 times longer than a hardware multiply. The 6809E MUL instruction multiplies the contents of the A and B registers to get a 16-bit product (with the result going into the D register, A and B combined) in 11 machine cycles, or about 12.4 microseconds. A comparable software multiply would take about 150 microseconds.

NOW FOR THE DIV INSTRUCTION . . .

What about the DIV instruction? Let's see DAA, DEC, EOR . . . What! No divide? Ha, ha . . . They must have made a mistake and left it off of the Instruction Set Summary "crib card". Surely it's in the EDTASM + manual . . . .

Don't bother looking any further! There is no DIVide. If you want a divide, you'll have to implement it yourself. In this column we'll look at how we can write a software divide in lieu of a built-in DIV and see how closely we can come to that MUL instruction time of 12.4 microseconds. You may be surprised. Or appalled . . . .

WHAT TYPE OF DIVIDE?

First of all, consider what type of divide should be implemented. The MUL is an "8 by 8" multiply. It takes two 8-bit operands, multiplies them, and puts the result into the D register. (If you're not familiar with the 6809E registers, you'd better go back to earlier columns, or the EDTASM + manual.)

Of course, in 8 bits we can hold values from 0000000 (0) through 11111111 (255). The product of the MUL will be 0000000000000000 through 1111111 00000001 ($FE01 or 65025 or 255 times 255).

Note that the multiply is an "unsigned" multiply. The two operands are considered unsigned numbers, rather than "two's complement" numbers. This is typical for rudimentary multiplies, as we can always test operands for their sign, take the absolute value, multiply, and then negate them for the proper product if necessary.

The MUL could have been a "16 by 8", or a "16 by 16", but larger multiplies can be built from the basic "8 by 8", and the "8 by 8" is convenient in terms of cpu register sizes.

The reason I'm going into detail on the MUL is that it seems reasonable to implement a divide that is somewhat equivalent to the MUL in terms of size of operands. Let's see, if we divided an 8-bit operand by another 8 bit operand, we couldn't do much. How about a "16 by 8" divide, dividing a 16-bit operand by an 8 bit operand? That seems reasonable. It would allow us to hold a maximum dividend of up to 65,535 (1111111111111111) and work with a maximum divisor of 255 (11111111).

Like the MUL, the divide should probably be an "unsigned" divide; we can test the dividend and divisor for sign initially, take the absolute value, divide, and then negate the quotient if necessary. Are there any other considerations? One thing that we have with the DIV that we didn't have in the MUL is a "second" result. We not only have a quotient, we have a "remainder", representing the fractional portion of the result. (Don't forget that the MUL and DIV will operate on "integer" values and not mixed numbers.) It would be nice to make some provision for retaining this remainder, if it's convenient.

What can we say about the sizes of the quotient and remainder? It appears that the maximum quotient will be 16 bits, as 65,535 divided by 1 is a legitimate division. The maximum remainder will be 254, produced when the divisor is 255 for various dividends. We can therefore hold the remainder in 8 bits.

Next question: Now that we have the divide "roughed out", how do we implement it?

WHICH REGISTERS SHALL WE USE?

The first order of business is to decide where to hold the operands. We could put them in RAM somewhere, but why not take advantage of the 6809E registers? In general, if we can hold operands in registers, we'll be able to speed up the program. Let's see, we have a somewhat limited choice of registers . . . Let's assume that the dividend is in the X register and that the divisor is in the A register on entry to the divide subroutine. On exit from the subroutine, we'll put the 16-bit quotient in X and the 8-bit remainder in the A register. The dividend and divisor will be destroyed, and we'll leave it up to the "calling program" to save them if necessary.

We now have the scheme shown in Figure 1. The DIV subroutine will be a machine-language program "callable" by other assembly-language code or by BASIC through the USR call.

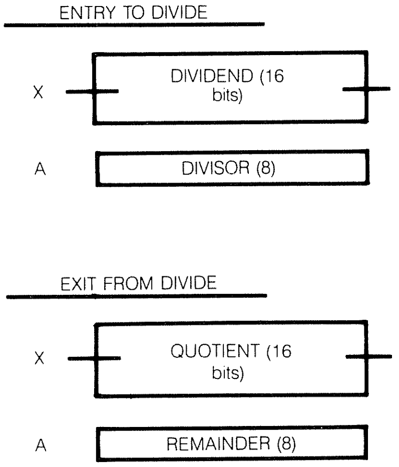


Figure 1.

WHAT'S THE ALGORITHM?

That was the easy part. The next question is: How do we actually do the divide?

There are some standard ways of doing software divides, developed over a number of years, so don't feel bad if obvious algorithms don't immediately spring to mind.

One algorithm we could use is to do "successive subtraction". In this method of division, the divisor is subtracted from the dividend until the result "goes negative". The quotient is the number of times that the divisor could be subtracted from the dividend before the result went below 0. The remainder is the "residue" on the last subtract. An example is shown in Figure 2.

```
                   DIVIDE 100/23
                                  QUOTIENT   REMAINDER
                          100        0
                          -23
                           77        1
                          -23
                           54        2
                          -23
                           31        3
                          -23
                            8        4
RESIDUE                   -23
 "WENT           -->      -15        4
NEGATIVE"                 +23
  HERE                      8                    8
```

Figure 2.

A divide by this "successive subtraction" is shown in Figure 3. The actual divide subroutine is in lines 160 through 370. The preceding code is a short "driver" routine to interface to BASIC. Let's look at the actual divide first.

```
0000 BE   3FF0     00100 DRIVER  LDX     $3FF0   GET DIVIDEND
0003 B6   3FF2     00110         LDA     $3FF2   GET DIVISOR
0006 8D   07       00120         BSR     DIVRSB  DO DIVIDE
0008 BF   3FF3     00130         STX     $3FF3   STORE QUOTIENT
000B B7   3FF5     00140         STA     $3FF5   STORE REMAINDER
000E 39            00150         RTS             RETURN
                   00160 ***********************************************
                   00170 * UNSIGNED DIVIDE. DIVIDES 16 BIT UNSIGNED NUM- *
                   00180 * BER BY 8 BIT UNSIGNED NUMBER. DIVIDE BY 0 NOT *
                   00190 * ALLOWED, AS IT WILL PRODUCE SAME RESULT AS    *
                   00200 * $FFFF/$01. QUOTIENT WILL BE $0000 - $FFFF.    *
                   00210 * REMAINDER WILL BE $00 - $FE (AS IN $00FE/$FF).*
                   00220 *     ENTRY:  X=DIVIDEND, $0000 - $FFFF         *
                   00230 *             A=DIVISOR, $01 - $FF              *
                   00240 *     EXIT:   X=QUOTIENT, $0000 - $FFFF         *
                   00250 *             A=REMAINDER, $00 - $FE            *
                   00260 ***********************************************
000F 34   02       00270 DIVRSB  PSHS    A       SAVE DIVISOR
0011 1F   10       00280         TFR     X,D     DIVIDEND NOW IN D (A,B)
0013 8E   FFFF     00290         LDX     #-1     INITIALIZE QUOTIENT
0016 30   01       00300 DIV010  LEAX    +1,X      BUMP QOUTIENT
0018 E0   E4       00310         SUBB    ,S        SUBTRACT DIVISOR
001A 82   00       00320         SBCA    #0        NOW ANY BORROW
001C 24   F8       00330         BCC     DIV010    GO IF RESULT + .
001E EB   E4       00340         ADDB    ,S      ADD BACK TO GET REMAINDER
0020 1F   98       00350         TFR     B,A     REMAINDER
0022 32   61       00360         LEAS    +1,S    RESET STACK
0024 39            00370         RTS             RETURN
          0000     00380         END
00000 TOTAL ERRORS

DIV010  0016
DIVRSB  000F
DRIVER  0000
```

Figure 3.

Coming into the DIVRSB subroutine, we have the dividend in X and the divisor in A. The PSHS A instruction transfers the divisor to the "stack". The dividend in X is now moved to the D register (A and B combined). The quotient will be held in X at the end, and X is loaded with -1 as an initial value.

We're now at DIV010. Lines 300 through 330 make up the loop for the successive subtract. For every "iteration" through the loop we do the following:

- Bump quotient in X by 1 (goes to 0 on first time)
- Subtract the divisor in the stack from B
- Subtract a possible "borrow" from A
- Branch back to the beginning if the result is not below 0

Some of these operations need explanation:

The divisor was stored in the stack and is accessed by "S", which says use the stack pointer register S as an index register to get the operand to be subtracted from the contents of the B register. The stack pointer always points to the last byte stored, and the stack, of course, "builds down" in memory. Why didn't we store the divisor in another cpu register? We only have Y left, and we can't use it to hold an operand for the subtract.

The subtract was an 8 bit subtract. The "residue", the result of each successive subtract, is held in D, which is A and B together, with A holding the most significant 8 bits, and with B holding the least significant 8 bits. As the divisor is only 8 bits, we can do the subtract by subtracting the divisor from B and then subtracting a possible "borrow" from the A register. This is done by doing a "subtract with carry (borrow) with an immediate operand of 0 (SBCA #0), which subtracts only a (possible) borrow from A. In other words, only 0 or 1 is subtracted from the upper 8 bits of the residue.

If the residue after the SBCA goes below 0, the carry bit in the condition codes will be set to a 1, and the loop ends. At this point an add of the divisor must be done to restore the proper remainder, and this is done by ADDB ,S. The remainder in B is then transferred to the A register. The X register holds the proper quotient. The LEAS +1,S resets the stack pointer by adding 1 to its contents, causing it to bypass the divisor byte and point back to the return address for the RTS.

The driver portion of this subroutine is an interface to BASIC. It assumes that the dividend has been stored in locations $3FF0 and $3FF1, and the divisor has been stored in location $3FF2. It then calls DIVRSB and on return stores the quotient in locations $3FF3 and $3FF4 and the remainder in location $3FF5.

The entire code for DIVRSB is "position independent", or "relocatable". It can be put anywhere in RAM and run properly, as long as the dividend and divisor are in the $3FFx area.

Figure 4 shows the code incorporated as BASIC program DATA values and moved to the $3F00 area. An excellent way to do this is as follows: Assemble the program using the EDTASM+ (Don't have one? Get one as an indispensable tool for assembly language, and I'm not just helping Radio Shack sell ROM packs either. EDTASM+ is a subset of the excellent EDTASM+ for the Model I and allows in-memory assembly, "symbolic" debugging, and a high degree of interactiveness. More on EDTASM+ in another column . . .)

```
100 DATA 190,63,240,182,63,242,141,7,191,63
110 DATA 243,183,63,245,57,52,2,31,16,142
120 DATA 255,255,48,1,224,228,130,Ø,36,248
130 DATA 235,228,31,152,5Ø,97,57
140 FOR I=&H3FØØ TO &H3F24
150 READ A: POKE I,A
160 NEXT I
170 DEFUSRØ=&H3FØØ
180 INPUT DD,DV
190 POKE &H3FFØ,INT(DD/256)
200 POKE &H3FF1,DD-INT(DD/256)*256
210 POKE &H3FF2,DV
220 A=USRØ(Ø)
230 PRINT DD;"/";DV; "S/B";INT(DD/DV);"REMAINDER";DD-INT(DD/DV)*DV
240 PRINT "WAS ";PEEK(&H3FF3)*256+PEEK(&H3FF4);" WITH REMAINDER ";PEEK(&H3FF5)
250 GOTO 180
260 FOR I=&H3FØØ TO &H3F26
270 PRINT HEX$(PEEK(I)),
280 NEXT I
```

Figure 4.

When you've assembled, check to see that all code is position independent. (Use BSRs and LBSRs in lieu of JMPs or JSRs.) Now assemble in memory and then go to Zbug by entering the "Z" response.

Once in Zbug, enter"O10" for decimal "output", "byte" mode ("B"), and numeric ("N"). Now display your program from the program start and record the decimal values (designated by a "T" suffix).

Use those decimal values in DATA statements in the BASIC program. Use code similar to Figure 4 to move the values to a protected area (by an initial "CLEAR"). Before execution, dump the area in BASIC by using a PRINT HEX$(PEEK(I)) command. Compare the values from the print with your original assembly-language listing. If they compare exactly, you have successfully incorporated the machine code and relocated it in a BASIC program.

To use this Extended Color BASIC program, CLEAR 200, &H3EFF to protect the $3F00 area and execute. The BASIC program will then relocate the program and ask you for a dividend and divisor. It will then call the DIVRSB program and compare the expected results with the actual.

Try the program with various values, and you'll see that it works for any legitimate operands.

A VOICE FROM THE AUDIENCE . . . "HOW FAST *IS* IT?"

That's an interesting question. If you think about it, you'll see that the speed of DIVRSB is related to the number of subtracts. Discounting "overhead", the speed of 10000/100 should be about 100 times faster than 10000/1, as 100 times the number of subtracts need to be done for the latter case. Can we measure the actual speed?

We could go through and calculate the speed based on the number of "cpu cycles" from Motorola documentation, but here's an easier way. Figure 5 shows a short BASIC program to POKE the operands and then call DIVRSB. It assumes that the program has been relocated to the $3F00 area. The program as shown POKEs a "57" into location $3FF6. A 57 is the "op code" value for an RTS, or Return From Subroutine instruction.

```
2ØØØ DEFUSRØ=&H3FF6
2Ø1Ø POKE &H3FFØ,3
2Ø2Ø POKE &H3FF1,&HE8
2Ø3Ø POKE &H3FF2,1ØØ
2Ø4Ø POKE &H3FF6,57
2Ø5Ø FOR I=1 TO 1ØØØØ
2Ø6Ø A=USRØ(Ø)
2Ø7Ø NEXT I
```

Figure 5.

As it stands, the program does 10,000 calls to the USR subroutine at location $3FF6, which is simply an RTS. If we time the execution of this program, we'll be able to measure the BASIC "overhead" before measuring the actual speed of DIVRSB. It turns out that this program takes about 55.5 seconds to go through 10,000 calls.

Now, substitute "&H3F00" for the "&H3FF6" in the DEFUSR and measure the time for 10000 iterations. The POKEs in the program setup a dividend of 1000 and a divisor of 100, so we'd have 10 subtracts using the successive subtraction method. If you measure it, you'll find that the divide takes about 58.1 seconds for 10,000 iterations (that's 10,000 divides of 1000 by 100). Subtracting 55.5 seconds (the BASIC "overhead" time from 58.1 seconds gives about 2.6 seconds to execute 10,000 divides in the subroutine itself, or about 260 microseconds per divide. The 260 microseconds is about 20 times more slow than the equivalent hardware multiply! Wait, it gets worse . . . .

If you modify the BASIC code at 2000 to POKE different values for the dividend and divisor, you'll find that a divide of 1000 by 10 takes about 1660 microseconds per divide! What is the absolute worst case? This occurs when 65,535 is divided by 1 and 65,535 subtracts are done, and the time required is about 95,000 microseconds or about 1/10 second! And you thought assembly language was fast!

This algorithm for division is therefore not too efficient, and Microsoft probably will not pick it up for inclusion into BASIC interpreter software. If the average quotient is 255 (statisticians may prove me wrong), the average division speed will be about 3 milliseconds, or about 333 divides per second. Can't we do better? Although assembly-language code doesn't have to be "tight" in most programs, it does have to be fast and efficient for commonly used operations such as divides.

ANOTHER ALGORITHM

Is there a faster algorithm that we can use for a "16 by 8" divide, or for that matter, for any divide? No, next question . . . .

Just kidding. One that has been used for many years is the "restoring" divide. It emulates a paper and pencil divide. (Hardware designers sometimes use a "non-restoring" divide to speed up hardware divides in larger microprocessors or computers.)

The restoring divide works on paper as shown in Figure 6. The divisor is subtracted from the first group of digits. If the divide "doesn't go", then the subtract is not done and a shift to the next digit position is done, where the process is repeated.

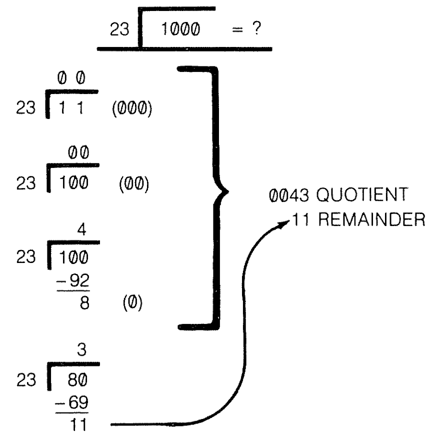


Figure 6.

We can do the same type of thing in the CoCo in binary. (See Figure 7.) Here, however, the computer can't inspect the numbers and tell that the subtract can't be accomplished. In this case the subtract is done, and if a negative result occurs the divisor is added back to the "residue". If the residue is still positive or 0 (0 is a positive number in the 6809E and other microprocessors), then the residue is not "restored" and a 1 bit is put into the quotient.

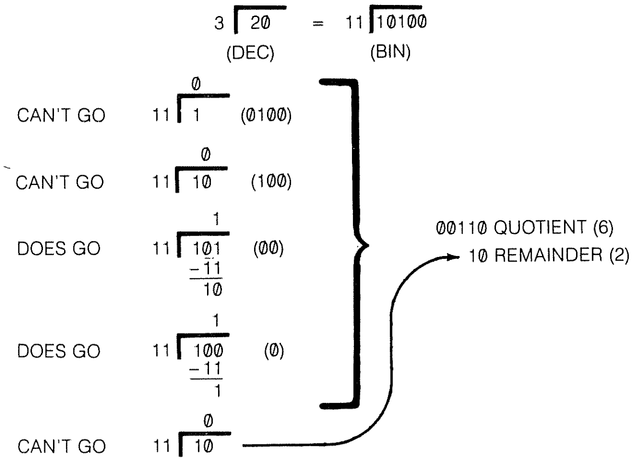


Figure 7.

As you can see from the figure, the operation is essentially a subtract, restore if necessary, put a 0 or 1 bit into the quotient, and shift the residue. Even within this method, there are a number of ways to do the divide. Either the divisor or residue may be shifted. We'll choose to shift the residue, as shown in Figure 8.

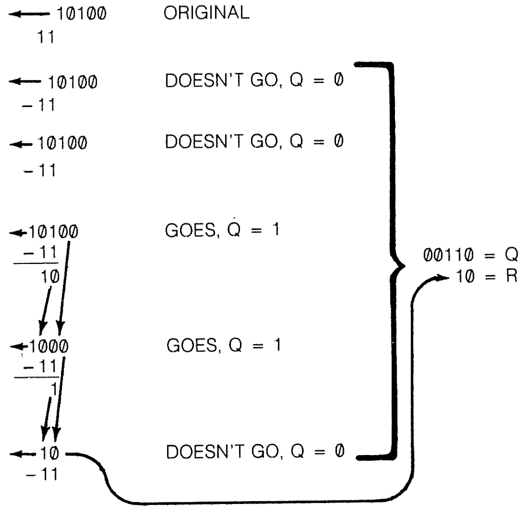


Figure 8.

Figure 9 shows the register setup for the DIV168, or the "16 by 8" divide. The residue is held in the D register which is the A and B registers taken together as one 16-bit register. The divisor is held in the "S" stack, the normal hardware stack. I could have used the "U" stack, or the user stack, but I guess I'm a traditionalist . . . .

The D register is shifted left one bit at a time by shifting B left, and then shifting A and the "carry" from B left. After each shift, the divisor from the stack is subtracted from the A register. We don't have to subtract anything from B because it cannot change until it is shifted into the upper 8 bits.

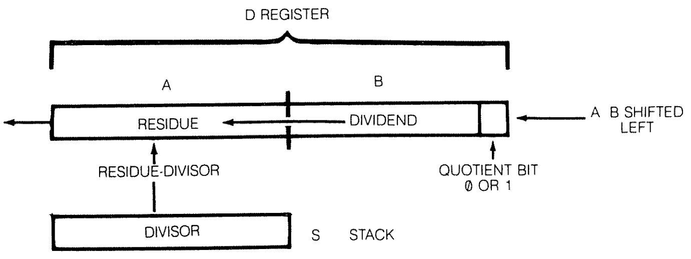


Figure 9.

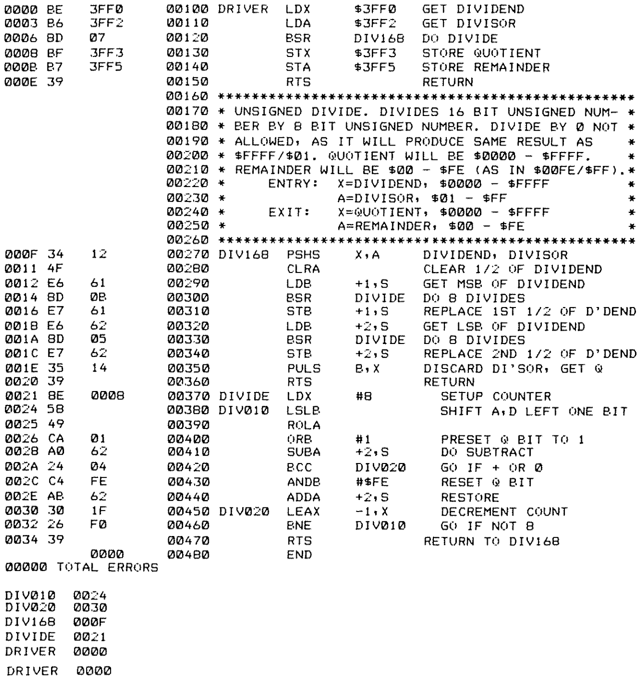
After the subtract and possible restore, the quotient bit is put into the least significant bit of the B register. The shift of D, therefore, not only shifts the "residue", but the accumulation of the quotient. The quotient can be stored in B because the shift leaves nothing in the 1s bit of the B register, and the residue is being shifted inexorably towards the left. Although at first glance it might seem that I'm being too darn clever with this scheme, it is a not uncommon way of accumulating the quotient, especially when one doesn't have an overabundance of registers, as in the 6809E.

Eight bits of the dividend are handled at one time. At the end of 8 iterations, the B register holds the upper 8 bits of the quotient. This is saved, and the B register is then loaded with the lower 8 bits of the divisor. After 8 more iterations, the 8 quotient bits in B are stored as the 8 1s quotient bits. At this point, the A register holds the remainder, the "dregs" of the residue.

So much for the algorithm. It seems complicated, but you might try actually working it out by playing computer with ones and zeroes. Its really not very profound. On to the actual code . . .

DIV168 CODE

The DIV168 code is shown in Figure 10. Lines 160 through 470 represent the actual subroutine which can be called by other assembly-language code. Lines 100 through 150 are a driver program to permit easy interfacing to BASIC.



```
ØØØØ BE    3FFØ    ØØ1ØØ DRIVER  LDX     $3FFØ   GET DIVIDEND
ØØØ3 B6    3FF2    ØØ11Ø         LDA     $3FF2   GET DIVISOR
ØØØ6 BD    Ø7      ØØ12Ø         BSR     DIV168  DO DIVIDE
ØØØ8 BF    3FF3    ØØ13Ø         STX     $3FF3   STORE QUOTIENT
ØØØB B7    3FF5    ØØ14Ø         STA     $3FF5   STORE REMAINDER
ØØØE 39            ØØ15Ø         RTS             RETURN
                   ØØ16Ø ***********************************************
                   ØØ17Ø * UNSIGNED DIVIDE. DIVIDES 16 BIT UNSIGNED NUM- *
                   ØØ18Ø * BER BY 8 BIT UNSIGNED NUMBER. DIVIDE BY Ø NOT *
                   ØØ19Ø * ALLOWED, AS IT WILL PRODUCE SAME RESULT AS    *
                   ØØ2ØØ * $FFFF/$Ø1. QUOTIENT WILL BE $ØØØØ - $FFFF.    *
                   ØØ21Ø * REMAINDER WILL BE $ØØ - $FE (AS IN $ØØFE/$FF).*
                   ØØ22Ø *     ENTRY:  X=DIVIDEND, $ØØØØ - $FFFF        *
                   ØØ23Ø *             A=DIVISOR, $Ø1 - $FF             *
                   ØØ24Ø *     EXIT:   X=QUOTIENT, $ØØØØ - $FFFF        *
                   ØØ25Ø *             A=REMAINDER, $ØØ - $FE           *
                   ØØ26Ø ***********************************************
ØØØF 34    12      ØØ27Ø DIV168  PSHS    X,A     DIVIDEND, DIVISOR
ØØ11 4F            ØØ28Ø         CLRA            CLEAR 1/2 OF DIVIDEND
ØØ12 E6    61      ØØ29Ø         LDB     +1,S    GET MSB OF DIVIDEND
ØØ14 8D    ØB      ØØ3ØØ         BSR     DIVIDE  DO 8 DIVIDES
ØØ16 E7    61      ØØ31Ø         STB     +1,S    REPLACE 1ST 1/2 OF D'DEND
ØØ18 E6    62      ØØ32Ø         LDB     +2,S    GET LSB OF DIVIDEND
ØØ1A 8D    Ø5      ØØ33Ø         BSR     DIVIDE  DO 8 DIVIDES
ØØ1C E7    62      ØØ34Ø         STB     +2,S    REPLACE 2ND 1/2 OF D'DEND
ØØ1E 35    14      ØØ35Ø         PULS    B,X     DISCARD DI'SOR, GET Q
ØØ2Ø 39            ØØ36Ø         RTS             RETURN
ØØ21 8E    ØØØ8    ØØ37Ø DIVIDE  LDX     #8        SETUP COUNTER
ØØ24 58            ØØ38Ø DIVØ1Ø  LSLB              SHIFT A,D LEFT ONE BIT
ØØ25 49            ØØ39Ø         ROLA
ØØ26 CA    Ø1      ØØ4ØØ         ORB     #1        PRESET Q BIT TO 1
ØØ28 AØ    62      ØØ41Ø         SUBA    +2,S      DO SUBTRACT
ØØ2A 24    Ø4      ØØ42Ø         BCC     DIVØ2Ø    GO IF + OR Ø
ØØ2C C4    FE      ØØ43Ø         ANDB    #$FE      RESET Q BIT
ØØ2E AB    62      ØØ44Ø         ADDA    +2,S      RESTORE
ØØ3Ø 3Ø    1F      ØØ45Ø DIVØ2Ø  LEAX    -1,X      DECREMENT COUNT
ØØ32 26    FØ      ØØ46Ø         BNE     DIVØ1Ø    GO IF NOT 8
ØØ34 39            ØØ47Ø         RTS             RETURN TO DIV168
           ØØØØ    ØØ48Ø         END
ØØØØØ TOTAL ERRORS

DIVØ1Ø  ØØ24
DIVØ2Ø  ØØ3Ø
DIV168  ØØØF
DIVIDE  ØØ21
DRIVER  ØØØØ
DRIVER  ØØØØ
```

Figure 10.

DIV168 is entered with the dividend in the X register and the divisor in the A register. The PSHS pushes X and A into the S stack. After the PSH, the stack looks as shown in Figure 11, with the S pointer pointing to the last byte (A) pushed.

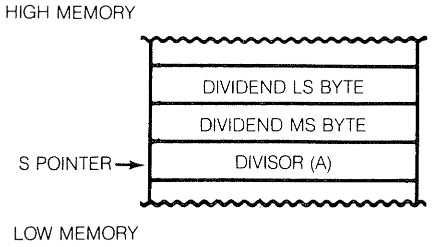


Figure 11.

The CLRA clears the A register. The LDB +1,S loads the upper 8 bits of the dividend from the stack. Subroutine DIVIDE is now called.

DIVIDE does 8 iterations of the divide. The X register is loaded with a count of 8. A shift of D is done by two shifts, the first of B and the second of A. The LSLB does a logical shift of B to the left, with a 0 going into the 1s bit of B, and the ms bit of B going into the carry condition code. The ROLA "rotates" A to the left; this has the effect of shifting A left one bit with the carry going into the 1s bit of A.

Now the Q bit in the 1s bit of B is set to a 1. This action makes the assumption that the subtract will be successful. If it is not, this bit will be reset.

A subtract of the divisor is now done from A. If the subtract left a result of 0 or greater, the C flag will be "clear" or 0, and the restore is not done. If the result was less than 0, the Q bit is reset and the divisor is added back to the A register by ADDA +2,S.

We're now at DIV020. At this point the Q bit has been stored. The count of 8 iterations in X is now decremented by the LEAX -1,X. If this count is not 0, the DIV010 loop is repeated. After 8 iterations, an RTS is done back to DIV168 code.

One tricky point in DIVIDE: Note that the subtract and add used +2,S. This is a difference of 2 from Figure 11 because the BSR to DIVIDE pushed 2 more bytes onto the stack and adjusted the S pointer. We're now back at line 310. At this point the first 8 iterations have been done and B holds the upper 8 bits of the quotient. This is now stored in the upper 8 bits of the divisor in the stack, as we'll no longer be needing them. The lower 8 bits of the divisor is now put into the B register and another call to DIVIDE is made. On the return from this call, the lower order quotient bits replace the lower-order divisor bits.

The last instruction before the return puts the divisor in the B register and the quotient in the X register.

DIV168 IN BASIC

The DIV168 subroutine can be called from other machine-language code, or it can be used from BASIC. If used from BASIC, the DRIVER code before DIV168 picks up the dividend and divisor before the divide and stores the quotient and remainder after the divide.

The "parameter block" area allocated for the input and output arguments is at RAM location $3FF0. Locations $3FF0 and $3FF1 hold the dividend. Location $3FF2 holds the divisor. Locations $3FF2 and $3FF3 hold the 16-bit quotient, while location $3FF5 holds the remainder.

