TRS-80®
Microcomputer News

- **Home Issue**
- **Graphics for the LP-8**

Fort Worth Scene

May is home issue time. Some homes are run much like businesses and some businesses are homey mom and pop operations. A wide variety of computer related activities occur in our homes — word processing, game playing, data base management, accounting, education, and innumerable others.

I know a man who uses Profile III + to organize recipes. That way he can quickly find any recipe in the data base when he needs it, and he can print copies on request. A little food for thought: beneath the business-like exteriors of that program you use at the office may lurk a useful home application heretofore unimagined.

WHAT'LL THEY THINK OF NEXT?

While some people ponder endlessly what to do with a home computer, others have no trouble at all creating at once handy and unique applications programs. We discover much to our relief that we don't have to be expert programmers to write our own useful routines. Consider some of the programs submitted to us by readers and published in this month's issue.

Spellum is similar to a popular word game where each player is given six letters from which to make a word. The more letters used, the more points gained.

Checkbook Balancing let's you check the accuracy of your checkbook arithmetic. Of course you'll have to enter the figures in the check book.

Clock Tutor prints a clock face with random times and prompts the user to enter the correct hour and minutes. If the answer is incorrect, you will be asked to try again. What a fun way to teach a child to tell time!

There's even a program for transferring Model 100 .DO files to OS-9.

I had so much fun with the series of programs in Graphics for the LP-8. Mr. Smith has captured my imagination about what could be done (a letterhead or signature for instance) with a little time and effort using his ideas.

There's a color pie chart for the Tandy 2000 that looks spectacular. Those colors! That resolution! That keyboard! Oh gosh! I'm gushing again. You'll have to excuse me, a Model 2000 just arrived recently in my office.

MAGAZINES

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

*Advanced Computing Magazine**
131 East Orange Street
Lancaster, PA 17602
(717)394-3364

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

Color Micro Journal
5900 Cassandra Smith Rd.
Hixson, TN 37343

Computer User
16704 Marquardt Ave.
Cerritos, CA 90701

80 Micro
P.O. Box 981
Farmingdale, NY 11737

Hot CoCo
P.O. Box 975
Farmingdale, NY 11737

Portable 100 — The Magazine for Model 100 Users
67 Elm Street
P.O. Box 250
Camden, ME 04843
(207)236-4365

PCM — The Portable Computing Magazine
9529 U.S. Highway 42
P.O. Box 209
Prospect, KY 40059

Rainbow (Covers the TRS-80 Color Computer)
P.O. Box 209
Prospect KY 40059
(502)228-4492

*Formerly *two/sixteen* magazine

Front cover photo by Steve Hamblin, Radio Shack Photography

TRS-80®
Microcomputer News

Contents:

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

Graphics for the LP-8*

Paul W. Smith
789 St. Charles Dr. #5
Thousand Oaks, CA 91360

Editor's note: The programs below take advantage of the graphics mode of the LP-8 to produce nice graphic printouts and provide helpful information for producing detailed graphics. Programs three, four, and five can be used as tools in creating your own graphic designs. Design suggestions include a customized letter head or a signature written in script. Mr. Smith provided us with programs one, two, three, four, and five. Other programs and text in this article came from our editorial staff.

*These programs were tested on the DMP-200 in addition to the LP-8. Except for the Color Computer the programs should run on most TRS-80 Computers. To run the programs on the Color Computer, change all LPRINTS to PRINT#-2,.

PROGRAM 1—A DANDY FORM

The five line form in Figure 1 was printed with this program. It is possible to select and print the form with up up to forty-five lines.

Figure 1.

```
100 CLS
110 CLEAR 1000
120 PRINT @0,STRING$(23,"*");
      : PRINT" LINE PRINTER 8 ";
      : PRINT STRING$(24,"*")
125 PRINT @896,STRING$(63,"*")
130 PRINT @128,"1. THERE ARE 5 LINES PER INCH"
140 PRINT @256,"2. THE PAGE IS 11 INCHES LONG"
150 PRINT @384,"3. YOU HAVE A LIMIT OF 45 LINES."
160 PRINT @512,"4. HOW MANY LINES DO YOU WANT";
      : INPUT E
170 PRINT @640,"5. HOW MANY BLANK LINES BEFORE
      PRINTING";
      : INPUT T
175 PRINT @768,"6. PUSH RESTART,RESET ON PRINTER,
      THEN <ENTER>";
      : INPUT H
180 FOR R=1 TO T
      : LPRINT
      : NEXT R
1005 LPRINT CHR$(18);
1010 FOR Z=1 TO E
1020 GOSUB 2000
1030 NEXT Z
1040 LPRINT CHR$(30)
1050 GOTO 100
1500 '
2000 ' PRINTS TOP LINE
2020 LPRINT CHR$(255);
```

```
2030    FOR A=1 TO 118
      : LPRINT CHR$(129);
      : NEXT A
2040    LPRINT CHR$(255);
2050      FOR A=1 TO 12
2060      FOR B=1 TO 29
      : LPRINT CHR$(129);
      : NEXT B
2070      LPRINT CHR$(255);
2080      NEXT A
2090 LPRINT
3000 LPRINT CHR$(255);
3010    FOR A=1 TO 118
      : LPRINT CHR$(192);
      : NEXT A
3020    LPRINT CHR$(255);
3030      FOR A=1 TO 12
3040      FOR B=1 TO 29: LPRINT CHR$(192);
      : NEXT B
3050      LPRINT CHR$(255);
3060      NEXT A
3070 LPRINT
3080 RETURN
```

PROGRAM 2—NUMERIC VARIABLES

Program 2 creates a table of all possible numeric variables. Can you find the five BASIC keywords in the table that are not legal numeric variables? Check the appendix of your computer owner's manual if you're not already sure which ones they are. When you find the five BASIC keywords, cross them out of the table.

```
1000 ' ........... ALL POSSIBLE
      VARIABLES..............
1010 ' ................................................
1020 ' ............ LINE PRINTER 8.................
1030 ' ............MODEL 3 COMPUTER................
1040 ' ................................................
1050 ' .............BY  PAUL W. SMITH........04/11/82
1060 ' ............789 ST. CHARLES DR. # 5..........
1070 ' .............THOUSAND OAKS, CA. 91360.........
1080 ' .............HOME PHONE (805) 496-2223........
1090 ' ................................................
2000 LPRINT CHR$(27);CHR$(20) ' START CONDENSED
2005 ' ................................................
2010 E=12 ' LEFT TAB MARGIN FOR VARIABLES
2015 ' ................................................
2020 A=65 ' (ASCII A)
2025 LPRINT TAB(E);
2030      FOR B=0 TO 25 ' (A TO B TO C ETC.)
2035      LPRINT TAB(E);
2040      LPRINT CHR$(A+B);" ";
```

```
2050          FOR C=65 TO 90 ' (ASCII A THRU Z)
2055          LPRINT CHR$(A+B);
2060          LPRINT CHR$(C);" ";
2070          NEXT C
2072            FOR D=48 TO 57 ' (ASCII 0 TO 9)
2073                LPRINT CHR$(A+B);CHR$(D);" ";
2074            NEXT D
2075      LPRINT
2080      NEXT B
2100 ' ................................................
2110 LPRINT CHR$(27);CHR$(19); ' END CONDENSED
3000 ' ................................................
```

PROGRAM THREE—GRAPHIC CHR$ VALUE CHART

This program uses the LP-8's graphic mode to print the chart shown in Figure 2. The decimal numbers in the chart indicate the CHR$ value used to print the dot configuration by the number. The chart shows the dots in the vertical configuration as they print. LPRINT CHR$(255) in the graphics mode of the LP-8 will cause the printer to print a solid vertical bar, seven dots high. LPRINT CHR$(128); CHR$(128); CHR$(201); will space over two dot columns (128 indicates that no dots are printed.) and then print the first, fourth, and bottom dots in the third column. This chart is a handy reference for creating graphics on the LP-8. Used in conjunction with the worksheets of Programs Four and Five, it is easy to create very detailed graphics.

```
10    LPRINT CHR$(18);  '   START GRAPHIC MODE
30    GOTO 17000
100   LPRINT CHR$(156); CHR$(190); CHR$(255);
      CHR$(255); 'SOLID
101   LPRINT CHR$(255); CHR$(190); CHR$(156);
      CHR$(128);
102   RETURN
200   LPRINT CHR$(156); CHR$(162); CHR$(193);
```

```
      CHR$(193); 'BLANK
201   LPRINT CHR$(193); CHR$(162); CHR$(156);
      CHR$(128);
202   RETURN
400   LPRINT
401   RETURN
500   LPRINT CHR$(128); CHR$(128); CHR$(128);
      CHR$(190); CHR$(128);
501   LPRINT CHR$(128); CHR$(128); CHR$(128);
502   RETURN
1000  LPRINT CHR$(136); CHR$(152); CHR$(136);
      CHR$(136);
1010  LPRINT CHR$(136); CHR$(136); CHR$(156);
      CHR$(128);
1020  RETURN
2000  LPRINT CHR$(188); CHR$(194); CHR$(130);
      CHR$(188);
2010  LPRINT CHR$(192); CHR$(192); CHR$(254);
      CHR$(128);
2020  RETURN
3000  LPRINT CHR$(254); CHR$(132); CHR$(136);
      CHR$(156);
3010  LPRINT CHR$(130); CHR$(194); CHR$(188);
      CHR$(128);
3020  RETURN
4000  LPRINT CHR$(132); CHR$(140); CHR$(148);
      CHR$(164);
4010  LPRINT CHR$(254); CHR$(132); CHR$(132);
      CHR$(128);
4020  RETURN
5000  LPRINT CHR$(254); CHR$(192); CHR$(252);
      CHR$(130);
5010  LPRINT CHR$(130); CHR$(194); CHR$(188);
      CHR$(128);
5020  RETURN
6000  LPRINT CHR$(132); CHR$(136); CHR$(144);
      CHR$(188);
6010  LPRINT CHR$(194); CHR$(194); CHR$(188);
      CHR$(128);
6020  RETURN
```


Figure 2.

```
7000 LPRINT CHR$(254); CHR$(130); CHR$(132);
     CHR$(136);
7010 LPRINT CHR$(144); CHR$(144); CHR$(144);
     CHR$(128);
7020 RETURN
8000 LPRINT CHR$(188); CHR$(194); CHR$(194);
     CHR$(188);
8010 LPRINT CHR$(194); CHR$(194); CHR$(188);
     CHR$(128);
8020 RETURN
9000 LPRINT CHR$(188); CHR$(194); CHR$(194);
     CHR$(188);
9010 LPRINT CHR$(136); CHR$(144); CHR$(160);
     CHR$(128);
9020 RETURN
10000 LPRINT CHR$(188); CHR$(198); CHR$(206);
     CHR$(218);
10010 LPRINT CHR$(242); CHR$(226); CHR$(188);
     CHR$(128);
10020 RETURN
15000 '
16000 '
17000 '
20000 GOSUB 8000
    : GOSUB 9000
    : GOSUB 10000
    : GOSUB 1000
20010 GOSUB 2000
    : GOSUB 3000
    : GOSUB 4000
    : GOSUB 5000
20020 GOSUB 6000
    : GOSUB 7000
    : GOSUB 8000
    : GOSUB 9000
20030 GOSUB 10000
    : GOSUB 1000
    : GOSUB 2000
    : GOSUB 3000
20040 GOSUB 4000
    : GOSUB 5000
    : GOSUB 6000
    : GOSUB 7000
20050 GOSUB 8000
    : GOSUB 9000
    : GOSUB 10000
    : GOSUB 1000
20060 GOSUB 2000
    : GOSUB 3000
    : GOSUB 4000
    : GOSUB 5000
20070 GOSUB 6000
    : GOSUB 7000
    : GOSUB 8000
    : GOSUB 9000
20080 GOSUB 10000
    : GOSUB 1000
    : GOSUB 2000
    : GOSUB 3000
20090 GOSUB 4000
    : GOSUB 5000
    : GOSUB 6000
    : GOSUB 7000
20100 GOSUB 8000
    : GOSUB 9000
    : GOSUB 10000
    : GOSUB 1000
20150 GOSUB 400 ' NEXT LINE
30000 GOSUB 2000
    : GOSUB 2000:
30010 GOSUB 3000
    : GOSUB 3000
    : GOSUB 3000
    : GOSUB 3000
    : GOSUB 3000

30020   GOSUB 3000
    : GOSUB 3000
    : GOSUB 3000
    : GOSUB 3000
    : GOSUB 3000
30030 GOSUB 4000
    : GOSUB 4000
    : GOSUB 4000
    : GOSUB 4000
    : GOSUB 4000
30040   GOSUB 4000
    : GOSUB 4000
    : GOSUB 4000
    : GOSUB 4000
    : GOSUB 4000
30050   GOSUB 5000
    : GOSUB 5000
    : GOSUB 5000
    : GOSUB 5000
    : GOSUB 5000
30060   GOSUB 5000
    : GOSUB 5000
    : GOSUB 5000
    : GOSUB 5000
    : GOSUB 5000
30070   GOSUB 6000
    : GOSUB 6000
    : GOSUB 6000
    : GOSUB 6000
30080 GOSUB 6000
    : GOSUB 6000
    : GOSUB 6000
    : GOSUB 6000
30090 GOSUB 7000
    : GOSUB 7000
30095 GOSUB 400 ' NEXT LINE
30100 FOR A=1 TO 44
30110 GOSUB 1000
30120 NEXT A
30121 GOSUB 400 ' NEXT LINE
30122 FOR A=1 TO 44
    : GOSUB 500
    : NEXT A
30123 '
40000 GOSUB 400 ' NEXT LINE
41000 FOR A=1 TO 22
41010 GOSUB 200
41020 GOSUB 100
41030 NEXT A
41040 '
41041 GOSUB 400 ' NEXT LINE
41050 FOR A=1 TO 11
41060 GOSUB 200
    : GOSUB 200
41070 GOSUB 100
    : GOSUB 100
41080 NEXT A
41090 '
41100 GOSUB 400 ' NEXT LINE
41110 FOR A=1 TO 5
41120 GOSUB 200
    : GOSUB 200
    : GOSUB 200
    : GOSUB 200
41130 GOSUB 100
    : GOSUB 100
    : GOSUB 100
    : GOSUB 100
41140 NEXT A
41150 GOSUB 200
    : GOSUB 200
    : GOSUB 200
    : GOSUB 200
41160 '
```

```
41165 GOSUB 400 ' NEXT LINE
41170 FOR A=1 TO 2
41180    FOR B=1 TO 8
41190       GOSUB 200
41200    NEXT B
41210    FOR B=1 TO 8
41220       GOSUB 100
41230    NEXT B
41240 NEXT A
41250 FOR A=1 TO 8
41260 GOSUB 200
41270 NEXT A
41280 FOR A=1TO 4
41290 GOSUB 100
41300 NEXT A
41310 '
41320 GOSUB 400 ' NEXT LINE
41330 FOR A=1 TO 16
41340 GOSUB 200
41350 NEXT A
41360 FOR A=1 TO 16
41370 GOSUB 100
41380 NEXT A
41390 FOR A=1 TO 12
41400 GOSUB 200
41410 NEXT A
41420 '
41430 GOSUB 400 ' NEXT LINE
41440 FOR A=1 TO 32
41450 GOSUB 200
41460 NEXT A
41470 FOR A=1 TO 12
41480 GOSUB 100
41490 NEXT A
41500 '
41510 GOSUB 400 ' NEXT LINE
41520 FOR A=1 TO 44
41530 GOSUB 200
41540 NEXT A
41550 '
41560 ' END OF FIRST SECTION ( 128 TO 171 )
41565 '
41570 FOR A=1 TO 8
     : LPRINT
     : NEXT A
42000 GOSUB 400 ' NEXT LINE
42010 GOSUB 2000
     : GOSUB 3000
     : GOSUB 4000
     : GOSUB 5000
42020 GOSUB 6000
     : GOSUB 7000
     : GOSUB 8000
     : GOSUB 9000
42030 GOSUB 10000
     : GOSUB 1000
     : GOSUB 2000
     : GOSUB 3000
42040 GOSUB 4000
     : GOSUB 5000
     : GOSUB 6000
     : GOSUB 7000
42050 GOSUB 8000
     : GOSUB 9000
     : GOSUB 10000
     : GOSUB 1000
42060 GOSUB 2000
     : GOSUB 3000
     : GOSUB 4000
     : GOSUB 5000
42070 GOSUB 6000
     : GOSUB 7000
     : GOSUB 8000
     : GOSUB 9000
42080 GOSUB 10000
     : GOSUB 1000

     : GOSUB 2000
     : GOSUB 3000
42090 GOSUB 4000
     : GOSUB 5000
     : GOSUB 6000
     : GOSUB 7000
42100 GOSUB 8000
     : GOSUB 9000
     : GOSUB 10000
     : GOSUB 1000
42110 GOSUB 2000
     : GOSUB 3000
     : GOSUB 4000
     : GOSUB 5000
42120 '
42130 GOSUB 400 ' NEXT LINE
42140 FOR A=1 TO 8
     : GOSUB 7000
     : NEXT A
42150 FOR A=1 TO 10
     : GOSUB 8000
     : NEXT A
42160 FOR A=1 TO 10
     : GOSUB 9000
     : NEXT A
42170 FOR A=1 TO 10
     : GOSUB 10000
     : NEXT A
42180 FOR A=1 TO 6
     : GOSUB 1000
     : NEXT A
42190 '
42200 GOSUB 400 ' NEXT LINE
42210 FOR A=1 TO 28
     : GOSUB 1000
     : NEXT A
42220 FOR A=1 TO 16
     : GOSUB 2000
     : NEXT A
42221 '
42222 GOSUB 400 ' NEXT LINE
42223 FOR A=1 TO 44
     : GOSUB 500
     : NEXT A
42230 '
42240 GOSUB 400 ' NEXT LINE
42250 FOR A=1 TO 22
     : GOSUB 200
     : GOSUB 100
     : NEXT A
42252 '
42254 GOSUB 400 ' NEXT LINE
42260 FOR A=1 TO 11
42270 GOSUB 200
     : GOSUB 200
     : GOSUB 100
     : GOSUB 100
42280 NEXT A
42290 '
42300 GOSUB 400 ' NEXT LINE
42310 FOR A=1 TO 4
     : GOSUB 100
     : NEXT A
42320 FOR A=1 TO 5
42330 GOSUB 200
     : GOSUB 200
     : GOSUB 200
     : GOSUB 200
42340 GOSUB 100
     : GOSUB 100
     : GOSUB 100
     : GOSUB 100
42350 NEXT A
42360 '
42370 GOSUB 400 ' NEXT LINE
```

```
42380 FOR A=1 TO 4
    : GOSUB 100
    : NEXT A
42390 FOR A=1 TO 2
42400    FOR B=1 TO 8
    : GOSUB 200
    : NEXT B
42410    FOR B=1 TO 8
    : GOSUB 100
    : NEXT B
42420 NEXT A
42430 FOR A=1 TO 8
    : GOSUB 200
    : NEXT A
42440 '
42450 GOSUB 400 ' NEXT LINE
42460 FOR A=1 TO 4
    : GOSUB 200
    : NEXT A
42470 FOR A=1 TO 16
    : GOSUB 100
    : NEXT A
42480 FOR A=1 TO 16
    : GOSUB 200
    : NEXT A
42490 FOR A=1 TO 8
    : GOSUB 100
    : NEXT A
42500 '
42510 GOSUB 400 ' NEXT LINE
42520 FOR A=1 TO 20
    : GOSUB 100
    : NEXT A
42530 FOR A=1 TO 24
    : GOSUB 200
    : NEXT A
42540 '
42550 GOSUB 400 ' NEXT LINE
42560 FOR A=1 TO 20
    : GOSUB 200
    : NEXT A
42570 FOR A=1 TO 24
    : GOSUB 100
    : NEXT A
42580 '
42590 ' END OF SECOND SECTION ( 172 TO 215 )
42600 '
42610 FOR A=1 TO 8
    : LPRINT
    : NEXT A
42620 GOSUB 400 ' NEXT LINE
42630 GOSUB 6000
    : GOSUB 7000
    : GOSUB 8000
    : GOSUB 9000
42640 GOSUB 10000
    : GOSUB 1000
    : GOSUB 2000
    : GOSUB 3000
42650 GOSUB 4000
    : GOSUB 5000
    : GOSUB 6000
    : GOSUB 7000
42660 GOSUB 8000
    : GOSUB 9000
    : GOSUB 10000
    : GOSUB 1000
42670 GOSUB 2000
    : GOSUB 3000
    : GOSUB 4000
    : GOSUB 5000
42680 GOSUB 6000
    : GOSUB 7000
    : GOSUB 8000
    : GOSUB 9000

42690 GOSUB 10000
    : GOSUB 1000
    : GOSUB 2000
    : GOSUB 3000
42700 GOSUB 4000
    : GOSUB 5000
    : GOSUB 6000
    : GOSUB 7000
42710 GOSUB 8000
    : GOSUB 9000
    : GOSUB 10000
    : GOSUB 1000
42720 GOSUB 2000
    : GOSUB 3000
    : GOSUB 4000
    : GOSUB 5000
42730 '
42740 GOSUB 400 ' NEXT LINE
42750 FOR A=1 TO 4
    : GOSUB 1000
    : NEXT A
42760 FOR A=1 TO 10
    : GOSUB 2000
    : NEXT A
42770 FOR A=1 TO 10
    : GOSUB 3000
    : NEXT A
42780 FOR A=1 TO 10
    : GOSUB 4000
    : NEXT A
42790 FOR A=1 TO 6
    : GOSUB 5000
    : NEXT A
42800 '
42810 GOSUB 400 ' NEXT LINE
42820 FOR A=1 TO 40
    : GOSUB 2000
    : NEXT A
42822 '
42823 GOSUB 400 ' NEXT LINE
42824 FOR A=1 TO 40
    : GOSUB 500
    : NEXT A
42826 '
42840 GOSUB 400 ' NEXT LINE
42850 FOR A=1 TO 20
    : GOSUB 200
    : GOSUB 100
    : NEXT A
42855 '
42856 GOSUB 400 ' NEXT LINE
42860 FOR A=1 TO 10
    : GOSUB 200
    : GOSUB 200
    : GOSUB 100
    : GOSUB 100
    : NEXT A
42862 '
42864 GOSUB 400 ' NEXT LINE
42870 FOR A=1 TO 5
    : GOSUB 200
    : GOSUB 200
    : GOSUB 200
    : GOSUB 200
    : GOSUB 100
    : GOSUB 100
    : GOSUB 100
    : GOSUB 100
    : NEXT A
42880 '
42890 GOSUB 400 ' NEXT LINE
42910 FOR A=1 TO 8
    : GOSUB 100
    : NEXT A
42920 FOR A=1 TO 8
    : GOSUB 200
```

```
        : NEXT A
42930 FOR A=1 TO 8
        : GOSUB 100
        : NEXT A
42940 FOR A=1 TO 8
        : GOSUB 200
        : NEXT A
42950 FOR A=1 TO 8
        : GOSUB 100
        : NEXT A
42960 '
42970 GOSUB 400 ' NEXT LINE
42980 FOR A=1 TO 8
        : GOSUB 100
        : NEXT A
42990 FOR A=1 TO 16
        : GOSUB 200
        : NEXT A
43000 FOR A=1 TO 16
        : GOSUB 100
        : NEXT A
43100 '
43200 GOSUB 400 ' NEXT LINE
43210 FOR A=1 TO 8
        : GOSUB 200
        : NEXT A
43220 FOR A=1 TO 32
        : GOSUB 100
        : NEXT A
43230 '
43240 GOSUB 400 ' NEXT LINE
43250 FOR A=1 TO 40
        : GOSUB 100
        : NEXT A
44000 LPRINT CHR$(30)
```
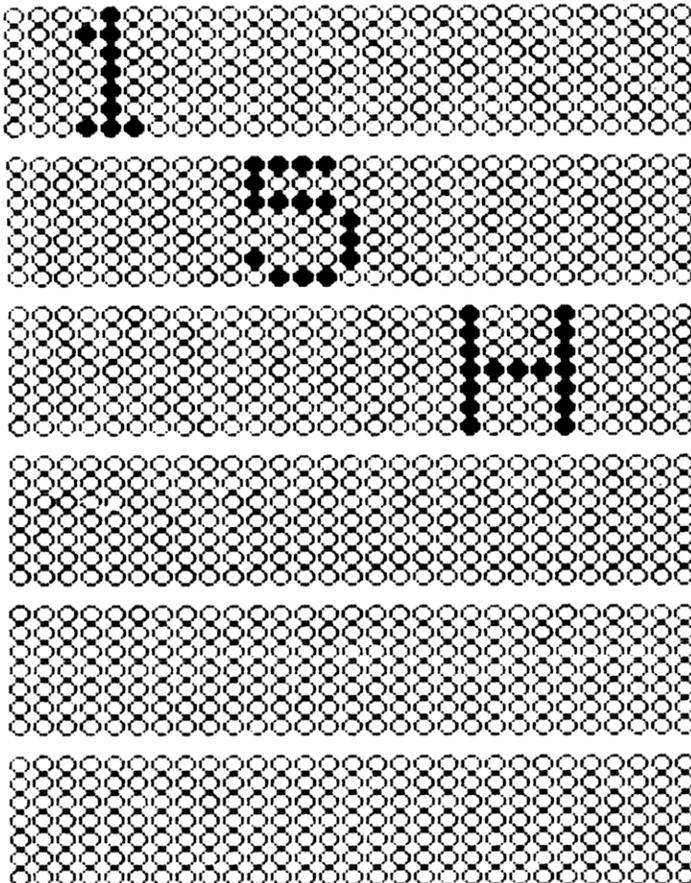

Figure 3.

PROGRAM 4—WORKSHEET 1

Except that none of the dots are filled in this worksheet consists of seven sections identical to the four shown in Figure 3. Worksheet 1 is useful for defining individual characters like the ones shown below (1, 5, and H). First comes the program to print the worksheet.

```
2 GOTO 1000
10    LPRINT CHR$(18);  '  START GRAPHIC
20 GOTO 300
200   LPRINT  CHR$(156); CHR$(162); CHR$(193);
      CHR$(193); 'BLANK
201   LPRINT CHR$(193); CHR$(162); CHR$(156);
      CHR$(128);
202 RETURN
300   FOR A=1 TO 7
310       FOR B=1 TO 7
320           FOR C=1 TO 42
330               GOSUB 200
340               NEXT C
345 LPRINT
350       NEXT B
355 LPRINT
360 NEXT A
400 END
1000 FOR V=1 TO 4
        : LPRINT
        : NEXT V
1117 LPRINT TAB(5);
1120 LPRINT CHR$(27); CHR$(20); ' START CONDENSED
1130 LPRINT CHR$(27); CHR$(14); ' START ELONGATED
1140 LPRINT "LINE PRINTER 8";TAB(50);"BASIC
      PROGRAMMING";
1150 LPRINT CHR$(27); CHR$(19); ' END CONDENSED
1160 LPRINT CHR$(27); CHR$(15)  ' END ELONGATED
1170 ' ......................................
1175 LPRINT TAB(5);
1180 FOR H=1 TO 70
1190     LPRINT CHR$(231); ' BLOCK GRAPHICS CODE 231
1200 NEXT H
1210 ' ......................................
1220 FOR I=1 TO 4
1230     LPRINT ' FOUR LINE SPACES AFTER TOP BLOCK
      LINE
1240 NEXT I
2000 GOTO 10
```

To provide a guideline for writing the program below, I darkened in the circles to make the numbers 1 and 5 and the letter H. Looking at the first row of dots I find that only the second and seventh dot should print. By referring to the chart in Figure 2 I see that to print only those two dots I need CHR$(194);. In the second column of the character one all the dots will be printed, so I need CHR$(255);, and for the third column of dots I need only the seventh dot to print so I use CHR$(192). Now I have my one. The other characters are printed by following a similar procedure. The following short program prints the three characters 1, 5, and H.

```
10 LPRINT CHR$(18);     ' SET GRAPHICS MODE
20 LPRINT CHR$(194); CHR$(255); CHR$(192) '1
30 LPRINT CHR$(167); CHR$(197); CHR$(197); CHR$(197);
      CHR$(184)                          ' 5
40 LPRINT CHR$(255); CHR$(136); CHR$(136); CHR$(136);
      CHR$(136); CHR$(255)               ' H
```

Using patience and perserverance it would be possible to create shapes (e.g. triangles and squares), special symbols (e.g. pi and infinity) and even special character sets.