A BASIC program that incorporates the machine-language code of DIV168 is shown in Figure 12. This program holds the codes as DATA values and relocates the code to the $3F00 area. Protect this area before execution by a CLEAR 200,&H3EFF.

```
100 DATA 190,63,240,182,63,242,141,7,191,63
110 DATA 243,183,63,245,57,52,18,79,230,97,141
120 DATA 11,231,97,230,98,141,5,231,98,53
130 DATA 20,57,142,0,8,88,73,202,1,160
140 DATA 98,36,4,196,254,171,98,48,31,38
150 DATA 240,57
160 FOR I=&H3FØØ TO &H3F34
170 READ A: POKE I,A
180 NEXT I
190 DEFUSRØ=&H3FØØ
200 INPUT DD,DV
210 POKE &H3FFØ,INT(DD/256)
220 POKE &H3FF1,DD-INT(DD/256)*256
230 POKE &H3FF2,DV
240 A=USRØ(Ø)
250 PRINT DD;"/";DV; "S/B";INT(DD/DV);"REMAINDER";DD-INT(DD/DV)*DV
260 PRINT "WAS ";PEEK(&H3FF3)*256+PEEK(&H3FF4);" WITH REMAINDER ";PEEK(&H3FF5)
270 GOTO 200
```

Figure 12.

The BASIC program asks for the dividend and divisor, calls DIV168, and then compares the expected result with the actual result. You really don't have as much faith as a grain of mustard seed, do you . . . .

DIV168 works, and works quite a bit faster than the successive subtraction program. Unlike the latter program DIV168 is about the same for the best case and worst case and runs between about 510 and 640 microseconds per divide. You can see from Figure 4 that the only difference in speed would be caused by the number of times that the "restore" was done.

Is this fast enough for a divide? It's pretty slow to my mind, although it's not that badly written (if one can be subjective about these things . . .). Can you do better? Why not? How about flexing your programming muscles and coming up with a faster divide? Good luck. You may advance the cause of the CoCo immeasurably . . .

Corrections to June Barden Article

Our apologies to Mr. Barden but there were typographical errors in the programs in Listing 1 (page 14) and Listing 2 (page 16) of the June article

Corrections to Listing 1
Change the following lines to read:

```
3F1A  E6   C4    ØØ31Ø          LDB      .U       USE SMALLER LENGTH
32F2  27   Ø4    ØØ42Ø          BEQ      SRTØ45   GO IF EQUAL
3F56  AE   61    ØØ64Ø          LDX      1,S      GET # ENTRIES
```

Corrections to Listing 2
Change lines 130 and 150 to read:

```
13Ø DATA 19Ø.63,254,52,16,238,228.174,94,48,31,79,52,
      18,166,196,39,42,166.196,23Ø.69,16Ø.69,36,2,23Ø,
      196,52,1,174,66,16,174,71
15Ø DATA 196,23Ø,69,231,196,167,69.234,69.234,228.
     231,228.51.69.174,97,48,31,175,97.38,176,166,228,
     5Ø.99,38.161,5Ø,98,57
```

Hex to Decimal

Charles M. Thonen
8005 B Gen Grant Dr.
Black River, NY 13612

Unlike the HEX$ function which converts the Hexadecimal number to decimal, the Color Computer has no built in Decimal to Hex function.

This conversion is easily accomplished by use of the value function and the &H symbols. This can be illustrated by the following short program listing.

```
1Ø INPUT "HEX NUMBER "; H$
2Ø PRINT "DECIMAL=";VAL("&H" + H$)
```

Landline

Rodney M. Hallam, Age 15
265 Rolfe Road
De Kalb, IL 60115

This program which I call Landline generates a "tree-dimensional" image. To me it looks like a landscape tableau. To my mom, it looks like someone "shaking a rug" . . . . Anyway, they are both good descriptions of an image that is hard to describe.

What the program actually consists of is the repetitive plotting of a sin wave. Each time it is drawn, however, the amplitude (AM) and pitch (P) change slightly. To make the image seem to come at you from left to right, change line 60:

```
6Ø P = P - .ØØ1
```

to

```
6Ø P = P + .ØØ1
```

This program also makes a great example for showing off the "Screen Print" routine on a Radio Shack printer capable of handling graphics.

If one wished to hear a little music to go along with the plotting, delete line 130. (You call that "music"?)

```
1Ø PMODE 4,1
2Ø P = (RND(6) + 1) / 1ØØ
3Ø AM = RND(2) + 1
4Ø PCLS (1)
5Ø AM = AM + 4
6Ø P = P + .ØØ1
7Ø COLOR Ø,1
8Ø SCREEN 1,Ø
9Ø LINE (Ø,94) - (Ø,95), PSET
1ØØ FOR T = Ø TO 255 STEP RND(15) + 5
11Ø A = SIN(T * P) * AM
12Ø LINE - (T, INT(A) + 95), PSET
13Ø GOTO 15Ø
14Ø SOUND 255 - (INT(A) + 95), 1
15Ø NEXT T
16Ø IF AM < 9Ø THEN 5Ø
17Ø FOR T = 1 TO 1ØØØ
     : NEXT T
18Ø GOTO 2Ø
```

Print to Screen and Wraparound

Gerald Plueard and Family
501 North 143rd Street
Seattle, WA 98133

I have the Line Printer VII and programs that use the printer. But I do not want to print all the time and have found that, with my Color BASIC 1.0 Computer, if the switch is put in the 8BS (UP) position, the computer will run and the information is seen on the screen. Then if I want to print on the printer all I have to do is run the program again but this time with the switch in the 7BS (MID) position.

```
42Ø PRINT
    : PRINT "      TYPE YOUR SENTENCE NOW"
45Ø A$=INKEY$
    : IF A$ = "" THEN 45Ø
48Ø IF A$ = CHR$(13) THEN 45Ø 'CHR$(13) = <ENTER>
    KEY
5ØØ IF A$ = CHR$(8) THEN 45Ø
51Ø PRINT A$;
52Ø A$(X) = A$(X) + A$
53Ø IF LEN(A$(X)) > 65 AND A$ = CHR$(32) THEN 54Ø
    ELSE 45Ø
54Ø X = X + 1
    : PRINT#-2, A$(X-1)
56Ø GOTO 45Ø
```

The next program allows the Line Printer VII to print without cutting words off on the right edge of the paper.

```
251Ø FOR Y = 1 TO 5Ø  'NUMBER OF SENTENCES
261Ø LINE INPUT D$(Y)
267Ø INPUT" DO YOU WANT TO PRINT IT?"; S$
     : IF LEFT$(S$,1) = "Y" THEN 268Ø ELSE 281Ø
268Ø LL = LEN(D$(Y))
269Ø IF LL > 255 THEN LL = 255
27ØØ FOR Z = LL TO 1 STEP -1
271Ø IF LEN(D$(Y)) < 1 THEN 283Ø
272Ø IF LEN(D$(Y)) < DW1 THEN 28ØØ 'DW1 = 4Ø FOR
     DOUBLE WIDTH, 8Ø FOR NORMAL
273Ø PLAY "L235;12"
274Ø IF MID$(D$(Y),Z,1) = CHR$(32) THEN IF
     LEN(MID$(D$(Y),1,Z)) < DW2 THEN 276Ø
275Ø NEXT Z
276Ø PRINT#-2, MID$(D$(Y),1,Z)
277Ø D$(Y) = MID$(D$(Y),Z,LL)
278Ø LL = LEN(D$(Y))
279Ø IF LEN(D$(Y)) > 1 THEN 27ØØ
28ØØ PRINT#-2, D$(Y)
281Ø CLS
282Ø X2 = Y
     : PRINT "SENTENCES TYPED" X2
     : PRINT "OUT OF 5Ø "
283Ø NEXT Y
```

Data Base Management Systems

This month we will discuss Data Base Management Systems. This field of "data-processing" has become a science in itself, so we obviously can't cover the subject completely in one short article.

The definition of a data base system is a computer-based record keeping system whose overall purpose is to record and maintain information. The information stored in the system can be anything that is considered to be important or of significance to the company or organization that is using the system. The manipulation of the information in the data base is done by the data base management system software. With a bit of programming knowledge, additional reports can be extracted from the data base.

Why should a company use a data base system, and what are some of the advantages of such a system? The primary reason for a company or organization to use a data base is centralized control of information. One of the primary advantages of a data base system is the elimination or reduction of redundant files. Inconsistencies in data can also be avoided to a great extent by reducing the number of times data is entered. This goes hand-in-hand with the elimination of redundant files. Another advantage to a data base is that data can be shared, i.e., the information captured by one group can be used by another.

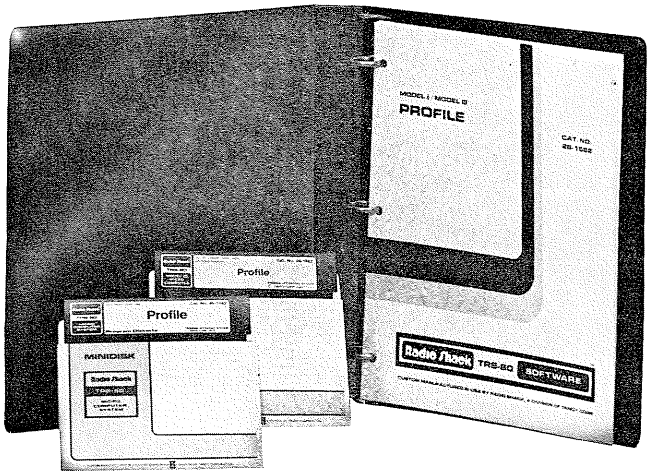
What type of information should be stored in a particular data base? Any information that would normally be put into an organized filing system is one type of information that could be put into a data base. A data base could also include complete company records—information to be used in the decision making processes of a company or organization. The information contained in the data base can range from a simple personnel file to a complete set of files containing employee information, project information, suppliers, inventory information, parts used in a project, and departments involved in a project. Many companies and organizations use more than one data base. Let's look at some of Radio Shack's data base management systems.

MODEL III DATA BASE SYSTEMS

Profile—Catalog No. 26-1562

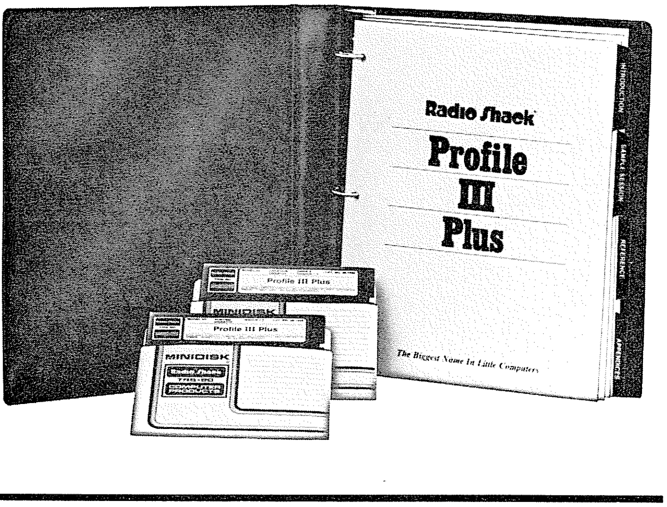
Profile for Model I/III allows the user to design a data entry screen to his or her own specifications, with up to 32 fields per record. Records can be variable in length up to a maximum of 254 characters in each record. With this type of versatility a company can create a custom record keeping system with ease.

One important consideration in choosing a data base system is the ease with which information can be retrieved from the system. Profile allows the user to find information by any field in the record. The system will also sort the records by any field in the record. Printed reports are also defined by the user, putting a report title and column headings at the top of the report with the system printing the data below the headings. The data chosen can be selected information or all of the information stored in the data base. Another feature of Profile is the accessibility of the data through the BASIC programming language. This feature allows the user to manipulate the data in any way that is required.



Profile III Plus—Catalog No. 26-1592

The Profile III Plus data base system (Model III only), like Profile, allows the user to define his/her own data entry screens. Each data base can have up to five separate screens to provide for additional flexibility. Profile limited the user to 32 fields on a single screen whereas Profile III Plus allows a total of 99 fields, with a maximum of 36 fields on any one of five separate screens. In Profile the user was limited to 254 characters of information, but Profile III Plus allows multiple segment records, providing space for more than 1000 characters of information per record.



Profile III Plus provides many of the same features of Profile with many advanced data manipulation features such as the ability to perform mathematical functions with the data in specified fields The searching and sorting features of Profile III Plus are much the same as Profile with the ability to search or sort by any of the 36 fields in the first segment. Profile III Plus gives the added advantage of being able to associate fields from the first segment with one another. A totally customized data base can be created with the user menu feature of Profile III Plus.

Information is easily accessed with Profile III Plus as the user may define up to five different reports where Profile allows only one report format to be stored. Additionally Profile III Plus can be accessed by other Radio Shack software packages such as VisiCalc and SuperScripsit adding to its report generating capabilities.

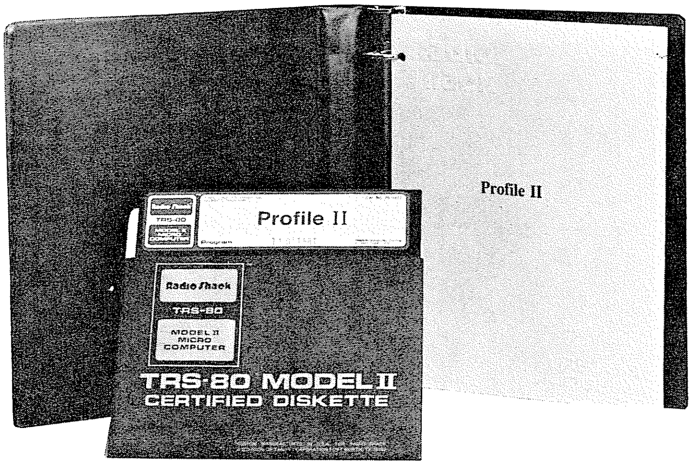
MODEL II DATA BASE SYSTEMS

Profile II—Catalog No.26-4512

Profile II is a very versatile data base system that lets the user format up to five data entry screens, each screen containing up to 36 fields of information with a maximum number of 99 fields. Profile II also has the ability to use multiple segment records. Multiple segment records allows the user to store up to 850 characters of information per record. The Model II owner with four floppy disk drives can store up to 1800 four segment records or a total of one and a half million characters of information.

Finding information is easy with Profile II. Profile II has the ability to find information by record number or search on any one of the 36 fields in the first segment. Profile II will sort the information in the data base by any of the fields in the first segment. Many times information for reports must meet more than one criterion and Profile II allows two fields to be associated with each other for greater searching and sorting capability. The system allows the addition and subtraction math functions to be performed on the data in the system to give some additional data manipulation capability.

Up to five user defined printed reports are allowed in Profile II. A report can consist of any combination of the fields of information in the system based on the data required in the report. Scripsit can also be used with Profile II for additional flexibility in generating reports.



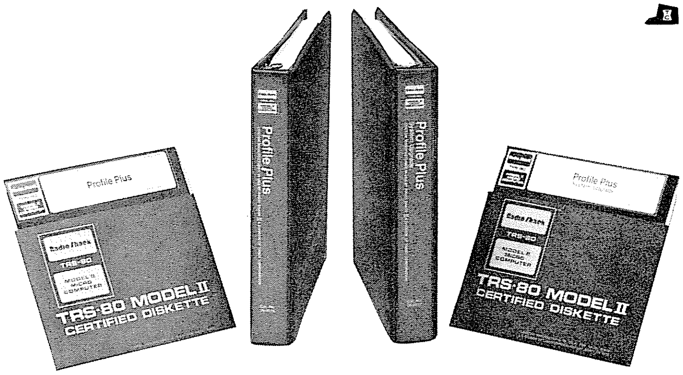
Profile Plus—Catalog No. 26-4517

Profile Plus for Model II is an extension of Profile II with all of the features mentioned above but with enhancements to make the user's data base even more manageable. Records can be defined by the user with five input screens for a total of 99 fields of information. One of the advantages of Profile Plus is the ability to define segments shorter than 256 characters whereas in Profile II the second, third, and fourth segments were 256 characters long with no option to allow for less space to be used. This means that if only 200 characters are required in a data base, the first segment would contain 85 characters and the second segment would have 115 characters instead of 256. This would save a great deal of storage space allowing for additional records to be stored. Profile Plus also has additional field indicators and the ability to associate fields in segment one into groups to allow for specialized data entry screens and data manipulation within the data base. Profile Plus also allows the user to define mathematical formulas that are to be used with certain pieces of information within the data base, making the system extremely flexible.

Profile Plus can be accessed in the same manner as Profile II. The added features of Profile Plus allow for specialized report generation and more versatile data manipulation. The user can design up to five report screens and using the advanced functions of Profile Plus can set up the specific reports required from the data base.

Conclusion

There are many reasons for a company to use a data base management system. Concise, accurate, well organized, and easily accessed information are the goals of a data base management system. Radio Shack's Profile data base management systems can help you achieve these goals.



Computer Customer Services Address and Phone Number

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

These are New Numbers and Were Effective Aug. 27

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

Welcome To Someday

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

"Someday we'll leave them speechless by coming up with a way to shop, choose and purchase merchandise from our own home."

"Sure."

"Someday we'll develop a way to order and load a computer program from our system into a home computer all by using a couple of commands and then what'll they say?"

"Right."

"Someday we'll develop a way to do all our banking from our living rooms and see what they say to that."

"Whatever you say."

"Then what if I say 'Welcome to Someday,' today?.

Someday is here and we'd like you to get to know us. We're the CompuServe Information Service (CIS), and we'd like to welcome you to a tour of someday today.

We at CompuServe feel we're riding on the crest of a technological tidal wave as the largest electronic information and entertainment service available to owners of personal computers and computer terminals. We're taking entrepreneurs up on their ideas of computerized home management and coming up with a few of our own.

CompuServe is a computer services company headquartered in Columbus, Ohio, and is a subsidiary of H&R Block. We serve corporate and government customers who share time on our computers and often depend on our application products to manage their information and data processing needs.

We've been serving some of the nation's largest industries in major metropolitan areas through coast-to-coast sales and service facilities since 1969. By the late 1970s, CompuServe had an established reputation for reliability in providing software solutions, operations assistance and research and development capabilities to its business customers. Having successfully introduced electronic information services to the business world, we then turned our efforts toward personal computing services and opened up a world of information to the hobbyist, the professional and the novice.

CompuServe Information Service is available in Radio Shack outlets from Maine to Modesto and Tallahassee to Toronto. As a new CIS customer you can purchase a starter kit containing a special access number, a secret password, a CIS User's Guide and a free hour of time on us. In addition to access to all of our services you'll receive a free subscription to Today magazine, a CompuServe publication which keeps you up to date on industry trends, plus our monthly newsletter, Update, which lists the latest innovations and additions to CIS.

Our primary and supplemental networks provide access to the information service through a local telephone call in more than 300 U.S. cities and Canada. At a cost of $5 per hour, (in the U.S.) customers can access the service anytime before 5 a.m. and after 6 p.m. and all day Saturday. Sunday and holidays, thus utilizing CompuServe's off peak-hours computer power.

CompuServe provides you with the ability to communicate with other CIS users either across the street or across the country by using electronic mail. CIS combines menu-choice and word-search technology to display information arranged in an easy-to-find, "Go" anywhere format. As well as providing you with information, our service enables you to enhance and dramatically increase the capabilities of your TRS-80 computer or Videotex terminal.

CompuServe categorizes its offerings in menu form and enables you to access topics of interest by making selections from more specific menus. We have designed a subject index which alphabetically lists our subject areas and provides a page number enabling you to go directly to your subject without searching through the separate menus. In addition to our alphabetized index, we provide a built in key word search capability which enables you to enter the key word at the appropriate prompt. The computer will then respond with your topic and its electronic page number.

The CIS main menu contains the following listings: home services, business and financial services, personal computing services, user information and the subject index.

EXTRA, EXTRA! ELECTRONIC NEWSPAPERS HIT THE SCREENS

The CompuServe Information Service offers a number of major daily newspapers and The Associated Press newswire service under the Home Services menu category.

All of the electronic papers offer basic news as well as special features created specifically for CompuServe customers including on-line updating of current news, video games and advertisements.

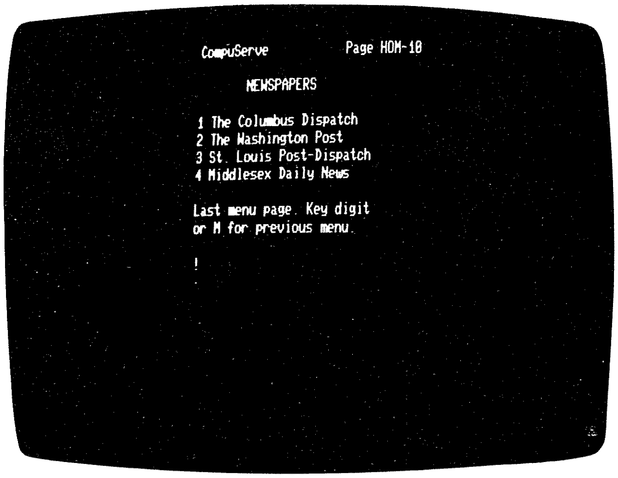
The electronic editions feature the whole gamut of world, U.S. and local news, sports, theater and movie reviews, books, financial and art news and many other standard newspaper features.

The Washington Post, for example, can help you find a new job in the Washington D.C. area when you input the job title desired, salary and geographic preference. The system provides you with a list of acceptable job titles and state abbreviations required for the search. The computer does the search for you using this information while you sit back. No walking, driving or pounding on doors.

The St. Louis Post-Dispatch offers a classified marketplace with almost 300 categories of daily updated advertisements. News events are updated hourly through the system and comments and suggestions are welcomed through the paper's own electronic mail "Feedback" service.

The Columbus Dispatch offers a wide variety of on-line entertainment including entertainment news, movies and other reviews, plus the local, state and national news. The Dispatch also offers two video games: Starforce Marauder and Remove: a logic game. Starforce Marauder, an interactive space mission could raise even the eyebrows of the skeptical Spock. To complete your mission, you must move through quadrants on a patrol route provided with an arsenal of weapons. You must complete your patrol route before your battle armor runs out of power.

Remove is based on a Chinese game of intelligence and is played with a certain number of objects stacked in a column. The key to the game is to remove at least one object from only one column per turn. But if you're the player who picks up the last remaining object, you lose.



```
CompuServe          Page HOM-10

         NEWSPAPERS

 1 The Columbus Dispatch
 2 The Washington Post
 3 St. Louis Post-Dispatch
 4 Middlesex Daily News

Last menu page. Key digit
or M for previous menu.

!
```

TAKE A "BYTE" OUT OF HOUSEHOLD CHORES

Many other services are available under our Home Services menu, such as weather information, a reference library, containing miscellaneous information such as video program reviews, movie reviews, and the Popular Science and Better Homes and Gardens magazines.

Also, under our Home Services menu is the communications section which offers the interactive services of Email, or electronic mail, "CB," the User's Directory, and the National Bulletin Board. This selection also includes Shop/Bank at Home, Groups and Clubs, Games and Entertainment, Education, which offers programs such as mind and memory builders, and Home Management programs which help you accomplish the more complex tasks of balancing your checkbook or amortizing a loan.

MIND YOUR OWN BUSINESS

Another main menu feature is Business and Financial Services providing news and reports from the business world. Information providers here include the Archer Commodities Report, the Small Business Report and the Business Information Wire, to name a few.

YOU SAY YOU WANT TO GET PERSONAL . . .

The Personal Computing Services listed on the main menu offer several on-line newsletters from companies like Tandy, Commodore, Atari and RCA, as well as the Micro Advisor and Popular Electronics magazine. There is also a reference section featuring information on languages, word processing and mathematics programs. The Communications section of PCS offers the interactive services of CB, Email and a shop-at-home section, which points you to the Software Exchange program and the Art Gallery where you can order or review the line printer art offerings. The Groups and Clubs section takes you into the Special Interest Groups (SIGs), and the Programming and Computing section enters you into the Command Mode allowing you to expand the programming capabilities of your personal computer by tapping into CompuServe's main frame computing power.

HELLO? INFORMATION . . .

The next main menu area, User Information, tells you immediately at log on "What's New" for the week in the way of our products, new telephone access numbers and new access locations. CompuServe's free Feedback section allows you to tell us what's on your mind. There are also instructions on how to change your terminal defaults or password and how to review your charges for the month. There is always a listing of telephone access numbers and the current rates which accompany them.

Also under User Information, CIS provides CompuServe Viewpoint, a series of articles which help explain our information service and Videotex. It offers answers to your billing questions, a short course in FILGE, (our text editing program), results of our user surveys plus an explanation of CompuServe and Videotex and career opportunities with CompuServe.

LET YOUR FINGERS DO THE KEYING THROUGH THE USER'S INDEX.

The last main menu item is the CompuServe Information Services Index. This is a complete alphabetical listing of all our subjects. The user index appears on the screen in the following form:

```
Access . . . . . . . . . . . . . . . . . . . . Go PCS-30
Access Phone Number . . . . . . . Go CIS-4
Adult Education . . . . . . . . . . . . . Go HOM-70
Adventure Game . . . . . . . . . . . . Go HOM-60
```

The "Go" command in these examples refers to the method used in CIS to go directly to a page rather than running through all the separate menus.

"Go n . . . " refers to Go directly to page "n". N stands for a number alone or an information provider/number combination such as CIS-4.

CompuServe is proud of its over 450 subject offerings and we'd like to highlight some of our more popular services from the multitude.

BREAKER BREAKER ONE-NINE. WHAT'S YOUR ID NUMBER?

CompuServe offers "CB". This is our simulation of the Citizens' Band radio which allows you to talk with other CompuServe users around the city, state, country or Canada.

Our CBers have "handles" just like the regular CB users for the purpose of identification to other CBers once on the system. To gain access to CB, users enter the system with their user ID number and password and select "Communications". CB users can send, receive, and monitor messages on any of the 40 channels offered over the system.

CBers can talk to as many people as they wish at one time or just one person in a private conversational mode. Everything typed onto the screen is seen by other CompuServe CB users logged on to the same channel. Becoming a skilled CB user is accomplished by following a few basic commands.

/TUNE changes the channel so that you can send and receive messages on a different channel.

/STATUS tells you how many people are on the channel at the time you issued the command.

/MONITOR allows you to follow conversations on one or more channels.

/SCRAMBLE allows you to have a private conversation with another CBer by issuing a special code which only you and that other person know. By doing this, you and the other person are reading your messages while others are just seeing the scrambled code.

CIS customers can find the CB Simulator under the main menu item 2, Communications.

SIGs: OUR MEMBERS ARE "SPECIALLY INTERESTED IN GROUPS."

CompuServe offers another area where you can share your interests with others on-line. These are the Special Interest Groups or SIGs. SIGs are electronic clubs for those of you who like to converse about a particular subject.

We offer many different SIGs. One such SIG is MNET80, a TRS-80 Users Group providing information and the latest news on software for TRS-80 microcomputers. Its members include professional programmers and companies using TRS-80 hardware and software. A membership fee of $20 allows access to TRS-80 experts and to other TRS-80 enthusiasts around the country.

Offerings in the MNET80 SIG include complete cassette based smart terminal programs, scores of original pieces of software written by its members and a section for the Color Computer, and TRS-80 Models I, II and III.

The MCONN SIG is sponsored by CompuServe and the Microperipheral Corporation, a manufacturer of modems for popular microcomputers. This SIG is a communication tool between Microperipheral Corporation and its dealers, customers and potential customers. The only requirement for joining this SIG is that your company has purchased either hardware or software from the Microperipheral Corporation. Membership can also be requested upon completion of an entry in the User Interest Log.

A non-SIG informational product close to the heart of TRS-80 users is the Tandy Newsletter. This selection offers information on product availability, hints and tips on equipment use, product descriptions and its own Feedback section where our customers can communicate directly with Tandy headquarters for the answers to their questions.

RAINDROPS KEEP FALLING ON MY HEAD . . .

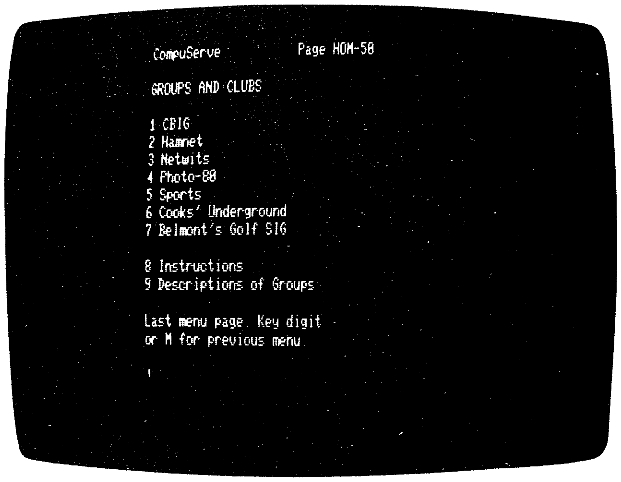
If you're going to discuss something with someone you don't know, it may as well be the weather. And CompuServe can have you discussing anything from whether it's going to rain in Topeka to the location of the nearest VORTAC station for the quick landing of your single engine aircraft.

CompuServe offers "Aviation and General Weather" on the Home Services menu. Public weather consists of Public and Marine weather reports written in plain English. The Aviation Weather is a service for pilots consisting of information taken from the National Oceanographic and Atmospheric Administration (NOAA) high-speed wire. There is help available for those of you who find you aren't able to read the listings with "flying" colors.

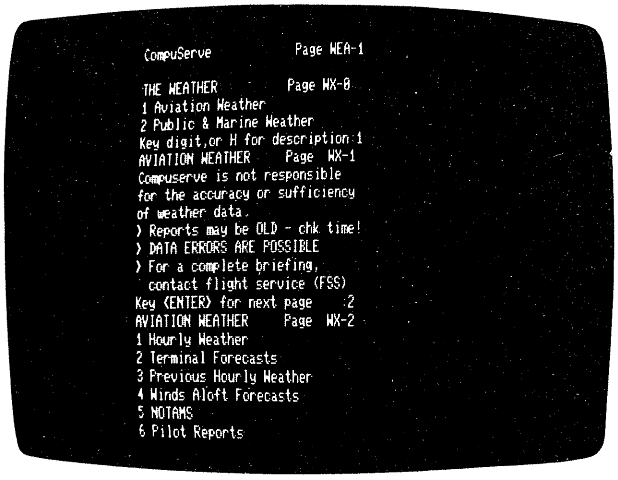
"Aviation" offers the hourly weather for different areas, terminal forecasts which contain data for specific airports and are issued three times daily; previous hourly weather, reports of weather conditions given by pilots in flight, and area forecasts, prepared twice a day for an area that may encompass several states.

PARLEZ-VOUS FORTRAN?

If weather is a little too broad a category for you, we offer a list of 11 computer languages for you to tinker with. CompuServe offers languages for the advanced computer hobbyist to the inexperienced novice including BASIC and XF4 FORTRAN, an extended FORTRAN language developed by CompuServe. Programming languages can be found by accessing Personal Computing Services.



```
CompuServe          Page HOM-58

GROUPS AND CLUBS

1 CBIG
2 Hamnet
3 Netwits
4 Photo-80
5 Sports
6 Cooks' Underground
7 Belmont's Golf SIG

8 Instructions
9 Descriptions of Groups

Last menu page. Key digit
or M for previous menu.

!
```



```
CompuServe          Page WEA-1

THE WEATHER         Page WX-0
1 Aviation Weather
2 Public & Marine Weather
Key digit,or H for description:1
AVIATION WEATHER    Page  WX-1
Compuserve is not responsible
for the accuracy or sufficiency
of weather data.
) Reports may be OLD - chk time!
) DATA ERRORS ARE POSSIBLE
) For a complete briefing,
  contact flight service (FSS)
Key <ENTER> for next page    :2
AVIATION WEATHER    Page  WX-2
1 Hourly Weather
2 Terminal Forecasts
3 Previous Hourly Weather
4 Winds Aloft Forecasts
5 NOTAMS
6 Pilot Reports
```

IS IT SECURITIES OR BONDS? WE'VE GOT YOUR SERVICE IN "STOCK"

CompuServe has a vast array of financial services. The MicroQuote service is an historic securities database for people who follow the stock market or are exploring the world of finance. It's easy to use and is updated each day to provide fast access to a variety of information on securities traded on the exchanges and over-the-counter. MicroQuote offers historical data dating back to Dec. 31, 1973 on dividends, pricing and volumes.

There is also information on bonds including yields, maturity dates and Moody's ratings. Options information features prices, expiration dates and underlying stock prices. Depending on the information accessed, MicroQuote users must pay a small charge above the $5 per hour CompuServe charge.

Another follower of the stock market is Quick Quote. This service provides high, low, closing, volume and net change figures on over 9,000 securities traded on the New York and American Stock Exchange and over-the-counter.

To use Quick Quote, you insert the ticker symbol or CUSIP (Committee on Uniform Security Identification Procedures) number, and the program gives you the latest information as reported from Wall Street.

Quick Quote also provides you with an easy way to find the name of a stock or whether or not it's being traded and through which exchange.

Another of our financial features is Value Line Database II. This product allows you to analyze the performance of more than 1,600 major industrial, transportation, utility, retail, banking, and insurance companies and includes detailed data from historical annual income statements, balance sheets, sources and uses of funds, as well as reported quarterly results.

Standard and Poor's is another financial product available under the Business and Financial Services section or by typing Go FIN-20. It's an easy way to obtain financial statistics on whatever company interests you. Included in this datafile are the separate features of the Business Summary, Important Business Developments, Product/Service Line and General Information.

WE'RE GAME IF YOU ARE

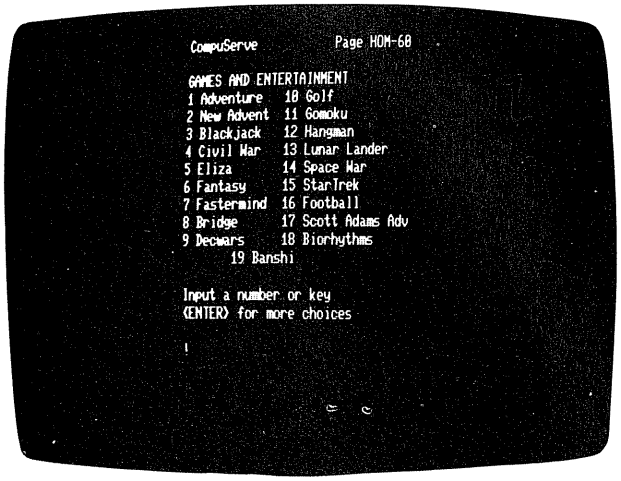
One of our more popular services with the young and old alike are the over 30 offerings of all types of games. Some of the selections include games of chance such as blackjack, craps or roulette, athletic games such as golf and football, and analytical or strategic games like Star Trek, backgammon and chess.

Other entertainment features are biorythmns, the cube solver and Eliza, an interactive program which features the computer in the role of the psychiatrist carrying on simple conversations with you, the "patient."

Two of the most popular games are Adventure and Megawars. Adventure uses your computer as your eyes and hands to battle snakes and dwarves in Colossal Cave in search of the lost treasure. Adventure has caused quite a stir with CIS customers and even has its own T-shirt, puzzle and two different Adventure maps to guide adventurers through the game (one is the advanced version), and a full-color poster depicting a confrontation between explorers and a large green snake in Colossal Cave.

Megawars is another popular game as a real-time space battle game where up to 10 players can play at one time. The object of this game is to choose either to captain a star ship or an enemy ship and thus proceed to attack your opponents' space ships and capture planets. Megawars also has its own accessories including T-shirts, brochures, and a full-color poster depicting a confrontation between Colonists and the Kryon Empire.

Adventure, Megawars and all the other CompuServe games can be found under Home Services, or by typing Go HOM-60.



```
CompuServe          Page HOM-60

GAMES AND ENTERTAINMENT
1 Adventure    10 Golf
2 New Advent   11 Gomoku
3 Blackjack    12 Hangman
4 Civil War    13 Lunar Lander
5 Eliza        14 Space War
6 Fantasy      15 StarTrek
7 Fastermind   16 Football
8 Bridge       17 Scott Adams Adv
9 Decwars      18 Biorhythms
        19 Banshi

Input a number or key
<ENTER> for more choices

!
```

HOW DO YOU SPELL CONVENIENCE?

Two features which welcome you to someday are the banking and shopping at home features. Customers may shop from the comfort of their own home through Comp-U-Store, our interactive electronic shopping service developed in conjunction with Comp-U-Card, a Connecticut-based company which allows customers to order merchandise electronically to be delivered to their homes. Comp-U-Store offers consumer products in a format which allows customers to either browse through or review items by description.

A customer can check price, features and receive information about a specific item all through Comp-U-Store. Comp-U-Store allows customers to search and comparison shop electronically.

CompuServe, Radio Shack and United American Bank Service Corporation have joined forces to offer Bank-At-Home service. Bank-At-Home service permits customers of the United American Bank to pay most of their bills, receive current bank information and view their accounts displayed on their screens. Bank-At-Home will also offer computerized bookkeeping and tax record services, and the ability to apply for loans on-line.

AND THERE'S STILL MORE, MORE, MORE, MORE, MORE . . .

The CompuServe Information Service offers beneficial subscriber services. These services include two publications which we provide in addition to CIS.

Update is our monthly newsletter which informs you about new products, programs, telephone number additions and corrections, plus other timely information. Update also provides you periodically with the complete CIS subject index as an on-hand reference.

Today is a full-color magazine which offers a more in-depth review of new services, information providers to CIS, customers and how they use the service and special announcements. Today is free to CIS subscribers and is available to noncustomers through subscription.

CompuServe's free on-line Feedback service allows you to communicate with CompuServe's Customer Service department. Any problems which you are experiencing can be quickly attended to through Feedback found under User Information.

In addition to our Feedback section, new customers to our system receive a free hour of CIS time with the starter kit purchased at any Radio Shack outlet. This free hour is of course accompanied by your own User ID, secret password and user's guide.

For those of you who have questions regarding microcomputers and personal computing, CompuServe offers the Micro Advisor. The Micro Advisor is a clearinghouse for information to aid the experienced or novice user. Micro Advisor discusses equipment usage, selection, peripheral devices and software. Micro Advisor utilizes its own user Feedback service to answer questions or to comment on groups and the latest developments in the microcomputing industry.

CompuServe also offers Access. This service is for those of you who want to share programs which you have authored with other CIS users. Access contains a wide selection of user-contributed programs, games and text files.

OUR CIS NETWORK . . . AND WELCOME TO IT.

CompuServe Information Service is known as one of the largest information and entertainment services available to home computer and computer terminal owners across the country. And CompuServe Incorporated goes even further by offering the benefits of its high quality communications services through its value-added network in 130 cities.

Our customers using this network are connected to a system of communications lines which we've leased in order to offer them the most efficient service we can develop. Customers with personal computers and terminals can dial a local phone number and hook up to our network. Because we control the growth of our network, we have the ability to offer a smoothly-running, error-free and always operational network. If there's a problem, we're able to detect it and either repair it or reroute our customers' data to ensure minimal interruption of service.

U.S. customers in cities we don't reach access CIS through the TYMNET system, a common carrier network. Customers accessing through TYMNET are charged an additional $2 per hour communication surcharge making the use of the service $7 an hour rather than the usual $5 an hour.

CompuServe plans to expand to 300 cities by 1983 to provide our customers with good quality communications in their city. It probably won't be long before we're in your neighborhood, too.

THINK YOU MIGHT BE CONVINCED?

All of the above information is just our way of introducing you to the CompuServe Information Service. We think you'd like having all this convenient information available from your living room, accessible through a local phone call for just $5 an hour.

CompuServe is proud of its comprehensive service, but we don't plan to stop here. We're researching and discovering new services and methods of operation every day in order to offer you even more than we've ever had before. And when we find these discoveries, we let you know through our system announcements, Update, and Today magazine. After all, we are an information, communications service. And we're bullish on keeping you informed at all times.

Go ahead and take advantage of us. Take advantage of our convenient information, the convenience of using us in your own home and accessing our ever-growing databases. Purchase CompuServe Information Service. We're found at any Radio Shack locations around the country and in Canada.

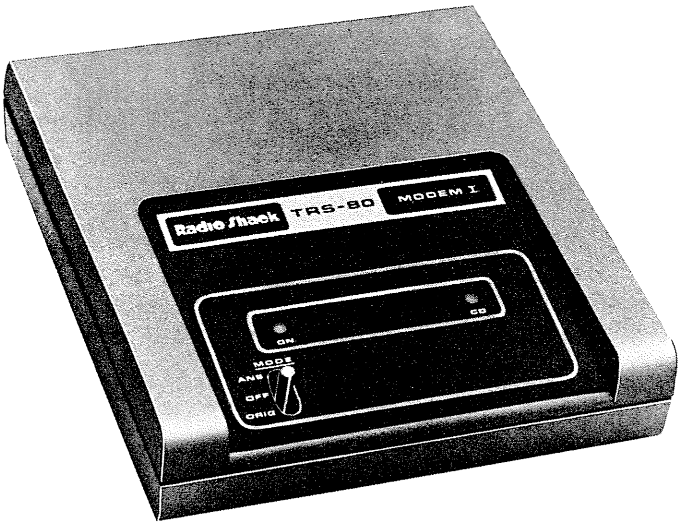
Questions and comments about the CompuServe Information Service can be sent to Richard A. Baker, Editorial Director, CompuServe Information Service, 5000 Arlington Centre Blvd., P.O. Box 20212, Columbus, Ohio 43220 or through Feedback, main menu item 5, CompuServe User Information.

Computer Clubs

East Coast Computer Club
c/o James Law Conway
198 Old River Road
Wilkes-Barre, PA 18702

Rhode Island Tandy Users Group (RITUG)
c/o McDonald & Ferdinandi
1441 Park Avenue
Cranston, RI 02920
(401) 943-1801

Western Massachusetts Computer Club
c/o Phillip C. Jachem
2275 Westfield Street
West Springfield, MASS 01089
(413) 736-3521 or (413) 592-2023



The Academic American Encyclopedia–A to Z Electronically

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

There is a new kind of encyclopedia these days—an encyclopedia that is never out of date and is always easy to handle. It is both original and widely praised. It is an electronic encyclopedia, and News/Retrieval has it.

With the new electronic edition of the highly respected Academic American Encyclopedia already online for subscribers, News/Retrieval has become the first electronic information service to offer a highly praised encyclopedia to the consumer in a form he or she can quickly grasp and easily use.

Because electronic publishing permits limitless page expansion and constant re-editing, the new encyclopedia can be completely updated whenever necessary—not with separate, printed handouts, but with actual changes in the original text to provide the most current information available. New articles will be added and others revised to supply adults as well as students with a current, comprehensive, authoritative reference tool for the home or office.

In addition to these revisions, the capabilities of electronic publishing will allow News/Retrieval journalists to cross-reference other data bases with the encyclopedia, directing users to the Academic American for background on breaking news and sports developments.

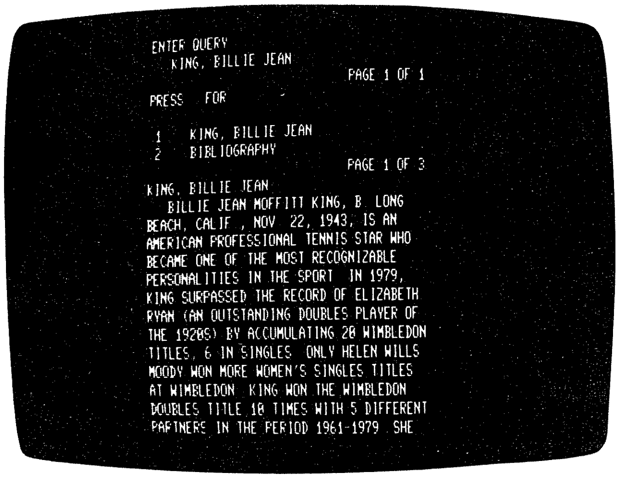
The electronic currency of the encyclopedia data base is blended with scholarly research of more than 2,500 distinguished contributors and editors. Produced by Arete Publishing Company, the original 20-volume Academic American contains more than nine million words in 28,000 carefully researched and concisely written articles on a vast array of subjects.

The first new multi-volume encyclopedia in more than a decade, this reference work received excellent reviews when it was introduced in 1980. The Encyclopedia Buying Guide praised the Academic American as "an innovative, forward-looking encyclopedia." The set was found to be "meticulously researched . . . both comprehensive and well organized," by Booklist, an American Library Association journal. Adds Library Journal: "The Academic American succeeds admirably . . . It is up-to-date, concise, authoritative, and well documented . . . Its most outstanding attribute is its timeliness."

Reviewers have been particularly impressed by the Academic American's coverage of volatile subjects—political changes around the world, scientific backgrounds, sports and industry trends—all of which are accompanied by the historical, philosophical or scientific background needed to put these developments in perspective. "In its treatment of bias-prone areas, the editors are successful in presenting balanced, open statements, free from editorializing," said Booklist, citing as evidence the entries covering the Ku Klux Klan, Khomeini, Sun Myung Moon, and Terrorism.

In the encyclopedia's regular revisions—which initially will be made twice a year—many new articles, especially biographies, will be created. And original articles will be revised to reflect new facts and information. Grolier Incorporated, the world's largest publisher of multi-volume reference sets, will produce these updates, with the first planned for early autumn. Grolier recently acquired the long-term rights to publish the Academic American in both its printed and electronic forms.

In addition to these continuing updates the encyclopedia will be made relevant to daily current events by the News/Retrieval staff. Newsroom writers and editors, keeping abreast of both the news and relevant encyclopedia information, will cross-refer news, sports and business readers to entries in the Academic American. An example was this summer's Wimbledon tennis coverage. On the day that saw Billie Jean King reach the semifinals, a sentence below the story in News/Retrieval Sports read: "For a closer look at Billie Jean King's tennis career, see King, Billie Jean in //ENCYC." Such cross-referencing is one way News/Retrieval interre-



```
ENTER QUERY
  KING, BILLIE JEAN
                                PAGE 1 OF 1
PRESS  FOR

 1   KING, BILLIE JEAN
 2   BIBLIOGRAPHY
                                PAGE 1 OF 3
KING, BILLIE JEAN
  BILLIE JEAN MOFFITT KING, B LONG
BEACH, CALIF , NOV 22, 1943, IS AN
AMERICAN PROFESSIONAL TENNIS STAR WHO
BECAME ONE OF THE MOST RECOGNIZABLE
PERSONALITIES IN THE SPORT  IN 1979,
KING SURPASSED THE RECORD OF ELIZABETH
RYAN (AN OUTSTANDING DOUBLES PLAYER OF
THE 1920S) BY ACCUMULATING 20 WIMBLEDON
TITLES, 6 IN SINGLES  ONLY HELEN WILLS
MOODY WON MORE WOMEN'S SINGLES TITLES
AT WIMBLEDON  KING WON THE WIMBLEDON
DOUBLES TITLE 10 TIMES WITH 5 DIFFERENT
PARTNERS IN THE PERIOD 1961-1979 . SHE
```

lates all its data bases to provide the in-depth coverage a complicated world requires.

The building of an information service capable of that coverage was the first consideration of News/Retrieval editors when they selected the Academic American as the encyclopedia they would put online. The Academic American is the first encyclopedia specifically conceived for electronic publication, with complete but relatively concise articles, and fact boxes and abundant statistical information designed for easy and accurate updating. The prose style, as described in a review by Choice magazine, "is commendable, is well written, concise, and informative without being condescending or obscure." Also, many articles conclude with bibliographies which have won special praise. Booklist called them "appropriate, current, and skillfully selected".

To make the electronic encyclopedia easy-to-use, News/Retrieval editors and computer specialists spent the past year preparing the Academic American for electronic distribution. For example, the chemical notation for water (H_2O) had to be written out completely since most computers cannot handle below-the-line notation. The methods for getting into the encyclopedia were painstakingly devised for exceptional ease of use.

Data base organization, search strategies, commands and help messages all have been tested on secondary-level students, many of whom had no experience with computers or online information services. This "child-proofing" of the electronic encyclopedia has made it easier for everyone to use, from junior high school level on up.

The electronic encyclopedia has been designed to resemble the familiar "subject outline," leading students from the broad to the specific. In effect, it helps teach children how to begin and develop a research project. The online bibliographies and various types of cross-references make it easy for users to find additional information.

Adults and students will find the electronic encyclopedia invaluable for quick fact checks. For example, detailed tables on major athletic events will help sports fans settle arguments. And special fact boxes about countries and states highlight the area's population, principal products, languages, climate, levels of education, and infrastructure.

The encyclopedia is part of the General News and Information Services group of data bases and is accessible by typing //ENCYC. It costs standard subscribers 60 cents per minute in prime time and 30 cents per minute in non-prime time.

//ENCYC How-To

There are two ways to find information in the electronic encyclopedia. After typing //ENCYC and pressing the ⟨ENTER⟩ key to access the encyclopedia you can "search by partial title of a subject heading" or "search by the complete title of a subject heading."

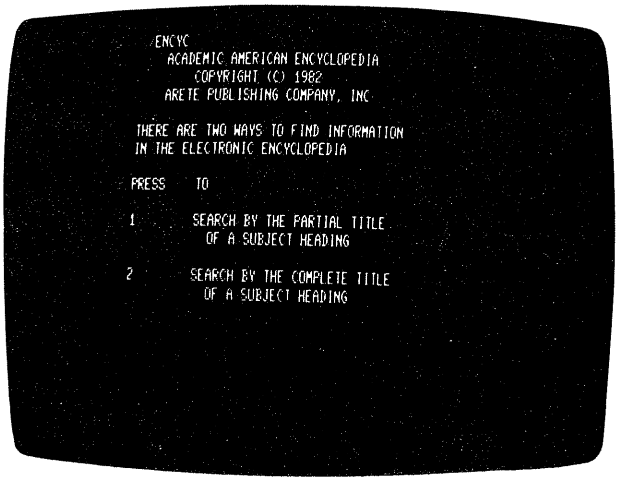
SEARCHING BY PARTIAL TITLE

Searching by partial title, the method called for in most cases (exceptions will be discussed below), is little different from turning to a subject in a familiar, printed encyclopedia.

If you were in a library reference section searching for information about automobile drag racing you would pick up the "A" volume and flip through the pages until you found the area of words whose beginning was AUTO, then AUTOM, then AUTOMO. Having narrowed your search area, you would look through headings beginning with AUTOMO until you found your chosen topic, automobile racing. Likewise with the electronic encyclopedia.

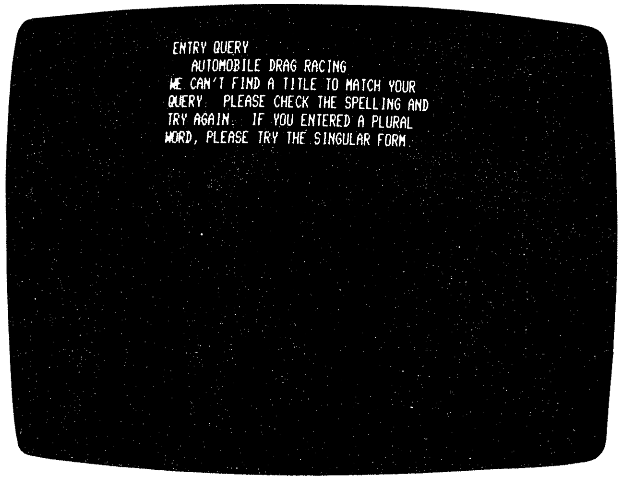
A SAMPLE SEARCH USING THE PARTIAL TITLE METHOD

After accessing the electronic encyclopedia you will be offered a choice of two search methods. Type 1 and press ⟨ENTER⟩ to search by the partial title.



```
/ENCYC
   ACADEMIC AMERICAN ENCYCLOPEDIA
      COPYRIGHT (C) 1982
   ARETE PUBLISHING COMPANY, INC

THERE ARE TWO WAYS TO FIND INFORMATION
IN THE ELECTRONIC ENCYCLOPEDIA

PRESS   TO

1       SEARCH BY THE PARTIAL TITLE
          OF A SUBJECT HEADING

2       SEARCH BY THE COMPLETE TITLE
          OF A SUBJECT HEADING
```

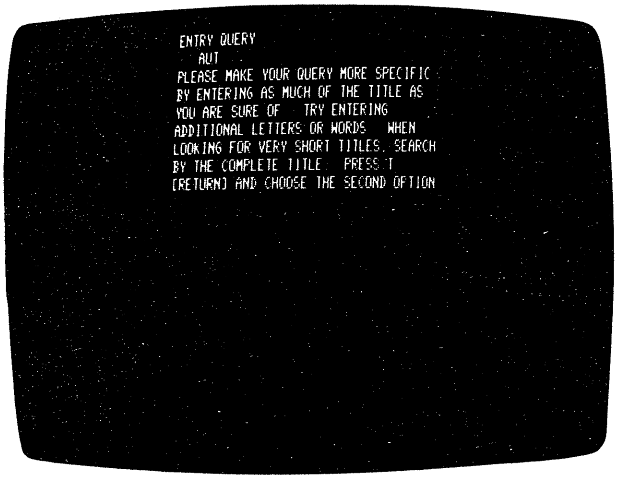
The display screen will show ENTER QUERY. If you type AUTOMOBILE DRAG RACING and press ⟨ENTER⟩, the display will say there is no title to match your query. This means your query either was too specific or doesn't have a main heading in the encyclopedia.



```
ENTRY QUERY
   AUTOMOBILE DRAG RACING
WE CAN'T FIND A TITLE TO MATCH YOUR
QUERY.  PLEASE CHECK THE SPELLING AND
TRY AGAIN.  IF YOU ENTERED A PLURAL
WORD, PLEASE TRY THE SINGULAR FORM.
```

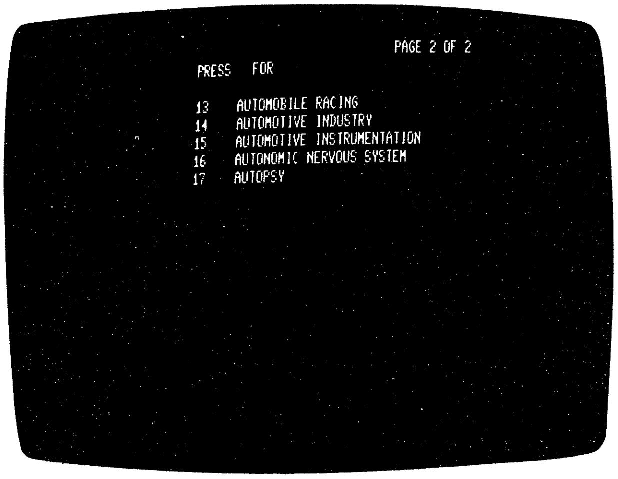
If you type only AUT (the beginning letters of the subject sought) and ⟨ENTER⟩ ,the display will request a more specific query because the range of titles beginning with AUT is too broad.

Add another letter in a new query—type AUTO and press ⟨ENTER⟩. The display then will show a numbered list of titles beginning with those letters, titles from AUTOWORKERS and AUTOBIOGRAPHY to AUTONOMIC NERVOUS SYSTEM and AUTOPSY. Altogether there are 17 titles in the list. AUTOMOBILE RACING, number 13, is the title



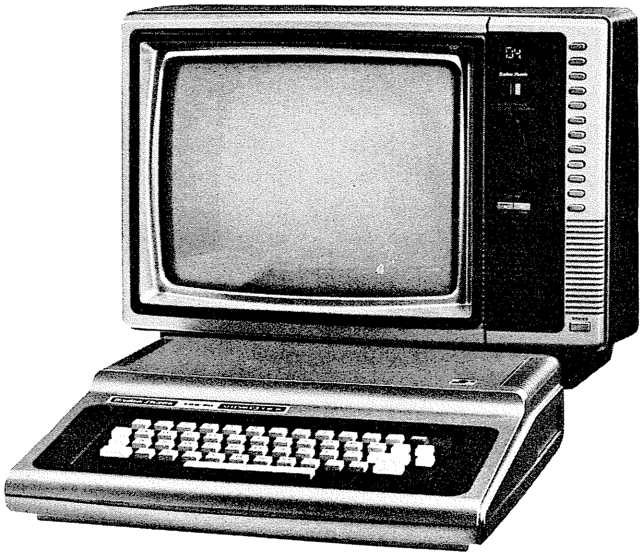
```
ENTRY QUERY
   AUT
PLEASE MAKE YOUR QUERY MORE SPECIFIC
BY ENTERING AS MUCH OF THE TITLE AS
YOU ARE SURE OF   TRY ENTERING
ADDITIONAL LETTERS OR WORDS   WHEN
LOOKING FOR VERY SHORT TITLES. SEARCH
BY THE COMPLETE TITLE   PRESS T
[RETURN] AND CHOOSE THE SECOND OPTION
```

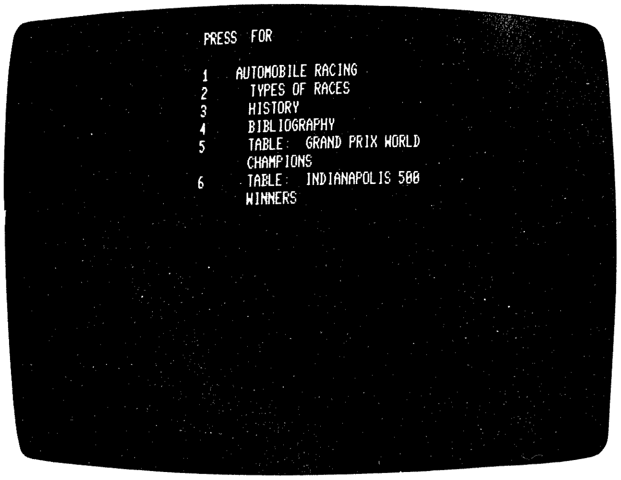
most likely to include information on automobile drag racing, the subject sought.



```
                              PAGE 2 OF 2
PRESS   FOR

13    AUTOMOBILE RACING
14    AUTOMOTIVE INDUSTRY
15    AUTOMOTIVE INSTRUMENTATION
16    AUTONOMIC NERVOUS SYSTEM
17    AUTOPSY
```

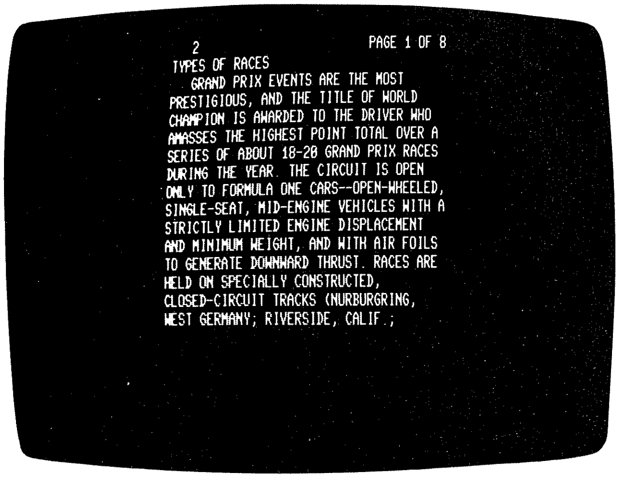
Type 13 and ⟨ENTER⟩. The display will show a list of six subheadings, number two of which is TYPES OF RACES.





```
PRESS  FOR

1   AUTOMOBILE RACING
2     TYPES OF RACES
3     HISTORY
4     BIBLIOGRAPHY
5     TABLE:  GRAND PRIX WORLD
      CHAMPIONS
6     TABLE:  INDIANAPOLIS 500
      WINNERS
```

Type 2 and ⟨ENTER⟩. The display will show page one of an eight-page article written under TYPES OF RACES. Press ⟨ENTER⟩ to proceed to the following page. Information on drag racing starts on page five.



```
  2                          PAGE 1 OF 8
TYPES OF RACES
  GRAND PRIX EVENTS ARE THE MOST
PRESTIGIOUS, AND THE TITLE OF WORLD
CHAMPION IS AWARDED TO THE DRIVER WHO
AMASSES THE HIGHEST POINT TOTAL OVER A
SERIES OF ABOUT 18-20 GRAND PRIX RACES
DURING THE YEAR. THE CIRCUIT IS OPEN
ONLY TO FORMULA ONE CARS--OPEN-WHEELED,
SINGLE-SEAT, MID-ENGINE VEHICLES WITH A
STRICTLY LIMITED ENGINE DISPLACEMENT
AND MINIMUM WEIGHT, AND WITH AIR FOILS
TO GENERATE DOWNWARD THRUST. RACES ARE
HELD ON SPECIALLY CONSTRUCTED,
CLOSED-CIRCUIT TRACKS (NURBURGRING,
WEST GERMANY; RIVERSIDE, CALIF.;
```

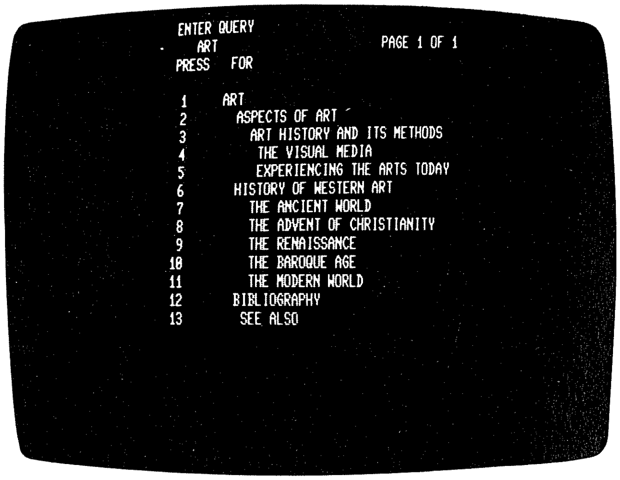
SEARCHING BY COMPLETE TITLE

Sometimes the commonly used partial-title search method won't work. For instance, if you type ART and press ⟨ENTER⟩ while using the partial title search method, the display screen will say PLEASE MAKE YOUR QUERY MORE SPECIFIC. It will go on to suggest that you enter additional letters or words or that you try searching by complete title, if the title of the subject you seek is very short.