PROGRAM FIVE—WORKSHEET TWO

Program five prints a second worksheet for defining larger graphic designs (Figure 4). The dots are printed with no spaces between rows. It's up to the user to remember to look at the dots in groups of seven when determining the dot configuration to be selected.

```
10    LPRINT CHR$(18);  '  START GRAPHIC MODE
20 GOTO 300
200   LPRINT  CHR$(156); CHR$(162); CHR$(193);
      CHR$(193); 'BLANK
201   LPRINT CHR$(193); CHR$(162); CHR$(156);
      CHR$(128);
202 RETURN
300   FOR A=1 TO 7
310       FOR B=1 TO 7
320           FOR C=1 TO 50
330               GOSUB 200
340           NEXT C
350       NEXT B
360 NEXT A
```

Figure 4 depicts worksheet two with a ship drawn in.

SHIP

Using the worksheet printed by program 5 and the chart in Figure 2, the program below was written to print the ship in Figure 5.

```
10 'FIRST PASS
20 LPRINT CHR$(18); 'ENTER GRAPHICS MODE
30 FOR N=1 TO 19
   : LPRINT CHR$(128);
   : NEXT N
40 FOR X=1 TO 2
:LPRINT CHR$(255);
50 FOR N=1 TO 8
   : LPRINT CHR$(129);
   : NEXT N
```

```
60 NEXT X
70 LPRINT CHR$(255)
80 'SECOND PASS
90 FOR N=1 TO 7
   : LPRINT CHR$(128);
   : NEXT N
100 FOR N=1 TO 2
    : LPRINT CHR$(252);
110 FOR X=1 TO 4
    : LPRINT CHR$(196);
    : NEXT X,N
120 LPRINT CHR$(252); CHR$(128); CHR$(135);
130 FOR N=1 TO 8
    : LPRINT CHR$(132);
    : NEXT N
140 LPRINT CHR$(255);
150 FOR N=1 TO 8
    : LPRINT CHR$(132);
    : NEXT N
:LPRINT CHR$(135)
160 'THIRD PASS
170 FOR N=1 TO 12
    : LPRINT CHR$(128);
    : NEXT N
    : LPRINT CHR$(255);
180 FOR N=1 TO 7
    : LPRINT CHR$(128);
    : NEXT N
    : LPRINT CHR$(191);
190 FOR N=1 TO 7
    : LPRINT CHR$(161);
    : NEXT N
    : LPRINT CHR$(255);
200 FOR N=1 TO 8
    : LPRINT CHR$(161);
    : NEXT N
    : LPRINT CHR$(191)
210 'FOURTH PASS
220 FOR N=1 TO 6
    : LPRINT CHR$(128);
    : NEXT N
```
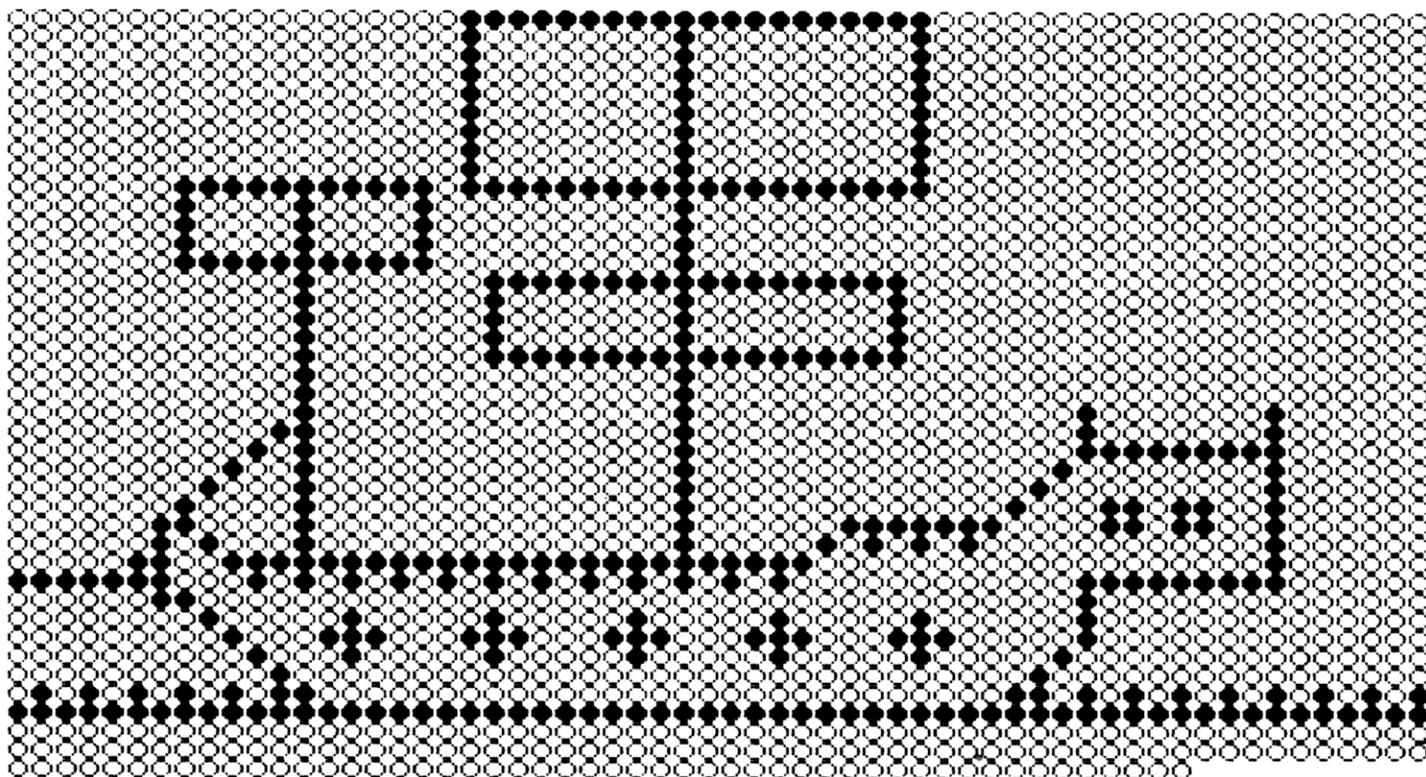

Figure 4.

```
230 LPRINT CHR$(192); CHR$(224); CHR$(144);
    CHR$(136); CHR$(132); CHR$(130); CHR$(255);
240 FOR N=1 TO 15:
LPRINT CHR$(128);
    : NEXT N
    : LPRINT CHR$(255);
250 FOR N=1 TO 6
    : LPRINT CHR$(128);
    : NEXT N
260 FOR N=1 TO 7
    : LPRINT CHR$(192);
    : NEXT N
270 LPRINT CHR$(192); CHR$(160); CHR$(144);
    CHR$(136); CHR$(135); CHR$(228); CHR$(228);
    CHR$(132); CHR$(228); CHR$(228); CHR$(132);
    CHR$(132); CHR$(255)
280 'FIFTH PASS
290 FOR N=1 TO 5
    : LPRINT CHR$(132);
    : NEXT N
300 LPRINT CHR$(134); CHR$(143); CHR$(136);
    CHR$(145); CHR$(162); CHR$(198); CHR$(130);
    CHR$(135); CHR$(162); CHR$(246); CHR$(162);
    CHR$(134);
310 LPRINT CHR$(130); CHR$(134); CHR$(162);
    CHR$(246); CHR$(162); CHR$(130); CHR$(134);
    CHR$(162); CHR$(246); CHR$(162); CHR$(135);
320 LPRINT CHR$(130); CHR$(134); CHR$(162);
    CHR$(246); CHR$(162); CHR$(129); CHR$(128);
    CHR$(129); CHR$(160);
330 LPRINT CHR$(241); CHR$(160); CHR$(129);
    CHR$(128); CHR$(128); CHR$(128); CHR$(192);
    CHR$(188);
340 FOR N=1 TO 8
    : LPRINT CHR$(132);
    : NEXT N
    : LPRINT CHR$(135)
350 'SIXTH PASS
360 LPRINT CHR$(132); CHR$(134); CHR$(132);
    CHR$(134); CHR$(132); CHR$(134); CHR$(132);
    CHR$(134); CHR$(132); CHR$(134); CHR$(132);
    CHR$(135); CHR$(134);
370 FOR N=1 TO 29
    : LPRINT CHR$(132);
    : NEXT N
380 LPRINT CHR$(134); CHR$(135); CHR$(132);
    CHR$(134); CHR$(132); CHR$(134); CHR$(132);
    CHR$(134); CHR$(132); CHR$(134); CHR$(132);
    CHR$(134); CHR$(132); CHR$(134); CHR$(132);
    CHR$(134); CHR$(132); CHR$(134); CHR$(132);
    CHR$(134); CHR$(132); CHR$(134);
390 LPRINT CHR$(30);
```
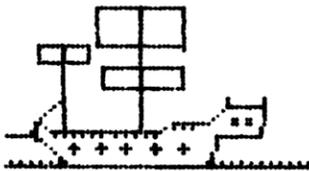

Figure 5.

Checkbook Balancing Program

Ted Magee
CompuServe ID 72315,343

In this checkbook program the instructions remain on the screen while the program is running. This program was tested on a Model II.

```
10 CLS
   : CLEAR 10000
   : DEFDBL B,C,D,N,J
20 PRINT
30 PRINT TAB(27);"CHECKBOOK"
40 PRINT
50 PRINT
60 PRINT"ENTER YOUR BALANCE"
70 PRINT
80 PRINT"ENTER CHECK #  AND AMT."
90 PRINT
100 PRINT"ENTER 0 IN CK.# WHEN YOU COME TO A
    DEPOSIT."
110 PRINT
120 PRINT"ENTER 0 IN DEPOSIT TO GO BACK TO CHECKS."
130 PRINT
140 PRINT"ENTER CREDIT CARD & SERVICE CHGS. AS CK.#
    55555,THEN ENTER AMT."
150 PRINT
160 PRINT"WHEN FINISHED--ENTER 1 IN CK.# TO GET
    CORRECT BALANCE."
170 PRINT
180 INPUT"ENTER YOUR CHECKBOOK BALANCE      ";BA
190 PRINT STRING$(32,32);CHR$(27);
200 PRINT USING"$$#,#######.##";BA
210 PRINT
220 N=N
230 INPUT"CHECK #     ";CN(N)
240 PRINT STRING$(11,32);CHR$(27);CN(N)
250 PRINT
260 IF CN(N)=0 THEN 420
270 IF CN(N)=1 THEN 530
280 INPUT"CHECK AMT.   ";CA(N)
290 PRINT STRING$(12,32);CHR$(27);
300 PRINT USING"$$#,#######.##";CA(N)
310 PRINT
320 BA=BA-CA(N)
330 PRINT"CK.BAL.     ";BA
340 PRINT STRING$(12,32);CHR$(27);
350 PRINT USING"$$#,#######.##";BA
360 PRINT
370 GOTO 220
380 PRINT
390 PRINT"ENTER A     0    WHEN FINISHED"
400 PRINT
410 M=M
420 INPUT"DEPOSIT AMT.";DT(M)
430 PRINT STRING$(12,32);CHR$(27);
440 PRINT USING"$$#,#######.##";DT(M)
450 BA=BA+DT(M)
460 PRINT
470 PRINT"CK.BAL.       $"BA
480 PRINT STRING$(12,32);CHR$(27);
490 PRINT USING"$$#,#######.##";BA
500 PRINT
510 IF DT(M)=0 THEN 230
520 GOTO 410
530 NB=BA-CB+DT
540 PRINT"NEW BALANCE     ";NB
550 PRINT STRING$(15,32);CHR$(27);
560 PRINT USING"$$#,#######.##";NB
570 PRINT
580 INPUT"CHECKBOOK BAL.     ";CK
590 PRINT STRING$(18,32);CHR$(27);
600 PRINT USING"$$#,#######.##";CK
610 PRINT
620 JM=NB-CK
630 PRINT"CORRECT BALANCE      ";CK
640 PRINT STRING$(20,32);CHR$(27);
650 PRINT USING"$$#,#######.##";JM
660 END
```

Orchestra-90's Compiler Format

by Bryan Eggers
Software Affair, Ltd.

The author of Orchestra-90, Jon Bokelman, created a powerful compiler within his music system. It accepts data in a format which, fortunately for me, is extremely forgiving! The relative spacing of symbols is not at all critical, and actual syntax errors stop the compiler with the cursor positioned at the problem location in the file. A couple of keystrokes fix the problem, followed by a Score (compile) and Play—all without leaving the master ORCH-90 program.

If you study various music files from our ORCH-90 SIG database on CompuServe (page HOM-13), you'll be surprised at the different formats accepted by the compiler. Each file represents a unique, yet correct, style of music symbol entry.

For example, some files have all the voices in a measure entered on one line:

```
M01 *V1Q1;2;3;4; V2Q2;3;4;5; V3Q3;4;5;6; V4Q4;5;6;7;
```

Other files contain one voice per line:

```
M01 *V1Q1;2;3;4;
V2Q2;3;4;5;
V3Q3;4;5;6;
V4Q4;5;6;7;
```

and some may even have voices out of sequence (within the measure):

```
M01 *V3Q3;4;5;6;
V4Q4;5;6;7;
V2Q2;3;4;5;
V1Q1;2;3;4;
```

All three measures will sound the same when compiled, demonstrating the ability of ORCH-90's compiler to accumulate and interpret the data correctly, regardless of the style of entry.

The compiler is exceptionally flexible in terms of spacing. In fact, there is only one rule governing the spacing of symbols in an ORCH-90 music file: the"measure string" must be terminated by a space. All other spaces are ignored during the compile.

I tend to waste spaces while transcribing, knowing I can eliminate them later with ORCHUTIL's Pack command. My typical measure format looks like this:

```
M01 *V1Q1;2;3;4;
V2Q2;3;4;5;
V3Q3;4;5;6;
    @V4Q4;5;6;7;
```

This helps me to visualize the relationship between voices and to spot obvious typos. Using one voice per line makes it easier to debug a measure, too. If you hear a "sour note" in the measure, you can use the / to "comment-out" various voices, making them silent:

```
M01 *V1Q1;2;3;4;
/   V2Q2;3;4;5;
/   V3Q3;4;5;6;
    @V4Q4;5;6;7;
```

Inserting a / turns the remainder of the line into a comment, ignored by the compiler. This time we forced Voices 2 and 3 to become silent, allowing only Voices 1 and 4 to play. You can eventually narrow down the "sour note" problem to a single voice. Use the ! command to replay the measure until you find and fix your transcription error. For example, make your change, position your cursor on the first line of the measure, press (BREAK) to return to Command mode, then enter an ! on the command line. This command does a "quick-compile" and automatic play from the cursor position!

Keep in mind, however, that each measure in this format contains an extra 14 or 15 spaces (3 or 4 spaces in front of Voices 2, 3 and 4). Not much, unless you're trying to enter a 200 measure classical piece on a 16K system. That's about 3K bytes wasted on spaces, or roughly 3 minutes of playing time.

Extra spaces are a luxury that simplify music entry, but contribute nothing to the sound of the compiled music file And yes, it does take a fraction of a second for the compiler to ignore each space, so eliminating them will reduce the compile time of the file as well as the size.

So, once satisfied with your new arrangement, you can use ORCHUTIL's Pack command to remove all the unnecessary spaces, along with the measure strings. A Packed file can still be edited, compiled and played normally. Packing of files is required for submission to the ORCH-90 SIG on CompuServe. No one wants to waste their connect time on a file loaded with thousands of unnecessary spaces!

MEASURE STRINGS

In the previous example, the measure string was M01. Any symbols after the M (measure symbol), in this case 01, are ignored by the compiler. The purpose of allowing a string of numbers or letters after the M symbol is to allow a reference back to your original sheet music. Most of us like to lightly pencil in reference numbers above each measure on the sheet music. This simplifies debugging of the piece later by using the string search function to locate the measure string. It also helps identify groups of measures which may be repeated with the R symbol instead of retyped.

To be valid, a measure string must begin with an M, be followed by optional letters or numbers and be terminated with a space. This is the only mandatory space in an ORCH-90 music file. Try not to get carried away creating monstrous measure strings like MEASURE1000, though They eat up memory fast, too.

OVERLAP

It's a good habit to be a "byte-miser" when you create your ORCH-90 files. Here's why: The text file created by the editor must allow enough room in memory for the compiled code created by the SCORE command. You can't save this compiled code (you never even see it), but ORCH-90's synthesizer actually plays the compiled code, not the text file. This gives you the powerful capability of entering or debugging music and listening to your changes almost instantaneously. No need to save the text file, compile it with a separate utility or play it with yet another utility.

If there isn't enough memory left to create the compiled code, ORCH-90 gives you the option of OVERLAP. The compiler needs more memory for its code, and answering Y to the "OVERLAP?" prompt will allow it to use the space currently occupied by your text file. In other words, your text file will be eliminated. Make sure to answer N to the "OVERLAP?" prompt if you haven't already saved your text file to tape or disk.

The OVERLAP feature allows you to create HUGE music files, perhaps playing for 45 minutes or longer in 48K of memory. OVERLAP occurs automatically when you retrieve files with the GET command.

CHANGING PARAMETERS

We've shown the flexibility of ORCH-90's compiler in terms of spaces, but where do we insert other symbols representing musical parameters like key signature, transposition, tempo, Clef, and registration? Although these symbols may be inserted just about anywhere in the music file, they take effect at specific boundaries. Let's examine them all.

NOTE: The following examples may contain some of the four articulation symbols ' " ; , which are present in almost all music files but have no effect on these parameters.

KEY SIGNATURE, TRANSPOSITION AND OPTION 1

The parameters K, <, >, U and O1 take effect at the beginning of the measure in which they are inserted, no matter where in the measure they appear.

For example, consider the following measure:

```
M25 *V1Q7;6;5;4;
    V2Q2;1%;0;-1;
    @V3H3,7,
    V4HA,E,
```

Let's add the following parameter changes to the /@ measure:

K1& - Define key signature as "one flat".
>3 - Transpose the entire measure up 3 half-steps.
O1 - Option 1. Accidentals now affect only the voice in which they appear.
V4U7 - Transpose Voice 4 down seven steps (one octave).

We can insert these parameters anywhere in the measure and the compiler will always interpret them correctly. Here are a few examples:

```
M25 *V1Q7;6;5;4;   K1& >3
    V2Q2;1%;0;-1;  O1 V4U7
    @V3H3,7,
    V4HA,E,

M25 *V1Q7;6;5;4;
    V2Q2;1%;0;-1;
```

```
    @V3H3,7,
    V4HA,E,
K1& >3 O1 V4U7

M25 *V1Q7;6;5;4; >3
    K1&  V2Q2;1%;0;-1;
O1  @V3H3,7,
V4U7 V4HA,E,


M25 *V1Q7;6;5;4;
K1& >3 O1 V4U7
    V2Q2;1%;0;-1;
    @V3H3,7,
    V4HA,E,

M25 K1& >3 O1 V4U7
    *V1Q7;6;5;4;
    V2Q2;1%;0;-1;
    @V3H3,7,
    V4HA,E,
```

OK, we know the system is flexible and the compiler accepts any of these formats so which one should we use? The last example is probably the easiest to read and debug later, but it's really a matter of personal preference. The compiler doesn't care. Quite a change from that picky BASIC compiler, isn't it?

By the way, you can reduce the last example to:

```
M25 K1& >3 O1
    *Q7;6;5;4;
    V2Q2;1%;0;-1;
    @V3H3,7,
    V4U7HA,E,
```

The first notes in a measure always belong to V1, unless otherwise specified, so V1 was eliminated.

We also take advantage of the V4 already existing in the file by adding U7 to it. When the compiler encounters the V4, all notes or transposition parameters that follow are assigned to Voice 4. Inserting the U7 after the V4 accomplished the same effect as inserting a separate V4U7; the voice plays an octave lower than the note symbols would normally indicate.

These parameter changes take effect at the beginning of the measure and remain in effect until changed. For example, Voice 4 will continue to be compiled an octave lower than the note symbols indicate throughout the remainder of the music file, or until the Voice 4 transposition is changed in another measure. Some common voice transpositions are:

V4U0 Resets the voice to normal (no transposition).
V4UE Transposes the voice down TWO octaves.
V4U + 7 Transposes the voice UP one octave.

ACCIDENTALS AND OPTION 2

Accidentals take effect immediately and remain in effect until the end of the measure. The affected notes are reset to normal at the end of the measure. The accidental #, & or % (sharp, flat or natural) must be inserted immediately after the note pitch symbol (obtained from the ORCH-90 music scale chart):

```
M10 *Q1#,3%,5&,7%,
```

Some music requires the accidentals to be carried over into the following measure. To do this, insert the Option 2 parameter symbol O2 anywhere in the measure:

```
M10 *Q1#,3%,5&,7%, O2
M11 *Q1,3,5,7,
```

The notes in M11 will be compiled with the accidentals from M10.

NOTE DURATION

Take another look at measure 11 in the previous example. It could have been written as:

```
M11 *Q1,Q3,Q5,Q7,
```

But, this is quicker and requires less memory:

```
M11 *Q1,3,5,7,
```

When the compiler encounters the Q, it compiles every note (or $ rest) that follows as having a Quarter note duration, until specified otherwise. The note duration parameter can only be reset by a different note duration symbol. Therefore, all notes in the following measure will be compiled as quarter notes (spaces for clarity):

```
M20 *Q 8,A,9,8,
    V2 6,8,7,6,
    V3 1,3,2,1,
    @V4 4,B,8,F,
```

The Q duration symbol sets a "switch" in the compiler and it stubbornly compiles to that switch setting until told otherwise, regardless of any other parameter symbols it encounters. The duration switch can only be reset by one of the other duration parameters, W, H, I, S, T, or X, alone or combined with their modifiers "." (for dotted notes) or ":" (for triplets).

CLEF

The Clef parameters, *, @ and _ (underline) are handled like the duration symbols. No other symbols affect the Clef switch. For example:

```
M30 *Q7'7'7'7'
    V2Q5'5'4'4'
    V3Q3'3'2'2'
    V4Q0'0'0'0'
```

The Clef switch is set to * (treble clef), so all notes in all four voices are compiled as treble clef notes.

A new measure does not automatically reset the clef switch to *. In the following two measures, all notes will be compiled in the @ (bass) clef:

```
M40 @Q7'7'7'7'
    V2Q5'5'4'4'
    V3Q3'3'2'2'
    V4Q0'0'0'0'
M41  Q7'7'7'7'
    V2Q5'5'4'4'
    V3Q3'3'2'2'
    V4Q0'0'0'0'
```

The Clef parameter takes effect immediately, and can be inserted anywhere:

```
M50 *Q7'7'@7'7'
    V2*Q5'5'@4'4'
    V3*Q3'3'2'2'
    @V4Q0'*0'@0'*0'
```

TEMPO AND REGISTRATION

A tempo parameter, like NQ = 80, takes effect at the current Part number. If you need a different tempo at every measure, you'll need to insert a different Part number before each measure string.

```
P10 NQ=80
M25 *V1Q7;6;5;4;
    V2Q2;1%;0;-1;
    @V3H3,7,
    V4HA,E,
P00 NQ=90
M26 *V1Q5,$Q.7,S7#
    V2Q3,$Q.5,S$
    V3Q0,$Q.0,S0#
    @V4QB,$HB
```

You could also use P00 which is kind of a "wild card" Part number. It can be inserted between measures and will never conflict with any other Part number or any other P00. It can also be used to terminate a group of measures for possible repeat. For example, we could repeat the measure in P10 (measure 25) in the previous example by inserting an R10 elsewhere in the file, but we cannot repeat measure 26 because P00 is considered an "empty" Part number. The P00 should only be used if a repeat of its associated measures are not required.

The tempo parameters are inserted on the same line as the Part number. I'd suggest this format for clarity. The tempo parameter could be inserted anywhere between P10 and P00, but it will take effect at the beginning of P10.

Voice/register assignments (registration) also take effect at Part number boundaries:

```
P10 V1YA V2YA V3YA V4YA
M25 *V1Q7;6;5;4;
    V2Q2;1%;0;-1;
    @V3H3,7,
    V4HA,E,
P00 V1YD V2YD V3YC V4YC
M26 *V1Q5,$Q.7,S7#
    V2Q3,$Q.5,S$
    V3Q0$Q.0,S0#
        @V4QB,$HB
```

All four voices in measure 25 will play using the A register sound (trumpet). In measure 26, Voices 1 and 2 switch to the D register (organ) and Voices 3 and 4 switch to the C register (clarinet).

STEREO MAPPING

With two stereo channels, and up to five voices to direct to these channels, there are several possible voice-to-channel combinations. The Z command is used to select the desired stereo mapping, according to the chart in the ORCH-90 manual. For example, Z5 positions Voices 3 and 4 to channel A and Voices 1, 2 and 5 to channel B.

The mapping should always be specified or changed at Part Number boundaries. This is critical, since the compiler must not only switch the Voices to new positions, but must also switch their current instrument registers along with them.

Add the Z command to the Part Number line, along with any other parameter changes. The compiler will accept all the parameter changes, regardless of their sequence.

Stereo mapping can be changed any number of times during the piece, as long it is specified at Part Number boundaries:

```
P10 Z5 NQ=90
M25 *V1Q7;6;5;4;
    V2Q2;1%;0;-1;
    @V3H3,7,
    V4HA,E,
P15 V1YC V2YE Z2 NQ=80 V4U7
```

```
M26 *V1Q5,$Q.7,S7#
    V2Q3,$Q.5,S$
    V3Q0,$Q.0,S0#
    @V4QB,$HB
```

REGISTER DEFINITION

This feature changes the timbre of a default instrument register. It allows the strengths of eight partial harmonics, as well as the loudness of the overall instrument, to be defined within the music file. Each of the five registers, A, B, C, D and E may be defined ONCE within the file. If a register is not defined, ORCH-90's internal default settings will be used.

A typical register definition for register A would be:

```
JASEFA5F000F
```

A register definition always takes effect at the beginning of the file, regardless of its position in the file. You'll find it convenient to insert these J commands near the top of the file, especially if you like to experiment with new sounds.

CROSSING THE BOUNDARIES

If you make a mistake (or simply can't resist the temptation to experiment with ORCH-90), you may end up trying to change the same parameter twice within a boundary where ORCH-90 will accept only a single change. This is not really a syntax error, and is not reported as such. It's important to know how the compiler interprets such ambiguous commands. For example:

```
P10 V1YA
M25 V1YB
   *V1Q7;6;5;4;
    V2Q2;1%;0;-1;
   @V3H3,7,
    V4HA,E,
M26 V1YC
   *V1Q5,$Q.7,S7#
    V2Q3,$Q.5,S$
    V3Q0,$Q.0,S0#
   @V4QB,$HB
P20
```

Instrument register assignments, like V1YA, are supposed to be made at Part Number boundaries. We have register A assigned to Voice 1 at Part 10, which is fine. But, we try to assign register B to Voice 1 within Measure 25, and in measure 26, we assign register C to Voice 1. We're trying to assign three different instruments to the same voice (V1) within the same Part number. Only V1YC will be accepted by the compiler.