Since you can be reasonably certain that most encyclopedias contain articles about ART, you wouldn't want to add any more letters or words to your search query. So in this case—trying to find a subject with a short title—you should try a complete title search, which is the second way to search for information in the electronic encyclopedia.

To look up "art" by using the complete title search method, you would have begun your search using the search by partial title search. Type T and press ⟨ENTER⟩ to get to the top encyclopedia menu to change your search method. When the display offers the two search methods, type 2 and press ⟨ENTER⟩ to search by the complete title. When the display shows ENTER QUERY, type ART and press ⟨ENTER⟩.

Under ART, the display will show a menu listing 13 different categories. Notice that two categories, ASPECTS OF ART and HISTORY OF WESTERN ART, are indented slightly. These are broad topics, corresponding to the upper-case letters in a conventional subject outline, which include information on the subcategories indented below them. If you want to read everything in the encyclopedia on art, including all the categories and subcategories listed on the menu as numbers 2 through 13, type 1 and press ⟨ENTER⟩.



```
ENTER QUERY
-   ART                             PAGE 1 OF 1
PRESS  FOR

 1    ART
 2      ASPECTS OF ART
 3        ART HISTORY AND ITS METHODS
 4         THE VISUAL MEDIA
 5         EXPERIENCING THE ARTS TODAY
 6      HISTORY OF WESTERN ART
 7        THE ANCIENT WORLD
 8        THE ADVENT OF CHRISTIANITY
 9        THE RENAISSANCE
10        THE BAROQUE AGE
11        THE MODERN WORLD
12      BIBLIOGRAPHY
13       SEE ALSO
```

If you want only the information specified by a certain category or subcategory, type the corresponding number and press ⟨ENTER⟩. If you were to type 6 and ⟨ENTER⟩, for HISTORY OF WESTERN ART, you would get information on all the subcategories indented below it: THE ANCIENT WORLD, THE ADVENT OF CHRISTIANITY, THE RENAISSANCE, THE BAROQUE AGE, and THE MODERN WORLD.

SOME BASIC COMMANDS

Press ⟨ENTER⟩ to go forward. This will always take you to the next page if there is more than one in a search.

Type R and press ⟨ENTER⟩ to go backward (reverse) to every page except the menu.

Type T and press ⟨ENTER⟩ to get to the top encyclopedia menu to change your search method.

Type Q and press ⟨ENTER⟩ to start a new search or enter a new query.

Type M and press ⟨ENTER⟩ to go to the previous menu.

Type P (NUMBER) and press ⟨ENTER⟩ to go to a specific page in an article. For example: type P12 and press ⟨ENTER⟩ to go to page 12.

Type //ENCYC HELP if you find yourself in trouble. It will display information that will help you use the electronic encyclopedia.

If you are not a Dow Jones News/Retrieval service subscriber but are interested in this or other data bases, call the Dow Jones Customer Service hotline number: 254-5114. It's toll free.

Coded Message

Peter L. Vogel
Notre Dame Regional Secondary
2855 Parker St.
Vancouver, B.C.
Canada
V5K 2T8

I have found David Snyder's Hex/ASCII problem, in the March 1982 Newsletter, to be an excellent exercise for students. The short program which I have written is a follow-up to Mr. Snyder's program.

This short program takes the concept one step further by automating the decoding process. A problem of this type would make an excellent short programming assignment.

The language used is Level II Microsoft BASIC.

```
100 REM HEX DECODER
110 DIM HEX$(15)
120 FOR I=0 TO 15
130 READ HEX$(I)
140 NEXT I
150 DATA 0,1,2,3,4,5,6,7,8,9,A.B.C,D,E,F
160 READ EX$
     : REM EX$=HEX EXPRESSION
170 FOR I=1 TO LEN(EX$) STEP 2
180 X1$=MID$(EX$,I.1)
     : X2$=MID$(EX$,I+1,1)
190 FOR J=0 TO 15
200 IF HEX$(J)=X1$ THEN 220
210 NEXT J
220 X1=J
230 FOR J=0 TO 15
240 IF HEX$(J)=X2$ THEN 260
250 NEXT J
260 X2=J
270 AC=X1*16+X2
     : REM AC=ASCII CODE
280 PRINT CHR$(AC);
290 NEXT I
300 DATA 434F4E47524154554C4154494F4E5321
```

SCRIPSIT Ideas

William Ashworth
201 Gresham
Ashland, OR 97520

The method you present, in the April 1982 Newsletter, of creating hanging indents for Model I/III is unnecessarily complicated and time-comsuming. A better way is as follows:

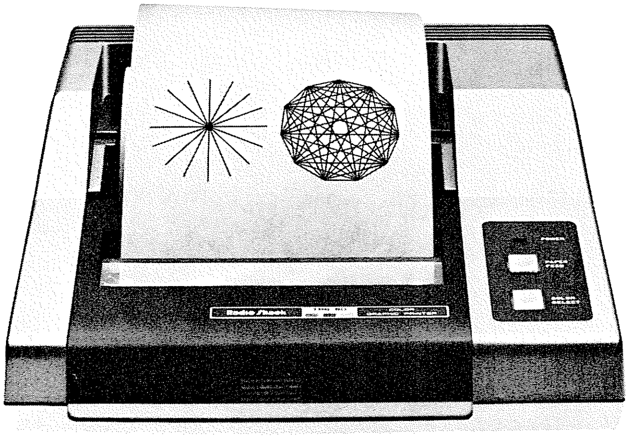
1. Set the cursor on the line BEFORE the line you wish to indent.
2. Use the TAB function to tab all the way to the right-hand end of the line. This prints spaces following the text right to the end of the 60-space line (or whatever your machine is set for).
3. Hit the space bar once. The space will print at the beginning of the next line, opening a new line and moving the text material beyond that point down one line.
4. TAB over to the spot where you want the indented material to begin, and punch CLEAR.

This process creates hanging indents very rapidly. It also avoids having to move the cursor from line to line manually to begin each new indent; after you have CLEARed the line into its indented location, the cursor is already positioned on the spot you want to begin Step One for working the next line.

Color Graphic Printer

When the Pocket Computer 2 was introduced in January of this year, one of the things people spent the most time looking at was the printer. The printer was small, compact and featured X-Y plotting and four colors. You could tell that many of these people were wondering how to interface the printer to their larger computers.

With the introduction of the Color Graphic Printer (CGP-115 26-1192 Suggested Retail Price $249.95) Radio Shack is providing the same technology with a wide 4 1/2" paper and dual interfaces (serial or parallel). The CGP-115 uses standard LPRINT commands on Models I/III, II, 16, and uses PRINT#-2 for Color Computers. The CGP-115 can also be used with the DT-1 Data Terminal.



OPERATION MODES

The CGP-115 has two operating modes, Text and Graphics as well as a self-test mode. The self-test mode is entered holding down the Paper Feed switch while turning on the printer.

TEXT MODE

This is the mode the printer powers up in. In the text mode, the CGP-115 print density is determined by a DIP switch which is read by the printer when the power is turned on.

Using the DIP switch (with power off), you can select either 40 or 80 characters per 4" line.

The text mode is used for note writing, program listings and most program output. In this mode, you can use the following commands to control the printer:

CHR$(8) - Backspace one character
CHR$(11) - Reverse Line Feed
CHR$(18) - Select Graphic Mode
CHR$(29) - Change Pen color

GRAPHIC MODE

In the Graphic mode, the following commands are available:

CHR$(17) - Select Text Mode

A - Reset Pen. This moves the pen back to the left margin without drawing a line. The "A" command also selects the Text mode.

Cn - Change Pen Color to the one specified by 'n'. In Text mode you must keep track of the current pen color since your only color command is to rotate the pen to the next color. In the Graphic mode, you specify which of the four colors (0-3) you want.

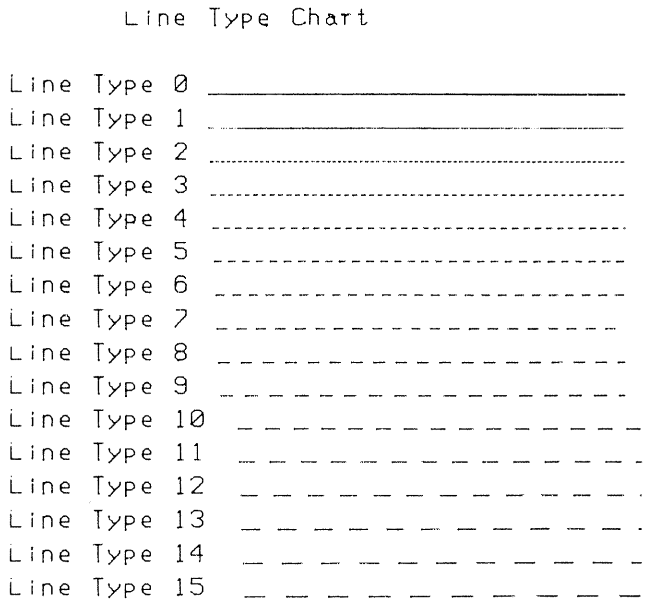
Dx,y - Draw a line from the current pen location to the X,Y coordinate location specified by 'x,y'. Multiple points can be specified, resulting in connected lines.

H - Moves the pen to the current origin without drawing a line.

I - Sets the current Pen location as the new origin.

Jx,y - Draws a line from the current Pen position to a point 'x' horizontal steps and 'y' vertical steps away. If 'x' is positive, the Pen moves to the right and if 'x' is negative, the Pen moves left. A positive 'y' moves up, while a negative 'y' moves down the page. Multiple 'x,y' destinations can be specified with a single 'J' command.

Ln - Change the line type to the type specified by 'n'. 'n' can be a value from 0 (solid) to 15.



Line Type Chart

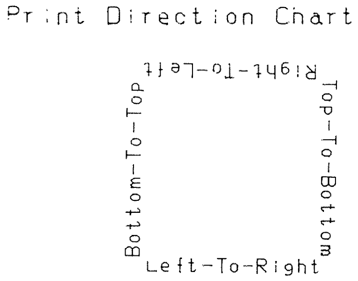
Line Type 0
Line Type 1
Line Type 2
Line Type 3
Line Type 4
Line Type 5
Line Type 6
Line Type 7
Line Type 8
Line Type 9
Line Type 10
Line Type 11
Line Type 12
Line Type 13
Line Type 14
Line Type 15

```
20000 LPRINT"        Line Type Chart"
20010 LPRINT
20020 FOR I=0 TO 15
20025 LPRINT CHR$(18)
20030 LPRINT"PLine Type";I
20050 LPRINT "I"
20060 LPRINT "L";I
20070 LPRINT "D300,0"
20090 LPRINT CHR$(17)
20110 NEXT
```

Mx,y - Move (without drawing a line) to a position 'x' horizontal steps and 'y' vertical steps from the current origin. Note that this move to an absolute position based on current origin, not on the current pen position.

Ptext - Print the text which follows the 'P' starting from the current pen location, using current character size and direction.

Qn - Change the print direction. 'n' can be a value from 0 to 3:
0 - Normal left-to-right
1 - Top-to-bottom
2 - Upside down, right-to-left
3 - Bottom-to-top

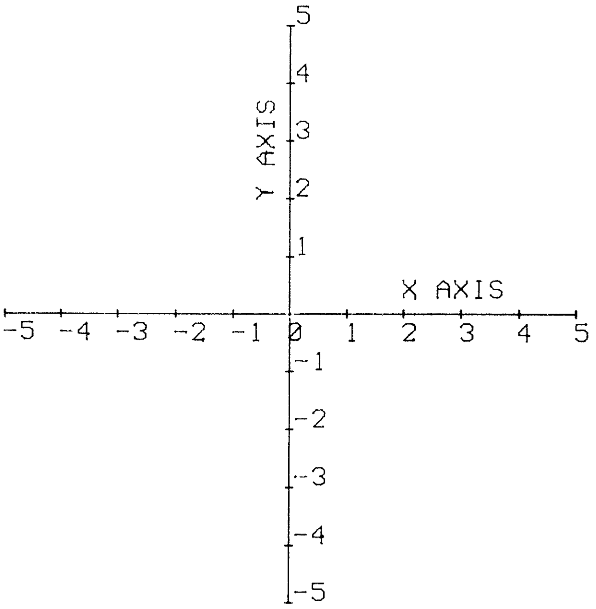


```
30000 LPRINT"      Print Direction Chart"
30010 LPRINT STRING$(3,13)
30020 LPRINT CHR$(18)
30030 LPRINT "M200,-100"
30040 LPRINT "I"
30050 LPRINT "M-50,-50"
30060 LPRINT "Q0"
30070 LPRINT "PLeft-To-Right"
30080 LPRINT "Q1"
30090 LPRINT "M100,120"
30100 LPRINT "PTop-To-Bottom"
30110 LPRINT "Q2"
30120 LPRINT "M95,135"
30130 LPRINT "PRight-To-Left"
30140 LPRINT "Q3"
30150 LPRINT "M-55,-35"
30160 LPRINT "PBottom-To-Top"
30170 LPRINT "Q0"
30180 LPRINT "A"
30185 LPRINTSTRING$(10,13)
30190 LLIST 30000-
```

Rx,y - Move (without drawing a line) to a position 'x' horizontal steps and 'y' vertical steps from the current Pen position. Note that this is a move relative to the current position, not the currently defined origin.

Sn - Sets the current character size for use with the 'P' command. 'n' can be a value from 0 (80 CPL) to 15 (5 CPL). To determine how many characters per line a given value of 'n' yields, use the following formula:
CPL = INT(80/(n + 1))
where CPL stands for Characters Per Line.

Xa,s,i - The 'X' command is used to draw an X-Y axis from the current Pen position. 'a' specifies which axis (0 = Y, 1 = X) is to be drawn. 's' specifies the number of steps between interval marks, and can have a value between -999 and 999. 'i' specifies how many intervals there will be (how many times 'step' is to be repeated), and can have a value between 1 and 255. If 's' is negative, an X axis would be drawn to the left, or a Y axis would be drawn down from the current Pen position.



Here is the program that produced this chart:

```
10 REM X-Y AXIS TO BE DRAWN
20 LPRINT: LPRINT CHR$(18): REM GRAPHIC C
30 LPRINT "M240,-240" : REM MOVE
40 LPRINT "I" : REM SET NEW ORIGIN
50 LPRINT "X1,40,5" : REM RT HALF
60 LPRINT "H" : REM RETURN TO ORIGIN
70 LPRINT "X1,-40,5" : REM LF HALF
80 LPRINT "H"
90 LPRINT "X0,40,5" : REM TOP HALF
100 LPRINT "H"
110 LPRINT "X0,-40,5" : REM BTM HALF
120 FOR I=-200 TO 200 STEP 40
```

```
13Ø LPRINT "M"; I; ",-2Ø" : REM LABEL X
14Ø READ A$ : REM READ LABEL
15Ø LPRINT "P" A$ : REM PRINT LABEL
153 IF A$="Ø" THEN 16Ø : REM 1 ZERO
155 LPRINT "M5,"; I : REM LABEL Y AXIS
157 LPRINT "P" A$ : REM PRINT LABEL
16Ø NEXT
17Ø LPRINT "M8Ø,1Ø" : REM MOVE
18Ø LPRINT "PX AXIS" : REM PRINT
19Ø LPRINT "M-1Ø,8Ø" : REM MOVE
2ØØ LPRINT "Q3" : REM ROTATE PRINTING
21Ø LPRINT "PY AXIS" : REM PRINT
22Ø LPRINT "MØ,-25Ø"
999 LPRINT "A" :REM RESET, MOVE LEFT
1ØØØ DATA -5, -4, -3, -2, -1, Ø
1Ø1Ø DATA 1, 2, 3, 4, 5
```

SPECIFICATIONS

Plotting/Printing System	- 4 Ball Point Pens
Plotting Speed	
Horizontal	- 52mm/sec (2.05 ips)
Vertical	- 73mm/sec (3.8 ips)
Printing Speed	- 12 Characters/sec
Resolution	- 0.2 mm/step
Effective plotting range	- X axis - 96mm (480 steps) - Y axis - Length of the paper
Characters Per Line	
Text Mode	- 40 or 80
Graphic Mode	- 5 to 80
Accuracy	
Repetition	- 0.2mm max
Movement	- 0.3mm max
Distance	- X axis 0.5% max Y axis 1% max
Dimensions	
Width	- 210mm (8.4 in)
Depth	- 216mm (8.64 in)
Height	- 75mm (3 in)
Pens	
3 Black Pens	- Radio Shack 26-1480
1ea Blue,Green,Red	- Radio Shack 26-1481
Pen Life	- 250 meters (825 ft)
Paper	- Radio Shack 26-1428
Parallel Interface	- 8-bit parallel BUSY handshaking, STROBE, and ACKNOWLEDGE
Serial Interface	- RS-232C Using DATA and BUSY. 600 Baud, 7-bit, no parity, two stop bits

Heart Health Programming Contest

In keeping with its reputation as a leader in promoting heart health education in the young, the American Heart Association, Greater Boston Division, is taking a new approach to education—an electronic approach. The American Heart Association in conjunction with **Classroom Computer News** has announced a "Heart Health Computer Programming Contest" to solicit the development of programs on health education for eventual national distribution to schools.

Programs should be aimed at an elementary, junior high or high school audience, and should deal with ways in which heart disease can be prevented. The Surgeon General of the United States has recommended prevention of heart disease as a national priority and the Heart Association has started its campaign to educate youngsters about the importance of good habits early in life.

The best program entry has the potential to receive $1500.00; eleven other entries may receive $500.00. Programs are to be written in BASIC language to run on the Apple, Atari, TRS-80, Texas Instruments or the Commodore PET microcomputers. All entries must be received by January 31, 1983. A panel of computer and education experts will act as judges and reach a decision by Spring 1983.

For further information, write to "Heart Health Computer Programming Contest", American Heart Association, Greater Boston Division, 33 Fourth Avenue, Needham, Massachusetts, 02194, before December 1, 1982.

Firewood Calculator

Gary A. Davis
Sugar Software
2153 Leah Lane
Reynoldsburg, Ohio 43063

I have written a program for the color computer which determines how many cubic feet or cords of firewood you have stacked up. Measure your stacks and input the measurements in feet and inches.

```
1Ø  '     FIREWOOD CALCULATOR
2Ø  '
3Ø  '    WRITTEN BY GARY DAVIS
4Ø  ' (C) 1982 BY SUGAR SOFTWARE
5Ø  '       2153 LEAH LANE
6Ø  ' REYNOLDSBURG, OHIO   43Ø68
7Ø  '
8Ø  CLS4
9Ø  PRINT @38,"FIREWOOD CALCULATOR";
1ØØ PRINT @96,"YOU MUST ENTER BOTH FEET AND"
11Ø PRINT "INCHES (SEPARATED BY A COMMA)"
12Ø PRINT "FOR EACH PROMPT."
13Ø PRINT @224,"ENTER WIDTH"
14Ø PRINT "ENTER LENGTH"
15Ø PRINT "ENTER HEIGHT"
16Ø PRINT @236,;
    :INPUT WF.WI
17Ø PRINT @268,;
    :INPUT LF.LI
18Ø PRINT @3ØØ,;
    :INPUT HF.HI
19Ø CF=(WF+WI/12)*(LF+LI/12)*(HF+HI/12)+CF
2ØØ PRINT @384,;
21Ø PRINT USING "####.# CUBIC FEET, ##.#
    CORDS";CF,CF/128
22Ø GOTO 13Ø
```

Profile III+: Introducing The Flexible Data Base Management System

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

Pentacle is a New York City-based non-profit service organization specializing in administrative services such as bookkeeping, grant-writing, tour management, public relations and booking representation for performing arts groups. We keep our member companies going while their artists are on stage, on tour, or in the rehearsal studio.

We've been running PROFILE III+ on our two-disk Model III (48K) for more than six months. So far, the program has made itself indispensable in two ways: 1) By enabling us to do certain jobs more efficiently, our small staff has been able to accomplish more for each member and at the same time offer our services to new members. 2) We have been able to create new services that we simply couldn't handle before.

We've used PROFILE III+ to create a half-dozen completely different files to serve different needs. Along the way we've learned a few tricks which should prove useful in other applications. I'll describe them as I go along.

WHY WE USE PROFILE III+

Mailings: One of our jobs is to send out press releases about the companies' performances and activities to a press-list of over 400 critics and publications. We used to use copier labels, which are expensive and look sloppy, especially after a few months of accumulated additions, deletions and corrections. PROFILE III+ lets us keep the list "clean" in both senses of the word. Saturation mailings which can be sent bulk-rate come out in zip code order, ready to go to the Post Office.

And we can use the selection feature to target sub-sets of the complete list for first-class mailings, saving money because there are no unused labels.

Trick: We use a four-character key field for each press-list record. The first character of the four identifies the entry as dance, theater or music press. The second identifies the entry as a monthly, weekly, daily and so forth—important because these different types receive releases according to different rhythms. The third character identifies the addressee as being on the "A" or "B" list or as accepting photos. The last character distinguishes between three- and four-line addresses so that the labels look their best. We do two runs—first we use the selection operation to pull out all the four-line addresses, then we use it to pull out the three-line addresses. We can select with incredible flexibility using the "wild card" (=) character: To send information to the entire "A" list, we simply type ==A= or ==A (in this case, the computer treats a blank and the wild-card symbol following the selection character the same way).

PROFILE III+ works so efficiently as a mailing list generator that it is now possible for us to manage the member companies' individual mailing lists for them, tailoring the lists to their particular needs—people who take dance classes, VIPs who might donate money, people who helped in a special project. The password protection offered by PROFILE III+ makes the companies confident in the security of this precious resource.

Fund Raising: We've made a special file for our own fund-raising solicitations, and that's password-protected to the hilt! PROFILE III+'s math capability lets us tally an individual's contributions during the year and then total them easily for end-of-year financial statements. People who donate certain amounts are entitled to various perks, and the PROFILE sort feature makes it easy to pluck them out for special mailings.

Trick: PROFILE III+ can interface with Radio Shack's new SuperSCRIPSIT to create personalized letters to these very important people. We anticipated this from the beginning, devoting a field to a salutation appropriate to each individual. It's silly to 'individualize' a letter to Mrs. Smith when it's Aunt Millie or an old acquaintance you're writing to.

Scheduling: Another important part of our work is to keep track of our companies' performing activity and schedules, both past and future. As we deal with over 30 groups, B.P. (Before Profile) it was very tedious to collect and arrange this information every time we had to do a grant application or final report. Each public funding agency requires a different assortment of these statistics. We designed a PROFILE III+ file which tells us the dates of each tour, the name of the theater, the fee paid, the type of performances and workshops involved and the attendance figures. PROFILE's flexible report formats let us tailor print-outs for each grant application.

Trick: By using the YY/MM/DD format, the tours come out in chronological order, and it's simple to summarize the information for a particular period or fiscal year.

Booking: Our first application of PROFILE III+, and the reason we haunted our local dealer until the program was available, is the most important of all. We try to get bookings for our touring groups, which means keeping track of hundreds of potential presenters across the country and abroad. Calls go in and out all the time, interruptions interrupt other interruptions, crucial information (leads, contacts, phone numbers possible dates and fees) gets scribbled on pieces of paper. Even a mild breeze could spell wasted effort or disaster.

PROFILE III+ lets us keep track of these bits of information, and nothing except a direct hit by a tornado could scatter them again. The information is all in one place, legible and accurate; plus we always know we're looking at the latest data. Not only does this system insure that I know what I'm talking about when I'm dealing with presenters and company heads, but it makes it possible to use assistants more effectively because the information is ordered and legible.

Here's an application which really flexes PROFILE's muscles—keeping on top of the hundreds of presenters interested in one or more of over 20 performing groups. A trick we learned from the airlines is to reduce each company's name to a three-letter airport-style code. For instance, James Cunningham's Acme Company becomes ACM. Then we set up associated fields to hold the codes of the company or companies in which the presenter is interested. So long as ACM is in any one of these associated fields, the record will surface when needed.

DESIGNING YOUR FILES

It's important to think through your file design carefully—the key fields especially, since later you will use them to sort and select records. In planning our presenter file, we first listed all the types of information we wanted to have on file and made preliminary assessments of the field lengths required for each.

Use Worksheets: This is also the time to anticipate screen, label and especially report formats. A good idea is to block out the formats on the video/program worksheets available from Radio Shack (cat. no. 26-2105). The program side of the worksheets has exactly 80 spaces across, counting the one between the listing and the variable sections—perfect for laying out report formats for your 80-column printer. Using the format sheets, you might find that by shortening a name field one or two characters, you'll have room for both address and phone number on the same line. (Don't forget to allow a space between fields.)

Choose Key Fields Carefully: We made the area code a key field because it's easier to use for geographic targeting than zip codes. The phone number itself is relegated to another segment. As for zip code, we allocated 10 characters—the new nine-digit codes have a dash. Also, since we deal with foreign presenters, we can use the zip field for country names (in this case, use the alphanumeric instead of the numeric indicator for the zip code field on the screen format). This works perfectly for both labels and sorted reports.

Our key segment has 34 fields! This means that I can walk up to my Model III and say, "Spit out mailing labels for all directors of student activities in Ohio and Illinois who booked the Acme Company in the last five years, and then send them personalized letters offering special rates on a return engagement." As long as I'm typing in the instructions at the same time as I say them, the computer will do exactly that.

This power and flexibility gets turned into increased opportunities for artists to perform—which is what it's all about. PROFILE III+ can do the same for your business.

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

ACCESSING PROFILE DATA FROM AN INDEX SEQUENTIALLY

In this part of our column, we'll show you how to move data from Model II PROFILE into BASIC programs.

We've written a program that will let you access PROFILE data sequentially from a PROFILE index for use with BASIC programs. You can also use the program as a module, building a BASIC program around it. The program shown is a demonstration program only—you should adapt it for your own purposes. Some possible uses:

*You have an inventory analysis program written in BASIC and you want to pull out data from your PROFILE stock file.

*You have a drug side-effects BASIC program and you want to access PROFILE patient information.

*Before you started your business, you analyzed your expected customer base; now you want to analyze your actual customer base with the same program to check the accuracy of your projections.