Beginning at Part 10, the compiler started accepting all Part-related parameter assignments until it reached Part 20. The last assignment accepted for a Part-related parameter becomes the effective parameter used for the entire Part. As a result, Voice 1 will use the C register (clarinet) throughout Part 10 (measures 25 and 26) because the V1YC assignment was the last change accepted before the compiler moved on to Part 20.

It's so much easier to insert your parameter changes where they actually take effect, than to debug a file with commands inserted haphazardly.

Hope you're enjoying your Orchestra-90. Next month I'll explain how to UPLOAD your music files into the ORCH-90 SIG database.

Orchestra-90 is a trademark of Software Affair, Ltd.

Computer Clubs

Adirondack Color Computer Club
Box 365
Bolton Landing, NY 12814

Chaves County Color Computer Club
% Lee Mitchell, Secretary
1102 Melrose Dr.
Roswell, NM 88201

Color C. H. I. P. S.
1325 Crescent Lake Road
Ortonville, MI 48462

Cursor Computer Club
% J. R. Foss
3816 Saxon Drive
New Smyrna Beach, FL 32069

East Texas Color Computer Club
% Glen Carlton Ford, President
Route 2 Box 527
Gilmer, TX 75644

Erie TRS-80 User's Group
% Tom Kuklinski
320 Maryland Ave.
Erie, PA 16505
(814) 456-4786

Evansville CoCo Club
% Brian Broyles
Box 462
Poseyville, IN 47633

HUB/80 TRS Users Group
% Sam Wells
530 Buschman St.
Hattiesburg, MS 39401

Laporte County TRS-80 Users Club
% Joyhn Rothert
Box 504
Westville, IN 46391
(291) 785-2445

Milwaukee Area TRS-80 User's Group
P. O. Box 184
South Milwaukee, WI 53172-184

Topeka Computer Club
% Kevin Cronister
2224 Hope
Topeka, KS 66614

Vancouver Color Computer Club
333 Boyne Street
New Westminster, B. C. V3M 5J9

Westchester Bulletin Board Service
% Bill Graspo, Sysop
10 Davenport Ave.
New Rochelle, NY 10805

Anyone interested in forming a XENIX user's club and/or exchanging information may contact.
Bill Beatty
TeleVisual
1287 Wabash Ave.
Springfield, IL 62704   (217) 787-4855

Notes on Previous Issues

APRIL 1982
Merge Cassette Programs

James Haher
303 Chestnut Street
Oneonta, NY 13820

The Merge Cassette program was excellent, but it did not work properly on my Color Computer with Extended Color BASIC.

The following procedure should work.
1. Number the second program so it has higher numbers than the first program.
2. Turn the computer off and then on again.
3. Type PCLEAR 1 unless graphic pages are needed.
4. CLOAD the first program.
5. Find the contents of memory locations 25, 26, 27, and 28 by peeking them. (As in P = PEEK(25))
6. For example, assume you found 12, 1, 40, and 15 in that order when you PEEKed the addresses in 5.
7. Subtract 2 from the value in 28 (15 in this example) in your head.
8. POKE 25,40: POKE 26,13
9. CLOAD the second program.
10. POKE 25,12: POKE 26,1

Locations 25 and 26 point to the start of the BASIC program; locations 27 and 28 point to the end of the program. The starting point of the second program is made two locations before the end of the first program and then the old starting point is finally re-inserted. As Jorge Mir mentioned in the April 1982 issue, you still may have work on DATA and DIM statements.

JUNE 1982
Texture

Patrick Chen
143 Clinton St.
Brooklyn, NY 11201

The program by George Fraser is really great. Now we can have many colors and textures in the highest resolution mode of the CoCo. One drawback in the program is that painting can only start from the top of the area. This makes the use of the procedure difficult in programming. I have added 83 bytes to the USR routine to overcome this limitation. The modified routine is still not as smart as the PAINT command in the Extended Color BASIC which can reach every corner of an odd shaped area. Add these lines:

```
570 DATA 189, 179, 237, 174, 141, 254, 209, 52, 22,
    23
580 DATA 254, 217, 53, 22, 131, 0, 128, 43, 42, 52
590 DATA 6, 31, 80, 74, 31, 139, 0, 2, 31, 16
600 DATA 30, 137, 221, 0, 48, 141, 0, 33, 141, 22
610 DATA 134, 57, 167, 88, 141, 164, 134, 49, 167, 88
620 DATA 53, 6, 23, 254, 174, 48, 141, 0, 18, 141
630 DATA 1, 57, 198, 6, 51, 140, 179, 189, 165, 154
640 DATA 57, 224, 16, 140, 6, 0, 37, 32, 16, 140
650 DATA 29, 224, 34,
```

The execution address now changes to the beginning of this addition, that is, DEFUSR7 = S + 296. The amended

routine is still relocatable. You can combine this routine with the other "High Resolution Character Generator" by George Fraser in the same issue to get the full benefit.

NOVEMBER 1982
Descenders on the LP VII

Joel Robbins
Route 5 Box 450
Syracuse, IN 46567

Those Color Computer users who want descenders might try the following changes to Joe Diel's program for the Model III and the LP VII.
1. Delete (DEFINT:) in line 40
2. Change all of the LPRINT statements to PRINT #-2,
3. Add ⌈ENTER⌉ after every two screen lines to line 330
4. Change the first part of line 340 to LINE INPUT A$:

JANUARY 1983
Day of the Week or Monthly Calendar

George Quellhorst and Son
1006 Robinhood
Painseville, OH 44077

There is an error in line 430. It should be:

```
430 IF Y/4=INT(Y/4) AND M<=2 THEN T=T-1
```

As published the Calendar program is only good for the years 1909-1999. The following changes make the calendar good for the years 1801-1999, providing you input a negative number for years <1900. For example February 1850 would be input as 2, -50.

Change lines 110 and 340 to read:

```
110 L=A(M)
    : IF (M=2) AND ((INT(Y/4)) * 4 = Y) AND Y > 0
    THEN L=29
340 ON M GOTO 400, 370, 370, 401, 390, 360, 401, 380,
    350, 400, 370, 350
```

Delete line 430.
Add:

```
401 IF Y/4=INT(Y/4) AND M<=2 THEN T=T-1
402 IF Y <= 0 THEN  T=T+1
```

The calendar is now good for almost 200 years and shows the right day in a leap year.

MAY 1983
Radio Shack Dot-Addressable Printers

Mitchell M. Sackson
2 Elmdorf Drive
Scarsdale, NY 10583

I ran this program on a DMP-2100. If successive RUNs are issued for the original coding, the image becomes "split." To avoid this the Carriage Return code CHR$(13) should be issued at the beginning of the RUN as shown in line 20.

```
10 'GRAPHICS PRINT FOR THE DMP-2100
20 LPRINT CHR$(18);CHR$(13)
30 GOTO 60
40 SAVE"MS02/BAS"
50 END
60 READ N
    : IF N=999 THEN LPRINT CHR$(30);
    : END
70 IF N=888 THEN LPRINT
    : GOTO 60
80 IF N >=0 THEN LPRINT CHR$(N+128);
    : GOTO 60
```

```
 90 READ NN
    : LPRINT CHR$(28);CHR$(-N);CHR$(NN+128);
100 GOTO 60
110 DATA 124, -3, 2, -8, 113, -4, 98, 124, 120, 112,
    888
120 DATA 119, -3, 28, 60, 60, -4, 56, -3, 57, -3, 1,
    127, 127, 124, 888
130 DATA 1, 2, 6, 14, 12, -8, 28, -3, 30, 15, 15, 7,
    999
```

JUNE 1983
Communications Corner

Bruce Benson
Customer Service Representative
Radio Shack Computer Center
57 West Towne Mall
Madison, WI 53719

The Bit Allocations for the Registers and Latches chart at the top of page 22 should have the Data Bit D0 and D1 functions reversed for the Model III so that D0 is Receiver Input, UART Pin 20 DB-25 and D1 is Not Used. This information comes from the Technical Reference Manual (26-2109).

JULY 1983
Invasion

Fred Haiback
2420 Beachview
Columbia, Missouri 65201

I enjoyed the program Invasion published in the July issue. Here are some changes that I made to the program.

```
500 PRINT "DO YOU WANT TO PLAY AGAIN (Y/N)"
501 M$=INKEY$
502 IF M$="Y" THEN 110
503 IF M$="N" THEN END
504 GOTO 501
780 IF S=A1 THEN GOSUB 800
```

In addition to the above changes, the direction keys might be made more comprehensible and easy to reach by changing lines 380 and 390 to:

```
390 IF I$=CHR$(9) THEN GOSUB 560
400 IF I$=CHR$(8) THEN GOSUB 600
```

Jason Stollings
421 N. Ridge
Liberty, MO 64068

I enjoyed the Invasion game by Marty Faivre. I don't like having to press a key for every space I move because the INKEY$ function doesn't have a key repeat. The answer is to change lines 380-410. This change will change the controls in the game. Instead of pressing the > and < keys to move, you press the right and left arrow keys.

```
380 Z=PEEK(14400)
    : IF Z=0 THEN 420
390 IF Z=64 THEN GOSUB 560
400 IF Z=32 THEN GOSUB 600
410 IF Z=128 THEN GOSUB 630
```

These changes will allow you to hold down the keys instead of pressing them over and over. Because this tool is very handy in BASIC games, I have included a full list of keys that will work with it and the numbers they produce.

⟵	32
⟶	64
↑	8
↓	16
Spacebar	128
ENTER	1
CLEAR	2

Asteroids - Modified

R. Spencer Soanes
31 Hi Mount Dr.
Willowdale, Ont, M2K 1X3
Canada

A few simple changes will remove a couple of errors and increase the challenge. While Manuel in his legend (lines 490-510) promises a bonus when you reach 1000 points, line 190 prevents you from passing 800 (line 35). All that happens is that you may start another game by pressing ENTER without benefit of a prompt, and you start at 0 again.

My modification automatically restarts the game when 1000 points are reached (lines 35, 190, 630), retains your score, and adds to it 5 points per move instead of 1 (line 185), and you are back to 3 chances (line 620). The screen advises you of the free game (lines 600, 610).
Change:

```
 35 C=1000
    : RV=2
    : M=0
185 IF M=0 THEN S=S+1 ELSE S=S+5
190 IF M=0 THEN IF S => C THEN 600
```

Add:

```
600 CLS
    : PRINT" YOU HAVE REACHED THE PLATEAU"
610 PRINT
    : PRINT" HAVE A FREE GAME WITH EACH POINT
    WORTH 5 POINTS THIS TIME."
615 FOR N=1 TO 500
    : NEXT N
620 RV=2
    : B=RND(63)
    : P=0
630 M=1
    : GOTO 50
```

Bombdrop

Jeff Squyres
71 Fox Valley Lane
Glen Mills, PA 19342

I modified this program to save high scores on Model III disk. I added the following lines.

```
  0 CLEAR 2000
    : OPEN "I",1,"DROPHS"
    : INPUT #1, HS
    : CLOSE
176 IF SC=HS THEN 440
440 CLS
    : PRINT"YOUR SCORE IS";HS:PRINT"YOU MADE THE
    HIGH SCORE!!! DO YOU WANT YOUR SCORE SAVED";
    : INPUT SCA$
445 IF LEFT$(SCA$,1)="Y" THEN OPEN "O", 1,"DROPHS"
    : PRINT #1,HA
    : CLOSE
450 GOTO 180
```

I deleted line 60 and changed line 5 to:

```
5 GOSUB 395
    : GOSUB 430
```

Bombdrop and Subdestroyer

Henry S. Tan
6417 Simon Drive
Cincinnati, OH 45238

An error in line 280 of Bombdrop always occurred when I ran the program so I changed the line.

```
280 BL$=MID$(CT$(J),(SP+6),5)
```

The statement ALL SUBS DESTROYED (line 510) appeared on the screen for only a fraction of a second. To make this statement appear longer I included a time counter in line 510.

```
510 CLS
    : PRINT@231, "ALL SUBS DESTROYED!!"
    : FOR L=1 TO 330
    : NEXT L
```

AUGUST 1983
Notes on Previous Issues
Matthew Belmonte
CompuServe ID 73165,171

In the correction to Renumbering on the I and III, A = Y/255 should be changed to A = Y/256 not A = Y-256.

Supershopper for the PC-1
Gene Trottot
304 L Street
Sparks, Nevada 89431

At the end of line 90 it should read NEXT B instead of just NEXT. Also in 90 A(B) should be A$(B). Line 140 should read INPUT"ITEM TO ENTER";A$.

I enjoyed running the program and found it very helpful when shopping.

Communications Corner
R. E. Jansen
Ark-Net PBBS Sysop

The telephone number listed for the Ark-Net PBBS is my voice line. The correct telephone number for Ark-Net PBBS is:

(501) 372-0576

I wish to thank Tandy Corporation, Al and Dru Simon, and Ric Manning for the publicity. And Ma Bell would probably like to thank whoever got the phone number wrong.

NOVEMBER 1983
Plotting Ratios
Gene Scanlon
130 N. Clifton Ave.
Aldan, PA 19018

I have found only one small problem with the program. When more than one set of bar graphs is used (statement 370 . . . Q$ = "YES". . .), the greatest value (B) is not reset. Thus the ratio used in all subsequent graphs within the run of the program are based upon the greatest value found in any of the graphs.

The program can be corrected by changing this statement.

```
130 B=0
    : CLS
    : PRINT
    : PRINT "ENTER VALUES FOR EACH ITEM"
```

DECEMBER 1983
Prime Numbers
Doug Bogia and Ward E. Canfield
3210 N. W. Tamblyn Terr.
Topeka, KS 66617

I was intrigued by the method used in this program. However, I noticed there was a lot of repetition. For instance, once all the multiples of two are crossed out then that also covers all the multiples of four, six, eight, etc. Therefore, to make this program run at almost twice the speed (and with the same accuracy) I added this line:

```
205 IF A(I)=1 THEN GOTO 240
```

This takes care of the repetition by checking to see if the number is prime or not. If it is not prime then it is not used in further checks. If it is prime the program runs on as it normally would.

DECEMBER 1983
Extended Multiplication
Martin F. Combs
2989 Sundance Circle
Las Cruces, NM 88001

I noticed that the program appears to have trouble with decimals. For instance, 1.23456 * 1.23456 = 1.46912.0550183936. Note the presence of two decimal points in the answer.

Changing line 1320 and adding the other lines below seems to solve the problem. In effect the lines convert the inputs to whole numbers, multiply, and then resupply the appropriate decimal point.

```
1054 L=LEN(N1$)
    : FOR K=1 TO L
1055 IF MID$(N1$,K,1)="." THEN D1=L-K
1056 NEXT
    : IF D1 <> 0 THEN
      N1$=LEFT$(N1$,L-D1-1)+RIGHT$(N1$,D1)
1064 L=LEN(N2$)
    : FOR K= 1 TO L
1065 IF MID$(N2$,K,1)="." THEN D2=L-K
1066 NEXT
    : IF D2 <> 0 THEN
      N2$=LEFT$(N2$,L-D2-1)+RIGHT$(N2$,D2)
1320 PRINT
    : PRINT "ANSWER       ";
    : D=D1+D2
    : M$=M$+N$
    : GOSUB 1400
    : PRINT LEFT$(M$,LEN(M$)-D)+ "." +RIGHT$(M$,D)
1400 IF D>LEN(M$) THEN M$="0" + M$
    : GOTO 1400
1410 RETURN
```

JANUARY 1984
Compounding Interest
Carlton Atherton
1211 Springbrook Ave.
Moosic, PA 18507

Below is a faster version of this program.

```
10 "A" AREAD D
20 PRINT"S&L Compounding Interest"
30 I=0
    : IF P>5000 THEN 50
40 P=5000
50 I=I+P*(((.162/360+1)^D)-1)
60 P=P*((.162/360+1)^D)
70 USING "#####.##"
80 LPRINT INT (I*100+.5)/100
90 LPRINT INT (P*100+.5)/100
100 END
```

FEBRUARY 1984
Notes on Previous Issues

In the Sieve of Eratosthenes listing, page 5, there is an error. The end of line 20 should be . . . (INT(SQR(N))-K1)/K2)

Improving Home and Small Business Productivity

As one involved in ongoing support for home and business microcomputer systems, I am often asked by the lay public just what a small computer can do besides play games or educate their children. They may own a small computer that was purchased to allay their fears about their children's education. Many of them have experienced frustration while trying to write a checkbook register in BASIC or another beginner's language. They emerge from this experience with the conviction that simple tasks are still better done by hand. This myopic view is also shared by some small businessmen who feel their computer can only be used for accounting, so they still employ a typewriter for business correspondence. Fortunately for all involved, the microcomputer does provide a means by which an individual household or a small business can improve productivity.

PRODUCTIVITY SOFTWARE

The answer lies in the realm of Productivity Software. This is software that is very user-friendly and capable of increasing the productivity of the average individual, household, or business. Typically included in this category of software are Word Processing Software (such as Scripsit and MultiMate), Data Base Management Software (such as Profile or dBaseII), Spreadsheets (such as VisiCalc or Multiplan), and Graphics packages.


I'LL TRACK DOWN THAT PROBLEM!

A ranking executive for a large home computer manufacturer recently said "The PC (Personal Computer) is a solution looking for a problem." One of the problems solved by computers in the sixties and seventies was the manipulation and analysis of large amounts of data. This led to improved productivity nationwide. Now in the eighties, smaller businesses and individual households can benefit from the frontiers blazed by the larger companies. Large business has traditionally analyzed massive amounts of data to determine future corporate direction. Now, you, the small businessperson or individual can analyze your data to determine the direction of your business.

As an example, on a recent PBS show dealing with microcomputers, a sandwich store entrepreneur related her tale of woe nationwide. She had set off into the world of data processing without any guidance, and the experience had left a bad taste in her mouth. She engaged the services of a small business microcomputer consultant, and he taught her the fundamentals of VisiCalc, her first piece of productivity software. Soon she was able to construct a financial picture of her business and could better direct her energies toward profit making. She began creating models on her own to experiment with new sandwich concepts and what her resultant profit or loss would be, something she had never been able to conceptualize without her personal computer. The program she used, VisiCalc, was probably responsible for the sale of more microcomputers during the period of 1979—1983 than any other piece of software.

SPREADSHEETS
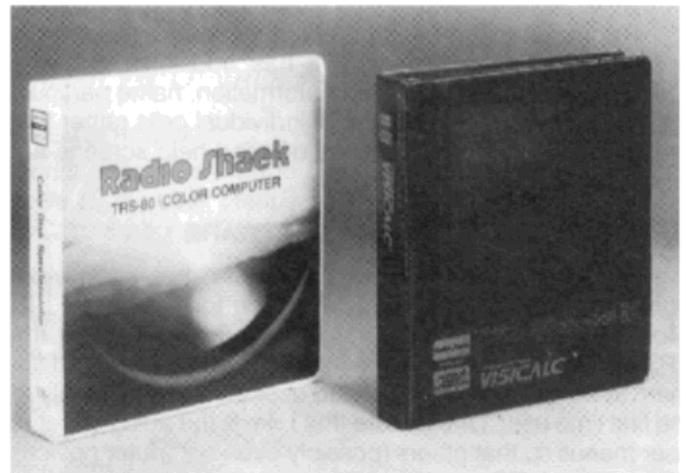
VisiCalc was the first electronic spreadsheet. The term spreadsheet has it's origins in accounting. It refers to large sheets of ruled paper used by accountants and auditors to assess the financial picture of a business. It was typically a neat, ordered series of balance sheets, income statements and ratios used to determine how successful or unsuccessful a company was. VisiCalc means Visible Calculator (Visi-Calc) and is itself a large "window" onto a larger spreadsheet that is 255 rows by 63 columns. On it you can display a sheet of

approximately 20 rows by 7 columns (depending on the format selected). Each part of the screen is called a cell and has an address specified by its row and column coordinates. You can type text or numbers on the screen. With simple to learn and remember commands, you can manipulate data or text and perform any number of math operations; the results will be displayed immediately. One of the best features of VisiCalc is that it lets you play "what if." If you make a change in a cell on the spreadsheet, the change is reflected in any other cells that are tied to that particular cell. For example, if your income increases and you are subtracting a fixed number of payments from it to get your discretionary income (the money you will have left after expenses), the figure reflecting discretionary income will increase as you increase your income. Color Computer and Model I/III owners may also use Spectaculator which is a simplified spreadsheet program.
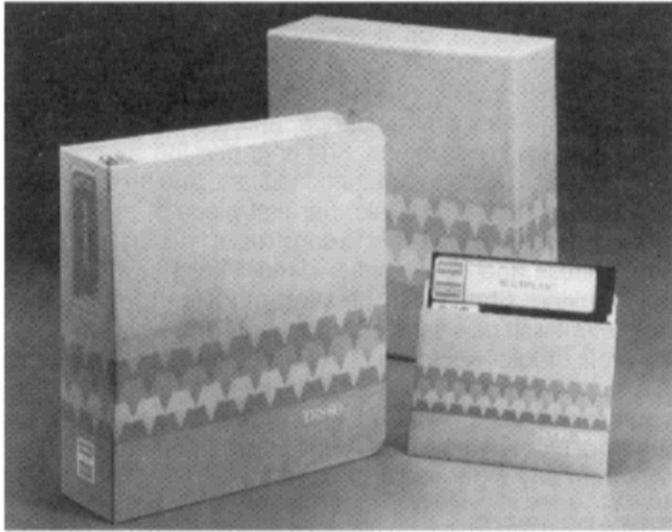
Radio Shack also markets a "second-generation" spreadsheet called Multiplan. It has additional enhancements which enable you to sort information, name particular cells, change the format (size) of individual cells rather than the entire sheet, and it also has detailed help screens and simplified commands.

DATA BASE MANAGEMENT SOFTWARE

Profile is also very easy to use and has a simplified command structure. Profile is available for the Model I/III, Model II/12, and in a new Multiuser version for the TRS-XENIX operating system. I have used all three and find them to be well documented and user friendly, especially for the first time user. One feature that I like is the ability to set up user menus so that others (possibly even computer novices) can use the data base without having to learn anything about it. If you are having trouble visualizing possible uses for a data base, look at this example. Let's say you own a small store. You could set up a customized data base to keep track of your inventory. With Profile (or any other data base) define the categories you will use, such as name, stock number, inventory amount, vendor, cost and retail price. Next set up entry screens, format reports, and possibly labels (with the vendors name and address). This example can be extended to the home. Imagine instead of an inventory you were keeping

track of a collection, anything from video cassettes and records to rare postage stamps and books.

Here are a couple of hints for first time data base designers:

1. Keep the amount of storage space used to a minimum. Don't make fields larger than they have to be.
2. When setting up a data base, give some thought to what information you will be sorting by. For example, if you have a data base consisting of record albums, you will probably want to sort by the album title, the artists name, the copyright date, and some sort of category you will assign such as Jazz. This information should always be placed in the key segment of a data base, or if there is no key segment, place it in the first fields. This will enable the information to be sorted at a faster rate with a resultant improvement in processing time for reports and catalog searches.
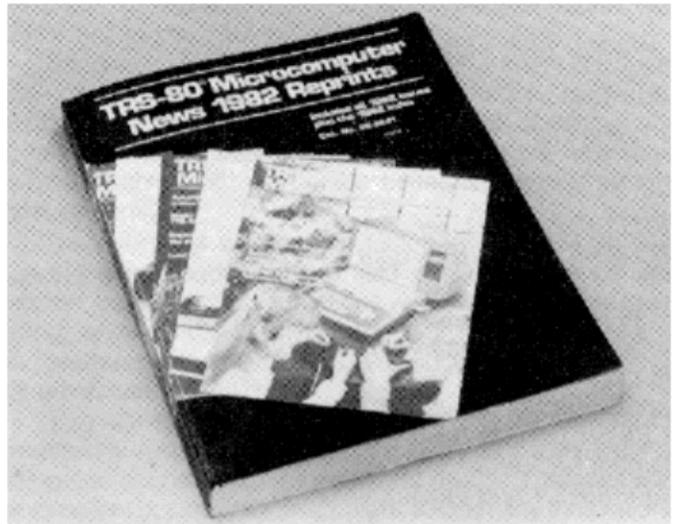

Other data base products for use on small business and home computers include PFS: File and PFS: Report, both available for the Model III/4 and the Tandy 2000 MS/DOS Computer. Color File and Color Profile are two data base managers available for the Color Computer.

While we are on the subject of data bases, there are several large national data bases that can be accessed by home computers using communications software. Some examples are CompuServe, Dow Jones, Tymenet and Videotex. They typically offer up-to-date news and weather as well as sports, movie information, games and general knowledge. One even has an on-line encyclopedia. One large national data base is AgriStar, a timesharing system for farmers and agricultural companies and cooperatives. It has very up-to-date weather information, and offers advice on selling crops, with up-to-the-minute forecasts of market prices.

WORD PROCESSING SOFTWARE

Last in the trilogy of productivity products is word processing software. SCRIPSIT and SuperSCRIPSIT are best known by users of TRSDOS machines. However, we now have MultiMate available for the Tandy 2000. Not enough can be said about the use of word processors in one article, let alone one devoted to the entire range of productivity soft-

ware. Suffice it to say that word processors have sounded the death knell for the manual typewriter. The last manual typewriter was manufactured sometime in July of 1983. A recent interview with a market analyst on PBS'S New Tech Times series cast gloom on home use of computers. The analyst felt most home computers were destined for the closet. The interviewer was inclined to agree with him, until the subject of word processors came up, which he defended to the death. I too will gladly defend my word processor to the doom sayers. I gleefully cast aside my supply of liquid paper the first day I touched the keyboard and haven't looked back since.

INTERACTIVE SOFTWARE

Instead of belaboring the obvious with descriptions of the function of a word processor, I would instead like to introduce the concept of interactive software. You can pass information between the various application programs named here. For example, you could first load information from a merge file created by your data base into your spreadsheet program. This could then be enhanced with new calculations within the spreadsheet. The resultant spreadsheet could generate a datafile that could be read into another database or into a word processor for the creation of a customized report. The sum becomes greater than the whole. Profile, SCRIPSIT, and VisiCalc already have these capabilities.

HELP IS AVAILABLE

Radio Shack Computer Customer Service has a dedicated group to help you with Productivity Software questions, and I have provided a list of recent Microcomputer News articles that will help you to use Profile, SCRIPSIT and VisiCalc to their fullest in improving your personal and business productivity.

JULY/AUGUST 1982

SuperSCRIPSIT and SCRIPSIT 2.0 (a comparison)
More About SCRISPIT 2.0
Merging Profile and SCRIPSIT
VisiCalc: Additional Uses of Lookup
Word Processing With the Keiths
More SCRIPSIT Ideas
Color SCRIPSIT and the LP VIII

SEPTEMBER 1982

Profile: Transferring Data from Records Into Shorter Records
VisiCalc: Personal Time Savers

OCTOBER 1982

Data Base Management Systems
Introducing Profile III +
SCRIPSIT Ideas

NOVEMBER 1982

Profile III +: Optimizing Storage Capacity
SCRIPSIT 2.0: LP VIII & the New Printers

DECEMBER 1982

Reprint Information (Page 34)
Profile III +: User Menus
VisiCalc: Personal Time Savers Revisited

JANUARY 1983

Profile III +: Designing Codes
VisiCalc: The Ides of April

FEBRUARY 1983

Profile II +: Math Appeal
Microsofts Multiplan

MARCH 1983

Profile III + and Your Printer

APRIL 1983

Profile: The Nature of Associated Fields
Data Bases and Communications for Starters
AGRISTAR: America's Agricultural Information and Computing Network

Computer Customer Service Address and Phone Numbers

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

Productivity/Special Applications	(817) 338-2390
Accounting Software	(817) 338-2391
O/S and Languages, Group No. 1	(817) 338-2392
O/S and Languages, Group No. 2	(817) 338-2393
Hardware and Communications	(817) 338-2394
Home Software	(817) 338-2395
Educational Software	(817) 338-2396
Newsletter Subscription Problems	(817) 870-0407

Bugs, Errors and Fixes

CORRECTION LETTERS FOR SOFTWARE

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

Model II, 12, 16

PROFILE PLUS (26-4515)

This letter contains modifications to correct a problem with printing double-line reports.
Letter dated January 18, 1984

PROFILE FORMS (26-4556)

This letter contains changes to correct underlining problems which occur with some printers.
Letter dated January 18, 1984

SCRIPSIT HARD DISK (26-4831)

Below are descriptions of four separate correction letters for Scripsit Hard Disk.