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

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

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

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

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

If you want an entire record, you can then access it by typing (between lines 1600 and 1700; make sure you have a third file):

```
GET 3, LR (where "LR" is the logical record)
```

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

```
1Ø  FIND/BAS
2Ø  The small Computer Company, Inc.
3Ø
4Ø  Index demo program
5Ø  Sequential access of PROFILE+ records from
     BASIC using a PROFILE+ index
6Ø
7Ø
1ØØ CLS : CLEAR 5ØØØ : DEFINT A-Z
2ØØ INPUT " ENTER FILE NAME"; F$
3ØØ F$ = LEFT$(F$+"ØØØØØØØØ",8) 'Pad with zeros to
     make 8-byte file name
4ØØ INPUT " ENTER KEY LENGTH"; KY
5ØØ OPEN "D", 1, F$ + "/IX1", KY+2 'Open index file
6ØØ FIELD 1, (KY) AS KY$, 2 AS LR$ 'Key field and

     two-byte pointer
```

```
700 OPEN "D", 2, F$ + "/KEY" 'Open /KEY segment
800 '
900 '
1000 FOR R= 4 TO LOF(1) 'Start of index is record #4
1100 GET 1, R ' Read the index
1200 LR = CVI(LR$) 'PROFILE+ record number from index
     pointer
1300 PR = INT((LR-1)/3) 'Deblock the KEY segment
     record:
1400 SR = LR - PR*3 - 1 'SR = sub-record number (0, 1
     or 2)
1500 PR = PR+1 'PR = physical record number
1600 GET 2, PR 'Get the /KEY record
1700 FIELD 2, (85*SR) AS DD$, 85 AS A$ 'A$ = /KEY
     segment data fields
1800 PRINT : PRINT LR; "  "; A$ 'Print record number
     and data
1900 NEXT R ' Advance pointer to next index record
2000 CLOSE
```

LETTERS

A few of our readers have written in with a similar problem: They've added a segment or segments to an existing file, rewritten their screens to accept the new fields, then found that the records wouldn't accept data in those fields when they tried to update records.

The answer is to expand the file to accept the new segment. However, you don't have to add any records. When the prompt "This File Is Currently Allocated nnnn Records. How Many Additional Records . . . ." comes up, simply press ENTER.

microRESEARCH Problems

Richard V. Andree
microRESEARCH
P.O. Box 2910
Fort Worth, Texas 76113-2910

Before we go on to a further discussion of Problem 2, here is another program which solves Problem 1. It was written by M. Harris, a high school student.

```
5 CLS
10 DEFDBL T,X,Y
14 PRINT "THIS PROGRAM FINDS SQUARES THAT CONTAIN
     EACH OF THE
15 PRINT "NINE NON-ZERO DIGITS ONCE AND ONLY ONCE
16 LPRINT "THIS PROGRAM FINDS SQUARES THAT
     CONTAIN EACH OF THE
17 LPRINT "NINE NON-ZERO DIGITS ONCE AND ONLY
     ONCE
18 FOR J=1 TO 1000
     : NEXT
     : CLS
19 LPRINT
20 X=11810
25 PRINT " NUMBER"," SQUARE"," NUMBER"," SQUARE"
26 LPRINT " NUMBER"," SQUARE"
27 PRINT
     : LPRINT
29 REM BEGINNING OF MAIN PROGRAM
30 Y=X*X
34 J=1
35 R=0
36 K=1
37 S=0
40 Z$=STR$(Y)
45 T=LEN(Z$)
46 REM THIS PART CHECKS TO SEE IF THERE ARE JUST
47 REM NINE WHOLE NUMBERS IN THE SQUARE
48 FOR I=1 TO T
49 L$=MID$(Z$,I,1)
50 IF VAL(L$)=0 THEN S=S+1
52 NEXT I
53 IF T-S<>9 THEN GOTO 120
55 IF K=T THEN K=1
     : GO TO 65
56 REM IF THERE ARE NINE WHOLE NUMBERS THEN THIS
57 REM PART CHECKS TO SEE IF THEY ARE THE NUMBERS
     1-9
60 IF K=T THEN K=1
     : J=J+1
65 IF J=10 THEN GOTO 120
70 K=K+1
80 L$=MID$(Z$,K,1)
90 IF VAL(L$)=J THEN R=R+1
     : J=J+1
     : K=1
     : GOTO 55
95 GOTO 60
110 IF THE NINE WHOLE NUBERS ARE 1-9 THE NUMBER
     IS PRINTED
120 IF R=9 THEN PRINT X,Z$,
125 IF R=9 THEN LPRINT X,Z$
130 X=X+1
140 GOTO 30
```

Last month's Problem 2 suggested a hunt for palindromic squares having an even number of digits. Mathematically inclined readers would cut their computer time by first proving that if S = N*N is a palindrome containing an even *number* of digits, then S, and hence N, must be divisible by 11. One proof uses a test for divisibility by eleven which forms the sum SE of all digits in even positions of S and the sum SO of all digits in odd positions of S. Then, if the absolute value of SE-SO is zero or is divisible by 11, the original number S is divisible by eleven. In a palindromic number having an even number of digits this sum is zero or divisible by 11. Hence 11 divides S. Since S = N*N and 11 is prime, 11 also divides N. This observation can save 90% of your computer time.

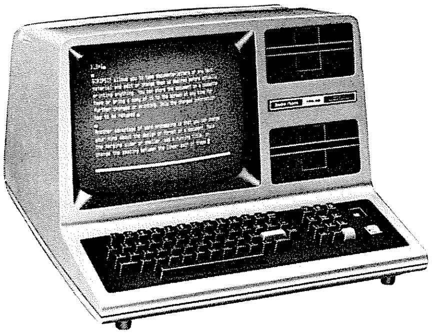
$(836)^2$ = 698896 has six digits
$(798644)^2$ = 637832238736 has twelve digits
$(\qquad)^2$ = 14 digits

microRESEARCH Problem #3:

The number 371 has the property that the sum of the cubes of its digits equals 371: $3^3 + 7^3 + 1^3 = 27 + 343 + 1 = 371$. Find other whole numbers N such that the sum of the cubes of the digits of N is equal to N.

There are four such numbers among the 3-digit numbers. Mathematically inclined readers may wish to show that there cannot be such an N with five (5) or more digits and then find *all* integers N such that the sum of the cubes of the digits of N is equal to N.

Possible extensions: for "cubes" read "squares" or "fourthpowers" or "k-th powers" or "factorials".



Model I Double Density

by Bruce Elliott

The Model I Double Density Upgrade (26-1143) is here, and we just had the two Microcomputer News Model I's upgraded. I have not had a lot of time to work with the system, but I would like to familiarize you with some of the new power that this upgrade brings to the Model I.

THE OBLIGATORY CAUTION:

While the double density option for Model I brings new power and storage space, it is important that you recognize that Radio Shack applications software is not available for double density. Further, Radio Shack can only support its programs on the TRSDOS version they are released on. This means we can not support applications programs which you move to double density.

Two additional cautions: First, while Model I Double Density TRSDOS is comparable to the Model III TRSDOS in many ways, Model I TRSDOS still cannot read a Model III diskette, and Model III still cannot read a Model I diskette directly. The Model III CONVERT utility has no provision for converting Model I double density diskettes.

Second, while Model I Double Density has the ability to format and use 40 tracks in double density, not all disk drives are able to support 40 track operations. Radio Shack Model I disk drives which are capable of supporting 40 track operations are designated by a '-1' after the serial number.

NOW THAT THAT IS OUT OF THE WAY . . .

One of the first questions I asked about the Double Density (DD) system was "What happens to all of my single density material?" The answer was both simple and pleasing. Once the DD system is installed, your Model I is able to detect which density is being used, and it will respond appropriately. This means that if you use TRSDOS 2.3 (single density—SD) your Model I will operate in single density just like it always did. If you use TRSDOS 2.7DD (double density) to boot with, your system will operate in double density.

TRSDOS 2.7DD will let you do two things with a single density diskette: look at the directory (DIR) and copy information from SD to DD or from DD to SD.

SPECIFICATIONS

Diskettes

Double Density diskettes are required for proper operation of the Double Density Adapter. Sounds reasonable, but how can I tell a SD diskette from a DD? Radio Shack 5¼" DD diskettes can be identified by a "hub ring" in the center of the diskette. The regular Radio Shack diskettes (26-305, 26-405, 26-406) are certified for double density operations.

Diskette Organization

Formatted Diskette

Double Density	—	Single Sided 256 Bytes/Sector 18 Sectors/Track 6 Granules/Track 35 or 40 Tracks
Formatted 35 Track Data Diskette	—	198 Grans
TRSDOS 35 Track without BASIC	—	128 Grans
with BASIC	—	120 Grans
Single Density	—	Single Sided 256 Bytes/Sector 10 Sectors/Track 2 Granules/Track 35 Tracks

MEMORY USAGE DIFFERENCES

In general, both the single and double density systems use lower memory the same way. The major exception is DEBUG. In SD, DEBUG loads below 5200 hex, and addresses above 5200H will not be affected by the operation of DEBUG. In the DD system, DEBUG loads from 4E00H to 53FFH. Double Density DEBUG can be used for memory between 5400H and TOP TOP is the memory location defined by the system as being the TOP of available memory. The value of TOP changes depending on what program files are loaded. In general, TOP will be above B000 in a 32K Model I and above F000 in a 48K Model I.

The following memory map gives an indication of how to view memory with the DD system operating:

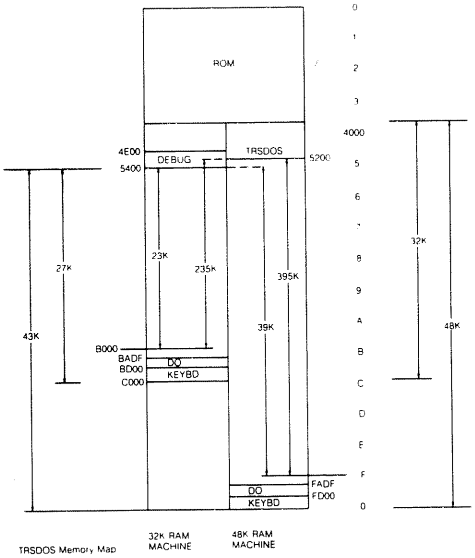


Figure 1. TRSDOS 2.7DD Memory Map

Four types of program files are defined for DD:

The Executive Program, which is resident in memory when TRSDOS is operating.

The Auxiliary System Files, which are the routines and commands which are needed to execute your commands and programs. These files load into an "overlay" area, and are present only when needed. These files will not affect memory above 51FFH.

The Library Command Files, which execute most of the operator commands. These files load into memory addresses from 5200H to 6FFFH. Because of the Library Files, machine language programs should generally be located above 6FFFH. Three library commands use ALL available RAM - BACKUP, FORMAT, and single drive COPY.

The Disk BASIC extensions to Level II BASIC. This file loads into memory beginning at 5200H.

In addition to low RAM, TRSDOS 2.7DD also uses some high memory locations. The keyboard driver routine resides above BD00H in a 32K system, and above FD00H in a 48K system. Also, DO uses memory above BADFH (32K) or FADFH (48K).

DOUBLE DENSITY EXTENSIONS TO TRSDOS

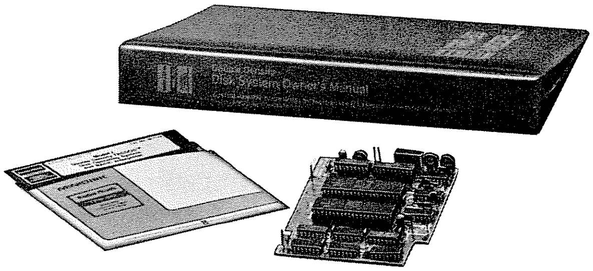
The following DD TRSDOS commands have been added beyond those commands which exist in SD Model I TRSDOS:

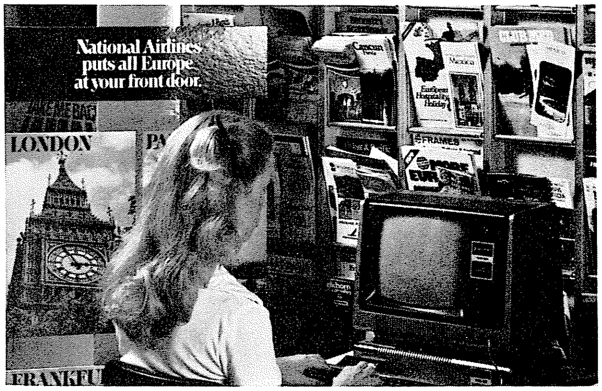
BLINK — Turn blinking cursor ON or OFF.
BUILD — Create an automatic command input file.
CLEAR — Clear user memory between start and end addresses, as well as setting the memory protect address.
CLS — Clear screen from TRSDOS.
CONFIG — Temporarily change the drive stepping rate and specify the number of tracks for a drive (if they can be changed for that drive.)
CREATE — Create a Pre-allocated disk file.
DO — Begin Auto command input from a BUILD file.
DUAL — Duplicate output to Video and Printer.
ERASE — Erase a file from a diskette by writing zeros in the data area.
ERROR — Display error message.
FILFIX — Load and modify the contents of a disk file using a full screen display.
FREE — Displays a disk allocation map.
HELP — Explanations of TRSDOS commands.
LPC — Line Printer Control program.
MASTER — Set Master read/write drive.
MEMTEST — Test ROM and RAM memory.
PATCH — Change the contents of a disk file.
PAUSE — Pause execution during execution of a BUILD file.
PURGE — Delete disk files.
RELO — Change where a machine language program loads into memory.
SETCOM — Initializes the RS-232C communications channel (if the RS-232 board is installed.)
SPOOL — Create an area in memory where output for the printer is dumped. Allows operation of the printer and some other operation
TAPE — Transfer machine language programs between tape, disk and RAM.
ULC — DD TRSDOS will automatically load the Upper/Lower case driver if the lower case modification is available.
UNKILL — Recover a KILLed file, if the directory and disk space have not been written over since the KILL command. You can not UNKILL an ERASEd file.
USER — Allows you to add a single machine language routine to the TRSDOS library.
WP — Write protect a disk drive via software.

DOUBLE DENSITY ENHANCEMENTS TO DISK BASIC

The following enhancements have been made to Disk BASIC:

BASIC filename -F: -M: When loading Disk BASIC, you can now specify a program to be executed (filename), you can specify the number of files that may be OPEN at one time (-F:) and you can specify the highest address available for use by BASIC (-M:).
CMD"A" — Return to TRSDOS with an error message.
CMD"B" — Enable or Disable the ⟨BREAK⟩ key.
CMD"C" — Compress the BASIC program in memory by removing spaces and remarks.
CMD"D" — Display the directory for the specified drive.
CMD"E" — Display previous TRSDOS error.
CMD"J" — Convert calendar dates.
CMD"K" — Turn CLOCK display on or off.
CMD"L" — Load a Z-80 subroutine or program file into RAM.
CMD"O" — Alphabetize (sort) a string array.
CMD"P" — Check printer status.
CMD"X" — Cross-reference of reserved words, string variables or strings in a program.
CMD"Z" — Duplicate output to video and printer.





A Baker's Dozen of Helpful Hints

**John F. Rogers
600 Seventh Street
Morgan City, LA 70380**

I own two 48K multi-disk Model I systems with printers. Since Spring 1978 I have been using my systems in various ways, read many magazines, talked to many other users, and learned quite a lot. (I'm just a beginner with assembly language, however.)

Although many tips have been published in Microcomputer News and in other magazines, perhaps some of the enclosed ideas have been missed along the way by enough of your readers that publishing them again would be helpful.

1. When using INKEY$ in a loop to read the keyboard, be sure to compare the string variable to a length of zero—i.e., use A$ = "", not A$ = ƀ (where ƀ represents a space.) Example:

```
1ØØ A$=INKEY$
   : IF A$="" THEN 1ØØ
```

will loop at line 100 until a key is pressed.

2. The opposite of the above is true for using LPRINT to create a blank line — i.e., use LPRINT "ƀ", not LPRINT"" to advance the printer one line. Example:

```
1ØØ LPRINT"LINE ONE"
   : LPRINT" "
   : LPRINT"LINE THREE"
```

Editors note: This is especially true in early Radio Shack printers. The newer printers will respond to LPRINT by printing the contents of the buffer (if any) and advancing one line. Thus if the print buffer is empty, sending just an LPRINT will result in a blank line.

3. In 32-character mode, PRINT@x must use an even number for 'x' — i.e., PRINTCHR$ (23): PRINT@21,"DOUBLE WIDE" will not print anything that is visible on the screen. Example:

```
1ØØ PRINTCHR$(23)
   : PRINT@74, "DOUBLE WIDE SCREEN PRINT"
```

4. When printing several items on the same line, it is not necessary to put the semicolon (;) between them — i.e., PRINT "PART NUMBER"; PN does not need the semicolon. Example: If X = 2, then

```
PRINT"SECTION ONE"X"SECTION THREE"
```

will print SECTION ONE 2 SECTION THREE on the same line. (The semicolon is still needed after the last element in a print statement if you want to suppress the cursor—dropdown.)

5. PRINT@p,STRING$ (n,c) is much faster than POKE for horizontal graphics. Example:

```
PRINT@64,STRING$(64,131)
```

will print a horizontal line in the blink of an eye, while

```
FOR I=15424 TO 15487
   : POKE I,131
   : NEXT I
```

allows one to see the line being formed, even with DEFINT I to speed up the loop.

6. The Reference Manual gives the impression that PRINT@p, allows only integer values for 'p', but variables may also be used for location 'p'. Example:

```
FOR J=192 TO 56Ø STEP 16
   : PRINT@J, "LOCATION" J
   : NEXT J
```

will print LOCATION 192, etc., four across on six lines starting with the fourth line.

7. LPRINTTAB(x) can go past TAB(63) on 132 column printers by using STRING$(n,c). Example:

```
LPRINTTAB(6Ø)"62"STRING$(48,32)"TAB 11Ø"
```

will print TAB 110 beginning at x = 110 (the 111th print position on the line.)

8. To print a heading that shows the LPRINTTAB(x) positions on a 132-column printer, use:

```
LPRINT"Ø   5";
   : FOR T=1Ø TO 95 STEP 5
   : LPRINT"  " RIGHT$(STR$(T),2);
   : NEXT T
   : FOR T=1ØØ TO 13Ø STEP 5
   : LPRINT" " RIGHT$(STR$(T),3);
   : NEXT T
   : LPRINT"1"
```

9. To get quotation marks inside a string, use CHR$ (34). Example:

```
PRINT"HERE ARE "CHR$(34)"QUOTES"CHR$(34)" WITHIN
QUOTES."
```

will print HERE ARE "QUOTES" WITHIN QUOTES.

```
A$="HERE ARE "+CHR$(34)+"QUOTES"+CHR$(34)+
   " WITHIN QUOTES."
   : PRINT A$
```

will do the same. (The quotation mark at the end of a string is not needed if the quotation mark would be the last character on the line.

10. This subroutine will print a solid border of graphic characters around the edge of the screen:

```
1Ø CLEAR 124
   : CLS
   : POKE 1536Ø, 191
   : PRINT@1, STRING$(62,131);
   : POKE 15423, 191
   : POKE 1632Ø, 191
   : PRINT@961, STRING$(62,176);
   : FOR I=15424 TO 16256 STEP 64
   : POKE I, 191
   : POKE I+63, 191
   : NEXT I
   : POKE 16383, 191
   : RETURN
```

When the routine is finished, the cursor will be at screen position 1023.

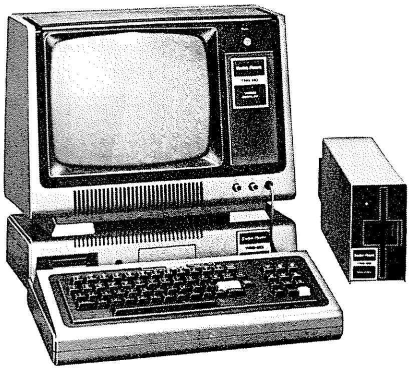
11. This subroutine creates a blinking prompt on the bottom line of the screen; the line is erased when a key is pressed. (The cursor will be on the last line; you should do a CLS or another PRINT@ whenever the program resumes.)

```
1 ZØ$=INKEY$
   : PRINT@978,"PRESS ANY KEY TO CONTINUE";
   : FOR I=1 TO 4Ø
   : NEXT I
   : IF ZØ$<>"" THEN PRINT@976,STRING$(31,32);
   : RETURN
2 PRINT@976,STRING$(31,143);
   : FOR I=1 TO 2
   : NEXT I
   : GOTO 1
```

12. This subroutine moves a rocket from bottom-right to top-right of the screen:

```
1ØØ DIM RK(12)
   : RK(1)=1612Ø
   : RK(2)=16121
   : RK(3)=16183
   : RK(4)=16185
   : RK(5)=16247
   : RK(6)=1625Ø
   : RK(7)=16311
   : RK(8)=16312
   : RK(9)=16313
   : RK(1Ø)=16314
   : RK(11)=16373
   : RK(12)=16377
11Ø FOR J=1 TO 12
   : POKE RK(1), 15Ø
   : POKE RK(2), 148
   : POKE RK(3), 17Ø
   : POKE RK(4), 17Ø
   : POKE RK(5), 149
   : POKE RK(6), 149
   : POKE RK(7), 181
   : POKE RK(8), 176
   : POKE RK(9), 176
   : POKE RK(1Ø), 149
   : POKE RK(11), 191
   : POKE RK(12), 149
   : FOR I=1 TO 12
   : RK(I)=RK(I)-64
   : NEXT I
12Ø NEXT J
   : PRINT@55, "B O O M !";
   : RETURN
```

13. Hey, reader, why not make your "trick of the trade" number thirteen, and send it to TRS-80 Microcomputer News?



Model I/III Bugs, Errors and Fixes

Note to Users:

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

DISK PAYROLL (26-1556 Version 3.0 and Prior)

Users of Disk Payroll, version 3.0 or prior, in Oklahoma or Montana should contact their local Computer Center, Expanded Computer Department or Dealer for information on changes needed to Disk Payroll:

1. Oklahoma—Changes are needed in the State Tax Table
2. Montana—Changes are needed to allow multiple "Standard Deductions as Percentage of Gross Pay"

MODEL III DISK SYSTEM MANUAL (26-2111)

On pages 78-79 of the Model III Disk System Manual, please note the following additional information about the $INIT I/O Call:

If the filespec name is found, $INIT simply opens the file for use and resets the pointers. $INIT is a destructive open of a file. The following pointers are reset: NRN, ERN, ODECR, EOFOLDLPR. If no writes are made, the disk copy is not changed. If a write is made, the directory is updated with the new information.

On page 80 please change the first paragraph under $READ to read:

> If LRL is not equal to zero, then $READ transfers the logical record whose number was placed in the DCB by $POSN into the RAM area addressed as UREC. The value of LRL is defined at open time. The record comes from "BUFFER" defined at open time. If TRSDOS must read a new physical record to satisfy the request, it will do so. "Spanned" logical records will be re-assembled as necessary. $READ will automatically increment in the DCB the offset to delimiter at end of current record (ODECR) by the value of LRL for each logical record and NRN by one for each physical record after each transfer is completed. $INIT/$OPEN will set NRN = X'0000' and ODECR = X'00' in order to read the first record with the first $READ.

Also on page 80, please change the first paragraph under $WRITE to read:

> If LRL is not equal to zero, then $WRITE transfers the one logical record from the RAM area addressed by UREC with the length LRL as defined at open time. The record goes into the 'BUFFER' which was defined at open time. If TRSDOS must write a physical record in order to satisfy the request, it will do so. "Spanning" will be handled by TRSDOS as necessary. At $INIT/$OPEN time the DCB value of NRN = X'0000' and the offset to delimiter at end of current record (ODECR) = X'00' so the first record can be written. After each logical record is transferred, the ODECR value in the DCB will be incremented by the value of LRL. After each physical record is transferred, the NRN value in the DCB will be incremented by one.

COBOL DEVELOPMENT SYSTEM (26-2203)

Page 23 in the user's guide (in the third paragraph of section 2.5.1) the manual states that a record can be any length from 2 to 255. Change 255 to 254, which is the maximum record length for variable records. If you try to create a file with a record length of 255, an ERROR 94 occurs.

Microcomputer-Based Data Management for Small Libraries

The Department of Library Science at Clarion State College in Clarion, Pennsylvania, is concerned about the data management needs of small libraries. Professor Bernard Vavrick at Clarion State has been experimenting with ways in which the TRS-80 Model III and off-the-shelf software can be used in school, rural, and small college libraries. He has developed some interesting applications for Radio Shack's MICROFILES program and has used the TRS-80 Model III in giving workshops and demonstrations at library conferences.

Microfiles is a software package for use with a TRS-80 Model I or Model III disk system with at least 32K RAM. It allows you to design a custom filing system based on your individual needs. You can build files in any format, make particular items in a file optional or required, index information and locate particular entries, update items in a file easily, and print out information in any format.

Using Microfiles with a 48K Model III disk system and a Daisy Wheel printer, Vavrick has worked out applications both for library circulation (check-outs and returns) and for inventory. A typical file for circulation might have a listing for each book held by the library, using the title, book number, and/or other information. "Name" and "date" fields would be "optional" items in the file, meaning that they need not be filled in all of the time for each book. When a book is checked out, the borrower's name and the date due can then be entered as part of the book's listing. Or, each listing could be entered as a book is checked out. When the book is returned, the entire listing can be deleted until the book is borrowed again. To find listings of overdue books, the librarian can index the file by date and/or use the Microfiles "FIND" command to locate listings for all books that were due on a certain date. Vavrick estimates that a data management system like this would work well for a small library with up to 1000 circulations per week.

A Microfiles-based filing system for recording the library's acquisitions (books on order or just received) would work similarly. Instead of searching through paperwork to find out about books ordered by the library, the librarian could use Microfiles to set up a quick, easily updated reference source. A file for acquisitions would have one listing per book. Information fields in the file would include the book's author, title, publisher, Library of Congress number or ISBN number, the date ordered, the supplier the book was ordered from, and the price. This would give the librarian a valuable record system that would allow him or her to locate information on ordered books quickly—by publisher, date ordered, or any of the other fields.

Similar filing systems could be used for small inventory needs, such as the library's list of community resources (other libraries and collections, charitable and educational institutions, and so on). For a small library, Microfiles could keep inventory records by author, title, and subject—similar to the card catalog. Vavrick noted that the time required to key in catalog information would be a disadvantage, but added that "once that is done, a tremendous facility is available." Other library departments that could use such a filing system include the audio/visual and serials/periodicals departments.

Vavrick mentioned the flexibility of Microfiles and its low cost as special advantages. He found the Microfiles user's manual very clear, and he remarked that "help is nearby," that Radio Shack store personnel in Erie, Pa. had been helpful in answering his specific questions. He remarked that in the field of library science there has been a general "dearth of attention paid to small libraries," but added that he has lately seen microcomputer programs emerging that can help small libraries—some of these programs specifically designed for library use and written to run on the TRS-80 microcomputer. Some of these programs, Vavrick noted, can even calculate numerical totals and library discounts. The microcomputer may well become an important tool for small libraries.

Note: Microfiles (Catalog Number 26-1565) is available from Radio Shack stores and dealers. The listed price is $99.95, though prices may vary at individual stores and dealers. Minimum system is a 32K Model III or Model I with at least one disk drive.

Introducing "Number Theory"

Radio Shack's new "Number Theory" program is an exciting addition to the growing library of TRS-80 courseware designed to aid students in learning math. With this program, educators can now use either the TRS-80 Model I or Model III to present one of the oldest and most elegant branches of mathematics to students at the junior high school level and up.

The developers of the "Number Theory" program, J. Liang and E. B. Saff, have spent many years working with high school students gifted in math. Their program has been classroom-tested at the University of South Florida—Hillsborough County and Pinellas County summer programs in mathematics and science.

Besides supplementing secondary and college courses, Radio Shack's "Number Theory" can also be used outside the classroom—as a self-study course by anyone interested in learning about the modern number system. The lessons concentrate almost exclusively on the computational or constructive areas of number theory, making the program accessible to users with a minimal mathematics background. In fact, Part I of the program requires only junior high school-level math skills; knowledge of high school algebra is sufficient for the rest of the program.

The field of number theory presents many purely mathematical challenges and is filled with unresolved or "open" problems of current interest to mathematicians. However, number theory also has many practical applications, such as coding theory and cryptography, and contains a wealth of examples and ideas which are useful in studying other branches of mathematics. Radio Shack's "Number Theory" program covers the fundamental topics in number theory. The lessons are designed to provide a clear overview of the essentials of number theory, especially the basic properties of integers.