The first letter contains a set of corrections to enable Scripsit Hard Disk to print multiple copies when performing a Merge.
Letter dated January 17, 1984

The second letter contains a set of corrections to enable Scripsit Hard Disk to function properly when headers and footers are used in a Merge document.
Letter dated January 18, 1984

The third letter contains a set of corrections to enable Scripsit Hard Disk to function properly when hyphenation is selected from the repagination menu.
Letter dated January 19, 1984

The fourth letter contains corrections to enable proper line spacing recall after changing the default to double line spacing, creating a document, exiting, and returning to the same document.
Letter dated January 20, 1984

SCRIPSIT FOR THINLINE FLOPPY AND HARD DISK (26-4835)

Following are four separate sets of corrections for Scripsit for Thinline Floppy and Hard Disk.

The first letter contains changes to correct problems that can occur when more than thirty-one document files are acquired and when text blocks are lost during repagination with hyphenation.
Letter dated January 10, 1984

The second letter contains changes to fix a problem with Directory page numbering when using Scripsit Utilities.
Letter dated January 16, 1984

The third letter contains changes to eliminate the erratic results that occur when opening a document to store 84 lines

per page.

Letter dated January 17, 1984

The fourth letter contains changes to enable the Merge function to work with headers and footers and allow printing of multiple copies of a Merge.

Letter dated January 18, 1984

OTHER CORRECTIONS

Peripherals

DAISY WHEEL II MANUAL (26-1158)

The seventh line, second column under the "5 UNITS" heading of the "Proportional Character Set Units Per Column" chart on page 24 should read "e-189/BD" not "e-187/BB."

DWP 410 (26-1250)

The seventh line, second column under the 5 UNITS heading of the "Proportional Character Set Units Per Column" chart on page 30 should read "e-189/BD" not "e-187/BB."

BAR CODE READER (26-1183)

A bug in the 3 of 9 decoder which causes a . (period) to be decoded as an ! (exclamation mark) can be fixed by following the steps below.

1. From BASIC type LOADM"B3OF9.CO" (ENTER)
2. Type ? PEEK (62608) (ENTER). The value returned should be 33. This step is used to verify that the correct program is in memory before making any changes to it.
3. Type POKE 62608, 46 (ENTER). This changes the table to decode a period instead of an exclamation mark.
4. Type ? PEEK (62608) (ENTER) to verify that the value stored at 62608 is now a 46.
5. Type SAVEM"B3OF9",61788,62611,61824 (ENTER). This saves the revised program to memory.

   Note: A CSAVEM must be done to save the changed program on tape. If the corrected program is not saved to tape, then the procedure above must be repeated any time the 3 of 9 program is read from cassette.

National Conference on Technology for Disabled Persons

An invitation is extended for presentations and/or exhibits at Discovery '84: Technology for Disabled Persons which will be held October 1-3, 1984 and is sponsored by the University of Wisconsin-Stout.

Last year Discovery '83: Computers for the Disabled, had over 500 participants from 38 states plus Canada, 60 presenters, 3 nationally renowned keynote speakers, and 40 exhibitors. In addition, hundreds of people attended the exhibitor hall to see only the show.

Secretary T.H. Bell, United States Department of Education, commended the conference planners for providing a place for professionals from a variety of disciplines to gather to share their knowledge and explore new ways to help disabled citizens pursue better lives and live more independently.

Discovery '84 will be held October 1-3, 1984, in Chicago, Illinois. Leaders in the field will address a general session; concurrent presentations will discuss areas of technology and services for disabled persons, and software and hardware representatives will be exhibiting new technological advances.

This conference will provide teachers, administrators, counselors, researchers, health professionals, purveyors of computer technology, engineers, and disabled persons with an excellent opportunity to learn about the many innovations in technology that are impacting special education and rehabilitation programs today.

For further information write to:

Discovery '84: Technology for Disabled Persons
Office of Continuing Education
University of Wisconsin-Stout
Menomonie, WI 54751

Newspaper Carrier Billing Labels

T. W. Talcott
129 Dorsey Station Rd.
Louisville, KY 40223

Every month your faithful paper carrier has to prepare bills to mail out to all subscribers. This is a program that will make the job easier.

The program prepares a file of all subscribers which can be printed out. All changes during one month can be noted. Then the file can be edited and brought up to date before printing the billing labels.

When preparing to print labels, a 'check' is provided to be sure that the labels are lined up.

This program was written on a 32K Model III single disk system, and inexpensive dry gum labels (26-1556) were used for printing. These labels are placed on the inside of an envelope addressed to the carrier and delivered with the paper about the first of the month.

```
10 CLS
   : CLEAR 500
20 PRINT"THIS PROGRAM PREPARES NEWSPAPER DELIVERY
   MAILING LABELS"
30 PRINT"    1. TO PREPARE MAILING FILE"
40 GOTO 60
50 CLS
   : PRINT"      1. TO PREPARE MAILING FILE"
60 PRINT"    2. TO ENTER NAMES AND ADDRESSES"
70 PRINT"    3. TO EDIT NAMES AND ADDRESSES"
75 PRINT"    4. TO PAGE PRINT SUBSCRIBERS"
80 PRINT"    5. TO PRINT MAILING LABELS"
90 PRINT"    6. TO TERMINATE PROGRAM"
100 S=0
   : INPUT"MAKE YOUR SELECTION";S
```

```
110 ON S GOTO 200,600,2000,3000,800,120
120 PRINT"PROGRAM TERMINATED"
    : END
200 OPEN "R",1,"DATA/A"
210 FIELD 1,8 AS BF$,8 AS EF$,4 AS TF$
215 B$=""
    : INPUT"ENTER BEGINNING DATE (8 CHAR. MAX)";B$
220 E$=""
    : INPUT"ENTER ENDING DATE (8 CHAR. MAX)";E$
230 T=0
    : INPUT"NUMBER OF WEEKS";T
240 LSET BF$=B$
    : LSET EF$=E$
    : LSET TF$=MKS$(T)
250 PUT 1
260 CLOSE 1
270 GOTO 50
600 OPEN "R",2,"DATA/B"
610 FIELD 2,30 AS NF$,30 AS AF$,4 AS CF$
620 LET R=LOF(2)+1
630 CLS
    : PRINT"SUBSCRIBER NO. "R
640 N$=""
    : INPUT"ENTER SUBSCRIBER NAME (30 CHAR. MAX)";N$
650 A$=""
    : INPUT"ENTER STREET ADDRESS (30 CHAR. MAX)";A$
660 C=0
    : INPUT"ENTER WEEKLY RATE";C
670 IC$=""
    : INPUT"IS INFORMATION CORRECT (Y OR N)";IC$
680 IF IC$="Y" THEN 710
690 IF IC$="N" THEN 640
700 GOTO 670
710 LSET NF$=N$
    : LSET AF$=A$
    : LSET CF$=MKS$(C)
720 PUT 2,R
730 AS$=""
    : INPUT"ANOTHER SUBSCRIBER (Y OR N)";AS$
740 IF AS$="Y" THEN 770
750 IF AS$="N" THEN 780
760 GOTO 730
770 R=R+1
    : GOTO 630
780 CLOSE 2
    : GOTO 50
800 R$=""
    : INPUT"PREPARE PRINTER TO PRINT LABELS (R)";R$
810 IF R$="R" THEN 820 ELSE 800
820 TS$=""
    : INPUT"TEST";TS$
830 LPRINT"T"
    : LPRINT"T"
    : LPRINT"T"
    : LPRINT"T"
    : LPRINT
840 LU$=""
    : INPUT"IS LINEUP OK (Y OR N)";LU$
850 IF LU$="Y" THEN 900
860 IF LU$="N" THEN 820
870 GOTO 840
900 OPEN "R",1,"DATA/A"
910 FIELD 1,8 AS BF$,8 AS EF$,4 AS TF$
930 GET 1
935 T=CVS(TF$)
940 OPEN "R",2,"DATA/B"
950 FIELD 2,30 AS NF$,30 AS AF$,4 AS CF$
970 X=1
975 GET 2,X
977 C=CVS(CF$)
1000 N1$=NF$
    : A1$=AF$
    : C1=C
1003 CLS
    : PRINT"PRINTING NAME "X

1005 PRINT N1$
    : PRINT A1$
1010 X=X+1
1012 GET 2,X
1014 C=CVS(CF$)
1200 N2$=NF$
    : A2$=AF$
    : C2=C
1210 PRINT
    : PRINT"PRINTING NAME "X
1220 PRINT N2$
    : PRINT A2$
1300 LPRINT TAB(1)N1$ TAB(43)N2$
1310 LPRINT TAB(1)A1$ TAB(43)A2$
1320 LPRINT TAB(1)"FROM "BF$"TO " EF$ TAB(43)"FROM
     "BF$"TO "EF$
1330 LPRINT TAB(0) T "WEEKS @";
1340 LPRINT USING"$#.##";C1;
1345 LPRINT " PER WEEK";
1350 LPRINT TAB(42) T "WEEKS @";
1360 LPRINT USING"$$#.##";C2;
1365 LPRINT " PER WEEK"
1370 LPRINT TAB(1)"AMOUNT DUE -";
1380 LPRINT USING"$$##.##";T*C1;
1390 LPRINT TAB(43)"AMOUNT DUE -";
1400 LPRINT USING"$$##.##";T*C2
1410 LPRINT
1420 IF X=>LOF(2) THEN 1440 ELSE 1430
1430 X=X+1
    : GOTO 975
1440 PRINT "PRINTOUT COMPLETE"
    : CLOSE 1
    : CLOSE 2
    : FOR D=1 TO 300
    : NEXT D
    : GOTO 50
2000 CLS
2010 OPEN "R",2,"DATA/B"
2020 FIELD 2,30 AS NF$,30 AS AF$,4 AS CF$
2030 X=0
    : INPUT"NO. OF SUBSCRIBER TO BE CHANGED";X
2040 R=X
2050 GET 2,R
    : CLS
2051 C=CVS(CF$)
2053 GOSUB 2500
2055 PRINT"CHANGE TO BE MADE:"
2060 PRINT"    1. TO CHANGE NAME"
2070 PRINT"    2. TO CHANGE ADDRESS"
2080 PRINT"    3. TO CHANGE WEEKLY RATE"
2090 PRINT"    4. NO CHANGE"
2100 S=0
    : INPUT"MAKE YOUR SELECTION";S
2110 ON S GOTO 2200,2220,2230,2350
2200 N$=""
    : INPUT"ENTER NEW NAME";N$
2210 NF$=N$
    : GOTO 2300
2220 A$=""
    : INPUT"ENTER NEW ADDRESS";A$
2225 AF$=A$
    : GOTO 2300
2230 C=0
    : INPUT"ENTER NEW WEEKLY RATE";C
2240 CF$=MKS$(C)
    : GOTO 2300
2300 GOSUB 2500
    : AC$=""
    : INPUT"ANOTHER CHANGE (Y OR N)";AC$
2310 IF AC$="Y" THEN 2055
2320 IF AC$="N" THEN 2350
2330 GOTO 2300
2350 CLOSE 2
    : PRINT"CHANGE COMPLETED"
    : FOR D=1 TO 300
```

Phone Pad

Gerald Landry
25 Broadoaks Drive #204
Downsview, Ont.
M3J-1E3
Canada

Phone pad is a program written for the Model 4 that may be of interest to your readers. It is set up to be used with either roll or single sheet paper. It is menu-driven so instructions are not required. There are just enough graphics for an attractive appearance.

```
10  '                          ** PHONEPAD **
11  '
20  '     LINE NUMBERS REPRESENT:
21  '                100              INTRO
22  '                200              PRINTER READY ?
23  '                300              PRINT
24  '                400              MESSAGE FOR WHOM ?
25  '                500              END ??
26  '     SUB-ROUTINE LINE NUMBER:
27  '                1000             INTRO
28  '
29  '
30  '
100 '              ** INTRO **
105 GOSUB 1000
200 '              ** PRINTER READY ? **
205 CLS
    : PRINT@ (9,0),STRING$(80,131)
210 PRINT@ (11,22),"PRESS <ENTER> WHEN PRINTER IS
    READY"
215 PRINT@ (13,0),STRING$(80,131)
220 AN$=INKEY$
    : IF AN$=""THEN 220
225 CLS
    : PRINT@ (9,0),STRING$(80,95)
230 PRINT@(11,25),"LINE UP PAPER AT 2ND HOLE."
235 PRINT@ (12,0),STRING$(80,95)
240 PRINT@(6,25),"HOW MANY COPIES DO YOU WANT "
    : INPUT X
245 FOR W=1 TO X
250 CLS
    : PRINT@ (9,0),STRING$(80,131)
255 PRINT@ (11,20),"PRINTING ..... "W" OF "X" COPIES
    ...."
260 PRINT@ (13,0),STRING$(80,131)
300 '         ** PRINT **
305 LPRINT TAB(30) STRING$(21,"*")
310 LPRINT TAB(30) "*";TAB(50)"*"
315 LPRINT TAB(30)"*";TAB(33)"TELEPHONE
    CALLS";TAB(50)"*"
320 LPRINT TAB(30)"*";TAB(50)"*"
325 LPRINT TAB(30) STRING$(21,"*")
    : LPRINT
    : LPRINT
330 LPRINT
    : LPRINT TAB(10) STRING$(4,"*")
335 LPRINT TAB(10)"*";TAB(13)"*";TAB(16)"MESSAGE ON
    ANSWERING SERVICE ... OR"
340 LPRINT TAB(10) STRING$(4,"*")
345 LPRINT
    : LPRINT
400 '     ** FOR WHOM **
405 LPRINT TAB(10)"FOR WHOM";TAB(20) STRING$(45,".")
    : LPRINT
410 LPRINT TAB(10)"WHO CALLED";TAB(20)
    STRING$(45,".")
    : LPRINT
415 LPRINT TAB(10)"TIME WAS";TAB(20) STRING$(45,".")
    : LPRINT
420 LPRINT TAB(10)"PHONE #";TAB(20) STRING$(45,".")
    : LPRINT
425 LPRINT TAB(10)"MESSAGE";TAB(20) STRING$(45,".")
    : LPRINT
    : LPRINT
430 LPRINT TAB(10) STRING$(55,".")
435 LPRINT TAB(12)"Tear between lines .... use ruler
    or straight edge"
440 LPRINT TAB(10) STRING$(55,".")
445 FOR P=1 TO 6
    : LPRINT
    : NEXT
450 NEXT W
500 '         ** END OR RESTART **
505 IF W=X THEN 510
510 CLS
    : PRINT@ (9,0),STRING$(80,95)
515 PRINT@ (11,20),"FINISHED PRINTING .... ANY MORE
    ??? (Y/N)"
520 PRINT@ (12,0),STRING$(80,95)
525 AN$=INKEY$
    : IF AN$=""THEN 525
530 IF AN$="Y" THEN 200
535 PRINT@ (9,0),STRING$(80,131)
540 PRINT@ (11,20),"THEN I'LL PACK AND GO HOME ...
    BYE "
545 PRINT@ (13,0),STRING$(80,131)
550 END
1000 '      ** PROG INTRO **
1005 CLS
1010 PRINT@ (7,10),STRING$(60,127)
1015 PRINT@ (8,10),"G"
    : PRINT@ (8,69),"L"
1020 PRINT@ (9,10),"E"
    : PRINT@ (9,69),"A"
1025 PRINT@ (10,10),"R"
    : PRINT@ (10,69),"N"
1030 PRINT@ (11,10),"A"
    : PRINT@ (11,30),"PHONE PAD PRINT-OUT"
    : PRINT@ (11,69),"D"
1035 PRINT@ (12,10),"L"
    : PRINT@ (12,69),"R"
1040 PRINT@ (13,10),"D"
    : PRINT@ (13,69),"Y"
1045 PRINT@ (14,10),STRING$(15,127)
    : PRINT@ (14,55),STRING$(15,127)
1050 FOR X=1 TO 10
1055 PRINT@ (14,30), "BY:    G.G. LANDRY"
1060 FOR DLY=1 TO 400
    : NEXT DLY
1065 PRINT@ (14,36),STRING$(11,128)
1070 FOR DLY=1 TO 200
    : NEXT DLY
1075 NEXT X
1080 RETURN
```

```
*********************
*                   *
*   TELEPHONE CALLS  *
*                   *
*********************


****
*  *   MESSAGE ON ANSWERING SERVICE ... OR
****


FOR WHOM  ...........................................

WHO CALLED..........................................

TIME WAS  ...........................................

PHONE #   ...........................................

MESSAGE   ...........................................
```

Radio Shack's Color LOGO Lab

Radio Shack's Color LOGO Lab is a series of five lessons for classroom use, designed to teach elementary-school children how to use the educational computer language LOGO.

The course was developed by a former elementary-school reading teacher and was field-tested in classrooms and computer camps. Entertaining hands-on activities in Color LOGO Lab guide the student through the fundamentals of Color LOGO and help keep the student's attention focused on the lesson content. The lesson series begins with the most simple LOGO commands and progresses to more complex activities such as changing colors in LOGO and writing LOGO programs.

The first lesson introduces Color LOGO's Doodle mode and some of Color LOGO's most basic regular commands. Doodle mode provides a way to draw LOGO pictures using simple one-key commands. This capability makes Doodle mode an ideal LOGO application for children who cannot yet read or type reliably. Also in lesson one, students explore the concepts of drawing forward and backward and changing direction with Color LOGO commands such as **FD** for "forward" and **RT** for "right turn." At the end of the lesson is an activity that challenges students to draw their initials or name using Color LOGO.

Lesson Two introduces students to Color LOGO "procedures," sequences of Color LOGO steps that are put together to make a program that will draw an entire picture. Students learn how to use Color LOGO's Edit mode to create procedures that will draw simple shapes such as a square or a triangle. Then they use Run mode to actually run the procedures. At the end of the lesson, the students are challenged to draw a geometric shape.

Lesson Three covers problem-solving with Color LOGO. Exercises in this lesson involve using Color LOGO to finish a picture, to draw a pattern of repeated shapes, and to move the Color LOGO turtle (the little character that does the on-screen drawing) without leaving a trail. The "Daily Challenge" activity for this lesson asks the student to draw a geometric shape that involves repetition.

Lesson Four introduces colors and advanced problem-solving. Students learn how to change the color they are drawing with and the color of the entire screen. They practice using Color LOGO to solve a graphical problem by dividing the problem into steps. Lesson Four's "Daily Challenge" involves drawing a geometric shape twice, using two different colors.

Lesson Five introduces advanced LOGO commands, and shows students how to make a group of procedures work together to draw a single picture. Students learn to use the HATCH command to create extra turtles, so that two or more turtles can be at work drawing simultaneously. The final "Daily Challenge" asks students to draw two shapes at the same time, using HATCH.

Color LOGO Lab assumes that students have some beginning reading and writing skills, but skills beyond the first few grade levels are not required.

The course assumes no previous programming knowledge on the part of the instructor. Overhead transparencies help the teacher convey the content of each lesson, and the transparencies serve as note-taking guides for the students. Students take brief notes based on each transparency in their student manuals (provided with the package). These student manuals parallel the overhead transparencies in form and content, giving each student a permanent record of what he or she has learned.

Hands-on activity is the most important part of the learning process, and so students are given specific steps to complete at their Color Computers. Ideally, each student or pair of students should have a Color Computer to work with.

Stand-alone Color Computers or a series of Color Computers in a Network 2 configuration may be used. A "Master Procedure Disk" is included in Color LOGO Lab, with procedures that can be sent by the teacher to the students, over the Network 2 system. The Disk or Program PAK™ version of Color LOGO is required:

- The disk version of Color LOGO (Cat. No. 26-2721) is needed for stand-alone Color Computer disk systems, or for the Host in a Network 2 system.
- The Program PAK™ version of Color LOGO (Cat. No. 26-2722) is needed for stand-alone non-disk systems or for the Student Stations in a Network 2 system.

Color LOGO Lab (Cat. No. 26-2770) is available through your local Radio Shack store or Computer Center, or through the Radio Shack Regional Educational Coordinator in your area. Color LOGO Lab includes a comprehensive teacher's manual with overhead transparencies; 25 student workbooks; and the Master Procedure Disk. Suggested retail price is $199.00, but prices may vary at individual stores and dealers. Additional student workbooks are available separately (Cat. No. 26-2771) at a suggested price of $2.00 each.

PlannerViews

By Gordon O. Rapkin

Target PlannerCalc© by Comshare is a new spreadsheet modeling program for the TRS-80 Model 4. Not only is it a new addition to the TRS-80 Software Library, it is a new approach to spreadsheet applications. PlannerCalc is designed for superior ease of use and learning, applying plain English statements instead of algebraic formulas.

One of the most significant developments in the field of finance in the last ten years has been the availability and use of computer-based financial modeling systems. Although computers are not new to industry, they are a relatively recent entry to the general business community. The computer was a natural to become the workhorse of the modeling and planning executives. Repetitive, what-if planning was easily, quickly, and relatively cost effectively performed using many of the first computers available for commercial use.

The first machines used by the business community for modeling were large mainframe computers which corporations used predominantly for other purposes, but which offered available computer time for financial managers.

These machines were operated internally by the individual company or shared in a conventional time-sharing environment. Under either environment, computer time was expensive, and accessibility was remote. Despite these detriments, computers served the needs of modelers and established the framework for computer-based modeling as an effective planning tool.

Computers allowed the analyst to create alternative situations within the modeling environment. They gave the modeler an environment complete with powerful built-in functionality and report generation capability. The modeler simply constructed the relationships between the various line items and generated sets of reports based upon these values.

The advantages of computer-based modeling, even in its infancy, far out weighed the disadvantages; modeling took off as a popular computer application. As modeling became more and more popular, computer time on mainframe computers became scarce. Modeling languages were developed with non-technical English language interface and additional functionality, taking much of the drudgery away from the modeling process.

A number of years later, the minicomputer emerged. The mini brought computer accessibility to smaller companies who could not afford a mainframe computer but who could benefit from the capabilities of a computer system. Modeling followed the change in the computing environment. Minicomputer modeling languages were developed as offspring of popular mainframe modeling languages. They maintained the non-technical English language interface, provided modelers additional access to computer capability, reduced cost of computation, and shortened the turn-around time in obtaining decision support information.

Technology again reduced the price tag of computing power. In the late 1970s, the personal computer was born. It brought with it the opportunity for anyone to access the power of a computer at an affordable cost. Initially, however, the calc programs were designed for microcomputers with minimal amounts of memory, processing power, and disk space. The programs were created with the emphasis on screen display and speed, and minimum dependence on the software to remove the drudgery from business modeling.

Early products were based on matrix algebra. The user developed a model by defining each row and column intersection either in terms of data values or other row and column intersections. The business user was thinking in terms of "Accounts" and "Time Periods" and "Reports," and the software forced the user to translate these terms into row and column locations.

PlannerCalc is a new generation spreadsheet modeler that returns to the non-technical English language interface of traditional large scale modeling, without sacrificing the gains made by early calcs in speed and use of the display screen.

The term MODEL is frequently used in discussing the application of PlannerCalc. A model is a logical or mathematical representation of reality. The model is intended to explain or predict behavior of the entity being modeled, especially when contemplating change. The financial model is a set of rules, data and inferences about the state of the business, why it is where it is, how it reacts to change, and how it will get to its next objective.

TARGET
PlannerCalc

In future columns, we will explore the concepts of building models and applying them to business analysis. We will also present ready-to-use application models and tips on creating your own models. If you have any questions or comments to share with other PlannerCalc users, send them to:

PlannerViews
Comshare, Inc.
1935 Cliff Valley Way, Suite 200
Atlanta, GA 303029

FFA Becomes Part of AgriData's Electronic Network for Students

The Future Farmers of America (FFA), with a membership of nearly half-a-million students, has announced plans to participate in the development of the nation's first on-line classroom.

The Ag Ed Network, an electronic database for vocational agricultural students, is part of AgriData Network (formerly AgriStar), the national agricultural business information and communications system.

The FFA will be an information provider to the network by putting news of its events, programs and awards in the database and will help develop on-line material to supplement the curriculum and, through the FFA Foundation, solicit industry sponsors for specific educational lessons.

Coleman Harris, National FFA Executive Secretary, said "We believe the success of our young people lies in acquiring the skills of business and management and in moving ahead with computers. The FFA is pleased to be part of the network which will contribute on both counts."

Fifteen vo-ag high schools and colleges are already participating in the pilot stage of the service which will be fully operational by September.

The Ag Ed Network was designed by an advisory board of vo-ag teachers, university faculty, and AgriData's educational staff.

Services offered on the Ag Ed Network include:

Presently On-line

Lessons-On-line instructions, questions, activities and projects on business, marketing and management concepts to supplement vo-ag curriculum.

Daily Reports-The daily agricultural news commentary for students with discussion questions for teachers.

User Exchange-A voluntary directory of on-line users to locate teachers and classes with similar interests using Star-Gram, the system's electronic mail service.

By September 1984

Idea Exchange-On-line bulletin boards for posting student and teacher ideas.

Student News Service-Student journalists providing timely and local new stories to the education network.

Simulations-"Live" management and production simulations for students.

Correlations-Curriculum topics correlated to textbooks, publications, and AgriData reports.

Software Reviews-Classroom tested, user-written reviews on applications of software and computer-assisted instruction related to agriculture and education.

"Vo-ag teachers in our pilot schools are already discovering the use of AgriData Network as a 'live' textbook. The up-to-the-minute market, business and weather information is playing a vital role in teaching students the real agriculture," said Mary Z. Holmes, Executive Vice President, AgriData Resources, Inc., the Milwaukee-based operator of AgriData Network.

The StarGram electronic mail feature of the AgriData network will serve as an electronic communications link between vo-ag students and teachers throughout the country.

In another joint activity, AgriData Resources, Inc., is sponsoring the first Future Farmers of America "Computers in Agriculture" award to encourage vo-ag students to develop computer software and applications which increase management and business skills in agriculture. The award program will culminate in a national Computer Awards Seminar for state winners and industry representatives to be held in Washington, D. C. in the summer of 1984.

AgriData Network is accessible through a personal computer or data terminal over a local or toll-free telephone line.

AgriData Resources is also the leading publisher of business information for farmers, ranchers, and agribusiness including the monthly magazine *Farm Futures*.

Defining User Edits for Profile 16

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

Edits control the type of data Profile 16 lets you enter into a field. Your ability to create these edits, that is, to specify the "field type" for each field you define in a data base, is one of the keys to the program's power and flexibility. These edits can limit, alter, format or even replace the data entered from the keyboard when you add or update records. They even work with the results of the processing you've defined for your file (see last month's column). This month, we'll examine the mechanics of defining edits and suggest some additional uses for this important function.

IN THE GOOD OLD DAYS

Field types in Profile Plus and Profile III Plus are defined when screens are set up. During file definition, the user indicates only the heading and the length of each field. You make decisions about field types and formats during screen creation by using special symbols that function as field indicators. For instance, "*" indicates an alphanumeric field. "#" and "." indicate numeric fields, which are right-justified. Date formats use "{" and "}". Finally, a parallel set of indicators is available for "must-fill" versions of these field types.