The program is divided into six major parts. Each part begins with a lesson menu listing the names of each lesson. The topics covered in these lessons are as follows:

Part I: Prime Factorization and Greatest Common Divisors
- Lesson 1: Divisibility of Integers
- Lesson 2: Perfect Numbers
- Lesson 3: Prime Numbers
- Lesson 4: Prime Factorization
- Lesson 5: Greatest Common Divisor

Part II: Solutions of Linear Diophantine Equations
- Lesson 1: Division Algorithm
- Lesson 2: Euclidean Algorithm
- Lesson 3. Diophantine Equations

Part III: Congruence MOD N
- Lesson 1: Congruence MOD N
- Lesson 2: Complete Residue System MOD N
- Lesson 3: Reducing an Integer MOD N

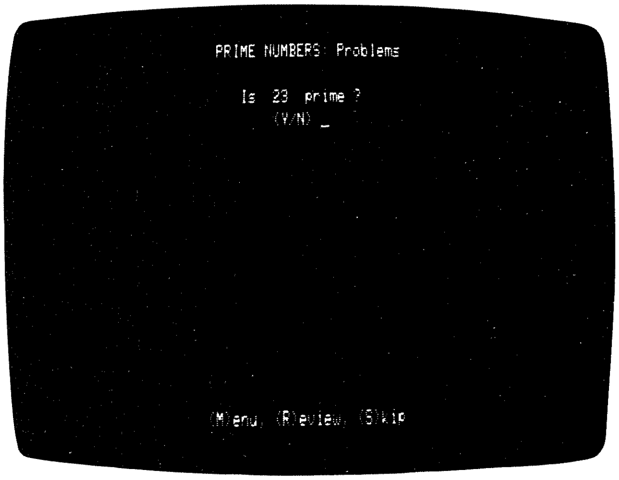
Part IV: Arithmetic MOD N
- Lesson 1: Addition, Multiplication, and Inverse MOD N
- Lesson 2: Evaluating a Power MOD N

Part V: Linear Congruence & Primitive Roots
- Lesson 1: Linear Congruence
- Lesson 2: Chinese Remainder Theorem
- Lesson 3: Primitive Root MOD a Prime

Part VI: Quadratic Residue & the Euler Phi-Function
- Lesson 1: Quadratic Residue
- Lesson 2: Continued Fraction Expansion
- Lesson 3: Euler Phi-Function

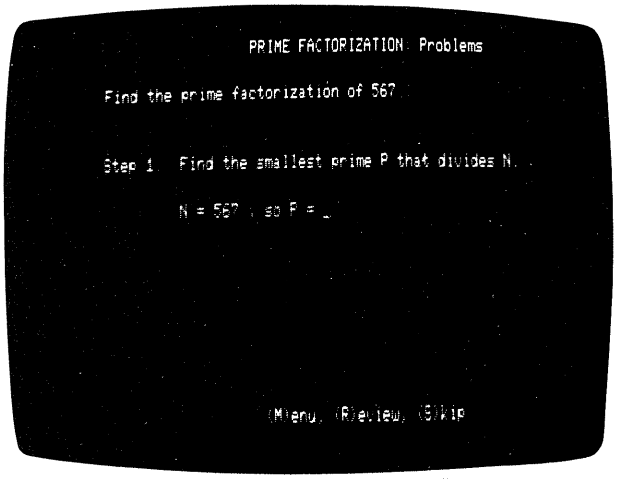
The "Number Theory" program has many features to aid and encourage the student. The lessons are progressive—arranged to build on the concepts and skills covered in the previous lessons; students are encouraged to approach the subject systematically, working from one lesson to the next. Each lesson provides definitions, explanations, examples, and problems over a specific area of number theory, to insure that the student fully understands each topic before moving on to the next lesson.

In the problem sections of each lesson, the program keeps track of the number of problems the student has successfully completed. The program provides reinforcement when the student's score is high, and advice for review when the score is low. Feedback messages appear after individual problems. The student sees a positive message for a correct response. For an incorrect response, the student is given another chance or the correct answer is displayed. The problems and many of the examples are randomly generated, so the program can be used again and again without repetition in the interactive sections. The type of problem used varies throughout the program, according to the topics covered. Some lessons use simple YES/NO questions to check student understanding of a basic concept. The problem screen below is from Part I, Lesson 3 on Prime Numbers:

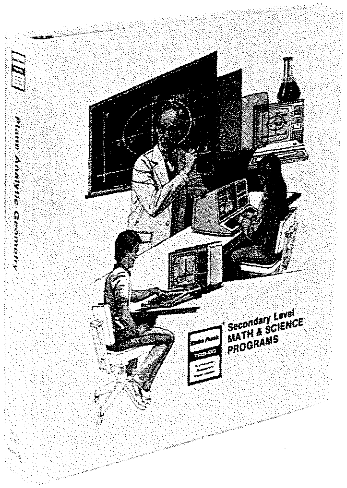


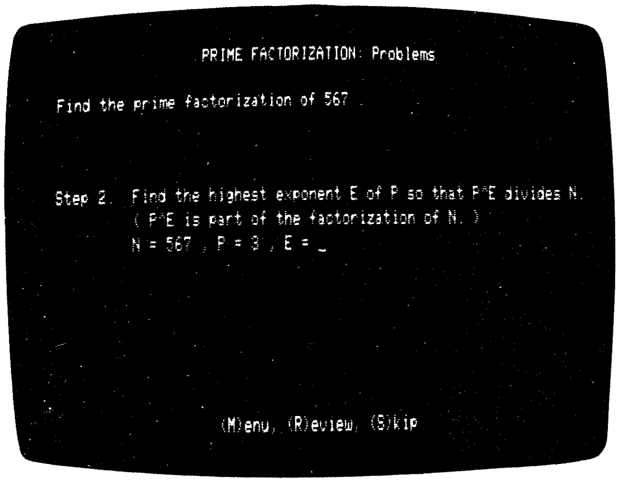
```
PRIME NUMBERS: Problems

Is 23 prime ?
(Y/N) _




(M)enu, (R)eview, (S)kip
```

Other lessons use long problems with several steps to check student ability to successfully complete a procedure. For instance, the following screen is from the problem section of Part I, Lesson 4, "Prime Factorization":



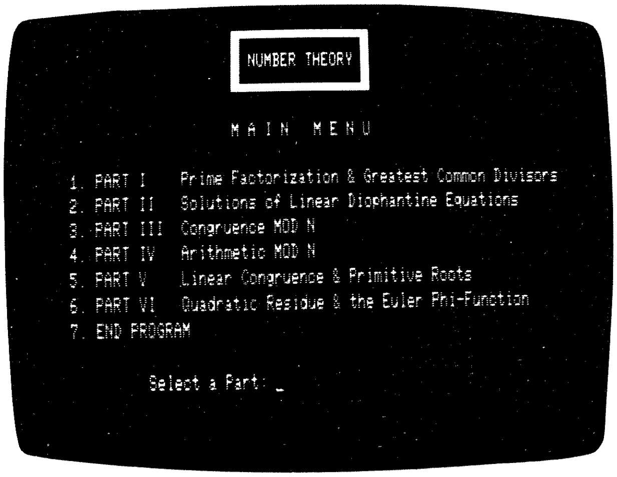
```
PRIME FACTORIZATION: Problems

Find the prime factorization of 567.

Step 1. Find the smallest prime P that divides N.

        N = 567 ; so P = _




(M)enu, (R)eview, (S)kip
```





```
                 PRIME FACTORIZATION: Problems

Find the prime factorization of 567 .



Step 2.  Find the highest exponent E of P so that P^E divides N.
         ( P^E is part of the factorization of N. )
         N = 567 , P = 3 , E = _




                 (M)enu, (R)eview, (S)kip
```

Another important aspect of this program is the availability of special keys. Throughout the program, the student is allowed to control the pace of the lessons, using four options: (M)enu, (R)eview, (S)kip and (C)ontinue. Pressing the "M" key takes the student back to the previous menu; that is, from a screen in the first lesson of Part I, "Divisibility of Integers," the "M" key gives the student the menu for Part I:



```
                 NUMBER THEORY

                 M A I N   M E N U

 1. PART I     Prime Factorization & Greatest Common Divisors
 2. PART II    Solutions of Linear Diophantine Equations
 3. PART III   Congruence MOD N
 4. PART IV    Arithmetic MOD N
 5. PART V     Linear Congruence & Primitive Roots
 6. PART VI    Quadratic Residue & the Euler Phi-Function
 7. END PROGRAM

          Select a Part: _
```

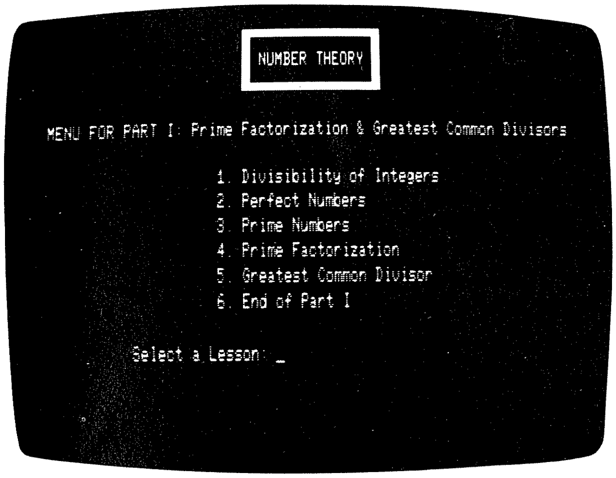
In those lessons divided into sections, the "M" key from within a section takes the student back to the menu for that lesson.

The "R" key returns the student to the first screen of a lesson; this "(R)eturn" function makes it possible for the student who wishes to review to return to the beginning of the lesson without going back through the menu. The student will thus feel free to progress through the lessons at his or her own pace, returning to the definitions and examples as often as is needed.

For those students who may not need to see all the explanations and examples for each lesson, the "S" key provides a way to skip from the initial definition screen of each lesson to the set of problems at the end of the lesson. Students can use this "(S)kip" option to scan through lessons they have already worked once, or to return to a set of problems after a break.

The "C" key allows the student to continue through the lessons one screen at a time, taking time to read each explanation and example as it appears. (M)enu, (R)eview, (S)kip, and (C)ontinue appear as prompts at the bottom of the screen whenever the options are available to the student, making the program self-explanatory and easy to get around in.

The manual included in the "Number Theory" package provides step-by-step instructions on how to use the program, along with a reference section with summaries of each lesson, sample problems, and instructional objectives. The manual also provides suggestions for additional projects in number theory which are designed to stimulate the advanced student to write his or her own algorithms and computer programs involving the techniques covered in the "Number Theory" program.



```
                 NUMBER THEORY

MENU FOR PART I: Prime Factorization & Greatest Common Divisors

                 1. Divisibility of Integers
                 2. Perfect Numbers
                 3. Prime Numbers
                 4. Prime Factorization
                 5. Greatest Common Divisor
                 6. End of Part I

          Select a Lesson: _
```

The "Number Theory" program is designed to provide a variety of applications. For instance, the program could be used by an advanced student for self-instruction. Or an entire algebra class could be structured around the material in this program, so that the teacher could introduce each concept to the class, present the initial examples and answer students' questions, and then allow the "Number Theory" program to provide students with the necessary review and skill-building practice. "Number Theory" can also be used with a Radio Shack Network System.

The "Number Theory" program (catalog no. 26-2613) is available—on 16K tape and 32K disk—at Radio Shack Stores and Computer Centers. $69.95 is the listed price; prices may vary at individual stores and dealers.

PATCHWRT

Richard A. Burke CSR
Radio Shack Computer Center 11-7554
Carmiachael, CA 95608

I would like to take this opportunity to introduce to you a program which I have written. The program's name is PATCHWRT (short for PATCHWRITER). As the name implies, the program is used to write, edit, save, and retrieve various Model II DO files.

The program ends the nasty frustrations of incorrect PATCHes by allowing the user to correct any portion of any line. With BUILD files, the user must replace the whole line even if only one character is incorrect. With PATCHWRT, he can not only correct lines, he may add lines, delete lines, and insert new lines into the file. The program is written in standard BASIC and from the TRSDOS READY level can be entered by typing the following:

```
BASIC PATCHWRT -F:1
```

Once BASIC has loaded the program the screen will ask what function you wish to perform. You may choose from the following options

(A)dd a Patch (Create a new BUILD file)
(E)dit an existing file
or you may
(Q)uit the program

When you choose (A) for add, the screen begins to input one set of commands at a time. To end each line, simply hit ⟨ENTER⟩ and begin inputting the next line. PATCHWRT will keep track of up to 20 command lines and will list which line the user is currently on

For example, enter the (A)dd section and type the following PATCH:

```
PATCH SYSTEM/SYR A=DEFF F=AB C=ØØ <ENTER>
```

When you have finished entering the line, simply hit the ⟨ENTER⟩ key at the beginning of the next line. The program will then immediately go into the edit phase.

The editor is quite simple to use. It allows you to select from the following options:

(Y) - make changes to a particular line
(N) - exit editing as no more changes are necessary
(I) - insert a line at a particular place (even at the end of the patch)
(D) - delete any line in the file

Each of the options will then ask you which line number you wish to work on. Give the line number and then hit ⟨ENTER⟩. If you were correcting part of a given line (Y), the screen will display that line only and ask for the "SEGMENT TO REPLACE." To insure proper editing, you should give at least a three character "section." For example:

The screen shows:

```
PATCH SYSTEM/SYR A=DEFF F=AB C=ØØ
        SEGMENT TO REPLACE:
```

The user should respond - SYR ⟨ENTER⟩

The screen then asks:

```
REPLACE WITH:
```

and the user should answer with the phrase - SYS ⟨ENTER⟩

The machine makes the corrections and takes the user back to edit the menu and re-displays all the lines. For our example we would see:

```
1 - PATCH SYSTEM/SYS A=DEFF F=AB C=ØØ
```

If you need to insert a new line, type (I) and then answer the line number prompt. If we need a blank line ahead of our current line we would type (I) and respond ⟨1⟩ ⟨ENTER⟩. The program then asks the user for what is to appear on the newly created line. For example we type:

```
PAUSE THIS PATCH DOES XXXXX XXXXX TO YYYYY FOR ZZZZ
```

and hit ⟨ENTER⟩

The display then reads:

```
1 - PAUSE THIS PATCH DOES XXXXX XXXXX TO YYYYY FOR
    ZZZZ
2 - PATCH SYSTEM/SYS A=DEFF F=AB C=Ø
```

In order to delete a whole line, we simply type (D) and enter which line is to be deleted. Suppose we typed (D) and then ⟨1⟩ ⟨ENTER⟩. The computer would then ask if we are sure we wish to delete this line. If we answer (N) for no we return to the edit menu. If we type (Y) for yes, the screen then shows

```
1 - PATCH SYSTEM/SYS A=DEFF F=AB C=Ø
```

and the line has been deleted

When there are no more corrections to be made, we type (N) at the edit menu and are taken to the output section of the program. The computer asks if we are sure the program is to be saved (answer Y or N as appropriate). If we do not save the file it is removed from memory and we are returned to the (A)dd (E)dit (Q)uit menu

If we are going to save the program we are then asked for a file name. This file name should observe all Model II filename protocols. After hitting ⟨ENTER⟩, the BUILD file is then saved under the given name and the user is returned to the top menu. If you wish to execute the file simply type (Q) to quit and then use the BUILD file as you would use any DO file (i.e. type DO filename)

If you have a file on the disk already, you can (E)dit it by typing (E) and entering the file's name. The program then goes immediately into edit phase and the user may proceed as listed above.

```
1 'Program to write PATCHes and DO files
2 'Written by Richard Burke -C.S.R.
```

```
3 '         Radio Shack Computer Center 11-7554
4 '         6305 Fair Oaks Blvd.
5 '         Carmiachael, Ca. 95608
6 '         (916)-484-6815
7 'Completed May 27,1982
8 '
9 'Clear memory for 20 strings
10 CLEAR 3000
    : DIM A$(20)
20 CLS
    : PRINT@(9,27),"PATCH WRITER - VERSION 3.3"
25 PRINT@(12,32),"RICHARD A. BURKE"
27 'Find out what user wishes to do
30 PRINT@(15,23),"(E)DIT PATCH (A)DD PATCH OR
    (Q)UIT";
40 Z$=INKEY$
    : IF Z$=""THEN 40
50 IFZ$="E" THEN 560 ELSE IF Z$="A" THEN 55 ELSE IF
    Z$="Q" THEN SYSTEM ELSE 30
55 FOR T=1 TO 20
    : A$(T)=""
    : NEXT
57 'Create new PATCH or DO file
58 'Begin inputting strings
60 CLS
    : B=1
65 PRINT "ENTER STRING -";B;
70 INPUT A$(B)
73 'If A$(C)="" then no more strings to be added -
    goto edit section
75 IF A$(B)="" THEN A=B-1
    : GOTO 100
80 B=B+1
    : GOTO 65
100 CLS
104 'Edit file section
105 'List out the file
110 FOR B=1 TO A
120 PRINT B;"- ";A$(B)
130 NEXT
135 'Find out if there are any changes
136 'Y = change a given line
137 'N = no more changes - goto output section
138 'I = insert a line
139 'D = delete a line
140 PRINT@(22,10),"ANY CHANGES (Y/N/I/D)";
145 Z$=INKEY$
    : IF Z$=""THEN 145
150 IF Z$="Y" THEN 155 ELSE IF Z$="N"THEN 330 ELSE IF
    Z$="I" THEN 205 ELSE IF Z$="D" THEN 230 ELSE 140
153 'Determine which line to be edited and make the
    correction
154 'If not valid line number then try again
155 PRINT@(22,10),"EDIT WHICH LINE NUMBER ";
    : INPUT C
    : IF C<1 OR C>A THEN 155
157 'Print line to be corrected
160 CLS
    : PRINT@(5,0),A$(C)
163 'Get segment to be replaced
165 PRINT@(9,0),"SEGMENT TO REPLACE"
    : INPUT"          ";D$
    : D=LEN(D$)
167 'Get correction to line
170 PRINT"REPLACE WITH"
    : INPUT "          ";E$
173 'Search to see if segment to be replaced is in
    the string
174 'If not then exit else make replacement
175 FOR B=1 TO (LEN(A$(C))-D+1)
180 IF MID$(A$(C),B,D)=D$ THEN 190 ELSE NEXT
185 PRINT "STRING NOT FOUND"
    : FOR X=1 TO 1000
    : NEXT
    : GOTO 100
190 E=B-1+D
```

```
195 A$(C)=LEFT$(A$(C), (B-1)) + E$ +  RIGHT$(A$(C),
    (LEN(A$(C))-E))
200 GOTO 100
203 'Insert line section
204 'Find out where to put new line
205 PRINT@(22,10),"INSERT A STRING WHERE";
    : INPUT C
207 'Get the new line
210 INPUT"ENTER NEW STRING";IN$
213 'Insert new line into the file
214 'Then exit
215 FOR B=A TO C STEP(-1)
220 A$(B+1)=A$(B)
    : NEXT
    : A=A+1
221 A$(C)=IN$
    : GOTO 100
225 'Delete line section
226 'Find out which line is to be deleted
227 'If user is sure, then delete line else exit
230 PRINT@(22,10),"DELETE WHICH LINE #";
235 INPUT C
236 PRINT@(23,10),"ARE YOU SURE YOU WISH TO DELETE
    LINE #";C;"(Y OR N)";
237 Z$=INKEY$
    : IF Z$=""THEN 237
238 IF Z$="Y" THEN 240 ELSE IF Z$="N" THEN 100 ELSE
    236
240 FOR B=C TO A
245 A$(B)=A$(B+1)
    : NEXT
250 A=A-1
260 GOTO 100
330 'Output section
335 'Add all necessary info to the front & back of
    file
340 A$(1)=CHR$(&HD0) + A$(1)
350 FOR F=1 TO A
360 A$(F)=A$(F) + CHR$(13)
370 NEXT
380 CLS
381 'If user wishes to save then continue else exit
    to beginning
383 PRINT@(22,10),"DO YOU WISH TO SAVE";
384 Z$=INKEY$
    : IF Z$="" THEN 384
385 IF Z$="Y" THEN 386 ELSE IF Z$="N" THEN 30 ELSE
    383
386 PRINT@(22,9)," ";
387 'Get filename for saving
388 'If filename exists then kill the file
389 'Then save file to disk
390 INPUT" SAVE AS WHAT FILENAME";F$
400 ON ERROR GOTO 730
410 KILL F$
420 OPEN "R",#1,F$,1
430 FIELD #1, 1 AS ZZ$
440 FOR F=1 TO A
450 FOR G=1 TO LEN(A$(F))
460 LSET ZZ$ = MID$(A$(F),G,1)
470 PUT #1,(LOF(1)+1)
480 NEXT
490 NEXT
500 CLOSE
510 CLS
520 GOTO 30
550 END
555 'Get file for editing
556 'If file doesn't exist then exit
560 CLS
    : INPUT "EDIT WHICH FILENAME";F$
565 FOR X=1 TO 20
    : A$(X)=""
    : NEXT
570 OPEN "R",#1,F$,1
580 FIELD #1,1 AS ZZ$
```

```
590 E=1
     : C=1
595 ON ERROR GOTO 73Ø
596 'Begin input of valid file
6ØØ GET #1,E
61Ø A$(C)=A$(C)+ZZ$
615 'Increment string count if end is detected
62Ø IF ZZ$=CHR$(13) THEN C=C+1
625 'Exit this section if end of file
63Ø IF E=LOF(1) THEN A=C-1
     : GOTO 65Ø
64Ø E=E+1
     : GOTO 6ØØ
65Ø CLOSE
655 'Begin stripping away file markers from the
     strings
656 '(Output replaces them)
66Ø A$(1)=MID$(A$(1),2,LEN(A$(1))-2)
67Ø IF A=1 THEN 1ØØ
68Ø FOR X=2 TO A
69Ø Y=LEN(A$(X))-1
7ØØ A$(X)=LEFT$(A$(X),Y)
71Ø NEXT
72Ø GOTO 1ØØ
725 'Handle all expected errors
73Ø IF ERL = 41Ø THEN RESUME 42Ø
74Ø IF ERL = 6ØØ THEN PRINT "NO SUCH FILE"
     : FOR M=1 TO 1ØØØ
     : NEXT
     : CLOSE
     : KILL F$
     : RESUME 3Ø
```

Model II Bugs, Errors, and Fixes

NOTE TO USERS:

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

PATCHES

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

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

SCRIPSIT 2.0 (26-4531)

Following are several series of patches to correct problems that have been found in Scripsit 2.0. They should be applied in the order that they appear.

Correction 1 - When repaginating a document and reformatting it, locked blocks are sometimes changed. The margins will not be adjusted, but tabs will be refigured, leading to strange results.

To prevent this from happening, apply the following patches. The date at the bottom of the STARTUP screen must be 02/23/1982 before you apply these patches; if it is not, obtain and apply all previous patches.

```
PATCH SCRIPSIT/SYS R=59 B=142 F=2ØBBF1FD3614ØØ
                              C=FD3614ØØ2ØB7F1
PATCH SCRIPSIT/SYS R=61 B=241 F=Ø1 C=Ø2
```

Sometimes user defined print codes are split by the "wraparound" feature of the system. This causes confusion in recognizing them as print codes and/or translating them properly. The following patches correct the problem:

```
PATCH SCRIPSIT/SYS R=157 B=22Ø F=D64147 C=CD58D6
PATCH SCRIPSIT A=9F67 F=E1FEØ7 C=C36ØD6
PATCH SCRIPSIT A=D658 F=ØØØØØØØØØØØØØØØØØØØØ
                      C=E67FD64147C9E17EFEØ7
PATCH SCRIPSIT A=D664 F=ØØØØØØ C=C36A9F
```

The Merge routine was failing to set up the proper data for alignment tabs. This sometimes caused the aligned text to print in the wrong column when merging. The following patches will correct the problem:

```
PATCH SCRIPSIT/SYS R=139 B=175 F=32FDDA C=CD4ED6
PATCH SCRIPSIT A=D64E F=ØØØØØØØØØØØØØØØØØØØØ
                      C=32FDDAFD7E163223FFC9
```

When using the SCRIPSIT backup utility, there is no prompt when the destination diskette contains data. Also, when doing a backup to the same drive other than drive 0, the source diskette would be formatted. The following patches will correct these problems:

```
PATCH SCRIPSIT/SYS R=37 B=147 F=32EDE2 C=C36BD6
PATCH SCRIPSIT A=D66B F=ØØØØØØØØØØØØØØØØØØØØ
                      C=32EDE26F3A1ØE3BD2ØØD
PATCH SCRIPSIT A=D675 F=ØØØØØØØØØØØØØØØØØØØØ
                      C=FE3Ø28Ø9212Ø2Ø22FØE2
PATCH SCRIPSIT A=D67F F=ØØØØØØØØØØØØ C=22F2E2C34ØE4
```

After applying these patches, apply the following to change the date in the STARTUP screen:

```
PATCH STARTUP A=E4ØE F=BØB2AFB2B3 C=BØB5AFB2BØ
```

Correction 2 - When repaginating a document with multiple headers or footers, repaginate sometimes uncenters centered-lines and reformats paragraphs even when no reformat was requested and when the paragraphs were locked.

Before applying the following patches to correct this problem, check to make certain that the date at the bottom of the STARTUP screen is 05/20/1982. If it is not, obtain and apply all previous patches.

```
PATCH SCRIPSIT A=D635 F=ØØØØØØØØØØØØØØØØØØØØ
                      C=E52A4CF47EE1FE8FC823
PATCH SCRIPSIT A=D63F F=ØØØØØØØØØØ C=232323C979
PATCH SCRIPSIT A=D644 F=ØØØØØØØØØØØØØØØØØØØØ
                      C=FEØ3D8CD79E4C4CFE4C9
PATCH SCRIPSIT/SYS R=59 B=2Ø9 F=23232323 C=CD35D6ØØ
PATCH SCRIPSIT/SYS R=59 B=24Ø F=CD79E4C4CFE4
                              C=CC43D6ØØØØØØ
PATCH SCRIPSIT/SYS R=6Ø B=51 F=79FEØ338 C=CC43D618
```

After making the above patches, apply the following to change the date in the STARTUP screen:

```
PATCH STARTUP A=E4ØE F=BØB5AFB2BØ C=BØB5AFB2B4
```

Correction 3 - When using a user defined print code to define a character or characters such as [ç] or [¶], the character(s) would not adhere to the current print mode (i.e. bold, underline, double underline, etc.).

The following patches will correct the problem:

Be sure that the date at the bottom of the STARTUP screen is 05/24/82. If not, be sure to obtain and do all previous patches.

```
PATCH SCRIPSIT/SYS R=163 B=196 F=F7F7 C=19D6
PATCH SCRIPSIT A=D619 F=ØØØØØØØØØØØØØØØØØØØØØØØØØØ
                      C=78FE213ØØ4CDF7F7C9C53AFDDA
PATCH SCRIPSIT A=D626 F=ØØØØØØØØØØØØØØØØØØØØØØØØØØ
                      C=ØEØ1CB5F28Ø43AFDDD4FCD35F9
PATCH SCRIPSIT A=D633 F=ØØØØ C=C1C9
```

After applying these patches, apply the following to change the date in the STARTUP screen:

```
PATCH STARTUP A=E4ØE F=BØB5AFB2B4 C=BØB5AFB2B7
```

Correction 4 - Scripsit will nang up if you try to edit a word that will not fit in the margins. The following patch corrects this problem. First be sure that the date at the bottom of the STARTUP screen if 05/27/1982. If not, obtain and make all previous patches.

```
PATCH SCRIPSIT A=9F68 F=6Ø C=5E
```

If a user defined print code contains a space, the space will not be underlined via the normal SCRIPSIT underline mode. The following patch corrects this problem:

```
PATCH SCRIPSIT A=D61B F=21 C=2Ø
```

In order to print more than one copy of a document when doing a merge printout, apply the following patches:

```
PATCH SCRIPSIT A=D5E5 F=ØØØØØØØØØØØØØØØØØØ
                      C=C2EBD532Ø4DB3A5ØDC
PATCH SCRIPSIT A=D5EE F=ØØØØØØØØØØØØ C=329ØE7C3BAEØ
PATCH SCRIPSIT A=D5F5 F=ØØØØØØØØØØØØØØØØØØ
                      C=CFC25CE13A9AE5FE43
PATCH SCRIPSIT A=D5FE F=ØØØØØØØØØØØØØØ
                      C=CA48E1219ØE735
PATCH SCRIPSIT A=D6Ø5 F=ØØØØØØØØØØØØ C=C2BAEØC348E1
PATCH SCRIPSIT/SYS R=135 B=186 F=2ØØ332 C=C3E5D5
PATCH SCRIPSIT/SYS R=136 B=78 F=CF2Ø14 C=C3F5D5
```

After applying these patches, apply the following to change the date in the STARTUP screen to 06/02/1982

```
PATCH STARTUP A=E4ØE F=BØB5AFB2B7 C=BØB6AFBØB2
```

Correction 5 - Scripsit locks up when the user attempts to use a serial printer with Merge. The following patches correct this problem:

The date at the bottom of the STARTUP screen must be 06/02/1982 before these patches are applied. If not, obtain and apply all previous patches.

```
PATCH SCRIPSIT/SYS R=176 B=2Ø F=9ØDA C=ØBD6
PATCH SCRIPSIT A=D6ØB F=ØØØØØØØØØØØØØØØØ
                      C=FDCB7F7EC8C39ØDA
```

Under some circumstances, when the user recalls text into a word, causing the text to exceed the margins, the cursor may be left off the text. Scripsit fails to flag the error and unpredictable results may occur. The following patches will correct the problem:

```
PATCH SCRIPSIT A=B228 F=2CA4 C=DFD5
PATCH SCRIPSIT A=D5DF F=ØØØØØØØØØØØØ C=CD2CA4C35CA9
PATCH SCRIPSIT A=B21C F=FE8D2Ø C=C3B8D5
PATCH SCRIPSIT A=D5B8 F=ØØØØØØØØØØØØØØØØØØØØ
                      C=FE8DCA2ØB2CD5CA9AFC9
```

After the preceding patches have been applied, make the following patch to change the date at the bottom of the STARTUP screen to 06/09/1982.

```
PATCH STARTUP A=E412 F=B2 C=B9
```

The Firmament

Waldo T. Boyd
P.O. Box 86
Geyerville, GA 95441

The Model II is becoming the workhorse of small business proprietors, professionals, and specialized departments within larger companies. As such it is rarely tested to anything like its ability to perform in what might be called non-business ways. Despite its long workday, there are times when the operator's attention is required elsewhere for short or medium "break" periods, and very few of these happy people turn off the CPU and its peripherals for such relatively brief interludes in their busy day.

Chances are, the life of the cathode-ray tube filament is prolonged by this practice since initial current surge at power-on is avoided. On the other hand, the screen endures a repetitive burning of a fixed pattern into the phosphor, a situation passersby sometimes interpret as an open invitation to start touching the various keycaps.