FIELD DAY

Profile 16 works differently. Where Profile Plus and Profile III Plus have a whole array of field indicators used for screens, Profile 16 has only three: "*" for a normal field; "!" for a protected field, which can not be updated from the screen in question; and "%" for a "must fill" field of any type. "Must fill" means that the information will not be recorded until data have been entered into the relevant field. This can insure, for instance, that zip codes are not inadvertently omitted from mailing label files.

The field types themselves are identified during file definition. After entering the heading and length of each field, you type the code name or symbol of its "edit type" in the third column, labeled "type." This can be one of the types supplied with the program or one you define yourself using the Define Edits program module. But before going into the details, there are some important points to make. First, you don't have to do anything more involved than specifying alphanumeric/numeric distinctions so that the program will know which fields to right-justify. Doing nothing causes the program to specify alphanumeric ("*") as the default. Second, if you are transferring data bases from Profile Plus using the convert utility, Profile 16 automatically supplies the default for each field. You can make necessary adjustments after the conversion process is complete. Third, you can change or add any edit later.

In sum, you don't have to bother with the edits until you are good and ready to do so. I emphasize this only because some users may be a bit overwhelmed when they first open the Profile 16 manual. There is a lot in it about a lot of features. Relax! You can be up and running in short order and bring the more advanced features into your operation as you go along.

TYPES OF TYPES

There are three main categories of edits in Profile 16. The first, a series of "system" edits, comes with the program. They are available for use with any data base you create, but they cannot be modified by the user. These include the default alphanumeric edit type ("*"), a series of numeric edits that format numbers with either floating decimal or from 1 to 8 decimal places, and an exhaustive series of date edits, plus time edits for both time lapsed and time of day.

Here's a nice touch: entering just "/" into a date field or ":" into a time field will automatically result in the current date or time being recorded. Here's another nice touch: abbreviated entries are expanded as appropriate. For instance, entering "5/2/84" into a date field that includes slashes will result in "05/02/84" being recorded. Similarly, "12:01" becomes "12:01:00."

The second type of edit is a series of "global" edits, also called supplied edits, that come with the program. They, too, are available for use with any data base, and they can be modified by the user. You can also add more: Profile is delivered with 16; you can invent another 84. They serve two functions. First, they offer a number of useful formats for commonly used data types. For instance, "ZIP" accepts five- or ten-character zip codes, automatically supplying the dash in the longer format. "PHONE" accepts numbers, automatically supplying the parenthesis and dashes, which are optional, and formats the entry as a phone number. That is, if you enter "2125551212" you get "(212) 555-1212." "SSNUM" performs a similar operation for Social Security numbers. Second, certain of the supplied edits are used as building blocks to construct more elaborate edits, as we'll see below. You can examine these edits from the Define Edits menu choice by specifying "global" as the file name.

The third and last category consists of "user" edits. You can devise up to 100 for each data base you create. In other words, this last category is "local." Any such edit takes precedence over a global edit that has the same name. (System edits take precedence over everything.)

LEARNING THE LANGUAGE

Learning to define user edits means learning a simple language with its own vocabulary (the global edits), its own

Punctuation

( )	handle this part of the edit separately (as in algebra)
" "	literal
{ }	accept one or more of these
< >	accept if present or supply if absent (literals only; punctuation marks used as literals must be enclosed in quotes; all other literals need not be)
! !	accept and discard
[ ]	optional; the material (literals or names) inside the brackets is acceptable but not necessary.
|	or; press <CTRL> <0> (on the DT-1 numeric keypad, <SHIFT> <.>)
&	and; used for multi-part edits such as ranges
-	from/to on the list of ASCII characters
\	right-justifies the output (usually a number) and accepts leading blanks if it has inserted them; press <CTRL> <9> (on the DT-1 numeric keypad, <SHIFT> <1>); must be _first_ on the line.
^	in comparisons, case is not significant (takes effect wherever it falls on the line); press <SHIFT> <6>
~	accepts lower-case, converts to upper-case; takes effect wherever it appears on the line; press <CTRL> <6> (on the DT-1 numeric keypad, <SHIFT> <3>)
_	the underscore accepts upper-case, converts to lower-case; takes effect wherever it appears on the line.
%	if either "~" or "_" is in effect, turns off case conversion

Figure 1. Table 1.

punctuation (the special symbols on your keyboard) and its own syntax (the rules for combining words and symbols). The user manual has a good explanation of these rules, but the syntax can get a bit tricky. I'm going to present the same information in a slightly different way in hopes that it will make things even clearer. Table 1 shows the punctuation symbols used to define edits and describes their functions. Refer to it as you examine the following examples. Seeing the punctuation in action is better than simply looking at a bunch of definitions. Table 2 shows the global edits supplied with the program. In line with the language analogy, it's appropriate that the list is called the edit "dictionary." The entries themselves form a list of "model sentences," which can guide you as you formulate your own edits. I'll group these by function in the following discussion.

EDITS HAVE CHARACTER

Some global edits define what a particular character can be, acting as a "filter" for keyboard entry.

N   any digit from 0 to 9
A   any letter from A to Z, upper or lower-case
ASCII   any ASCII character, including a space
CHR   any ASCII character except a space

None of the above edits alter the character entered from the keyboard.

Another group of global edits filter character entry, as above, but may also modify the entry, usually for cosmetic reasons.

NUM   any digit from 0 to 9; supplies a "0" if position is blank
SEX   accepts only "M", "m", "F" or "f"; converts to upper-case
YESNO   accepts only "Y", "y", "N" or "n"; converts to upper-case

The above two groups can be used alone to edit one-character fields or in combination to edit more complex fields. You can easily add other edits of this type, either to the list of global edits (see Table 2) or to local edit tables for particular files.

--Name--	------------------------------ Edit -------------------------
N	"0"- "9"
A	^ "A"-"Z"
#	\ { {N} | ("."} | ("-"} | ("/") | (" "} }
NUM	N | <"0">
$	\ <"$"> NUM [(N)] <"."> NUM NUM
ZIP	N N N N N <-> N N N N | N N N N N
SSNUM	N N N <-> N N <-> N N N N
ALNUM	{ (A) | (N) | (" ") }
ASCII	{ " "-"~" }
CHR	"!"-"~"
SEX	^~ "M" | "F"
YESNO	^~ "Y" | "N"
PHONE	<"("> N N N <")"> <" "> N N N <"-"> N N N N | N N N <"-"> N N N N
ALLUP	~ ASCII
UPL	~CHR _ [(CHR)]
UPLOW	[(" "}] [{ UPL (" ") }] [UPL]

Figure 2. Table 2.

MAGIC FILTER

Another group of global edits performs operations similar to those above, but is not limited to one character at a time. Rather, this group filters data entry for fields of varying lengths.

\#   any quantity of digits and associated punctuation; right-justifies
ALNUM   alphanumeric: any digit, upper- or lower-case letter or space, but no punctuation or special characters

The next group works as above, but alters input in some way:

ALLUP   converts any lower-case letter to upper-case
UPLOW   capitalizes the first letter of each word (defined as a string of characters terminated by a space), and converts the remaining characters to lower-case; watch out—it can be tricky ("USA" becomes "Usa")

THE PROPER FORMAT

None of the above edits cares about the position of any particular character. They simply filter character entry one by one when the entered field is recorded. The rest of the global edits format data entry in various ways. This means that the position a particular character occupies in its field may be significant.

$	converts to dollars and cents format, supplying dollar sign and decimal point if not entered manually (does not round numbers)
ZIP	zip code format; supplies dash in 10-character code
SSNUM	Social Security Number format; supplies dashes.
PHONE	converts 7 digits to the form, nnn-nnnn; 10 digits to the form, (nnn) nnn-nnnn

AS TIME GOES BY

These models suggest other edit types that you might find useful. The first one, which I call HRMIN (hour/minute), is a time of day format that excludes seconds and might, therefore, be more appropriate for an appointment calendar than the time edits supplied with the system.

```
HRMIN    \ N N <":"> N N | N <":"> N N | N N <":00"> | N <":00">
```
Figure 3.

We'll "parse" this dictionary entry so you can really see how the edit is put together.

First, the name of the edit can be up to seven characters long, not including punctuation symbols and embedded or leading blanks. The distinction between upper- and lower-case is not significant. The chosen name is entered in the left column on the edit definition screen. The "sentence" that defines the edit goes on the right. The names of other edits used in building the new edit must be surrounded either by spaces or punctuation marks. Otherwise (as in this example), the program wouldn't be able to distinguish between "N" and "NN," a possible but different edit. Elsewhere, spaces are optional, but useful for clarity.

The backslash entered by typing (CTRL) (9) on the console, (SHIFT) (1) on the DT-1 numeric keypad, tells the program to right-justify the entry in the field. The field length for an hour/minute field would be five spaces ("12:30") although an entry might have as few as four characters ("1:00"). The backslash must come first on the line if it is used.

"N" allows entry of digits from 0 to 9. The pipe symbol separates the "or" possibilities from each other. The first of the four allowed entry formats (N N <":"> N N) says: the user can enter four digits, each of which can be any number from 0 to 9. If indeed four digits and only four digits are entered, the program will insert a colon (":") between the second and third digits. If the user enters a colon manually in that position, the program will let it be. Angle brackets ("<" and ">") surround any character that the field should accept if present in the proper position or supply if absent. If the character involved is a punctuation mark, it must be enclosed in quotes to distinguish it from the punctuation that the edit dictionary uses as instructions. Other characters need not be enclosed in quotes.

The second "or" group performs a similar operation, but deals with the case in which the user enters only three digits instead of four. The third group deals with the case of two digits by assuming that if only two digits are entered, the user is referring to the hour as opposed to the minutes, and so it fills in the format by adding ":00". The fourth case takes care of one-digit hours. If you enter a "2" the edit will expand the entry to "2:00".

The order of the "or" groups is important. The program examines each group one at a time, from left to right. If the data you try to enter into the field "fails" the first possibility, the program moves on to the second and so on. Therefore, the longest group should come first, with all other possibilities following in descending order. In my HRMIN edit, the four-digit possibility is first, followed by three, two, then one. It's easier to follow this rule religiously rather than try to predict when order matters and when it doesn't. Conversely, if you invent an edit that doesn't work the way you intended, and you've checked the punctuation, then examine the order of any "or" groups.

(P.S. At the last minute, this edit was added to the Profile 16 as a system edit under the name "HM".)

AN EDIT APART

Here's an edit suitable for a field that contains a part or catalog number. Since there are so many variants, I've chosen the Radio Shack catalog format as an example:
```
PARTNO N N NUM <"-"> (N N N N | <"0"> N N N)
```
If you check the table of global edits, you'll see that there is an important distinction between "NUM" and "N". "NUM" supplies a place-holding "0", which is needed here because Radio Shack's system includes such numbers as 26-506, 26-6412 and 700-1111. The PARTNO edit will convert these to 260-0506, 260-6412, and 700-1111.

A CHANGE OF MIND

I said above that Profile 16 lets you change your mind about edits after the fact. You can add or revise edits after data has been entered into your data base. This is because you declare field types during file definition. After you make changes and exit the Define Files program, Profile automatically restructures your records. If the new or revised edit causes the existing contents of a field to "fail" the edit test, the program reacts in intelligent ways.

In the case of a global or local edit, it leaves the data alone, making no changes. In the case of system edits, the program's reaction depends on the types of fields involved. If you inadvertently change a date field into a time field, the program ignores the directive because it "knows" you made a mistake. If you swap date formats, the program juggles the field appropriately. It deals internally with dates in a consistent fashion, which is why Profile 16 can sort dates chronologically no matter which display format you've chosen. However, if you change an alphanumeric field into a numeric field, existing alphanumeric data will be zeroed out.

WHY BOTHER?

Is it worth the extra effort to design and test user edits? The question is intelligent, and I think the intelligent answer is yes. It saves time and effort in the long run. Entering "1" instead of "1:00" and "1235" instead of "12:35" saves a second each time, since you have to shift for the colon. Hundreds of records equals hundreds of seconds. It promotes accuracy. The fewer keystrokes required, the fewer chances of mis-typing them. It's elegant and fun. How many things can you say that about these days?

TWO MORE FOR THE ROAD

My DAY edit accepts abbreviated entries such as "m" and "th" and converts them to the standard three-character abbreviations for the days of the week:

DAY   ^_"th"<u>|"su"<n>|"m"<o><n>|"t"<u><e>|"w"<e><d>|"f"<r><i>|"s"<a><t>

Figure 4.

Note that the longer two-character possibilities must come first so that the program can distinguish between "t" for Tuesday and "th" for Thursday.

MONTH was designed by Ken Brody at The Small Computer Company to expand abbreviated entries to the complete month name.

```
MONTH    ^"Ja"<n><uary>|"F"<e><b><r><uary>|"M"<a>"r"<ch>|Mon2
Mon2     "Ap"<r><il>|"M"<a>"y"|"J"<u>"n"<e>|"J"<u>"l"<y>|Mon3
Mon3     "Au"<g><ust>|"S"<ep><t><ember>|"O"<c><t><ober>|Mon4
Mon4     "N"<o><v><ember>|"D"<e><c><ember>
```

Figure 5.

Yes, edits can be "daisy-chained" if they will not fit on one line. Simply give each continuation a name and have that name be the last "or" group on the line.

Try entering these edits into your own Profile 16 system. You can test them from the Define Edit screen. That last cursor position is simply a parking space for the cursor after you enter the field. Then make up your own global or local edits. And send us your best ones so we can all share them. If we don't edit together, we'll all edit separately.

PROFILE Editor's Note: This is Mr. Sygoda's eighteenth article in a series of 'how-to' Profile articles. We hope that you enjoy this feature, and we look forward to your comments and questions on Profile.

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

Amortization on the MC-10

Dennis O Dorrity
844 38th Street
Ogden, UT 84403

This amortization program was originally designed for the first Color Computer with standard BASIC. It was written to prove to a friend that 4K was enough memory for a useful program. Once you have the program entered, debugged, and running, you will be asked for three figures: the amount borrowed, the number of months, and the rate of interest. The amount borrowed is the amount that you need minus trade-in, down-payment, etc. The number of months is obtained by multiplying the number of years you will pay on the loan by twelve and adding any additional months. For example, two years and six months equals thirty months (2 × 12 + 6 = 30). The interest rate is entered like this: 11.5% becomes 11.5.

If you make a mistake on the entry, don't sweat it. When you get through entering the three figures from above, you are asked if you need to make changes. If you do, just enter the number of the item you need to change. The old item is replaced by the new entry. When you are satisfied with the results, press ④. You will now see the results of the first month. Pressing Ⓝ will give you the next month's results. Pressing Ⓜ will bring you back to the menu where you can make changes. I hope this will help you in your financial dealings.

```
10 REM ******************
20 REM **** MC-AMORT ****
30 REM ******************
40 REM BY DENNIS DORRITY
50 GOTO 690
60 GOSUB 160
70 GOSUB 180
80 GOSUB 210
90 GOSUB 230
100 PRINT@320, " 4. NO CHANGES. RUN PROGRAM."
110 PRINT@384, " 5. STOP PROGRAM."
120 PRINT@449, " ANY CHANGES? <ENTER 1-5>";
    : INPUT A
130 ON A GOSUB 180, 210, 230, 270, 690
140 GOTO 120
150 GOTO 150
160 CLS 3
    : PRINT@35, " $ AMORTIZATION PROGRAM $ ";
170 RETURN
180 PRINT@96, " "
    : PRINT@96,"1. AMOUNT BORROWED";
    : INPUT OP
190 PV=OP
200 RETURN
210 PRINT@160," "
    : PRINT@160," 2. TOTAL MONTHS TO PAY:";
    : INPUT N
220 RETURN
230 PRINT@224." 3. ANNUAL PERCENTAGE RATE"
240 PRINT@256," "
    : PRINT@256,"   (11.5% = 11.5)";
    : INPUT PR
250 PI=PR
260 RETURN
270 PI=PI*.01/12
280 N1=(1+PI)
    : N2=N1
290 FOR X=1 TO (N-1)
300 N1=N1*N2
310 NEXT X
320 N2=N1-1
330 MP=PV*PI*N1/N2
340 MP=INT(MP*100)/100
350 FOR X=1 TO N
360 IP=PV*PI
370 IP=INT(IP*100)/100
380 PP=MP-IP
390 PP=INT(PP*100)/100
400 TI=TI+IP
410 TI=INT(TI*100)/100
420 TP=TP+PP
430 PT=INT(PT*100)/100
440 CLS 3
    : PRINT@37," MONTH NO."X"SCHEDULE ";
450 SP=PV-PP
    : IF X=N THEN SP=0
460 PRINT@96," PRINCIPLE OWING    " SP
470 PRINT@160,"   INTEREST PAYMENT " IP
480 PRINT@192," + PRINCIPLE PAYMENT" PP
490 PRINT@224," = MONTHLY PAYMENT   " MP
500 PRINT@288,"   TOTAL INTEREST · " TI
510 IF X=N THEN TP=OP
520 PRINT@320," + TOTAL PRINCIPAL   " TP
530 PT=TI+TP
    : PT=PT+RF
    : PT=INT(PT*100)/100
540 PRINT@352," = TOTAL PAYMENTS    " PT
550 PV=PV-PP
560 PV=INT(PV*100)/100
570 PRINT@419," TYPE <N> FOR NEXT MONTH ";
580 PRINT@451," TYPE <M> TO MAKE CHANGES ";
590 A$=INKEY$
    : IF A$="" THEN 590
600 IF A$="M" THEN 630
610 IF X <= N AND A$="N" THEN NEXT X
620 GOTO 590
```

```
630 CLS 3
    : PRINT@96," 1. AMOUNT BORROWED?" OP
640 PRINT@160," 2. TOTAL MONTHS TO PAY?" N
650 PRINT@224," 3. ANNUAL PERCENTAGE RATE"
660 PRINT@256,"    (11.5% =11.5)?" PR
670 TI=0
    : TP=0
    : PT=0
    : PV=OP
    : PI=PR
680 GOTO 100
700 PRINT@103," BY DENNIS O DORRITY ";
760 PRINT@450," PRESS <ENTER> TO CONTINUE ";
770 IF INKEY$=CHR$(13) THEN 60
780 GOTO 770
```


Clock Tutor on the PC-2

A.E. Clarke
16648 35th Avenue
Seattle, WA 98188

I've noticed that computer generated clock faces seem to be a popular pastime, so I let my PC-2 have a go at it. The result is an educational program that draws a clock face, puts a random time on it and requests the correct time to be entered. I hope someone may be able to use it to help teach their children to tell time.

```
10 WAIT 100
    : CURSOR 2
    : PRINT "**** TIME TUTOR ****"
20 PRINT "by AL CLARKE   Sept. 1983
30 CLS
    : CLEAR
    : RANDOM
    : GRAPH
    : BEEP ON
    : DEGREE
40 WAIT 0
    : CURSOR 5
    : PRINT "< PLEASE WAIT >
50 LINE (8,30)-(208,-170),,,B
60 R=80
    : GLCURSOR (108,-70)
    : SORGN
70 GLCURSOR (0,R)
80 FOR J=0TO 354STEP 6
90 I=75
    : IF J/5=INT (J/5)LET I=70
100 X=R*SIN J
    : Y=R*COS J
    : A=I*SIN J
    : B=I*COS J
110 LINE (A,B)-(X,Y)
    : NEXT J
120 FOR J=1TO 12
    : RESTORE
130 FOR K=1TO 12
140 READ R,A
    : IF K=J THEN 160
150 NEXT K
160 P=J*30+A
    : X=R*SIN P
    : Y=R*COS P
170 GLCURSOR (X,Y)
    : LPRINT STR$ J
180 NEXT J
190 LINE (3,3)-(-3,-3),,,B
200 GLCURSOR (0,0)
210 M=RND 60
    : IF M=60 LET M=0
220 H=RND (12)
230 R=50
    : X=R*SIN(H*30+(M/2))
    : Y=R*COS(H*30+(M/2))
240 LINE -(X,Y)
    : GLCURSOR (0,0)
250 R=70
    : X=R*SIN(M*6)
    : Y=R*COS (M*6)
260 LINE -(X,Y)
    : GLCURSOR (0,0)
270 TEXT
    : LF 9
280 M$=STR$ M
    : IF M<10 LET M$="0"+STR$ M
290 H$=STR$ H+":"
300 WAIT 150
    : PRINT "**** What time is it? ****
310 INPUT "Enter the hour ";A
320 IF A=H GOTO 340
330 GOSUB "M"
    : GOTO 310
340 T=0
    : BEEP 1
    : WAIT 128
    : PRINT "That is correct!"
350 WAIT 0
    : PRINT "Now the minutes ";H$
    : CURSOR 16+LEN H$
    : INPUT B
    : CLS
360 IF B=M GOTO 380
370 GOSUB "M"
    : GOTO 350
380 BEEP 5
    : WAIT 128
    : PRINT "YOU ARE CORRECT
390 WAIT 128
    : PRINT "THE TIME SHOWN IS ";H$;M$
400 LF -3
    : LPRINT TAB 2;H$;M$
    : LF 3
    : GOTO "R"
410 "M"T=T+1
420 IF T=3 WAIT 0
    : PRINT "YOU'VE MISSED 3 TIMES"
    : GCURSOR 20
    : WAIT 128
    : GPRINT 11;7
    : GOTO 390
430 BEEP 2
    : WAIT 100
    : PRINT "SORRY, TRY AGAIN!"
    : RETURN
```

```
440 "R"Y$="(Yes/No)",N$="( es/ o)"
450 WAIT 0
    : PRINT "Do another?  ";Y$
460 FOR K=1 TO 60
    : I$=INKEY$
    : IF LEN I$ LET K=60
470 NEXT K
480 IF ASC I$=78 GOTO "N"
490 IF ASC I$=89 GOTO 30
500 WAIT 9
    : CURSOR 13
    : PRINT N$
    : GOTO 450
510 "N"CLS
    : WAIT 64
    : CURSOR 7
    : PRINT CHR$ 123;" GOOD BYE ";CHR$ 125
520 BEEP 1
    : FOR J=38 TO 110
    : GCURSOR J
    : WAIT 0
    : GPRINT 127-POINT J
    : NEXT J
    : BEEP 1
    : END
530 DATA 85, 0, 85, 3, 85, 4, 90, 5, 95, 3, 98, 2
540 DATA 100, 0, 97, -2, 95, -4, 100, -7, 93, -7, 83,
    -7
```

PC-2 Clock

C. M. MacLeod
10 North Crane Avenue
Taunton, MA 02780

This program has some interesting graphics to go with the clock. I have kept this program in my PC-2 for several months, and it always seems to impress those I show it to.

```
5 ARUN
10 "A"
    : LOCK
    : CLEAR
    : RESTORE
    : RANDOM
    : CLS
    : WAIT 0
15 G$=CHR$(126)
    : A$=CHR$(127)+CHR$(127)+CHR$(127)+CHR$
    (127)+CHR$(127)+CHR$(127)+CHR$(127)
17 A$=A$+A$
    : K$=CHR$(123)
    : J$=CHR$(125)
20 T$=STR$ TIME
    : IF LEN T$<11 THEN LET T$="0"+T$
25 MTH$=MID$(T$,1,2)
    : MTH=VAL MTH$
    : DAY$=MID$ (T$,3,2)
    : DAY=VAL DAY$
27 YEAR=83
30 HR$=MID$(T$,5,2)
    : MN$=MID$(T$,8,2)
    : HR=VAL HR$
    : MN=VAL MN$
31 IF HR>=12 THEN LET HR=HR-12
    : PM$="pm"
    : GOTO 40
32 PM$="am"
40 HR$=STR$ HR
    : MN$=STR$ MN
41 IF VAL HR$=0 THEN LET HR$="12"
42 IF VAL HR$<10 THEN LET HR$=G$+HR$
45 IF VAL MN$<10 THEN LET MN$="0"+MN$
```

```
50 TM$=G$+HR$+":"+MN$+PM$
55 CURSOR 8
60 PRINT TM$
62 WAIT 0
70 FOR X=46 TO 101
80 GCURSOR X
90 GPRINT 127-POINT X
91 NEXT X
100 WAIT 0
110 DIM N(12),D$(6),MTH$(12)*6
113 FOR X=1TO 12
    : READ N(X)
    : NEXT X
115 FOR X=0TO 6
    : READ D$(X)
    : NEXT X
117 FOR X=1TO 12
    : READ MTH$(X)
    : NEXT X
125 Y1=YEAR
127 IF YEAR/4=INT (YEAR/4)THEN 129
128 YEAR=YEAR-1
    : GOTO 127
129 YEAR=YEAR/4
130 T=YEAR+Y1+N(MTH)+DAY
132 T1=INT(T/7)
    : T2=T-(T1*7)
    : IF Y1=00 THEN 150
140 IF Y1/4=INT(Y1/4) AND MTH=1 THEN T2=T2-1
    : GOTO 150
142 IF Y1/4=INT(Y1/4) AND M=2 THEN T2=T2-1
150 REM
160 GCURSOR 1
161 IF TZ=1 THEN RETURN
162 PRINT D$(T2)
163 IF LEN (D$(T2))<=6 THEN GCURSOR 120
    : GOTO 167
164 GCURSOR 108
167 WAIT 100
    : PRINT D$(T2)
168 DATE$=K$+" "+MTH$(MTH)+STR$ DAY+" "+J$
169 WAIT 40
    : GCURSOR 42
    : PRINT DATE$
    : WAIT 0
170 FOR X=0 TO 40
    : GCURSOR X
    : GPRINT 127-POINT X
    : NEXT X
172 FOR X=155 TO 118 STEP -1
    : GCURSOR X
    : GPRINT 127-POINT X
    : NEXT X
    : GCURSOR 35
    : FOR T=1 TO 95
    : NEXT T
    : WAIT 50
    : PRINT A$
    : GOTO 178
173 FOR X=155 TO 1STEP -1
    : GCURSOR X
    : GPRINT 127
174 IF X<=45 THEN GCURSOR X+110
    : GPRINT 0
175 NEXT X
    : WAIT 0
    : FOR X=110 TO 0 STEP -1
    : GCURSOR X
    : GPRINT 0
    : NEXT X
178 CLS
    : CURSOR 9
    : WAIT
    : PRINT "memory=";INT ((MEM
    /1000+.005)*100)/100;"k"
    : END
```

```
18Ø DATA 1, 4, 4, Ø, 2, 5, Ø, 3, 6, 1, 4, 6
19Ø DATA "S a t.","Sunday","Monday","Tues.","W e
    d.","Thurs.","Friday"
195 DATA "J a n.","F e b.","March","April","M a
    y","June","July","A u g.","Sept."
2ØØ DATA "O c t.","N o v.","D e c."
21Ø END
```
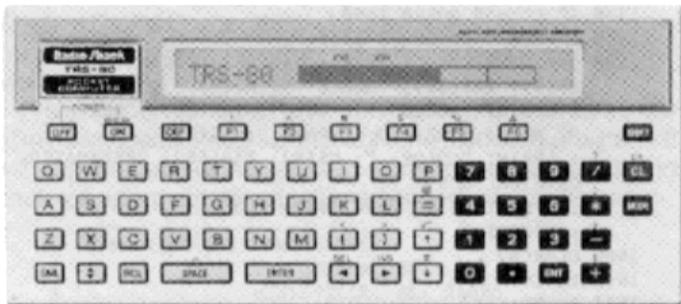
Household Inventory

Arnold E. van Beverhoudt, Jr.
P. O. Box 56
St. Thomas, VI 00801

How lo g have you been putting off preparing an inventory of your personal property? Well, now is your big chance. Why not let your TRS-80 and this program help you? No, your TRS-80 won't run around the house getting serial numbers, nor will it go through those old receipts looking for purchase dates and prices. But once you've done that, your TRS-80 can help you organize it all and give you a neat listing of all your worldly goods.