Here's a simple, neat little program that uses only one 256-byte record, which can be loaded into working memory in a few seconds just before taking that coffee break. When running, it looks like space at night, with the stars twinkling in the distance. Somehow a dynamic screen looks much more business like and less inviting to itchy fingers, making this program more practical than it might at first appear. Certainly it is anything but frivolous.

Four "star" types are depicted with the plus sign, the caret, the period, and the asterisk. A "wink-out" blank follows. PRINT@ statements assign print positions based upon random selection of row and column numbers. The cursor is inhibited in line 10, and the program loops back upon itself at line 20. The field never becomes full because, again at random, the line feed is activated and the field moves up a notch, and stars disappear. The constant exchange of asterisk for period, cross for caret—48 combinations happening sequentially so rapidly as to appear almost simultaneous—gives the effect of twinkling stars on a moonless night.

But be careful! The result is so fascinating to watch that operators have been known to forget the coffee break entirely . . . .

```
1Ø CLS
     : PRINT CHR$(Ø2)
2Ø X=RND(23)
     : Y=RND(79)
3Ø PRINT@ (X,Y), " + ";
4Ø X=RND(23)
     : Y=RND(79)
5Ø PRINT@ (X,Y), " ^ ";
6Ø X=RND(23)
     : Y=RND(79)
7Ø PRINT@(X,Y), " . ";
8Ø X=RND(23)
     : Y=RND(79)
9Ø PRINT@(X,Y), " * ";
92 X=RND(23)
     : Y=RND(79)
94 PRINT@(X,Y), "   ";
98 GOTO 2Ø
```

DATA FILE RECORDING PROGRAM

Allen G. Barclay
6402 Tamalpais Avenue
San Jose, CA 95120

Although this program was developed for use on a PC-1 pocket computer without a printer, it could be easily adapted to any other computer with or without a printer.

This program provides for memorizing, reviewing, and correcting data before recording it on magnetic tape. It also provides for recalling previously recorded data from tape for making additions or corrections before re-recording it on tape.

Other programs would be used to recall the data from the tape and to perform the desired analysis.

DATA FORMAT

Memory	X (Sequence No.)	Y (The Data)
A(10)	1	73.12
A(11)	2	8.03
A(12)	3	127.44
.	.	.
.	.	.
.	.	.
A(110)	101	11.723
A(111)	102	E99

The X values, or sequence numbers, are generated by the program. The Y values are the input data that you want to store and recall for future use. The last value is always E99, so it can be utilized in the using programs as a trailer to indicate the end of the data.

OPERATING SEQUENCE

1. Set to DEF mode, ⟨SHFT⟩ ⟨Z⟩
2. "DATA RECORD FILE" ⟨ENTER⟩
3. "1 = YES, 0 = NO" ⟨ENTER⟩
4. "NEW DATA FILE?"
   ⟨1⟩ ⟨ENTER⟩ denotes that a new file is to be compiled for recording onto magnetic tape. After each "Y = " enter the next item of data. If any error is entered, ignore it and continue making entries. After the last data has been entered, enter E99.
   ⟨0⟩ ⟨ENTER⟩ denotes that an existing file on magnetic tape is to be loaded into the computer memory for expansion or correction. The next message is, "GET DATA TAPE READY." ⟨ENTER⟩ will start the loading from the magnetic tape into the computer memory.
5. "LAST X = ,Y = " is printed after all the data has been entered into memory either manually or from a tape. ⟨ENTER⟩.
6. "ADD MORE DATA?"
   ⟨1⟩ ⟨ENTER⟩ will provide for entering more data, exactly as described above for compiling a new file. Again, enter E99 after all the new data has been entered.
   ⟨0⟩ ⟨ENTER⟩ moves to the next step if no new data is to be added.
7. "REVIEW DATA?"
   ⟨1⟩ ⟨ENTER⟩ initiates a print of the X and Y values for each data entry. If a larger number is entered, the review will start with that sequence number. If a number, decimal point, and two-digit numbers are entered, the review will be limited to the range of data starting with the number to the left of the decimal point and ending with the sequence number of the number to the right of the decimal. (i.e., 4.09 would limit the review of the data to the X and Y starting with X = 4 and ending with X = 9.)
   ⟨0⟩ ⟨ENTER⟩ denotes that no review is required and that the data is ready for recording. The next message is, "GET TAPE READY." ⟨ENTER⟩ will start recording of the data on magnetic tape.
8. "ALL VALUES CORRECT?" This question always follows the completion of the review of data.
   ⟨1⟩ ⟨ENTER⟩ denotes that all the data is correct and is ready for recording. The next message is, "GET TAPE READY." ⟨ENTER⟩ will start the recording of the data on magnetic tape.
   ⟨0⟩ ⟨ENTER⟩ will lead to three methods of correcting the data, as described below. Each of these three methods may be skipped, or ended, by entering 0 for the sequence number.
9. "SEQ # OF ERROR?" In response to this question, enter the sequence number of the data that is in error, or 0 if there is no data that needs to be corrected. When "CORRECT Y = " appears, enter the correct value for the data.
10. "DELETE SEQ #" Enter the sequence number of any data that is to be deleted, such as a data value that was entered twice by mistake. The data will be deleted and all the data values that follow will be moved up to close up the gap in the sequence. (A caution: If you are going to correct more than one item of data, remember that the sequence numbers have now been changed.) Enter ⟨0⟩ to move to the next step.

11. "INSERT NEW SEQ #" Enter the sequence number for the data to be inserted. Then enter the new data in response to "NEW Y = ". All the data that follows will be moved to the next higher sequence number to make room for this new data in its correct location. When 0 is entered for the sequence number, the program returns to "REVIEW DATA".

Note: In this article the 'E' in 'E99' is the ⟨Exp⟩ key found on the PC-1 next to the ⟨ + ⟩ key.

DATA RECORD FILE PROGRAM ("DATAFIL")

```
1 "Z" CLEAR
      : D=9
      : PRINT "DATA RECORD FILE"
      : PRINT "1=YES, Ø=NO"
      : INPUT "NEW DATA FILE?";B
      : IF B=1 THEN 2Ø
5 PRINT "GET DATA TAPE READY"
      : INPUT # "MI";A(1Ø)
1Ø FOR F=1Ø TO 1ØØ
      : IF A(F)=E99 THEN 3Ø
15 NEXT F
2Ø FOR F=D+1 TO 11Ø
      : INPUT "Y= ";A(F)
      : L=F-9 IF A(F)=E99 THEN 3Ø
25 PAUSE C, A(F)
      : NEXT F
      : A(111)=E99
      : PRINT "MAX DATA"
3Ø A=F-1Ø
      : D=F-1
      : PRINT "LAST X=";A;"Y=";A(D)
      : INPUT "ADD MORE DATA?";B
      : IF B=1 THEN 2Ø
35 INPUT "REVIEW DATA?";B
      : IF B>Ø THEN 45
4Ø PRINT "GET TAPE READY"
      : PRINT#"MI";A(1Ø)
      : PRINT "END"
      : END
45 E=INT B
      : H=(B-E)*1ØØ
      : IF H>Ø THEN 55
5Ø H=A
55 FOR F=9+E TO 9+H
      : C=F-9
      : PRINT C,A(F)
      : NEXT F
6Ø INPUT "ALL VALUES CORRECT?";B
      : IF B=1 THEN 4Ø
65 INPUT "SEQ # OF ERROR=";B
      : IF B=Ø THEN 75
7Ø F=B+9
      : INPUT "CORRECT Y= ";A(F)
      : GOTO 65
75 INPUT "DELETE SEQ # ";B
      : IF B=Ø THEN 85

8Ø FOR F=B+9 TO A+1Ø
      : A(F)=A(F+1)
      : NEXT F
      : A=A-1
      : GOTO 75
85 INPUT "INSERT NEW SEQ #";B
      : IF B=Ø THEN 35
9Ø FOR F=A+1Ø TO B+9 STEP -1
      : A(F+1) = A(F)
      : NEXT F
      : F=B+9
      : A=A+1
      : INPUT "NEW Y= ";A(F)
      : GOTO 85
```

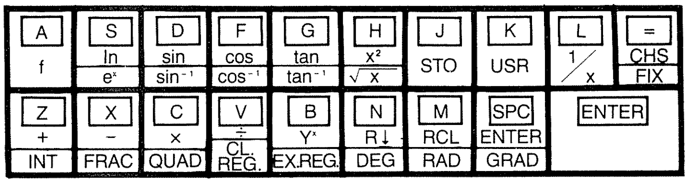
Reverse Polish Notation on a Pocket Computer

Joe H. Morrison
Radio Shack Computer Centre
1706 Dundas Street East
Mississauga, Ontario
CANADA

For all you Reverse Polish Notation lovers, here is a program for the PC-1 pocket computer (PC-2 changes follow the main article) that will replace the old Hewlett Packard you threw out! It is an almost perfect simulation of an RPN calculator, and includes a register shift key, an "exchange X and Y register" key, keys to store and recall up to 38 numbers (in the data memories), and all of the functions you would expect to find in any self-respecting calculator. Oh, and there is a quadratic notation equation solution program, as well!

OPERATION

You will need to make an overlay like this:



Keyboard Overlay for PC-1
(Hope you can write small!)

To operate the program once it is in the PC-1, use the DEF mode. You should recall that in the DEF mode, the ⟨SHFT⟩ key tells the computer which of the defined keys you want. Note that some of the keys have two functions defined. (Note: we will use ⟨UPPER CASE⟩ letters to designate actual keys on the Pocket Computers, and ⟨lower case⟩ to designate RPN functions. This is especially important because the RPN calculator has its own ⟨enter⟩ key.)

For the two function keys, the first function is called by keying ⟨SHFT⟩ and the function key, and the second function is called with ⟨SHFT⟩ ⟨A⟩, then ⟨SHFT⟩ and the key you want to execute. As in an ordinary calculator, to execute single-operand functions (like SIN, COS, etc.), you must key in the operand, and then press the function key. (For those of you that are very familiar with RPN, all of the functions provided in this program (including CHS and 1/x) automatically enter the result into the stack. The stack, by the way, is contained in variables A through D, A being the 'x' register, and D being the 't' register.)

The arithmetic functions work exactly as in an RPN calculator with one minor difference that I will get to in a minute. That is, to add two and three, you type ⟨2⟩ ⟨SHFT⟩ ⟨SPC⟩ ⟨3⟩ ⟨SHFT⟩ ⟨Z⟩. And, of course, the result is stored in the 'stack' enabling you to do calculations like: (2 + 3) × (5 + 6) by typing: ⟨2⟩ ⟨enter⟩ ⟨3⟩ ⟨plus⟩ ⟨5⟩ ⟨enter⟩ ⟨6⟩ ⟨plus⟩ ⟨multiply⟩. Now, that minor difference I mentioned. It is simply that because of the way the stack works, YOU CANNOT PERFORM A 2-OPERAND FUNCTION USING THE SAME NUMBER FOR BOTH OPERANDS. YOU WILL GET

WRONG ANSWER IF YOU DO! (Therefore, you cannot add 2 and 2, multiply 3 by 3, subtract 1 from 1, etc.) However, do not panic! That problem is taken care of by an interesting design feature of the pocket computer. You can intermix the RPN program with the algebraic operating system of the pocket computer. This means you can key in things like:

(2+2)*4 ⟨enter⟩ 8*16 ⟨ / ⟩.

Get it? Here is a more complicated example. Solve:

$$\frac{(\frac{3}{8})^3 + (4 - (\frac{5}{3} \times (3)^3)}{2 \times (\frac{4}{9}\mathrm{SIN}46^\circ\ )}$$

The required sequence (using the combination of AOS and RPN) is:

3/8 ⟨enter⟩ 3 ⟨Y^x⟩ 4 ⟨enter⟩ 5/3 ⟨enter⟩ 3^3 ⟨multiply⟩ ⟨minus⟩ ⟨plus⟩ 2 ⟨enter⟩ 4/9 ⟨enter⟩ 46 ⟨sin⟩ ⟨multiply⟩ ⟨multiply⟩ ⟨divide⟩.

On the pocket computer, the actual keystrokes are:

3/8 ⟨SHFT⟩ ⟨SPC⟩ 3 ⟨SHFT⟩ ⟨B⟩ 4 ⟨SHFT⟩ ⟨SPC⟩ 5/3 ⟨SHFT⟩ ⟨SPC⟩ 3^3 ⟨SHFT⟩ ⟨C⟩ ⟨SHFT⟩ ⟨X⟩ ⟨SHFT⟩ ⟨Z⟩ 2 ⟨SHFT⟩ ⟨SPC⟩ 4/9 ⟨SHFT⟩ ⟨SPC⟩ 46 ⟨SHFT⟩⟨D⟩ ⟨SHFT⟩ ⟨C⟩ ⟨SHFT⟩⟨C⟩ ⟨SHFT⟩ ⟨V⟩

EXTRA FEATURES

User key: The RPN program ends at line 199, so if you want to enter an additional program, you merely have to go to PROgram mode, and start your program at line 200. To execute your program, just type ⟨SHFT⟩ ⟨K⟩ (the key labeled 'user').

Data memories: If you wish to store the number in the display, simply type ⟨SHFT⟩ ⟨J⟩ (labeled 'sto') and when the question mark appears, key in the number of the memory you wish to use. (The number of available memories depends on the length of any additional programs in memory. If there are no additional programs in memory, there are 38 memories available.) Similarly, you can recall STOred numbers by typing ⟨SHFT⟩ ⟨M⟩ (labeled 'rcl') and keying in the number of the memory you wish to recall.

Trigonometry: The second functions of the ⟨N⟩, ⟨M⟩ and ⟨SPC⟩ keys switch the calculator into DEGREE, RADIAN and GRAD modes, respectively.

Formatting: The second function of the ⟨ = ⟩ key switches the display back and forth between floating point format and fixed format. (In fixed format the display is fixed to 5 decimal places.) In fixed format, you will get an error if the result of a calculation is greater than 999,999,999. However, the fixed format makes it easier to read the results of the trig functions, since they normally come out in exponential notation.

Quadratic equation solution: The second function of the ⟨C⟩ key will allow you to find the two roots of a quadratic equation (provided they are not imaginary). To use it, type ⟨SHFT⟩ ⟨A⟩ ⟨SHFT⟩ ⟨C⟩, and then you get 'A=?' enter the X^2 coefficient. When you get 'B=?', enter the X coefficient, and when you get 'C=?', enter the constant. After a few seconds, both roots will appear simultaneously on the display.

Miscellaneous: The second functions of the ⟨V⟩ and ⟨B⟩ keys are respectively, the 'clear register' and the 'exchange X and Y register' keys. When using the latter, the ⟨X⟩ number must be entered before you use the function (i.e., To calculate 6/(3+2), you type 3 ⟨enter⟩ 2 ⟨plus⟩ 6 ⟨enter⟩ ⟨exch. reg.⟩ ⟨divide⟩). The same is true of the ⟨N⟩ key, which is the register shift key—you must enter the number in the display before shifting registers.

An interesting side effect of this system is that you can turn off the pocket computer in the middle of a calculation, and can later pick up where you left off!

All in all, though the system requires more keystrokes for a given calculation than a normal RPN calculator would, I have found that it is faster to use than the regular system of the pocket computer, because I no longer need to type out 'SIN' and 'LOG', etc. Have fun with the program!

In the program listing, SQR should be replaced by the square root symbol.

Note to PC-2 users: The RPN calculator will function properly on the PC-2 with three changes. First, The PC-1 uses single character variable names only. When ever you see two variables or a number and a variable together (e.g., YY or 4X in line 51) separate the items with an '*' (for multiplication.) Second, because of the added *'s, line 51 will be too long. Split the line into two parts:

```
51 F=Ø
      : INPUT "A=?";X, "B=?";Y, "C=?";Z
      : W=SQR(Y*Y-4*X*Z)
      : U=(-Y+W)/(2*X)
      : V=(-Y-W)/(2*X)
52 PRINT U,V
      : END
```

The third change is that where you see the word ⟨SHFT⟩ in the article, substitute ⟨DEF⟩. To add two and three, you type ⟨2⟩ ⟨DEF⟩ ⟨SPC⟩ ⟨3⟩ ⟨DEF⟩ ⟨Z⟩.

```
1Ø "A" AREAD X
      : F=ABS (F-1)
      : PRINT X
      : END
2Ø " " AREAD X
      : IF F=Ø LET D=C
      : C=B
      : B=A
      : A=X
      : PRINT A
      : END
25 GRAD
      : F=Ø
      : PRINT X
      : END
3Ø "Z" AREAD X
      : IF F=1 LET D=C
      : F=Ø
      : C=B
      : B=A
      : A=INT X
      : PRINT A
      : END
33 IF X<>A LET A=A+X
      : PRINT A
      : END
35 A=B+A
      : B=C
      : C=D
      : PRINT A
      : END

4Ø "X" AREAD X
      : IF F=1 LET D=C
      : F=Ø
```

```
     : C=B
     : B=A
     : A=X-INT X
     : PRINT A
     : END
43 IF X<>A LET A=A-X
     : PRINT A
     : END
45 A=B-A
     : B=C
     : C=D
     : PRINT A
     : END
5Ø "C" AREAD X
     : IF F=Ø GOTO 53
51 F=Ø
     : INPUT "A=?"; X, "B=?"; Y, "C=?"; Z
     : W=SQR(YY-4XZ)
     : U=(-Y+W)/2X
     : V=(-Y-W)/2X
     : PRINT U, V
     : END
53 IF X<>A LET A=AX
     : PRINT A
     : END
55 A=BA
     : B=C
     : C=D
     : PRINT A
     : END
6Ø "V" AREAD X
     : IF F=1 LET D=Ø
     : F=Ø
     : C=Ø
     : B=Ø
     : A=Ø
     : X=Ø
     : PRINT A
     : END
63 IF X<>A LET A=A/X
     : PRINT A
     : END
65 A=B/A
     : B=C
     : C=D
     : PRINT A
     : END
7Ø "B" AREAD X
     : IF F=1 LET X=B
     : B=A
     : A=X
     : F=Ø
     : PRINT A
     : END
73 IF X<>A LET A=A^X
     : PRINT A
     : END
75 A=B^A
     : B=C
     : C=D
     : PRINT A
     : END
8Ø "H" AREAD X
     : IF F=1 LET D=C
     : F=Ø
     : C=B
     : B=A
     : A=SQR X
     : PRINT A
     : END
83 D=C
     : F=Ø
     : C=B
     : B=A
     : A=XX
```

```
     : A=XX
```

```
     : PRINT A
     : END
9Ø "D" AREAD X
     : IF F=1 LET D=C
     : F=Ø
     : C=B
     : B=A
     : A=ASN X
     : PRINT A
     : END
93 D=C
     : F=Ø
     : C=B
     : B=A
     : A=SIN X
     : PRINT A
     : END
1ØØ "F" AREAD X
     : IF F=1 LET D=C
     : B=Ø
     : C=B
     : B=A
     : A=ACS X
     : PRINT A
     : END
1Ø3 D=C
     : F=Ø
     : C=B
     : B=A
     : A=COS X
     : PRINT A
     : END
11Ø "G" AREAD X
     : IF F=1 LET D=C
     : F=Ø
     : C=B
     : B=A
     : A=ATN X
     : PRINT A
     : END
113 D=C
     : F=Ø
     : C=B
     : B=A
     : A=TAN X
     : PRINT A
     : END
12Ø "S" AREAD X
     : IF F=1 LET D=C
     : F=Ø
     : C=B
     : B=A
     : A=EXP X
     : PRINT A
     : END
123 D=C
     : F=Ø
     : C=B
     : B=A
     : A=LN X
     : PRINT A
     : END
13Ø "=" AREAD X
     : IF F=Ø LET D=C
     : C=B
     : B=A
     : A=-X
     : PRINT A
     : END

132 F=Ø
     : USING
     : R=ABS (R-1)
     : IF R=1 USING "##########.#####"
133 PRINT X
     : END
```

```
14Ø "L" AREAD X
    : IF F=Ø LET D=C
    : C=B
    : B=A
    : A=1/X
    : PRINT A
    : END
141 F=Ø
    : END
15Ø "J" AREAD X
    : IF F=Ø INPUT Y
    : A(Y+26)=X
    : PRINT X
    : END
151 F=Ø
    : END
16Ø "M" AREAD X
    : IF F=Ø INPUT Y
    : D=C
    : C=B
    : B=A
    : A=A(Y+26)
    : PRINT A
    : END
161 F=Ø
    : RADIAN
    : PRINT X
    : END
17Ø "N" AREAD X
    : IF F=1 LET F=Ø
    : DEGREE
    : PRINT X
    : END
172 X=A
    : A=B
    : B=C
    : C=D
    : D=X
    : PRINT A
    : END
199 "K"
```

Pocket Computer Bugs, Errors and Fixes

Calendar (26-3529)

Please note the following correction to page 7, example 3 in the Calendar program manual:

⟨SHFT⟩ ⟨K⟩ should read ⟨SHFT⟩ ⟨X⟩

Ode to Micro Idiom

Joe Carr
3950 Cromart Ct. N.
Ft.Worth, TX 76133

When speaking the lore of computer technology,
I offer the layman no mean apology.
Handshaking, polling, and flag waving appear,
Even if it's not an election year.
Addressing, bus drivers, and integration,
Have nothing to do with desegregation.
Booting the terminal hurts not a Bit,
But the CMOS Ram might throw a fit.
Wares, soft, firm, and hard, you'll find by the bunch,
But the dreaded hidden bug could still eat your lunch.
So grab your joysticks and hold onto your hair,
'Cause interchangeable programs are still pretty rare.
Memory mapped might be the key,
But what's it all mean?
Don't ASCII* me.

*Pronounced ASKY

Pearson ⟨R⟩ Correlation

Galen Currah
11023 N.E. Glisan St.
Portland, OR 97220

The Pocket Computer has gone with me to West Africa on demographic and other survey research. This program figures amongst several I have written for statistical analysis.

The Pearson Product-Moment Coefficient of Correlation (r) is calculated, showing the degree of association between up to 50 pairs of numbers. 616 steps and 103 memories are utilized.

The formula followed reads:

$$r = \frac{n\sum xy - \sum x \sum y}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

```
1Ø: CLEAR
     : PRINT "PROD MOMT CORRL COEF (R)"
     : INPUT " NO. OF PAIRS: "; A(1Ø1)
12: FOR N = 1 TO A(1Ø1)
     : M = N + 5Ø
     : PAUSE "PAIR ";N;":"
     : INPUT A(N),A(M)
     : NEXT N
14: FOR N = 1 TO A(1Ø1)
     : M = N + 5Ø
     : A(1Ø2) = A(1Ø2) + A(N) * A(M)
     : A(1Ø3) = A(1Ø3) + A(N) * A(N)
     : PAUSE "WAIT"
16: A(1Ø4) = A(1Ø4) + A(N)
     :A(1Ø5) = A(1Ø5) + A(M) * A(M)
     : A(1Ø6) = A(1Ø6) + A(M)
     : NEXT N
18: A(1Ø7) = A(1Ø2) * A(1Ø1)
     : A(1Ø8) = A(1Ø3) * A(1Ø1)
     : A(1Ø3) = A(1Ø4) * A(1Ø4)
2Ø: A(11Ø) + A(1Ø5) * A(1Ø1)
     : A(115) = A(1Ø6) * A(1Ø6)
     : A(116) = A(1Ø6) * A(1Ø4)
22: A(113) = A(1Ø7) - A(116)
     : A(114) = A(1Ø8) - A(1Ø3)
     : A(117) = A(11Ø) - A(115)
24: A(118) = A(114) * A(117)
     : LET A(11Ø) = |A(118)
     : A(118) = A(113) / A(11Ø)
26: A(118) = INT(A(118) * 1ØØ) / 1ØØ
     : BEEP 1
     : PRINT "      R= "; A(118)
```

LOADCOPY and LOADDATA

George Fraser
3403 Bear Creek Drive
Newbury Park, CA 91320

LOADCOPY and LOADDATA are two programs which will load a stand alone machine language utility program which I have written and assembled. I have tested the programs only with Extended Color BASIC 1.0, but I believe they should also work for the 4K computer if they are relocated in memory.

LOADCOPY will create a stand-alone machine language program. Everyone, whether a casual hobbyist or professional programmer, should find this program extremely useful. This program will change a BASIC program in memory by copying any requested program line to a new line. This is useful when you want to move a line to somewhere else in the program. Just copy the old line to the new location and delete the old line. You don't have to retype the whole thing. Or if you need a series of BASIC lines which are almost the same, you can use the copy repeatedly and then use "edit" to put the lines into final form. It is a very valuable tool for program development and maintenance.

I keep my "COPY" program on a cassette tape in machine language format. Then whenever I need it, I do the following:

```
CLEAR 1ØØ,16Ø85       'THIS RESERVES 29Ø BYTES FOR COPY
CLOADM "COPY"
EXEC                  'OR EXEC 16Ø86
```

To get your own copy of the program, type in my BASIC program "LOADCOPY" carefully. After it will run O.K. and you have double checked it for accuracy, CSAVE it to tape. Then put in a cassette for recording and save the machine language program with:

```
CSAVEM "COPY", 16Ø36. 16383, 16Ø86
```

Make sure you have program "LOADCOPY" CSAVEd on tape before you try to "EXEC 16086" for the first time. Any mistake in the machine language code can easily wipe out everything in memory. Owners of the 4K Color Computer should change lines 30 and 40 to:

```
3Ø CLEAR 1ØØ, 3797
4Ø S = 3798
```

and then "EXEC 3798." The machine language program can be relocated to any protected area of memory by changing lines 30 and 40.

It overlays the "EXEC" line on the screen with "COPY" followed by a cursor character. Type in the line number to be copied (the source line), a comma, and the new line number (the target line number). Press ⟨**ENTER**⟩. If you omit the target line number, it defaults to one greater than the source.

```
COPY 3Ø, 2ØØ     'WILL COPY LINE 3Ø TO LINE 2ØØ
COPY 12Ø         'WILL COPY LINE 12Ø TO LINE 121.
```

If COPY generates one of the following error messages, your BASIC program was not changed.

?OV Overflow. Source of target line number is greater than 63999.

?UL Undefined line. The source line number doesn't exist.

?TG Target error. The target line number already exists. Delete it first, and then do the copy.

?OM Out of memory. Try to free up more memory by clearing less string space or deleting some program comments.

The backspace key doesn't work in COPY, but you can interrupt the program with the ⟨**BREAK**⟩ key which causes an "?OV ERROR."

The machine language program is the shortest and most refined version of "COPY" that I have developed. It uses the hardware stack to store intermediate results, which guarantees a complex assembly language program.

The first half of the program is occupied with keying in the line numbers and converting them to binary. It scans the BASIC program to find the source line number; then it scans the BASIC program again to find where the target line is to be inserted and to make sure that the target line number does not already exist.

It has to make sure there is enough free memory left before it starts to do the copy; then it opens up enough room in your BASIC program at the target address for the copy. At long last, it copies the source line to the target area and slaps in your target line number. Finally, it has to update BASIC's next line pointer in each BASIC line from the target line to the end of the program.

All in all, this little program has a lot of very complex things to do just to copy one line!

```
1Ø '  "LOADCOPY" FOR THE COLOR COMPUTER
2Ø '   BY G.FRASER  MARCH 1982
3Ø CLEAR 1ØØ, 16Ø85
4Ø S=16Ø86   'START ADDRESS
5Ø E=S+297   'END ADDRESS
6Ø FOR A=S TO E
7Ø READ Bc POKE A,B
     : C=C+B
8Ø NEXT
9Ø IF C=34462 THEN PRINI "CSAVEM";S;E;S
     : END ELSE PRINT C
     : STOP
1ØØ DATA 222, 136, 51, 2ØØ, 224, 2Ø4, 67, 79, 237, 193
11Ø DATA 2Ø4, 8Ø, 89, 237, 193, 2Ø4, 96, 141, 237, 193
12Ø DATA 198, 26, 167, 192, 9Ø, 38, 251, 51, 2ØØ, 229
13Ø DATA 5Ø, 12Ø, 141, 22, 175, 1ØØ, 129, 13, 39, 8
14Ø DATA 141, 14, 1Ø9, 97, 42, 6, 174, 1ØØ, 48, 1
15Ø DATA 141, 71, 175, 1Ø2, 32, 8Ø, 2Ø4, Ø, 255, 237
16Ø DATA 98, 95, 237, 1ØØ, 189, 161, 193, 129, 3, 39
17Ø DATA 39, 129, 13, 39, 46, 129, 44, 39, 4Ø, 129
18Ø DATA 48, 37, 237, 129, 57, 34, 233, 141, 4Ø, 128
19Ø DATA 112, 167, 99, 236, 1ØØ, 142, Ø, 9, 227, 1ØØ
2ØØ DATA 37, 8, 48, 31, 38, 248, 227, 98, 36, 2Ø8
21Ø DATA 134, 96, 167, 196, 2Ø4, 79, 86, 32, 76, 141
22Ø DATA 8, 174, 1ØØ, 14Ø, 249, 255, 34, 238, 57, 138
```

```
230 DATA 64, 198, 141, 237, 192, 57, 134, 96, 167, 196
240 DATA 16, 158, 25, 31, 35, 16, 174, 196, 39, 37
250 DATA 174, 66, 172, 100, 38, 243, 239, 228, 236, 196
260 DATA 163, 228, 237, 98, 16, 158, 25, 31, 35, 16
270 DATA 174, 196, 39, 46, 174, 66, 172, 102, 34, 40
280 DATA 38, 241, 204, 84, 71, 32, 10, 204, 85, 76
290 DATA 32, 5, 204, 79, 77, 50, 98, 48, 140, 10
300 DATA 237, 1, 198, 10, 189, 185, 166, 50, 104, 57
310 DATA 63, 88, 88, 32, 69, 82, 82, 79, 82, 13
320 DATA 17, 163, 228, 34, 6, 236, 228, 227, 98, 237
330 DATA 228, 220, 27, 31, 2, 227, 98, 52, 70, 48
340 DATA 232, 206, 172, 225, 37, 202, 31, 1, 159, 27
350 DATA 159, 29, 159, 31, 16, 172, 228, 39, 6, 166
360 DATA 162, 167, 130, 32, 245, 52, 16, 174, 100, 16
370 DATA 174, 102, 166, 128, 167, 192, 49, 63, 38, 248
380 DATA 53, 70, 174, 102, 175, 66, 237, 196, 31, 3
390 DATA 236, 196, 39, 169, 227, 98, 32, 244
```

"LOADDATA" will create a stand-alone machine language program I call "DATA". I use this program whenever I want to include a machine language subroutine in a BASIC program in the form of "DATA" statements which will be read and POKEd into memory to load the subroutine.