In order to save the overhead of file load and save routines and add-delete-change routine, I chose to put the data file in DATA statements. This leaves more room for your file. Each record contains the following elements:

    room number
    item description
    brand
    year bought
    place bought
    model number
    serial number
    color
    cost

The DATA statements containing this information should begin at line 2000, and the records must be in order by room number. To make it easier to update the file later, I suggest setting aside a block of line numbers for each room, such as 2000-2990 for room 1, 3000-3990 for room 2, etc. The last DATA statement (line 10000) must be the end of file marker.

Household Inventory was set up for the configuration of my home. You may want to make the following changes to suit your needs. In line 110, A$ should equal your home address. In lines 1010 to 1080, record the rooms of your house (8 rooms only).

Finally, in lines 1810 to 1920, place the same room names you used above into DATA statements.

The program has the following sections:
  100 -   210 initialize arrays and print title page
  500 -   560 main menu
 1000 - 1230 display inventory on screen (includes selection menu)
 1500 - 1640 print inventory records to printer
 1650 - 1670 print grand total cost
 1690 - 1730 print page headings
 1740 - 1750 print room total cost
 1760 - 1800 print room name sub-headings
 1810 - 1820 DATA statements for room names
 2000 - 2090 inventory DATA file
 10000   end of file marker

The variables used in the program are:
A$ - your home address
B$ - array for brand
C   - array for time cost
C$ - array for color
D$ - current date
F   - maximum number of times in the file
I$ - array for item description
LC - line counter
M$ - array for model number
N   - loop counter
N$ - array for serial number
OL - old room number (for sub-total breaks)
R   - array for room number
R$ - array for room name
RC - total room cost
S   - menu choice variable
S$ - array for place bought (source)
SS - array index for room names
TC - grand total cost
TS - tab stop to center room name headings
Y$ - array for year bought

```
1Ø REM  **********************************
         ****** HOUSEHOLD INVENTORY *******
      ************* (1981) BY ************
              ** ARNOLD E. VAN BEVERHOUDT, JR. **
              **********************************
99 REM *** INITIALIZATION ROUTINE ***
1ØØ F = 125
      : REM ** MAXIMUM NUMBER OF RECORDS IN FILE **
11Ø A$="YOUR ADDRESS HERE"
12Ø CLS
      : PRINT
      : PRINT
      : PRINT
      : PRINT CHR$(23); TAB(7); "HOUSEHOLD INVENTORY"
      : PRINT
      : PRINT
      : PRINT
      : PRINT TAB(15); "BY"
      : PRINT
      : PRINT
      : PRINT
      : PRINT TAB(2); "ARNOLD E. VAN BEVERHOUDT, JR."
13Ø DIM R$(9), R(F), I$(F), B$(F), Y$(F), S$(F),
      M$(F), N$(F), C$(F),C(F)
14Ø RESTORE
15Ø FOR N=1 TO 8
      : READ R$(N)
      : NEXT N
16Ø FOR N=1 TO F
17Ø READ  R(N), I$(N), B$(N), Y$(N), S$(N), M$(N),
      N$(N), C$(N), C(N)
18Ø IF R(N)=9 GOTO 19Ø  ELSE NEXT N
```

```
190 CLS
200 PRINT "HOUSEHOLD ADDRESS: ";A$
210 INPUT "ENTER TODAY'S DATE IN THE FORMAT
    MM/DD/YY";D$
499 REM *** MAIN MENU ROUTINE ***"
500 CLS
    : PRINT
    : PRINT
    : PRINT TAB(23);"HOUSEHOLD INVENTORY"
    : PRINT
    : PRINT
510 PRINT TAB(5);"1. TO SEE THE INVENTORY ON THE
    VIDEO DISPLAY"
520 PRINT TAB(5);"2. TO PRINT THE INVENTORY ON THE
    LINE PRINTER"
    : PRINT TAB(5);"3. TO END THE PROGRAM"
530 PRINT
    : PRINT
    : INPUT"ENTER YOUR CHOICE (1 - 3)";S
540 IF S<1 OR S>3 GOTO 500
550 IF S=1 GOSUB 1000 ELSE IF S=2 GOSUB 1500 ELSE RUN
    "MENU"
560 GOTO 500
999 REM *** INVENTORY SELECTION MENU ROUTINE ***
1000 CLS
    : PRINT TAB(23);"HOUSEHOLD INVENTORY"
    : PRINT
1010 PRINT" 1. LIVING ROOM"
1020 PRINT" 2. DINING ROOM"
1030 PRINT" 3. KITCHEN"
1040 PRINT" 4. MASTER BEDROOM"
1050 PRINT" 5. BEDROOM NO. 1"
1060 PRINT" 6. OFFICE"
1070 PRINT" 7. HALLWAY"
1080 PRINT" 8. OUTDOORS"
1090 PRINT
    : INPUT "ENTER YOUR CHOICE OF LOCATION (1 TO 8)
    OR '9' TO END";S
1100 IF S<1 OR S>9 GOTO 1000 ELSE GOTO 1110
1110 IF S=9 GOTO 1230
1120 CLS
    : FOR N=1 TO F
1130 IF R(N)=9 GOTO 1200
1140 IF R(N)=S GOTO 1150 ELSE GOTO 1190
1150 PRINT "ITEM: "; I$(N); TAB(28); "BRAND: ";
    B$(N); TAB(52); "YEAR: "; Y$(N)
    : PRINT "SOURCE: "; S$(N); TAB(30); "MODEL: ";
    M$(N); TAB(51); "COST: ";
    : PRINT USING "$##,###"; C(N);
1160 PRINT "SERIAL #: "; N$(N); TAB(27); "COLOR: ";
    C$(N); TAB(52); "ROOM #:"; R(N)
    : PRINT
1170 R$=INKEY$
    : IF R$="" GOTO 1170 ELSE 1190
1180 RC=RC+C(N)
1190 NEXT N
1200 PRINT
    : PRINT"          TOTAL PROPERTY VALUE - ROOM # ";
    R$(S);": ";
    : PRINT USING"$##,###"; RC
    : RC=0
1210 PRINT
    : PRINT "PRESS 'ENTER' TO RETURN TO MENU"
1220 R$=INKEY$
    : IF R$="" GOTO 1220 ELSE GOTO 1000
1230 CLS
    : RETURN
1499 REM *** PRINT INVENTORY ROUTINE ***
1500 POKE 16424,66
    : POKE 16425,1
1510 CLS
    : PRINT TAB(5); "DO YOU WANT TO CANCEL THE PRINT
    COMMAND (Y/N) ?"
1520 R$=INKEY$
    : IF R$="" THEN 1520 ELSE IF R$="Y" GOTO 500
    ELSE 1530
```

```
1530 OL=1
    : LC=50
    : RC=0
    : TC=0
1540 PRINT
    : PRINT"          THE HOUSEHOLD INVENTORY IS NOW
    BEING PRINTED."
1550 FOR N=1 TO F
1560 IF R(N)=9 GOTO 1640
1570 IF LC>=50 GOSUB 1690
1580 IF R(N)<>OL GOSUB 1730
1590 LPRINT"ITEM: "; I$(N); TAB(28); "BRAND: ";
    B$(N); TAB(52); "MODEL: "; M$(N)
    : LPRINT "SOURCE: "; S$(N); TAB(30); "SERIAL #:
    "; N$(N); TAB(57); "YEAR: "; Y$(N)
    : LPRINT "COLOR: "; C$(N); TAB(48); "ESTIMATED
    ORIGINAL COST: ";
    : LPRINT USING "$##,###"; C(N)
    : LPRINT
1600 OL=R(N)
1610 LC=LC+4
1620 RC=RC+C(N)
1630 NEXT N
1640 GOSUB 1730
1650 TC=TC+RC
    : LPRINT
    : LPRINT TAB(32); "GRAND TOTAL COST - ALL ROOMS:
    ";
    : LPRINT USING "$##,###"; TC
1660 LPRINT CHR$(11);
1670 RETURN
1690 LPRINT TAB(31); "HOUSEHOLD INVENTORY"
1700 LPRINT TAB((80-LEN(A$))/2); A$
    : LPRINT TAB(33); "AS OF "; D$
    : LPRINT
1710 LC=0
1720 IF R(N)<>OL THEN GOTO 1740 ELSE GOTO 1750
1730 IF LC>45 LPRINT CHR$(11);
    : GOTO 1690 ELSE LC=LC+5
1740 LPRINT
    : LPRINT TAB(32); "TOTAL COST - ROOM #
    ";R$(SS);":";
    : LPRINT TAB(73);
    : LPRINT USING "$##,###"; RC
1750 SS=R(N)
    : IF R(N)=9 RETURN
1760 TS=(80-LEN(R$(SS)))/2
1770 LPRINT
    : LPRINT TAB(TS);R$(SS)
1780 LPRINT
1790 TC=TC+RC
    : RC=0
1800 RETURN
1810 DATA 1. LIVING ROOM,2. DINING ROOM, 3.
    KITCHEN,4. MASTER BEDROOM
1820 DATA 5. BEDROOM NO. 1,6. OFFICE,7. HALLWAY,8.
    OUTDOORS
1999 REM *** DATA FILE - 125 ITEMS MAXIMUM ***
2000 REM *** SAMPLE DATA ***:2000 REM *** DATE
    STATEMENT FORMAT ***2000 REM **** SAMPLE DATA
    STATEMENT FORMAT ****          DATA
    1,AM/FM-PHONO-TAPE UNIT,SONY,1980,HIFI
    SHOP,HP-319,S-1234,WALNUT,250
10000 DATA 9," "," "," "," "," "," "," "," ",0
```

Model III Head Cleaner

Dewey Lusk
161 Country Rd.
Huntsville, AL 35806

Although I had been considering writing this program some time, it was Mr. Johnson's Model I head cleaner pro-

gram in the April '82 issue that inspired me to go ahead and to try. In Johnson's article he stated that he could find no source for the disk controller addresses for the Model III. That's because the controller is port driven. The port address information may be found in the Model III Technical Reference Manual (26-2109). However, this manual does not provide enough information to program the controller. I was able to obtain the data sheet on the 179x Disk Controller series from Western Digital Corporation which provides detailed information on programming the 1793 controller used in the Model III.

Using the information, Mr. Johnson's program and a few of my own ideas, I was able to write the following program. It will step the head of the selected drive from track 0 to 39 and back for approximately as many seconds as requested.

```
1 'SUBMITTED BY: DEWEY LUSK
2 '                161 COUNTRY RD.
3 '                HUNTSVILLE, AL. 35806
10 DEFINT A-Z
20 CLS
30 PRINT "******** M3 DRIVER PROGRAM FOR HEAD
      CLEANING DISKETTES ********"
40 PRINT
50 PRINT "REMOVE PROGRAM DISKETTES FROM ALL DRIVES."
60 INPUT "WHICH DISK DRIVE"; A
70 IF A<0 OR A>3 THEN 1000
80 DR=2[A
90 INPUT "HOW MANY SECONDS TO CLEAN (0 TO QUIT)"; A
100 IF A=0 THEN 310
110 TM=INT(A/2.5)
120 CLS
130 PRINT "INSERT CLEANING DISKETTE AND PRESS ANY KEY
      TO CONTINUE."
140 A$=INKEY$
150 IF A$="" THEN 140
160 GOSUB 2000
161 '   RESTORE HEAD TO TRACK 0
170 OUT 240,0
180 SR=INP(240)
190 BU = SR AND 1
200 IF BU = 1 THEN 180
201 '  TEST TO SEE IF HEAD IS AT TRACK 0
210 TR = SR AND 4
220 IF TR <> 4 THEN 1100
221 '  LOOP TO CYCLE HEAD TO TRACK 39 AND BACK
222 '  EACH CYCLE IS APPROXIMATELY 2.5 SECONDS
230 FOR I=1 TO TM
240 OUT 243,39
250 GOSUB 2000
260 OUT 240,19
270 GOSUB 2000
280 OUT 240,3
290 GOSUB 2000
300 NEXT I
310 END
999 '  DRIVE NO. ERROR MESSAGE
1000 CLS
1010 PRINT "NUMBER MUST BE 0-3. PRESS ANY KEY TO
      CONTINUE."
1020 A$=INKEY$
1030 IF A$="" THEN 1020
1040 CLS
1050 GOTO 60
1099 ' TRACK 0 FAIL ERROR MESSAGE
1100 CLS
1110 PRINT "DISK DRIVE MALFUNCTION OR DEVICE NOT
      AVAILABLE."
1120 PRINT "PRESS ANY KEY TO CONTINUE."
1130 A$=INKEY$
1140 IF A$="" THEN 1130 ELSE 310
1998 '  SUBROUTINE TO SELECT DRIVE AND
1999 '     CHECK DRIVE BUSY FLAG.
```

```
2000 OUT 244,DR
2010 SR=INP(240)
2020 BU=SR AND 1
2030 IF BU=1 THEN 2010
2040 RETURN
```

Clock for the CoCo

Charles M. MacLeod
10 North Crane Ave.
Tuanton, MA 02780

This Color Computer Clock program requires Extended Color BASIC because it uses the TIMER function. Just enter the Hours and Minutes as prompted.

```
10 REM *************************
11 REM *************************
12 REM ********TIME*************
13 REM *********BY*************
14 REM ****CHUCK*MACLEOD********
15 REM ***10 N. CRANE AVE.******
16 REM **TAUNTON, MA. 02780*****
17 REM *****MAY 7, 1983*********
18 REM *TRS-80  COLOR  COMPUTER*
19 REM ***32K-EXTENDED  BASIC***
20 REM *************************
21 REM *************************
22 CLS 0
23 INPUT "HOUR=";HR
     : INPUT "MIN=";MN
24 CLS 0
25 GOSUB 37
26 TIMER=0
27 A=TIMER
28 SEC=INT(A/60)
     : SEC$=STR$(SEC)
     : IF VAL(SEC$)<10 THEN SEC$=" "+SEC$+" "
     ELSE SEC$=STR$(SEC)+" "
29 PRINT@234, SEC$;
30 PRINT@436, "" ;TIME$;
31 IF A>=3600 THEN TIMER=0
     : MN=MN+1
32 IF MN=60 THEN HR=HR+1
     : MN=0
33 IF HR>12 THEN HR=HR-12
34 HR$=STR$(HR)
     : IF LEN(HR$)<3 THEN HR$=" "+HR$
35 MN$=STR$(MN)
     : IF LEN(MN$)<3 THEN MN$="0"+MN$
36 TIME$=" "+HR$+":"+MN$+" "
     : GOTO 27
37 Y=23
     : FOR X=0 TO 6
38 SET(X,Y,7)
39 NEXT X
40 X=3
     : FOR Y=24 TO 30
41 SET(X,Y,7)
42 NEXT Y
43 X=10
     : Y=24
     : SET(X,Y,7)
     : FOR Y=26 TO 30
44 SET(X,Y,7)
45 NEXT Y
46 X=15
     : FOR Y=24 TO 30
47 SET(X,Y,7)
48 NEXT Y
49 X=16
     : Y=25
```

```
        : SET(X,Y,7)
        : X=17
        : Y=26
        : SET(X,Y,7)
        : X=18
        : Y=27
        : SET(X,Y,7)
        : X=19
        : Y=26
        : SET(X,Y,7)
        : X=20
        : Y=25
        : SET(X,Y,7)
        : X=21
        : FOR Y=24 TO 30
        : SET(X,Y,7)
        : NEXT Y
50 X=26
        : FOR Y=24 TO 30
        : SET(X,Y,7)
        : NEXT Y
        : Y=24
        : FOR X=26 TO 30
        : SET(X,Y,7)
        : NEXT X
51 Y=27
        : FOR X=26 TO 30
        : SET(X,Y,7)
        : NEXT X
52 Y=30
        : FOR X=26 TO 30
        : SET(X,Y,7)
        : NEXT X
53 Y=30
        : FOR X=35 TO 61 STEP 2
        : SET(X,Y,7)
        : NEXT X
        : Y=24
        : FOR X=35 TO 61 STEP 2
        : SET(X,Y,7)
        : NEXT X
        : X=35
        : FOR Y=24 TO 30 STEP 2
        : SET(X,Y,7)
        : NEXT Y
        : X=63
        : FOR Y=24 TO 30 STEP 2
        : SET(X,Y,7)
        : NEXT Y
54 Y=5
        : FOR X=15 TO 32
        : SET(X,Y,5)
        : NEXT X
        : Y=6
        : FOR X=16 TO 31
        : SET(X,Y,5)
        : NEXT X
        : Y=7
        : FOR X=17 TO 30
        : SET(X,Y,5)
        : NEXT X
        : Y=8
        : FOR X=18 TO 29
        : SET(X,Y,5)
        : NEXT X
        : Y =9
        : FOR X=19 TO 28
        : SET(X,Y,5)
        : NEXT X
        : Y=10
        : FOR X=20 TO 27
        : SET(X,Y,5)
        : NEXT X
        : Y=11
        : FOR X=21 TO 26
        : SET(X,Y,5)
```

```
        : NEXT X
        : Y=12
        : FOR X=20 TO 27
55 SET(X,Y,5)
        : NEXT X
56 Y=13
        : FOR X=19 TO 28
        : SET(X,Y,5)
        : NEXT X
        : Y=14
        : FOR X=18 TO 29
        : SET(X,Y,5)
        : NEXT X
        : Y=15
        : FOR X=17 TO 30
        : SET(X,Y,5)
        : NEXT X
        : Y=16
        : FOR X=16 TO 31
        : SET(X,Y,5)
        : NEXT X
        : Y=17
        : FOR X=15 TO 32
        : SET(X,Y,5)
        : NEXT X
57 RETURN
```

Scrabble® Scorekeeper for the PC-2

Victor Wright
8065 N. Meridian Street
Indianapolis, IN 46260

I have written a program to keep scores for Scrabble games. Keeping up with the input of each player's score each turn has been a problem, particularly since the score-keeper is busy thinking about his/her next turn as well as running the PC-2 program. The "?" asking for input all by itself is a problem: the scorekeeper asks "Whose turn is it?" or "Did I enter M's score last time?" The instruction with the present player's initial comes up with the ? and stays while the scorekeeper enters the score. (If there is a boo-boo, the red button clears the wrong score.) The score then is verified for a little while, and the next person's score is asked for. Lines 8030, 8031 and 8034 set off the "bells and whistles" extravaganza! The PC-2 beeps, adds the total game points and draws a red box around it, makes a frame out of yen symbols around "n IS THE WINNER.", and computes the elapsed time of the game.

The dialog between the PC-2 and the scorekeeper is straightforward until the end of the game. When a player finally "goes out" his last word is scored plus the value of all the files of the opponents that still remain in the racks. Press ⌷ENTER⌷ until the PC-2 prints out the scores of all playing. Then when "Enter score of n?" appears (this will be the initials of the first player) then:

> If the number of players is 2 enter 200 or
> If the number of players is 3 enter 300 or
> If the number of players is 4 enter 400.

This initializes the "bells and whistles" elements of the scorer.

```
7500 CLEAR
7510 INPUT "HOW MANY PLAYERS? ";Y
```

```
7513 IF Y<2 PRINT "NOT ENOUGH TO PLAY"
     : GOTO 7510
7515 IF Y>4 PRINT "TOO MANY!"
     : GOTO 7510
8000 INPUT "1ST PLAYER INIT=";F$
8005 INPUT "2ND PLAYER INIT=";G$
8007 IF Y=2 GOTO 8018
8010 INPUT "3RD PLAYER INIT=";H$
8013 IF Y=3 GOTO 8018
8015 INPUT "4TH PLAYER INIT=";I$
8017 IF Y=4 GOTO 8018
8018 PRINT "WHEN ";F$;" HAS 7 TILES IN RACK"
     : PRINT "PRESS ENTER"
8019 AA=TIME
     : LPRINT "START";AA
     : LF 1
8020 WAIT 0
     : PRINT "SCORE OF ";F$;" THIS TURN ";
8025 INPUT M
8030 IF M=200 BEEP 5
     : CLS
     : LF 1
     : BB=TIME
     : LPRINT "END";BB
     : GOTO 8115
8031 IF M=300 BEEP 5
     : CLS
     : LF 1
     : BB=TIME
     : LPRINT "END";BB
     : GOTO 12500
8034 IF M=400 BEEP 5
     : CLS
     : LF 1
     : BB=TIME
     : LPRINT "END";BB
     : GOTO 12500
8035 WAIT 0
     : PRINT
8036 N=N+M
     : M=0
8040 WAIT 50
     : PRINT "SCORE OF ";F$;" IS ";N
8045 WAIT 0
     : PRINT "SCORE OF ";G$;" THIS TURN ";
8050 INPUT O
8051 WAIT 0
     : PRINT
8055 P=P+O
     : O=0
8060 WAIT 50
     : PRINT "SCORE OF ";G$;" IS ";P
8062 IF Y=2 GOTO 8107
8065 WAIT 0
     : PRINT "SCORE OF ";H$;" THIS TURN ";
8070 INPUT Q
8071 WAIT 0
     : PRINT
8075 R=R+Q
     : Q=0
8080 WAIT 50
     : PRINT "SCORE OF ";H$;" IS ";R
8082 IF Y=3 GOTO 8108
8085 WAIT 0
     : PRINT "SCORE OF ";I$;" THIS TURN ";
8090 INPUT S
8092 WAIT 0
     : PRINT
8095 T=T+S
     : S=0
8100 WAIT 50
     : PRINT "SCORE OF ";I$;" IS ";T
     : GOTO 8110
8105 USING "####"
8107 LPRINT F$;"=";N;" ";G$;"=";P
     : GOTO 8020

8108 LPRINT F$;"=";N;" ";G$;"=";P;" ";H$;"=";R
     : GOTO 8020
8110 LPRINT F$;"=";N;" ";G$;"=";P;" ";H$;"=";R;"
     ";I$;"=";T
     : GOTO 8020
8115 U=N-P
     : LF 1
8120 IF N>P LPRINT F$;" WINS BY";U;" PTS"
     : GOTO 8150
8125 V=P-N
     : LF 1
8130 LPRINT G$;" WINS BY";V;" PTS"
     : GOTO 8150
8150 GOTO 12500
8170 IF N>P LET Z=N
     : GOTO 11065
8180 Z=P
     : GOTO 11065
10000 GOSUB 13000
10010 LPRINT "GAME TIME=";HH
10030 A$=STR$ (HH)
      : PRINT A$
10031 KK=LEN (A$)
10032 IF KK<6
      T HEN 10034
10033 GOTO 10040
10034 HH=HH+.0001
      : GOTO 10030
10040 B$=LEFT$ (A$,2)
10050 C$=RIGHT$ (A$,2)
10060 D$=MID$ (A$,3,2)
10065 LPRINT B$;"HR/";D$;"MINS/";C$;"SECS"
10066 LF 3
10070 END
10500 DIM A(Y)
10510 A(1)=N
10520 A(2)=P
10530 A(3)=R
10540 GOTO 11030
11000 DIM A(Y)
11005 A(1)=N
11006 A(2)=P
11007 A(3)=R
11009 A(4)=T
11030 Z=A(1)
11040 FOR X=2TO Y
11050 IF A(X)>Z LET Z=A(X)
11060 NEXT X
11065 LF 1
      : B$=" IS WINNER!"
11070 C$=F$+B$
11071 D$=G$+B$
11072 E$=H$+B$
11073 Q$=I$+B$
11080 DIM A$(0)*18
11085 IF N=Z GOTO 12000
11086 IF P=Z GOTO 12001
11087 IF R=Z GOTO 12002
11088 IF T=Z GOTO 12003
12000 T$=C$
      : GOTO 12010
12001 T$=D$
      : GOTO 12010
12002 T$=E$
      : GOTO 12010
12003 T$=Q$
      : GOTO 12010
12010 LT=LEN T$
      : LB=LT+6
12050 FOR L=1 TO LB
12060 A$(0)=A$(0)+CHR$ 92
12070 NEXT L
12080 SP=(18-LB)/2
12090 LPRINT TAB (SP);A$(0)
12100 LPRINT TAB (SP);CHR$ 92;CHR$ 92;" ";T$;" ";CHR$
      92;CHR$ 92
```

```
12110 LPRINT TAB (SP);A$(0)
12120 LF 1
      : GOTO 10000
12500 LF 1
      : W=N+P+R+T
12510 TEXT
12520 LPRINT " TOTAL PTS=";W
12530 LF -1
12540 GRAPH
12550 LINE (1,-5)-(185,15),0,3,B
12560 TEXT
      : COLOR 0
12570 IF Y=2 GOTO 8170
12571 IF Y=3 GOTO 10500
12572 GOTO 11000
13000 CC=DEG AA
13010 DD=DEG BB
13020 EE=DD-CC
13030 FF=DMS EE
13040 USING "##.####"
13050 GG=FF+.00005
      : LPRINT GG
13052 PRINT "AFTER ? TYPE NO. ON TAPE"
13054 INPUT "LAST NO. ON TAPE=";HH
13070 RETURN
```

Transfer Mod 100 .DO Files to OS-9

Peter Jablow
CompuServe ID 70405,471

I just finished the following program which makes the Model 100 a dumb terminal and transfers .DO files to a Color Computer running OS-9.

From line 400 up is the transfer program which you get by entering "*". The program is a compilation of parts taken from the manual on page 199, Larry Gensch's xmodem pch program, and Bill Walter's file transfer program.

```
1 ' THIS PROGRAM WILL RUN THE MODEL 100 FROM BASIC AS
    DUMB TERMINAL AND
2 ' TRANSFER A .DO FILE TO OS9. TO TRANSFER THE FILE
    ENTER "*" AT
3 ' THE ? PROMPT. USE TMODE -ECHO TO TRANSFER FILES.
5 MAXFILES = 3
20 INPUT "ENTER PHONE NO. AND <>";PH$
30 M = VARPTR(PH$)
40 AD = PEEK(M + 1) + (PEEK(M + 2)*256)
50 CALL 21200
60 CALL 21293,0,AD
70 CLS
110 OPEN "MDM:7I1D" FOR OUTPUT AS 2
120 OPEN "MDM:7I1D" FOR INPUT AS 1
130 ON MDM GOSUB 170
     : MDMON:
150 A$ = INKEY$
     : IF A$ = ""THEN 150
155 IF A$ = "*" THEN 400
160 PRINT#2,A$;
     : GOTO 150
170 MDMOFF
     : PRINT CHR$(ASC(INPUT$(1,1))AND127);
180 MDMON
     : RETURN
400 MDMOFF
     : INPUT"FILE TO SEND";F$
```

```
410 E$ = CHR$(63)
420 OPEN F$ FOR INPUT AS 3
430 IF EOF(3) THEN 510
440 LINEINPUT #3,A$
450 PRINT A$
460 PRINT #2,A$
470 B$ = INPUT$(1,1)
480 PRINT B$;
490 IF B$ <> E$ GOTO 470
500 GOTO 430
510 PRINT #2, CHR$(13)
520 CLOSE 3
530 GOTO 130
```

NEWSPAPER LABELS (From page 24)

```
      : NEXT D
      : GOTO 50
2500 CLS
      : PRINT NF$
2510 PRINT AF$
2520 PRINT"WEEKLY RATE "C
2530 RETURN
3000 CLS
      : PRINT "PAGE PRINT SUBSCRIBERS"
3010 R$=""
      : INPUT"POSITION PAPER IN PRINTER (R)";R$
3015 D$=""
      : INPUT "WHAT IS THE DATE";D$
3016 PRINT "PRINTING"
      : PRINT
3017 LPRINT "LIST OF SUBSCRIBERS
      "D$
3018 LPRINT
3020 IF R$="R" THEN 3030 ELSE 3010
3030 OPEN "R",2,"DATA/B"
3040 FIELD 2,30 AS NF$,30 AS AF$,4 AS CF$
3045 X=1
3050 GET 2,X
3060 C=CVS(CF$)
3070 CLS
      : PRINT TAB(0)XTAB(5)NF$TAB(35)AF$TAB(65)"RATE
      -";
3080 PRINT USING"$$#.##";C
3100 LPRINT TAB(0)XTAB(5)NF$TAB(35)AF$TAB(65)"RATE
      -";
3110 LPRINT USING"$$#.##";C
3120 IF X=LOF(2) THEN 3140 ELSE 3130
3130 X=X+1
      : GOTO 3050
3140 PRINT"PRINTOUT COMPLETE"
      : CLOSE 2
      : FOR D=1 TO 300
      : NEXT D
      : GOTO 50
```

```
I. M. SMART                    R. U. WISE
100 E. MAIN ST.                425 MAIN ST.
FROM 03/05/84TO 03/31/84       FROM 03/05/84TO 03/31/84
4 WEEKS @$2.89 PER WEEK        4 WEEKS @ $1.28 PER WEEK
AMOUNT DUE - $11.56            AMOUNT DUE - $5.12

A. HUCKLEBERRY                 CHERI TREE
228 TREE ST.                   404 N. FIRST ST
FROM 03/05/84TO 03/31/84       FROM 03/05/84TO 03/31/84
4 WEEKS @$2.89 PER WEEK        4 WEEKS @ $1.27 PER WEEK
AMOUNT DUE - $11.56            AMOUNT DUE - $5.08
```

Problems with COBOL Errors 98 and 94

How does one identify a crashed index? RM COBOL generates COBOL Error 98 which alerts the operator that there has been an error causing the program to terminate.