I usually keep "DATA" on a cassette tape in machine language format which will load starting at address 15360.

Suppose I have written a machine language subroutine which will be called by BASIC program "X", and I have the machine language code already in memory locations 16200-16383. (For example, I may have hand assembled it and POKEd it into memory.) Now I want to convert it into BASIC data statements to go on to the end of program "X".

Suppose I also have the calling program "X" in memory, complete except for the data statements. Here is where my program "DATA" is really a big help. I will now do the following:

```
CLEAR 100,15359      'THIS WILL PROTECT MEMORY FOR
                      "DATA".
CLOADM "DATA"        'LOADS AT ADDRESSES 15360-15676
EXEC                 'OR EXECD 15360
```

Of course, I have to make sure that "DATA" does not overlay my other machine language subroutine. As soon as I enter "EXEC", that line on the screen is overlaid with "DATA" followed by a cursor character. I now key in the desired range of memory that I want to capture so that the line now looks like:

```
DATA 16200,16383
```

The two numbers are the starting and ending address in decimal. When I press ⟨**ENTER**⟩, "DATA" instantly creates the data line on the end of my BASIC program "X". "DATA" can generate three error messages:

?OV Overflow. You typed an address greater that 65535.

?SQ Sequence Error. The second address is less than the first.

?OM Out of Memory. Perhaps some, but not all, of the DATA statements were created.

The backspace key doesn't work in "DATA" but you can interrupt the program with the ⟨**BREAK**⟩ key, which causes a "?OV ERROR".

To get your own copy of the program, type in my BASIC program "LOADDATA" carefully. After it will run O.K. and you have double checked it for accuracy, CSAVE it to tape; then put in a cassette for recording and save the machine language program with:

```
CSAVEM "DATA", 15360, 15676, 15360
```

Make sure you have program "LOADDATA" CSAVEd on tape before you try to "EXEC 15360" for the first time. Any mistake in the machine language code can easily wipe out everything in memory. Owners of the 4K Color Computer may change lines 30 and 40 to:

```
30 CLEAR 100,3778
40 S=3779
```

and then "EXEC 3779". The machine language program can be relocated to any protected area of memory by changing lines 30 and 40.

I created my first copy of "DATA" with my 6809 assembler. I then used the machine language program to create the DATA statements in the "LOADDATA" BASIC program.

You can customize "DATA" to some extent with the following patches. In each case, "S" is the start address you put in line 40 of "LOADDATA".

1. The line number increment for successive "DATA" line is set to 10. You can vary that value from 1 to 15 by POKEing that value into S+218. You must not exceed 15.
2. Ten numbers are put on a DATA line. If you want more or fewer, POKE that number into S+228. If you exceed 15, it probably won't catch an "OM" error in time.
3. "DATA" generates a space after each comma for readability. To save memory, you can eliminate the space with POKE S+284,160.

The program LOADDATA.

```
10 '  "LOADDATA" FOR THE COLOR COMPUTER
20 '   BY G.FRASER  MARCH 1982
30 CLEAR 100, 15359
40 S=15360    'START ADDRESS
50 E=S+316    'END ADDRESS
60 FOR A=S TO E
70 READ B
     : POKE A,B
     : C=C+B
80 NEXT
90 IF C=31862 THEN PRINT "CSAVEM";S;E;S
     : END ELSE PRINT C
     : STOP
100 DATA 222, 136, 51, 200, 224, 134, 2, 31, 139,
     204
110 DATA 68, 65, 237, 193, 134, 84, 237, 193,
     204, 96
120 DATA 141, 237, 193, 198, 26, 167, 192, 90,
     38, 251
130 DATA 51, 200, 229, 141, 49, 159, 4, 129, 13,
     39
140 DATA 24, 141, 41, 13, 1, 43, 16, 156, 4, 36
150 DATA 14, 141, 5, 204, 83, 81, 32, 89, 134, 96
160 DATA 167, 196, 57, 158, 4, 159, 6, 141, 245,
     190
170 DATA 0, 25, 31, 18, 174, 164, 39, 119, 236,
     34
180 DATA 221, 8, 32, 244, 204, 0, 255, 221, 0, 95
190 DATA 221, 8, 221, 2, 189, 161, 193, 129, 3,
     39
200 DATA 39, 129, 13, 39, 67, 129, 44, 39, 61,
     129
210 DATA 48, 37, 237, 129, 57, 34, 233, 141, 56,
     128
220 DATA 112, 151, 1, 220, 2, 142, 0, 9, 211, 2
230 DATA 37, 8, 48, 31, 38, 248, 211, 0, 36, 208
240 DATA 50, 98, 141, 170, 204, 79, 86, 48, 140,
     10
250 DATA 237, 1, 204, 0, 10, 31, 139, 126, 185,
     166
260 DATA 63, 88, 88, 32, 69, 82, 82, 79, 82, 13
270 DATA 141, 3, 158, 2, 57, 138, 64, 198, 141,
     237
280 DATA 192, 57, 79, 95, 237, 161, 31, 139, 31,
     33
```

```
290 DATA 159, 27, 159, 29, 159, 31, 57, 52, 32,
     48
300 DATA 232, 142, 172, 225, 34, 7, 141, 230,
     204, 79
310 DATA 77, 32, 190, 31, 35, 158, 8, 48, 10, 49
320 DATA 34, 175, 161, 159, 8, 158, 4, 198, 10,
     215
330 DATA 10, 204, 134, 32, 237, 161, 15, 11, 166,
     128
340 DATA 198, 48, 129, 100, 37, 5, 128, 100, 92,
     32
350 DATA 247, 141, 51, 198, 48, 129, 10, 37, 5,
     128
360 DATA 10, 92, 32, 247, 141, 38, 31, 137, 203,
     48
370 DATA 231, 160, 156, 6, 34, 17, 10, 10, 39, 7
380 DATA 204, 44, 32, 237, 161, 32, 205, 141, 8,
     159
390 DATA 4, 32, 160, 141, 2, 32, 141, 79, 167,
     160
400 DATA 16, 175, 196, 57, 13, 11, 38, 6, 193, 48
410 DATA 39, 4, 12, 11, 231, 160, 57
```

Mailing Labels on the CoCo

David L. Banaszak
168 Stratmore
New Carlisle, OH 45355

The program MAIL allows me to print mailing address labels for up to 105 names and addresses. Before loading or entering MAIL, type PCLEAR 1 or POKE 25, 6 NEW to ensure sufficient memory. Basically MAIL is an expansion of the "Inventory Shopping List" (ISL) program on page 295 of the Color BASIC Manual. Also, several finer features of MAIL were inspired by previous newsletters. I'm still refining MAIL, but I have used this version to print address labels for our Christmas card list and several non-profit club's mailing lists. The equipment and supplies I use to run MAIL are:

1). Line Printer VII
2). Color Computer with 16K Extended Color BASIC
3). Tape recorder and tape
4). Television
5). 9¹/₂" fanfold paper (26-1433)
6). Dry gum labels (26-1456)

After pressing ⟨ENTER⟩, the screen displays:

1). The memory left
2). My name, street, city, state, zip, and phone number
3). The 9 option main menu.

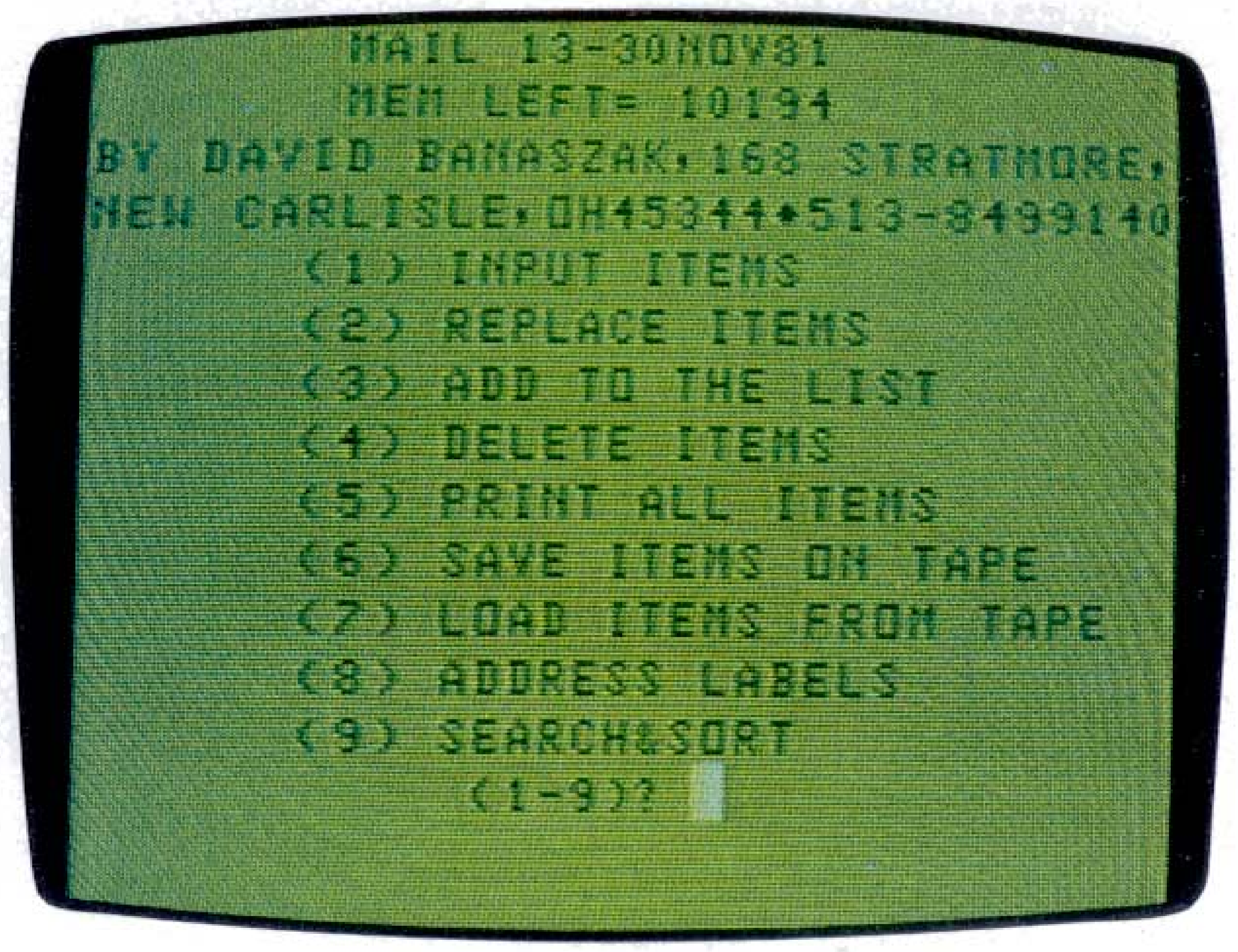
Option 1 on the main menu (lines 190-340) requests input items in the following format:
LAST NAME, FIRST NAME, STREET, CITY, STATE ZIP * ACPH CODE:.

For each item, the four commas and * must be typed. Characters after the * are for additional information such as phone numbers and the : serves as a separator if one wants to include 2 or 3 addresses per input item. L is used as a current item string length counter and should not be allowed to exceed 255. LL is a coarse total string character counter which should be kept less than about 7000 to avoid running out of string space. INKEY$ is used for input and PEEK(343) is used to detect left arrow backspace ←. Option 2 (lines 350-460) replaces items, Option 3 (lines 220-340) adds items to the list and Option 4 (lines 470-590) deletes items all similar to program ISL. Option 5 (lines 600-780) allows the user to view the items on the screen as in ISL and/or on the printer. Option 6 (lines 790-950) saves items on a user provided file name and Option 7 (lines 960-1150) loads items from a user specified file name.

Options 8 and 9 are not included in ISAL. Option 8 (lines 1160-1300) separates each string item S$(n) into 5 ports LN$(1) through LN$(5). The array LN$ is then printed in address label format as directed by the choices selected in lines 1170, 1175 and 1177. Currently option 9 (lines 1330-4000) contains two working functions. Lines 1400-1590 will reformat N items containing three addresses each separated by colons (:), in 3N times containing 1 address per item. Lines 1600 to 1999 alphabetized the items by last names since the last name is first in each element of S$. Expanded functions I hope to add to the option 9 section are the ability to search the entire list for some user specified character string and an ability to sort by user specified parameter such as street or city or phone or zip.

```
10 CLEAR 8000
     : DIM S$(105)
     : DIM LN$(5)
20 LL = 0
30 CLS
33 PRINT @7, "MAIL 13-30NOV81"
34 PRINT @39, "MEM LEFT="; MEM
35 PRINT @64, "BY DAVID BANASZAK, 168 STRATMORE,
   NEW CARLISLE, OH45344*513-8499140"
50 PRINT @134, "(1) INPUT ITEMS"
60 PRINT @166, "(2) REPLACE ITEMS"
70 PRINT @198, "(3) ADD TO THE LIST"
80 PRINT @230, "(4) DELETE ITEMS
90 PRINT @262, "(5) PRINT ALL ITEMS"
100 PRINT @294, "(6) SAVE ITEMS ON TAPE"
110 PRINT @326, "(7) LOAD ITEMS FROM TAPE"
120 PRINT @358, "(8) ADDRESS LABELS"
130 PRINT @390, "(9) SEARCH&SORT"
140 PRINT @427, "(1-9)";
150 INPUT M
160 IF M < 0 OR M > 9 THEN 20
170 ON M GOSUB 190, 350, 220, 470, 600, 790, 960, 1160, 1330
180 GOTO 30
190 'INPUT/ADD
200 IF Y > 1 THEN RETURN
210 Y = 1
220 L = 0
     : CLS
     : PRINT @3, "LN, FN, ST, CY, STZP * ACPHCODE:
230 PRINT @34, "PRESS <ENTER> WHEN DONE"
240 PRINT
     : PRINT "ITEM" Y;
250 S$(Y) = ""
260 A$ = INKEY$
     : IFA$ = "" THEN 260
274 IF PEEK(343) <> 247 THEN 280
275 L = L - 1
     : IF L < 0 THEN 220
276 S$(Y) = LEFT$(S$(Y),L)
279 GOTO 300
280 IF A$ = CHR$(13) THEN 310
290 S$(Y) = S$(Y) + A$
300 L = LEN(S$(Y))
301 PRINT @129, "LL="; LL; "L="; L
     : PRINT @160, S$(Y)
302 IF L = 0 THEN 220
305 GOTO 260
310 IF S$(Y) = ""THEN RETURN
320 Y = Y + 1
330 LL = LL + L
     : GOTO 220
340 '
350 'REPLACE
360 N = 0
     : L = 0
370 CLS
     :  PRINT @9, "REPLACE ITEMS"
380 PRINT @34, "PRESS <ENTER> WHEN FINISHED"
390 PRINT
     : INPUT "ITEM NO. TO REPLACE"; N
400 PRINT "OLD ITEM WAS: "; S$(N)
     : LL=LL-LEN(S$(N))
     : S$(N) = ""
405 IF N = 0 THEN RETURN
410 PRINT "REPLACEMENT";
420 A$ = INKEY$
     : IF A$ = "" THEN 420
430 IF A$ = CHR$(13) THEN 460
431 IF PEEK(343) <> 247 THEN 440
432 L = L - 1
     : IF L < 0 THEN 360
```

```
433 S$(N) = LEFT$(S$(N), L)
434 GOTO 441
44Ø S$(N) = S$(N) + A$
441 L = LEN(S$(N))
444 PRINT @129, "LL="; LL; "L="; L
      : PRINT @16Ø, S$(N)
445 IF L = Ø THEN 42Ø
45Ø GOTO 42Ø
46Ø LL = LL + L
      : GOTO 35Ø
47Ø 'DELETE
48Ø N = Ø
49Ø CLS
      : PRINT @9, "DELETE ITEMS"
5ØØ PRINT @34, "PRESS <ENTER> WHEN FINISHED
51Ø PRINT
      : INPUT "ITEM TO DELETE"; N
52Ø IF N > Y - 1 THEN 51Ø
53Ø IF N = Ø THEN RETURN
54Ø FOR X = N TO Y - 2
55Ø S$(X) = S$(X+1)
56Ø NEXT X
57Ø S$(X) = ""
58Ø Y = Y - 1
59Ø GOTO 47Ø
6ØØ ' PRINT
61Ø CLS
62Ø PRINT @134, "1 SCREEN ONLY"
63Ø PRINT @166, "2 PRINTER ONLY"
64Ø PRINT @198, "3 SCREEN & PRINTER"
65Ø INPUT M
      : IF M < 1 OR M > 3 THEN 62Ø
66Ø ON M GOTO 67Ø, 75Ø, 67Ø
67Ø FOR X = 1 TO Y - 1 STEP 15
68Ø FOR Z = X TO X + 14
69Ø PRINT Z; S$(Z)
7ØØ NEXT Z
71Ø INPUT "PRESS <ENTER> TO CONTINUE"; C$
72Ø NEXT X
73Ø IF M <> 3 THEN 78Ø
75Ø FOR X = 1 TO 1Ø5
76Ø IF S$(X) <> "" THEN PRINT#-2, X; S$(X)
77Ø NEXT X
78Ø RETURN
79Ø 'SAVE ITEMS
8ØØ CLS
      : PRINT @135,"SAVE ITEMS ON TAPE"
81Ø INPUT "TAPE FILE NAME"; FF$
82Ø MOTOR ON
      : AUDIO ON
83Ø PRINT @224, "POSITION TAPE MOTOR & AUDIO IS ON PRESS <ENTER> WHEN READY"
      : INPUT R$
84Ø MOTOR OFF
      : AUDIO OFF
85Ø PRINT @356, "PRESS PLAY AND RECORD"
86Ø PRINT @388, "PRESS <ENTER> WHEN READY"
87Ø INPUT R$
88Ø OPEN "O", #-1, FF$
89Ø FOR X = 1 TO Y - 1
9ØØ PRINT #-1, S$(X)
91Ø NEXT X
92Ø CLOSE #-1
93Ø CLS
      : PRINT"FILENAME IS "; FF$
94Ø PRINT "STOP RECORDER & <ENTER> WHEN READY"
      : INPUT R$
95Ø RETURN
96Ø ' LOAD ITEMS
97Ø CLS
      : PRINT @136, "LOAD ITEMS FROM TAPE"
98Ø INPUT "TAPE FILE NAME"; FF$
99Ø MOTOR ON
      : AUDIO ON
1ØØØ PRINT @235, "REWIND TAPE MOTOR & AUDIO ON"
1Ø1Ø PRINT "PRESS <ENTER> WHEN READY"
      : INPUT R$
1Ø2Ø MOTOR OFF
      : AUDIO OFF
1Ø3Ø PRINT @388, "PRESS PLAY THEN <ENTER> WHEN READY"
1Ø4Ø INPUT R$
1Ø5Ø OPEN "I", #-1, FF$
1Ø6Ø Y = 1
1Ø7Ø IF EOF(-1) THEN 112Ø
1Ø8Ø INPUT #-1, S$(Y)
1Ø9Ø PRINT S$(Y)
1Ø95 LL = LL + LEN(S$(Y))
11ØØ Y = Y + 1
111Ø GOTO 1Ø7Ø
112Ø CLOSE #-1
113Ø PRINT "FILE "; FF$; " HAS BEEN LOADED     LL="; LL
114Ø PRINT "STOP RECORDER & <ENTER> WHEN READY"
      : INPUT R$
115Ø RETURN
116Ø CLS
      : PRINT @2, "THIS IS THE AREA FOR ADDRESS LABELS"
117Ø INPUT "FIRST RECORD,LAST RECORD"; I1, I2
1175 INPUT "Ø FOR SCREEN 2 FOR PRINTER"; U
      : IF U =1 OR U > 2 OR U < Ø THEN 1175
1177 CLS
      : PRINT @33, "INSERT LABEL PAPER HERE-# OF COLUMNS 1 OR 2 <ENTER>"
      : INPUT CC
      : IF CC < 1 OR CC > 2 THEN 1177
118Ø FOR I = I1 TO I2
119Ø R$ = S$(I)
12ØØ FOR J = 1 TO 5
121Ø P = INSTR(1, R$, ",")
1215 IF J = 5 THEN P = INSTR(1,R$,"*")
122Ø IF P = Ø THEN 126Ø
123Ø L = P - 1
124Ø LN$(J) = LEFT$(R$,L)
125Ø R$ = MID$(R$,P+1)
126Ø NEXT J
1265 LN$(1) = LN$(2) = " " + LN$(1)
      : LN$(5) = LN$(4) + "," + LN$(5)
1268 ON CC GOTO 1272, 128Ø
1272 PRINT#-U, ""
1273 PRINT#-U, TAB(1) LN$(1)
1274 PRINT#-U, TAB(1) LN$(3)
1275 PRINT#-U, TAB(1) LN$(5)
1276 PRINT#-U, ""
      : PRINT#-U, ""
1277 GOTO 129Ø
128Ø PRINT#-U, ""
1281 PRINT#-U, TAB(1) LN$(1)
      : TAB(41) LN$(1)
1282 PRINT#-U, TAB(1) LN$(3); TAB(41) LN$(3)
1283 PRINT#-U, TAB(1) LN$(5); TAB(41) LN$(5)
1284 PRINT#-U,""
      : PRINT#-U,""
1289 FOR J = 1 TO 999
      : NEXT J
129Ø NEXT I
13ØØ RETURN
133Ø CLS
      : PRINT@5, "SEARCH & SORT AREA"
134Ø PRINT @134, "1-3 LINE TO 1"
135Ø PRINT @166, "2-SORT BY LAST NAME"
136Ø PRINT @198, "3-SEARCH"
137Ø PRINT @23Ø, "4-SORT BY ITEM#"
138Ø PRINT @262, "5-DONE"
139Ø INPUT M
      : IF M < Ø OR M > 5 THEN 133Ø
14ØØ ON M GOTO 142Ø, 161Ø, 133Ø, 133Ø, 4ØØØ
141Ø '3 TO 1
142Ø IF Y < 2 OR Y > 34 THEN 133Ø
143Ø FOR X = Y TO 1 STEP-1
144Ø XN = X * 3
145Ø 'DERIVE 3 PARTS HERE
146Ø S$(XN) = S$(X)
      : S$(X) = ""
      : 'PRACTICE
147Ø FOR I = 2 TO 1 STEP-1
148Ø P = INSTR(1, S$(XN), ":")
149Ø IF P = Ø THEN 155Ø
15ØØ L = P - 1
151Ø S$(XN-I) = LEFT$(S$(XN), L)
152Ø S$(XN) = MID$(S$(XN), P + 1)
155Ø NEXT I
157Ø NEXT X
158Ø Y = Y * 3 - 2
1585 SOUND 128, 99
159Ø GOTO 133Ø
16ØØ 'SORT BY LAST NAME
161Ø CLS
      : PRINT "SORTING-WAIT"
162Ø FOR X = 1 TO Y - 2
163Ø FOR I = X + 1 TO Y - 1
164Ø IF S$(I) < S$(X) THEN R$ = S$(X)
      : S$(X) = S$(I)
      : S$(I) = R$
165Ø NEXT I
      : NEXT X
1999 GOTO 133Ø
4ØØØ RETURN
```



```
         MAIL 13-30NOV81
         MEM LEFT= 10194
BY DAVID BANASZAK,168 STRATMORE,
NEW CARLISLE,OH45344*513-8499140
      (1) INPUT ITEMS
      (2) REPLACE ITEMS
      (3) ADD TO THE LIST
      (4) DELETE ITEMS
      (5) PRINT ALL ITEMS
      (6) SAVE ITEMS ON TAPE
      (7) LOAD ITEMS FROM TAPE
      (8) ADDRESS LABELS
      (9) SEARCH&SORT
           (1-9)?
```

Radio Shack Computer Center Addresses and Telephone Numbers

ALABAMA

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

ARIZONA

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

ARKANSAS

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

CALIFORNIA

ANAHEIM 509 Katella, (714) 776-9540
BAKERSFIELD 2018 Chester Ave., (805) 323-8074
BERKELEY 1922 Grove St., (415) 848-9170
BEVERLY HILLS 8500 Wilshire Blvd., (213) 659-8870
CANOGA PARK 8371 Topanga Canyon, (213) 347-9800
CARMICHAEL 6305 Fairoaks Blvd., (916) 484-6815
CHICO 1834 Mangrove Ave., (916) 894-0101
CHULA VISTA 1201 3rd Ave., (714) 420-3810
CITRUS HEIGHTS 7405 Green Back Ln. at San Juan, (916) 723-5183, 969-8752
DOWNEY 8031 Florence Ave., (213) 927-7882
ESCONDIDO 347 W. Mission Ave., (714) 741-6032
FREMONT Fremont Hub, 39114 Fremont, (408) 790-1112
FRESNO Princeton S/C, 2721 N. Blackstone Ave., (209) 225-5551
GARDEN GROVE 12821 Knott St., (714) 891-1160
GLENDALE 236 N. Brand Blvd., (213) 246-9310
HAYWARD 20942 Mission Blvd., (415) 278-2888
LAKEWOOD 5830 Lakewood Blvd., (213) 920-9671
LA MESA 5346 Jackson Dr., (714) 460-3610
LONG BEACH 2119 Bellflower Blvd., (213) 597-3377
LOS ANGELES 740 S. Olive St., (213) 629-2455
MONTCLAIR 5237 Arrow Hwy., (714) 625-1541
MONTEREY 484 Washington St., (408) 375-8430
MOUNTAIN VIEW 1933 El Camino Real W., (415) 961-0542
OAKLAND 1733 Broadway, (415) 763-3183
PASADENA 575 S. Lake Ave., (213) 449-5424
RIVERSIDE 3844 La Sierra Ave., (714) 689-0340
SACRAMENTO 4749 J. St., (916) 454-3287
SAN BERNARDINO 764 Inland Center Dr., (714) 884-6871
SAN DIEGO 3062 Clairemont Dr., (714) 276-6050; 3902 El Cajon Blvd., (714) 280-0227
SAN FRANCISCO One Market Place, (415) 777-9810; 2920 Geary Blvd., (415) 221-8716
SAN JOSE 1228 S. Bascom Ave., (408) 297-2603
SAN MATEO 3180 Campus Dr., (415) 573-8607
SANTA ANA 2320 S. Fairview St. (at Warner), (714) 966-2787
SANTA BARBARA 4141 State St. A-1, (805) 967-4538
SANTA FE SPRINGS 14138 East Firestone Blvd., (213) 921-0702
STOCKTON College Sq. S/C, 963 West March Lane, (209) 957-3676
TARZANA 18545 Ventura Blvd., (213) 343-1696
TORRANCE 3840 Sepulveda at Hawthorne, (213) 373-0306
VENTURA 4005 E. Main St., (805) 654-0196
WEST COVINA 2516 E. Workman St., (213) 915-5791

COLORADO

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

CONNECTICUT

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

DELAWARE

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

DISTRICT OF COLUMBIA

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

FLORIDA

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

GEORGIA

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

IDAHO

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

ILLINOIS

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

INDIANA

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

IOWA

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

KANSAS

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

KENTUCKY

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

LOUISIANA

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

MAINE

BANGOR Maine Square, (207) 945-6491

MARYLAND

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

MASSACHUSETTS

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

MICHIGAN

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

MINNESOTA

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

MISSISSIPPI

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

MISSOURI

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

NEBRASKA

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

NEVADA

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

NEW HAMPSHIRE

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

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

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

NEW JERSEY

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

NEW MEXICO

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

NEW YORK

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

NORTH CAROLINA

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

OHIO

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

OKLAHOMA

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

OREGON

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

PENNSYLVANIA

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

PUERTO RICO

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

RHODE ISLAND

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

SOUTH CAROLINA

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

TENNESSEE

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

TEXAS

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

UTAH

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

VIRGINIA

ALEXANDRIA 3425 King St. at Quaker Ln., (703) 870-2600
ARLINGTON Crystal City, 2301 So. Jefferson Davis Hwy., (703) 521-7479
FAIRFAX Westfair Center, 11027 Lee Hwy., (703) 273-6500
LYNCHBURG Hill's Plaza, Ward's Rd., (804) 237-1664
NEWPORT NEWS Newmarket South Shop. Ctr., (804) 245-8203
NORFOLK 5731 Poplar Hall Dr., (804) 461-0798
RICHMOND Willow Lawn S/C, 1617 Willow Lawn Dr., (804) 282-3453; 7728 Midlothian Turnpike, (804) 272-8803
ROANOKE Franklin Bldg., 3561 Franklin Rd. S.W., (703) 342-6335
ROSSLYN 1911 N. Ft. Myer Dr. at Rt. 29, (703) 243-5955

WASHINGTON

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

WEST VIRGINIA

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

WISCONSIN

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