Crashed indices (which leads to data file corruption) is a problem which plagues the computer industry, be it micro, mini, or mainframe environments. Our COBOL-based products which employ the use of ISAM files are no exception in that regard.

COBOL Errors 98 and 94 are two of many messages which indicate to the operator that there has been an error, thus causing the program to terminate.

COBOL I/O Error 98 means that the operating system has detected an invalid index in the ISAM file being referenced. This indicates that the index structure of the file has been damaged.

WHAT CAN CAUSE AN ERROR 98 OR 94?

The causes of an Error 98 are varied and difficult to pinpoint. However, some of the more common causes are:

1. Pressing either the `BREAK` key or the `RESET` button during file access or at any time while the file is open (our COBOL accounting packages have the `BREAK` key disenabled).
2. Power fluctuations and/or electrical spikes which can cause the system to reboot or lock up while the file is open.
3. Defective or badly formatted media.
4. A misaligned or defective disk drive.
5. Static electricity transferred from the operator into the system.

NOTE: It is possible to have a crashed index and not be alerted to that condition by an Error 98. This can occur when the program accesses a good index which "points" back to the damaged index. The result is that the system goes into an endless loop. Should that occur, a crashed index is the most likely cause.

COBOL I/O Error 94 means that the system is unable to find and/or open the requested file.

The most common cause of an Error 94 is that the file being requested simply does not exist. Another possible cause is that the file has been password protected and the program references only the file name and not the password. Another possible cause is that the program may specify a particular drive number when referencing a file. If the file does not exist on that particular drive, an Error 94 will be the result (For example, if you are using Model II Accounts Receivable (26-4604) on a Hard Disk with TRSDOS-II, the program may attempt to access another program on Drive 1. If Floppy is On, the system will only check Floppy Drive 1 for the file. Actually, the file exists on the Hard Disk [Drive 4]).

On XENIX systems, a common cause of an Error 94 is incorrect or inappropriate permission level settings. If the user involved does not have read and/or write permission to the file, the result will always be an Error 94.

WHAT CAN BE DONE TO PREVENT ERRORS 98 AND 94?

1. Make sure all operators are thoroughly familiar with the operation of the program being run, as well as the operating system.
2. Plan your steps ahead of time. Do not venture into a program phase unless you know precisely what it is you want to accomplish.
3. If AC line conditions are suspect (e.g. spikes, brownouts, interruptions, electro-magnetic interference) the user should ask the local power company to monitor the AC line. If any problems are found, an isolated AC line (to the transformer if necessary) should be considered.
4. As a routine matter, hardware and media should be scrutinized to make sure the errors are not the result of misaligned or defective drives, or badly formatted or defective media.
5. The combination of dry winter air and carpet means static electricity. The frequency of crashed indices increases, generally speaking, during winter months (but static electricity is not confined solely to winter months).

The system should be located, ideally, in a room without carpet or in a humidity-controlled environment. Humidity control through the use of relatively inexpensive room humidifiers, coupled with other known preventive measures such as anti-static mats, can be the answer to static problems.

Anti-static carpet is now becoming more widely available and while it's a more costly solution, it is nonetheless one which is available.

HOW DOES ONE RECOVER FROM AN ERROR 98 OR 94?

1. Preparation is the key to avoiding the inconvenience and cost of lost data when an Error 98 occurs. Probably the best form of preparation is a strict program consisting of daily, weekly, and monthly BACKUPS. Keeping archive data is the ONLY reasonably safe way of preventing the cost of re-inputting data in the event an Error 98 occurs.

   Recovering from an Error 98 is difficult at best. In some instances, an ISAM recovery program can be executed which generally has a 95% chance of successfully rebuilding the index. However, it is very unwise to expect the ISAM recovery program to recover more than 70-80% of the records in the file. An ISAM recovery module should only be used in cases where BACKUPS are non-existent. DO NOT DISREGARD THE NEED FOR BACKUPS THINKING THAT AN ISAM RECONSTRUCT WILL DO THE TRICK. IT WON'T!!
2. Depending on the cause, recovering from an Error 94 is usually a fairly simple process. If the cause is a non-existent file, then the solution is to insert the proper diskette into the drive and try again. If the file has been password protected, then remove the password using the ATTRIB command.

   If the problem is permission level settings on XENIX, use the chmod command to correct this condition. For example, the command:

   chmod 777 filename

   gives read, write, and execute permission for all users in all groups. ◢

Phone Sort

Micah Altman
16213 Corbett Village Lane
Monkton, MD 21111

This program let's you set up an electronic phone book on your Model I, III, or 4 computer. The files may be edited and stored to and loaded from tape.

```
2 REM FILE SYSTEM --- ELECTRONIC PHONE BOOK
3 REM *******************************
          PHONE 1-BY JOHN SAMPLE
          EDITED AND REWORKED BY
          MICAH ALTMAN
       *******************************
4 REM  MICAH ALTMAN
       16213 CORBETT VILLAGE LANE
       MONKTON, MD 21111
       (301) 472-2810
10 CLS
       : CLEAR 10000
       : DEF INT E-Z
       : DEF STR A-D
20 DIM A1(50), A2(50), B(50), C1(50), C2(50), D(50),
   A(50), E(50)
30 CLS
       : PRINT TAB(13) "PHONE BY JOHN SAMPLE AND MICAH
       ALTMAN"
40 PRINT TAB(13) "***************>-MENU-<********
   *******"
50 PRINT TAB(13) "*
   *"
60 PRINT TAB(13) "*   TO SET UP A FILE            <1>
   *"
70 PRINT TAB(13) "*   TO SEE A FILE               <2>
   *"
80 PRINT TAB(13) "*   TO SEE ALL FILES            <3>
   *"
90 PRINT TAB(13) "*   TO CHANGE A FILE            <4>
   *"
100 PRINT TAB(13) "*   TO STORE ON TAPE           <5>
    *"
110 PRINT TAB(13) "*   TO LOAD FROM TAPE          <6>
    *"
120 PRINT TAB(13) "*   TO CHANGE FILE NAME        <7>
    *"
121 PRINT TAB(13) "*   TO PRINT DIRECTORY         <8>
    *"
122 PRINT TAB(13) "*   TO ALPHABETIZE DIRECTORY   <9>
    *"
130 PRINT TAB(13) "********************************
    ****"
135 PRINT STRING$(64,191);
140 A$=INKEY$
       : IF A$="" THEN 140 ELSE Z=VAL(A$)
150 ON Z GOTO 500, 1000, 1500, 2000, 2500, 3000,
    4000, 5000, 6000
155 GOTO 140
500 FOR L=K TO 50
       : CLS
       : PRINT STRING$(64,191)
510 PRINT TAB(13) "TO RETURN TO MENU TYPE <999>"
520 PRINT TAB(10) "TYPE IN LAST NAME ";
       : INPUT C1(L)
525 IF C1(L)="999" THEN K=L
       : C1(L)=""
       : GOTO 30
530 PRINT TAB(10) "TYPE IN FIRST NAME ";
       : INPUT C2(L)
535 PRINT TAB(10) "TYPE IN PHONE # ";
       : INPUT D(L)
540 PRINT TAB(10) "TYPE IN STREET ADDRESS ";
       : INPUT A1(L)
550 PRINT TAB(10) "TYPE IN CITY, STATE, ZIP ";
       : INPUT A2(L)
560 PRINT TAB(10) "TYPE IN BIRTH DATE ";
       : INPUT B(L)
565 IF FRE(X$)<100 THEN 650
570 FOR T=1 TO 250
       : NEXT T
580 NEXT L
650 PRINT
       : PRINT TAB(10) "OUT OF STRING$ STORAGE SPACE.
       RETURN TO MENU "
660 PRINT TAB(10) "AND STORE INFO ON TAPE"
665 INPUT Z
670 GOTO 30
1000 CLS
       : PRINT STRING$(64,191)
       : M=1
1010 PRINT TAB(13) "<<< FILE SEARCH >>>"
1020 PRINT
1030 PRINT TAB(10) "TYPE IN THE LAST NAME OF FILE
     DESIRED ";
       : INPUT C3
1040 FOR L=0 TO K
       : IF LEFT$(C3,3)=LEFT$(C1(L),3) THEN 1200
1050 NEXT L
1053 IF M=2 THEN 1060
1055 PRINT TAB(10)"----<NAME NOT IN FILE>----"
1060 PRINT
       : PRINT TAB(10)"TO SEE MENU TYPE    <1>"
1061 PRINT TAB(10) "FOR ANOTHER FILE TYPE  <2>";
       : INPUT X
1062 IF X=2 THEN 1000
1070 GOTO 30
1200 IF M=2 THEN 1210
       : CLS
       : PRINT STRING$(64,191)
1205 M=2
1210 PRINT TAB(10) "NAME          "C2(L)" "C1(L)
1220 PRINT TAB(10) "ADDRESS       "A1(L)
1230 PRINT TAB(10) "              "A2(L)
1240 PRINT TAB(10) "PHONE #       "D(L)
1250 PRINT TAB(10) "B'DATE        "B(L)
1255 PRINT
1260 GOTO 1050
1500 CLS
       : PRINT TAB(20);NM$
       : PRINT STRING$(63,191)
1510 FOR L=0 TO K
1520 PRINT, C1(L),C2(L)
1530 NEXT L
1540 PRINT TAB(10)"----<TO SEE MENU HIT ENTER>----";
1550 INPUT Z
       : GOTO 30
2000 CLS
       : PRINT STRING$(64,191)
2010 PRINT TAB(5) "ENTER THE 'LAST' NAME OF THE FILE
     YOU WISH TO CHANGE "
2015 PRINT TAB(5) "";
2020 INPUT C4
2030 PRINT TAB(5) "ENTER THE 'FIRST' NAME OF THE FILE
     YOU WISH TO CHANGE "
       : PRINT TAB(5)"";
       : INPUT C5
2040 FOR L=0 TO K
2050 IF C4=C1(L) AND C5=C2(L) THEN 2200
2060 NEXT L
2070 CLS
       : PRINT STRING$(64,191)
       : PRINT TAB(10) "---<NAME NOT IN FILE>---"
2080 PRINT TAB(10) "TO RETURN TO MENU TYPE
     <1>"
2090 PRINT TAB(10)"TO SEARCH ANOTHER NAME TYPE
     <2>";
       : INPUT X
2100 IF X=1 THEN 30 ELSE 2000
2200 CLS
```

```
         : PRINT STRING$(32,"<");
         : PRINT STRING$(32,">")
2210 PRINT "THE FILE NOW APPEARS AS FOLLOWS"
2220 PRINT "NAME    "C2(L)" "C1(L);
2230 PRINT TAB(32) "PHONE  "D(L)
2240 PRINT "ADDRESS   "A1(L),A2(L)
2250 PRINT "BIRTH DATE  "B(L)
2260 PRINT "IF THE INFO BELOW IS CORRECT HIT <ENTER>"
2270 PRINT "OTHERWISE TYPE IN CORRECT INFO"
2280 PRINT "LAST NAME   << "C1(L)" >>";
         : INPUT C1(L)
2290 PRINT "FIRST NAME   << "C2(L)" >>";
         : INPUT C2(L)
2300 PRINT "STREET ADDRESS   << "A1(L)" >>";
         : INPUT A1(L)
2310 PRINT "CITY, STATE, ZIP   << "A2(L)" >>";
         : INPUT A2(L)
2320 PRINT "PHONE   << "D(L)" >>";
         : INPUT D(L)
         : PRINT "B'DATE   << "B(L)" >>";
         : INPUT B(L)
2330 FOR T=1 TO 250
         : NEXT T
2340 CLS
         : PRINT STRING$(64,191)
         : PRINT TAB(10) "THE FILE NOW APPEARS"
2350 PRINT "NAME "C2(L)" "C1(L);
         : PRINT TAB(32) "PHONE   "D(L)
2360 PRINT "ADDRESS   "A1(L), A2(L)
2370 PRINT "BIRTH DATE " B(L)
2380 PRINT
         : PRINT TAB(10) "TO RETURN TO MENU          <1>"
2390 PRINT TAB(10) "TO MAKE MORE CHANGES        <2>";
         : INPUT Z
2400 IF Z=1 THEN 30
2410 GOTO 2000
2500 CLS
         : PRINT STRING$(64,92)
2510 PRINT TAB(10) "TO RETURN TO MENU TYPE <999> AND
     ENTER."
2520 PRINT TAB(10) "OTHERWISE PREPARE TAPE FOR FILE
     STORAGE."
2530 PRINT TAB(10) "WHEN READY TYPE <111> AND
     ENTER.";
         : INPUT Z
2540 IF Z=999 THEN 30
2550 PRINT
         : PRINT
         : PRINT TAB(10) "DATA IS NOW BEING STORED ON
     TAPE."
2555 PRINT "STORING-"; NM$
         : PRINT#-1, NM$
2560 PRINT #-1,K
2570 FOR T=0 TO K
         : PRINT #-1, A1(T), A2(T), B(T), C1(T), C2(T),
     D(T)
2580 NEXT T
2590 PRINT
         : PRINT TAB(10) "<<< DATA NOW STORED ON TAPE
     >>>"
2600 PRINT
         : PRINT TAB(10) " TO RETURN TO MENU HIT ENTER";
         : INPUT Z
         : GOTO 30
3000 CLS
         : PRINT STRING$(64,34)
3010 PRINT TAB(5) "PREPARE TAPE AND COMPUTER TO LOAD
     TAPE DATA INTO"
3020 PRINT TAB(5) "COMPUTER.   TO RETURN TO MENU TYPE
     <999> AND ENTER."
3030 PRINT TAB(5) "OTHERWISE TO LOAD DATA TYPE <111>
     AND ENTER";
         : INPUT Z
3040 IF Z=999 GOTO 30
3045 ON ERROR GOTO 10000
3050 PRINT
```

```
         : PRINT TAB(10) "<<<<-GETTING LOADED->>>>"
3055 INPUT#-1, NM$
         : PRINT "LOADING-"; NM$
3060 INPUT #-1,K
3070 FOR T=0 TO K
         : INPUT #-1, A1(T), A2(T), B(T), C1(T), C2(T),
     D(T)
3080 NEXT T
3090 PRINT TAB(10) "--->>>>>  LOADED  <<<<<---"
3100 ON ERROR GOTO 0
         : PRINT
         : PRINT
         : PRINT TAB(10) "TO RETURN TO MENU TYPE ENTER";
         : INPUT Z
         : GOTO 30
3110 END
4000 INPUT "FILE NAME";NM$
         : GOTO 30
5000 INPUT "SHORT OR LONG PRINTED DIRECTORY";A
         : IF A="L" OR A="1" THEN 5500 ELSE LPRINT
     TAB(20); NM$
         : FOR L=0 TO K
         : LPRINT C1(L), C2(L), D(L)
         : NEXT
         : GOTO 30
5500 LPRINT TAB(20); NM$
         : FOR L=0 TO K
         : LPRINT "Name:"; C2(L);" "; C1(L)
         : LPRINT "Phone:"; D(L)
         : LPRINT "Address:"; A1(L), A2(L)
         : LPRINT "Birth date:"; B(L)
         : LPRINT
         : LPRINT
         : NEXT
         : GOTO 30
6000 FOR L=0 TO K
         : N=0
         : A=STRING$(255,255)
         : FOR M=0 TO K
         : IF C1(M)<A THEN A=C1(M)
         : N=M
         : NEXT ELSE NEXT
6010 E(L)=N
         : A(L)=C1(N)
         : C1(N)=STRING$(255,255)
         : NEXT L
         : FOR L=0 TO K
         : C1(L)=A(L)
         : NEXT
         : FOR L=0 TO K
         : A(L)=A1(E(L))
         : NEXT
         : FOR L=0 TO K
         : A1(L)=A(L)
         : NEXT
         : FOR L=0 TO K
         : A(L)=B(E(L))
         : NEXT
         : FOR L=0 TO K
         : B(L)=A(L)
         : NEXT
         : FOR L=0 TO K
         : A(L)=A2(E(L))
         : NEXT
         : FOR L=0 TO K
         : A2(L)=A(L)
         : NEXT
6020 FOR L=0 TO K
         : A(L)=C2(E(L))
         : NEXT
         : FOR L=0 TO K
         : C2(L)=A(L)
         : NEXT
         : FOR L=0 TO K
         : A(L)=D(E(L))
         : NEXT
```

```
       : FOR L=0 TO K
       : D(L)=A(L)
       : NEXT
6030 GOTO 1500
10000 IF ERR/2+1=4 OR ERR/2+1=22 THEN PRINT"
      -CHECK FILE DATA-"
      : RESUME 3000 ELSE ON ERROR GOTO 0
```

Spellum

Jeff Wright
145 Roosevelt, #42
Redwood City, CA 94061

Here is a game that two or more players can play. The object of the game is to make as many points as you can by spelling words from the letters shown on the 6×6 letter board. Each letter used in a word counts as one point and, if a five or more letter word is spelled, that player gets an additional five points. But each letter board will also have a "gotcha" letter which keeps changing as the game goes along. If a player encounters a "gotcha" letter, his score is reduced by five points. The player with the most points after two rounds is the winner. There are no official rules as to the types of words which can be spelled (proper names, slang words, etc.), but I find that following the 'scrabble' word spelling rules makes the game more challenging. If you do not want to publish this in your magazine, have fun with it anyway!

```
10 REM 'SPELLUM' BY JEFF WRIGHT
20 REM   145 ROOSEVELT #42
30 REM   REDWOOD CITY, CA. 94061
40 REM
50 REM
60 REM   7/31/82
70 REM
80 REM
90 REM
100 REM        MAIN PROGRAM
110 REM
120 REM
130 REM
140 DIM A$(6, 6)
150 REM TO READ COLOR ASCII CODES
160 FOR L=1 TO 5
170 READ VEC(L)
180 NEXT L
190 CLS
200 A$=INKEY$
210 PRINT @258, "PRESS ANY KEY TO START GAME"
220 X=RND(100)
230 IF A$<>"" THEN 240 ELSE 200
240 REM
250 CLS
260 PRINT @258, "DO YOU WANT INSTRUCTIONS?"
270 Z$=INKEY$
    : IF Z$="Y" THEN GOSUB 1180 ELSE IF Z$="N" THEN
    280 ELSE 270
280 CLS
    : PRINT
    : PRINT
    : PRINT
    : PRINT
290 PRINT @258, "HOW MANY PLAYERS?"
300 Z$=INKEY$
310 IF Z$="" THEN 300
320 NUM=VAL(Z$)
330 IF NUM<2 OR NUM>5 THEN 300
340 CLS
350 FOR K1=1 TO 2
360 FOR K=1 TO NUM
370 PRINT @258, "ONE MOMENT, PLEASE....."
380 PRINT
390 REM TO GET RANDOM LETTERS FOR THE LETTER BOARD
400 FOR I=1 TO 6
410 FOR J=1 TO 6
420 X=RND(26)
430 GOSUB 830 ' TO DEFINE LETTER
440 A$(I, J)=B$
450 NEXT J
460 NEXT I
470 CLS
480 GOSUB 1360 ' TO PICK NEW GOTCHA LETTER
490 PRINT @54, C$
500 PRINT "ROUND";K1;"PLAYER";K
510 PRINT
    : PRINT
520 GOSUB 1100 ' TO PRINT TABLE
530 PRINT @352, "("COUN(K)"PTS.)"
540 GOSUB 1400
550 Z$=INKEY$
560 IF Z$="1" THEN 710
570 IF Z$=CHR$(13) THEN 670
580 IF Z$=A$(T9, T8) THEN GOSUB 1320
    : C$=""
    : GOSUB 1360
    : GOTO 690
590 FOR R=1 TO 6
    : FOR C=1 TO 6
600 IF Z$=A$(R, C) THEN 630
610 NEXT C, R
620 GOTO 550
630 C$=C$+Z$
640 A$(R, C)=CHR$(VEC(RND(5)))
650 COUN(K)=COUN(K)+1
660 GOTO 470
670 IF LEN(C$)>4 THEN COUN(K)=COUN(K)+5
    : PRINTTAB(5);"***BONUS POINTS!!***"
    : FOR X=1 TO 460*2
    : NEXT X
680 C$=""
690 CLS
700 GOTO 500
710 CLS
720 C$=""
730 NEXT K
740 NEXT K1
750 CLS
760 PRINT
    : PRINT TAB(10);"FINAL SCORES"
    : PRINT TAB(10)"------------"
770 PRINT
780 FOR V=1 TO NUM
790 PRINT "PLAYER #";V;"GOT";COUN(V);"POINTS"
800 PRINT
810 NEXT V
820 GOTO 1530
830 IF X=1 THEN B$="A"
840 IF X=2 THEN B$="B"
850 IF X=3 THEN B$="C"
860 IF X=4 THEN B$="D"
870 IF X=5 THEN B$="E"
880 IF X=6 THEN B$="F"
890 IF X=7 THEN B$="G"
900 IF X=8 THEN B$="H"
910 IF X=9 THEN B$="I"
920 IF X=10 THEN B$="J"
930 IF X=11 THEN B$="K"
940 IF X=12 THEN B$="L"
950 IF X=13 THEN B$="M"
960 IF X=14 THEN B$="N"
970 IF X=15 THEN B$="O"
980 IF X=16 THEN B$="P"
990 IF X=17 THEN B$="Q"
1000 IF X=18 THEN B$="R"
1010 IF X=19 THEN B$="S"
```

```
1020 IF X=20 THEN B$="T"
1030 IF X=21 THEN B$="U"
1040 IF X=22 THEN B$="V"
1050 IF X=23 THEN B$="W"
1060 IF X=24 THEN B$="X"
1070 IF X=25 THEN B$="Y"
1080 IF X=26 THEN B$="Z"
1090 RETURN
1100 FOR I=1 TO 6
     : FOR J=1 TO 6
1110 PRINTTAB(10);A$(I, J);
1120 PRINT" ";
1130 NEXT J
1140 PRINT
1150 NEXT I
1160 PRINT
1170 RETURN
1180 CLS
     : PRINT
1190 PRINT "THIS IS A WORD GAME 'SPELLUM'."
     : PRINT "THE OBJECT OF THE GAME IS TO"
     : PRINT "SPELL AS MANY WORDS AS YOU CAN"
     : PRINT "FROM THE 6X6 LETTER BOARD. EACH"
     : PRINT "LETTER YOU USE COUNTS AS ONE"
1200 PRINT "POINT.  IF YOU SPELL A WORD"
     : PRINT "WITH FIVE OR MORE LETTERS, THEN"
     : PRINT "YOU GET AN EXTRA FIVE POINTS."
     : PRINT "EACH BOARD WILL HAVE A HIDDEN"
     : PRINT "'GOTCHA' LETTER, AND IF YOU"
1210 PRINT "HIT IT, YOU LOSE FIVE POINTS."
1220 PRINT
     : PRINT "PRESS ANY KEY TO CONTINUE"
1230 Z$=INKEY$
     : IF Z$="" THEN 1230
1240 CLS
     : PRINT
     : PRINT "THE GAME GOES FOR TWO ROUNDS."
     : PRINT "THE FINAL SCORES ARE THEN PRIN-"
     : PRINT "TED."
     : PRINT
     : PRINT "GOOD LUCK, TOOTS!!"
1250 PRINT
     : PRINT "PRESS ANY KEY TO CONTINUE"
1260 Z$=INKEY$
     : IF Z$=""THEN 1260
1270 RETURN
1280 DATA 159, 128, 191, 239, 255
1290 FOR I=33 TO 255
1300 PRINT I"=";CHR$(I)
1310 NEXT I
1320 CLS
     : PRINT
     : PRINT
     : PRINT
     : PRINT
     : PRINT
     : PRINT
     :
1330 PRINT "SORRY, YOU HIT THE 'GOTCHA' KEY"
     : PRINT "YOU LOSE 5 POINTS"
1340 COUN(K)=COUN(K)-5
     : FOR X=1 TO 460*2
     : NEXT X
1350 RETURN
1360 REM TO MAKE A NEW T9 & T8
1370 T9=RND(6)
     : T8=RND(6)
1380 IF A$(T9, T8)=CHR$(128) OR A$(T9, T8)=CHR$(159)
     OR A$(T9, T8)=CHR$(255) OR A$(T9, T8)=CHR$(239)
     OR A$(T9, T8)=CHR$(191) THEN 1370
1390 RETURN
1400 PRINT @385, "TYPE WORD. PRESS <ENTER> WHEN"
     : PRINT " DONE, OR 1 IF YOU CAN'T GO ON"
1410 RETURN
1420 REM
1430 REM TO SAVE PROGRAM 5 TIMES
1440 REM
1450 CLS
     : PRINT @258, "CSAVING PROGRAM.  PLEASE WAIT"
1460 FOR I=1 TO 5
1470 CSAVE "SPELLUM"
1480 MOTOR ON
1490 FOR X=1 TO 460*5
     : NEXT X
1500 MOTOR OFF
1510 NEXT I
1520 PRINT @258, "CASSETTE SAVE DONE"
1530 END
```

Loan Terms for the Model 100

James H. Johnson
8370 Pippin Rd.
Cincinnati, OH 45239

This program is a conversion of the Loan Terms program by C. Giordano published in the January 1984 issue of TRS-80 Microcomputer News. When compressed the program uses 1564 bytes. When inputting the items any one of those marked with an asterisk may be computed by entering a zero. Balance may be computed by entering "Y". Enter the target period at Number of Periods. All other items, just hit ENTER for Balance.

For hard copy press the PRINT key on the Model 100.

```
10 'AMORTIZATION - BALANCE
   : N=TARGET PERIOD - PRINT COMMAND LPRINTS -
   <<LOAN.BA>>
20 CLS
   : PRINT "  ENTER <0> @ <*> TO COMPUTE AN
   ITEM"
   : PRINT
30 INPUT "  AMOUNT OF LOAN... * "; A
40 INPUT "   INTEREST RATE.... * "; I
50 INPUT "  NUMBER OF PERIODS * "; N
60 INPUT"   PERIODS PER YEAR.    "; M
70 INPUT"  AMOUNT OF PAYMENT * "; P
80 INPUT"   BALANCE   Y/ENTER    "; Q$
   : IF Q$="Y" THEN 400
90 IF A=0 THEN 100 ELSE IF I=0 THEN 110 ELSE IF
   N=0 THEN 160 ELSE IF P=0 THEN 190
100 I=I*.01/M
   : A=P*((1-(1+I)^-N)/I).
   : A=INT(A*100+.5)/100
   : GOTO 210  'A
110 ON ERROR GOTO 200
   : F=A/P
   : Q=1/F-F/(N^2)  'I
120 L=1-(1+Q)^-N
   : D=(Q*F)-L
130 Z=D/(L/Q-(N*(1+Q)^(-N-1)))
140 Q=Q-Z
   : IF ABS(Z)-10^-7=>0 THEN 120
150 I=Q*M*100
   : S=I
   : S=INT(S)+INT((S-INT(S))*100+.5)/100
   : I=S
   : I=I*.01/M
   : GOTO 210
160 ON ERROR GOTO 200
   : I=I*.01/M  'N
170 N=-((LOG(1-I*A/P))/(LOG(1+I)))
180 N=INT(N*100+.5)/100
   : GOTO 210
```

```
190 I=I*.01/M
    : P=A*(I/(1-(1+I)^-N))
    : P=INT(P*100+.5)/100
    : GOTO 210   'P
200 CLS
    : PRINT"  CANNOT BE AMORTIZED - TRY AGAIN"
    : FOR X=1 TO 900
    : NEXT
    : GOTO 20
210 J=INT((P*N)*100+.5)/100
    : W=INT((J-A)*100+.5)/100   'J,W
220 U$="$ ######.##"  'PRINT
240 CLS
    : PRINT "  AMOUNT OF LOAN... "USING U$; A
250 PRINT "  INTEREST RATE.... "USING "%
    ##.##";I*100*M
260 PRINT "  NUMBER OF PERIODS "USING "    ####";
    N
270 PRINT "  PERIODS PER YEAR. "USING "    ##";
    M
280 PRINT "  AMOUNT OF PAYMENT "USING U$; P
290 PRINT "  TOTAL PAYMENTS... "USING U$; J
300 PRINT "  TOTAL INTEREST... "USING U$; W
310 INPUT "  PRESS <PRINT> OR <ENTER TO END> "; X
    : END
400 I=I*.01/M
    : B=((A*I-P)*(1+I)^N+P)/I
    : B=INT(B*100+.5)/100   'BALANCE
410 C=INT((A-B)*100+.5)/100
    : J=INT((P*N)*100+.5)/100
    : W=INT((J-C)*100+.5)/100
420 V$="$ ######.##"
430 CLS
    : PRINT "  BALANCE....... "USING V$; B
440 PRINT "  PAID AMOUNT... "USING V$; C
450 PRINT "  TOTAL PAYMENTS "USING V$; J
460 PRINT "  TOTAL INTEREST "USING V$; W
470 PRINT
    : INPUT"  PRESS <PRINT> OR <ENTER TO END> ";
    X
    : END
```

Midwest Computer Camp for Young People

Midwest Computer Camp is the largest, independent, year around, residential camp exclusively devoted to computer education in the United States. It boasts of a staff to camper ratio of 1:4 and computer to student ratio of better than 1:1. Campers spend over six hours per day in computer training. Additional time is spent in open sessions working on programs or playing computer games. In addition outdoor recreation is available for a well-rounded camping experience.

Midwest Computer Camp is located in the center of the United States, only minutes from Indianapolis International Airport in Indiana. It is just north of Eagle Creek reservoir, in a rugged, wooded area.

While roughly half of the campers have no previous computer experience, others may attend the camp to learn assembly language or solve problems with machine graphics. Campers are distributed evenly throughout the 8-16 year age range. The percentage of girls attending the camp (roughly 30% currently) is growing steadily.

Parents and potential campers are invited to visit Midwest Computer Camp any time. The facilities are open seven days a week, year around. Any questions regarding curriculum, programs, facilities, or staffing can be answered by calling (317) 297-2700.

Fancy Print for the Model 100

James H. Johnson
8370 Pippin Rd.
Cincinnati, OH 45239-3902

This program is a take off on the "Fancy" program on page 204 in the Model 100 Manual. It LLISTs a program in the format used in *Microcomputer News*.

The file to LLIST must be in ASCII format. All colons (:) will generate a new line. For colons other than those that appear between commands, substitute a period in the ASCII file. Then on the printout make it a colon. The LPRINT CHR$(140) is Top of Form for my non-Radio Shack printer.

A Model 4 version of this program involves the changes below.

Line 20: Delete FILES
Lines 60, 90, 200, 220: Change CHR$(10) to CHR$(13)
Line 220: Change G-2 to G-1

```
10 CLS
   : CLEAR 2000 'LLIST PROGRAMS - <<LLIST.BA>>
20 PG=1
   : FILES
30 PRINT
   : INPUT "NAME OF PROGRAM ";N$
   : OPEN N$ FOR INPUT AS 1
40 LPRINT "***** LISTING OF PROGRAM *****<"N$">
   *****      ";
   : LPRINT DATE$ " - "TIME$"...Page"PG
50 LPRINT
   : LC=2
   : IF IN$=":" THEN 80
60 GOSUB 200
   : IF IN$=CHR$(10) THEN GOSUB 220
   : GOSUB 250
   : GOSUB 210
   : IF LC=56 THEN 100 ELSE 60
70 IF IN$=":" THEN GOSUB 230
   : GOSUB 250
   : IF LC=56 THEN 100
80 GOSUB 200
   : IF IN$=":" THEN GOSUB 230
   : GOSUB 240
   : IF LC=56 THEN 100 ELSE 80
90 IF IN$=CHR$(10) THEN GOSUB 220
   : GOSUB 240
   : GOSUB 210
   : IF LC=56 THEN 100 ELSE 60
100 LPRINT CHR$(140);
    : IF Z=132 THEN CLS
    : END
110 PG=PG+1
    : GOTO 40
200 IN$=INPUT$(1,1)
    : PR$=PR$+IN$
    : IF IN$<>":" AND IN$<>CHR$(10) THEN 200
    ELSE RETURN
210 IF EOF(1) AND LC=56 THEN Z=132
    : GOTO 100 ELSE IF EOF(1) THEN Z=132
    : GOTO 100 ELSE RETURN
220 G=INSTR(PR$,CHR$(10))
    : PR$=LEFT$(PR$,G-2)
    : RETURN
```

```
230 G=INSTR(PR$,":")
      : PR$=LEFT$(FR$,G-1)
      : RETURN
240 LPRINT "       : " PR$
      : PR$=""
      : LC=LC+1
      : RETURN
250 LPRINT PR$
      : PR$=""
      : LC=LC+1
      : RETURN
```

DBM, Metric Conversion, and the Tandy Logo

Eric Tremblay
215 Rue Marchette
STE-Emile De L' Energie
J0K 2K0 Quebec Canada

The first program DBM (Data Base Manager) let's you make an inventory or a shopping list or anything you can imagine. You can save or create files so you can load your inventory from tape. You can even send your list to the printer. I have found this program very useful.

```
10 ' DATA-BASE MANAGER BY
20 '
30 ' ERIC TREMBLAY
40 ' 215 RUE MARCHETTE
50 ' STE-EMILE DE L'ENERGIE
60 ' QUEBEC JOK-2KO CANADA
70 '
80 ' USE WITH 16K EXTENDED BASIC
90 ' COLOR COMPUTER
100 ' MAY BE USE BY ANYONE
110 ' WHO READS THE TRS-80
120 ' MICROCOMPUTER NEWS
130 '
140 ' THIS PROGRAM IS PRINTER
150 ' OPTIONAL.MADE TO USE WITH
160 ' THE CGP-115 COLOR PRINTER
170 '
180 PCLEAR 1
190 PMODE 0
200 CLEAR 2000
      : DIM S$(200)
210 CLS
220 PRINT "       DATA-BASE MANAGER"
230 PRINT "       -----------------"
240 PRINT
250 PRINT " PLEASE SELECT (1-8)"
260 PRINT
270 PRINT " [1] INPUT ITEMS"
280 PRINT " [2] REPLACE ITEMS"
290 PRINT " [3] ADD TO THE LIST"
300 PRINT " [4] DELETE ITEMS"
310 PRINT " [5] PRINT ALL ITEMS"
320 PRINT " [6] SAVE FILE ON TAPE"
330 PRINT " [7] LOAD FILE FROM TAPE"
340 PRINT " [8] PRINT ITEMS TO PRINTER"
350 PRINT
360 PRINT " YOUR SELECTION";
370 INPUT M
380 IF M<0 OR M>8 THEN 210
390 ON M GOSUB 410, 500, 420, 560, 680, 750, 920,
      1120
400 GOTO 210
410 Y=1
420 CLS
      : PRINT @ 8,"INPUT/ADD ITEMS"
```

```
430 PRINT @ 34, "PRESS <ENTER> WHEN FINISHED"
440 PRINT
      : PRINT "ITEM" Y;
450 INPUT S$(Y)
460 IF S$(Y)="" THEN RETURN
470 Y=Y+1
480 GOTO 440
490 N=0
500 CLS
      : PRINT @ 9, "REPLACE ITEMS"
510 PRINT @ 34, "PRESS <ENTER> WHEN FINISHED"
520 PRINT
      : INPUT "ITEM NO. TO REPLACE";N
530 IF N=0 THEN RETURN
540 INPUT "REPLACEMENT ITEM"; S$(N)
550 GOTO 500
560 N=0
570 CLS
      : PRINT @ 9, "DELETE ITEMS"
580 PRINT @ 34, "PRESS <ENTER> WHEN FINISHED"
590 PRINT
      : INPUT "ITEM TO DELETE";N
600 IF N>Y-1 THEN 590
610 IF N=0 THEN RETURN
620 FOR X=N TO Y-2
630 S$(X)=S$(X+1)
640 NEXT X
650 S$(X)=""
660 Y=Y-1
670 GOTO 560
680 FOR X=1 TO Y-1 STEP 15
690 FOR Z=X TO X+14
700 PRINT Z;S$(Z)
710 NEXT Z
720 INPUT "    PRESS <ENTER> TO CONTINUE";C$
730 NEXT X
740 RETURN
750 CLS
760 PRINT "       DATA-BASE MANAGER"
770 PRINT "       -----------------"
780 PRINT @ 136,"SAVE FILE ON TAPE"
790 PRINT @ 234,"POSITION TAPE"
800 PRINT @ 294, "PRESS PLAY AND RECORD"
810 PRINT
      : PRINT " TYPE 'STOP' FOR MAIN MENU"
820 PRINT
830 PRINT "FILE NAME (NOT MORE THEN 8 CHAR)";
840 INPUT R$
850 IF R$="" THEN 840
860 IF R$="STOP" OR R$="'STOP'"THEN 210
870 OPEN "O",#-1,R$
880 FOR X=1 TO Y-1
890 PRINT #-1,S$(X)
900 NEXT X
910 CLOSE #-1
      : RETURN
920 CLS
930 PRINT "       DATA-BASE MANAGER"
940 PRINT "       -----------------"
950 PRINT @ 135, "LOAD FILE FROM TAPE"
960 PRINT @ 235, "REWIND TAPE"
970 PRINT @ 299, "PRESS PLAY"
980 PRINT
      : PRINT " TYPE 'STOP' FOR MAIN MENU"
990 PRINT
1000 PRINT " WHAT IS THE FILE NAME TO INPUT"
1010 INPUT R$
1020 IF R$="" THEN 1010
1030 IF R$="STOP" OR R$="'STOP'"THEN 210
1040 OPEN "I",#-1,R$
1050 Y=1
1060 IF EOF(-1) THEN 1110
1070 INPUT #-1,S$(Y)
1080 PRINT S$(Y)
1090 Y=Y+1
1100 GOTO 1060
```

```
1110 CLOSE #-1
     : RETURN
1120 CLS
1130 PRINT "     PRINT LIST TO PRINTER"
1140 PRINT "     --------------------"
1150 PRINT
     : PRINT "TURN YOUR PRINTER 'ON'"
1160 PRINT
     : PRINT "INSERT YOUR PENS AND PAPER"
1170 PRINT
     : PRINT "TYPE 'STOP' FOR MAIN MENU"
1180 PRINT
     : INPUT "PRESS <ENTER> WHEN READY";VC$
1190 IF VC$="STOP" OR VC$="'STOP'" THEN 210
1200 IF PEEK(65314)<>4 THEN 1250 ELSE PRINT
     "PRINTING THE LIST ON THE CGP-115"
1210 FOR X=1 TO Y-1
1220 PRINT#-2,S$(X)
1230 NEXT X
1240 RETURN
1250 PRINT "PRINTER NOT READY"
     : FOR F=1 TO 500
     : NEXT F
     : GOTO 1120
```

The second program is called Metric. It makes metric conversion easy. You can get a printout with this also.

```
10 'METRIC CONVERSIONS BY
20 '
30 'ERIC TREMBLAY
40 '215 RUE MARCHETTE
50 'STE-EMILE DE L'ENERGIE
60 'JOK-2KO QUEBEC CANADA
70 '
80 'MAY BE USED BY ANY PERSON
90 'WHO READS THE TRS-80
100 'MICROCOMPUTER NEWS
110 '
120 'TO BE USED WITH STANDARD
130 'OR EXTENDED COLOR BASIC
140 'COLOR COMPUTER WITH 16K
150 'THIS PROGRAM MAY BE
160 'USED WITH THE CGP-115
170 CLS
     : PRINT "METRIC CONVERSION"
180 PRINT
     : PRINT "WOULD YOU LIKE A COPY          OF
     THE RESULTS ON THE PRINTER"
190 INPUT "YES OR NO";Z$
200 IF LEFT$(Z$,1)="Y" THEN X=-2 ELSE X=0
210 CLS
220 PRINT "METRIC CONVERSIONS"
230 PRINT:
240 PRINT "WHICH CONVERSIONS DO YOU WANT:"
250 PRINT:
260 PRINT "[1] LENGTH"
270 PRINT:
280 PRINT "[2] AREA"
290 PRINT:
300 PRINT "[3] WEIGHT"
310 PRINT:
320 PRINT "[4] VOLUME"
330 PRINT:
340 PRINT "[5] TEMPERATURE"
     : PRINT
350 PRINT "[6] END PROGRAM"
     : PRINT;
360 A$=INKEY$
     : IF A$="" THEN 360
370 IF A$="1" THEN GOTO 430
380 IF A$="2" THEN GOTO 1320
390 IF A$="3" THEN GOTO 1670
400 IF A$="4" THEN GOTO 2030
410 IF A$="5" THEN GOTO 2560
420 IF A$="6" THEN END
430 CLS
```

```
440 PRINT "LENGTH"
450 PRINT
460 PRINT "[1] INCHES AND MILLIMETERS"
470 PRINT
480 PRINT "[2] FEET TO METERS "
490 PRINT
500 PRINT "[3] YARDS TO METERS "
510 PRINT
520 PRINT "[4] MILES TO KILOMETERS "
530 PRINT
540 PRINT "[5] INCHES TO CENTIMETER "
     : PRINT
550 PRINT "WHICH ONE DO YOU WANT"
560 A$=INKEY$
     : IF A$=""THEN 560
570 IF A$="1" THEN GOTO 620
580 IF A$="2" THEN GOTO 760
590 IF A$="3" THEN GOTO 910
600 IF A$="4" THEN GOTO 1050
610 IF A$="5" THEN GOTO 1180
620 CLS
630 PRINT
640 PRINT#X, "CONVERT INCHES TO MILLIMETERS:"
650 PRINT
660 INPUT "HOW MANY INCHES:";I
670 PRINT#X, I;"INCHES IS";I*25.4;"MILLIMETERS"
680 PRINT:
690 PRINT#X, "CONVERT MILLIMETERS TO INCHES:"
700 PRINT
710 INPUT "HOW MANY MILLIMETERS:";M
720 PRINT#X, M;"MILLIMETERS
     IS";M*0.039370;"INCHES"
730 PRINT
     : PRINT "PRESS ANY KEY"
740 A$=INKEY$
     : IF A$=""THEN 740
750 GOTO 210
760 CLS
770 PRINT#X, "FEET TO METERS:"
780 PRINT
790 PRINT#X, "CONVERT FEET TO METERS"
800 PRINT
810 INPUT "HOW MANY FEET:";F
820 PRINT#X, F;"FEET IS";F*0.3048;"METERS"
830 PRINT
840 PRINT#X, "CONVERT METERS TO FEET"
850 PRINT
860 INPUT "HOW MANY METERS:";M
870 PRINT#X, M;"METERS IS";M*3.28084;"FEET"
880 PRINT
     : PRINT "PRESS ANY KEY"
890 A$=INKEY$
     : IF A$=""THEN 890
900 GOTO 210
910 CLS
920 PRINT#X, "YARDS AND METERS"
930 PRINT
940 PRINT#X, "CONVERT YARDS TO METERS"
950 PRINT
960 INPUT "HOW MANY YARDS:";Y
970 PRINT#X, Y;"YARDS IS";Y*0.9144;"METERS"
980 PRINT
990 PRINT#X, "CONVERT METERS TO YARDS"
     : PRINT
1000 INPUT "HOW MANY METERS:";M
1010 PRINT#X, M;"METERS IS";M*1.09361;"YARDS"
1020 PRINT
     : PRINT "PRESS ANY KEY"
1030 A$=INKEY$
     : IF A$="" THEN 1030
1040 GOTO 210
1050 CLS
1060 PRINT#X, "MILES AND KILOMETERS"
1070 PRINT
1080 PRINT#X, "CONVERT MILES TO KILOMETERS"
     : PRINT
```

```
1090 INPUT "HOW MANY MILES:";M
1100 PRINT#X, M;"MILES IS";M*1.60934;"KILOMETERS"
1110 PRINT
1120 PRINT#X, "CONVERT KILOMETERS TO MILES"
     : PRINT
1130 INPUT "HOW MANY KILOMETERS:";K
1140 PRINT#X, K;"KILOMETERS
     IS";K*0.621371;"MILES"
1150 PRINT
     : PRINT "PRESS ANY KEY"
1160 A$=INKEY$
     : IF A$="" THEN 1160
1170 GOTO 210
1180 CLS
     : PRINT
     : PRINT#X,"INCHES TO CENTIMETER"
1190 PRINT
1200 INPUT "HOW MANY INCHES:";I
1210 PRINT
1220 PRINT#X,I" INCHES IS "I*2.54" CENTIMETERS"
1230 PRINT
1240 PRINT#X,"CENTIMETERS TO INCHES"
1250 PRINT
1260 INPUT "HOW MANY CENTIMETERS:";C
1270 PRINT
1280 PRINT#X,C" CENTIMETERS IS "C*.3937" INCHES"
1290 PRINT
     : PRINT "PRESS ANY KEY";
1300 A$=INKEY$
     : IF A$=""THEN 1300
1310 GOTO 210
1320 CLS
1330 PRINT "(1) YARDS SQ. AND METERS SQ."
1340 PRINT:
1350 PRINT "(2) HECTARES AND ACRES"
1360 PRINT
     : PRINT "WHICH ONE DO YOU WANT:"
1370 A$=INKEY$
     : IF A$="" THEN 1370
1380 IF A$="1" THEN GOTO 1400
1390 IF A$="2" THEN GOTO 1540
1400 CLS
1410 PRINT#X, "SQ. YARDS AND SQ. METERS"
1420 PRINT
1430 PRINT#X, "CONVERT SQ. YARDS TO SQ. METERS "
1440 INPUT "HOW MANY SQ YARDS:";Y
1450 PRINT#X, Y;"SQ YARDS IS";Y*0.836127;"SQ
     METERS"
1460 PRINT
1470 PRINT#X, "CONVERT SQ METERS TO SQ YARDS"
1480 PRINT
1490 INPUT "HOW MANY SQ METERS:";M
1500 PRINT#X, M;"SQ METERS IS";M*1.19599;"SQ
     YARDS"
1510 PRINT
     : PRINT "PRESS ANY KEY"
1520 A$=INKEY$
     : IF A$="" THEN 1520
1530 GOTO 210
1540 CLS
1550 PRINT#X, "ACRES AND HECTARES"
1560 PRINT
1570 PRINT#X, "CONVERT ACRES TO HECTARES"
     : PRINT
1580 INPUT "HOW MANY ACRES:";A
1590 PRINT#X, A;"ACRES IS";A*0.404686;"HECTARES"
1600 PRINT
1610 PRINT#X, "CONVERT HECTARES TO ACRES"
     : PRINT
1620 INPUT "HOW MANY HECTARES:";H
1630 PRINT#X, H;"HECTARES IS";H*2.47106;"ACRES"
1640 PRINT
     : PRINT "PRESS ANY KEY"
1650 A$=INKEY$
     : IF A$=""THEN 1650
1660 GOTO 210
```

```
1670 CLS
1680 PRINT
1690 PRINT "(1) OUNCES AND GRAMS"
1700 PRINT:
1710 PRINT "(2) POUNDS AND KILOGRAMS"
1720 PRINT
     : PRINT "WHICH ONE DO YOU WANT:"
1730 A$=INKEY$
     : IF A$="" THEN 1730
1740 IF A$="1" THEN GOTO 1760
1750 IF A$="2" THEN GOTO 1890
1760 CLS
1770 PRINT#X, "OUNCES AND GRAMS"
1780 PRINT
1790 PRINT#X, "CONVERT OUNCES TO GRAMS"
     : PRINT
1800 INPUT "HOW MANY OUNCES:";O
1810 PRINT#X, O;"OUNCES IS";O*28.3495;"GRAMS"
1820 PRINT
1830 PRINT#X, "CONVERT GRAMS TO OUNCES"
     : PRINT
1840 INPUT "HOW MANY GRAMS:";G
1850 PRINT#X, G;"GRAMS IS";G*0.035274;"OUNCES"
1860 PRINT
     : PRINT "PRESS ANY KEY"
1870 A$=INKEY$
     : IF A$="" THEN 1870
1880 GOTO 210
1890 CLS
1900 PRINT#X, "POUNDS AND KILOGRAMS"
1910 PRINT
1920 PRINT#X, "CONVERT POUNDS TO KILOGRAMS"
     : PRINT
1930 INPUT "HOW MANY POUNDS:";P
1940 PRINT#X, P;"POUNDS
     IS";P*0.453592;"KILOGRAMS"
1950 PRINT
1960 PRINT#X, "CONVERT KILOGRAMS TO POUNDS"
1970 PRINT
1980 INPUT "HOW MANY KILOGRAMS:";K
1990 PRINT#X, K;"KILOGRAMS IS";K*2.20462;"POUNDS"
2000 PRINT
     : PRINT "PRESS ANY KEY"
2010 A$=INKEY$
     : IF A$="" THEN 2010
2020 GOTO 210
2030 CLS
2040 PRINT "VOLUME"
2050 PRINT:
2060 PRINT "(1) CUB. YARDS AND CUB. METERS"
2070 PRINT
2080 PRINT "(2) QUARTS AND LITERS"
2090 PRINT
2100 PRINT "(3) GALLONS AND LITERS":PRINT
2110 PRINT "WHICH ONE DO YOU WANT ?"
2120 A$=INKEY$
     : IF A$="" THEN 2120
2130 IF A$="1" THEN GOTO 2160
2140 IF A$="2" THEN GOTO 2300
2150 IF A$="3" THEN GOTO2430
2160 CLS
2170 PRINT#X, "CUB.YARDS AND CUB.METERS"
2180 PRINT
2190 PRINT#X, "CONVERT CUB.YARDS TO CUB.METERS"
2200 PRINT
2210 INPUT "HOW MANY CU YARDS:";Y
2220 PRINT#X, Y;"YARDS
     IS";Y*0.764555;"CUB.METERS"
2230 PRINT
2240 PRINT#X, "CONVERT CUB.METERS TO CUB.YARDS"
     : PRINT
2250 INPUT "HOW MANY CUB.METERS:";M
2260 PRINT#X, M;"CUB.METERS
     IS";M*1.30795;"CUB.YARDS"
2270 PRINT
     : PRINT "PRESS ANY KEY"
```

```
2280 A$=INKEY$
     : IF A$="" THEN 2280
2290 GOTO 210
2300 CLS
2310 PRINT#X, "QUARTS AND LITERS"
2320 PRINT
2330 PRINT#X, "CONVERT QUARTS TO LITERS"
     : PRINT
2340 INPUT "HOW MANY QUARTS:";Q
2350 PRINT#X, Q;"QUARTS IS";Q*0.946353;"LITERS"
2360 PRINT
2370 PRINT#X, "CONVERT LITERS TO QUARTS"
     : PRINT
2380 INPUT "HOW MANY LITERS:";L
2390 PRINT#X, L;"LITERS IS";L*1.05669;"QUARTS"
2400 PRINT
     : PRINT "PRESS ANY KEY"
2410 A$=INKEY$
     : IF A$=""THEN 2410
2420 GOTO 210
2430 CLS
2440 PRINT
     : PRINT#X,"GALLONS TO LITERS"
     : PRINT
2450 INPUT "HOW MANY GALLONS:";G
2460 PRINT
     : PRINT#X,G" GALLONS IS "G*4.546" LITERS"
2470 PRINT
2480 PRINT#X, "LITERS TO GALLONS"
2490 PRINT
2500 INPUT "HOW MANY LITERS:";L
2510 PRINT
2520 PRINT#X,L" LITERS IS "L*.22" GALLONS"
2530 PRINT
     : PRINT "PRESS ANY KEY";
2540 A$=INKEY$
     : IF A$=""THEN 2540
2550 GOTO 210
2560 CLS
2570 PRINT
     : PRINT#X,"FAHRENHEIT TO CELSIUS"
2580 PRINT
2590 INPUT "HOW MANY FAHRENHEIT:";F
2600 PRINT
2610 PRINT#X,F" DEGREES FAHRENHEIT IS
     "(F-32)*5/9" DEGREES CELSIUS"
2620 PRINT
2630 PRINT#X, "CELSIUS TO FAHRENHEIT"
2640 PRINT
2650 INPUT "HOW MANY DEGREES CELSIUS:";C
2660 PRINT
2670 PRINT#X,C" DEGREES CELSIUS IS "(C*9/5)+32"
     DEGREES FAHRENHEIT"
2680 PRINT
     : PRINT "PRESS ANY KEY";
2690 IF INKEY$=""THEN 2690
2700 GOTO 210
```

The final program draws the Tandy Corporation logo. I find it interesting, because every time I open a book or catalog from Radio Shack, I see this logo on the big business computers. So I decided to make this logo for the Color Computer.

```
10 'THE TANDY CORPORATION LOGO
20 '
30 'ERIC TREMBLAY
40 '215 RUE MARCHETTE
50 'STE-EMILE DE L'ENERGIE
60 'JOK-2KO QUEBEC CANADA
70 '
80 'MAY BE USED BY ANY PERSON
90 'WHO READS THE TRS-80
100 'MICROCOMPUTER NEWS
110 '
120 'TO USE WITH ANY STANDARD
130 'OR EXTENDED COLOR BASIC
140 'COLOR COMPUTERS
150 'YOU MAY PRINT THIS LOGO
160 'ON, OR HAVE A HARDCOPY
170 'ON THE CGP-115
180 '
190 CLS
200 PRINT "THE TANDY CORPORATION LOGO"
210 PRINT "--------------------------"
220 PRINT
     : PRINT "DO YOU WANT THE LOGO TO BE
     DRAWN ON THE SCREEN OR ON        THE PRINTER"
230 PRINT
240 INPUT "ENTER SCREEN OR PRINTER (S/P)";C$
250 IF LEFT$(C$,1)="S" THEN X=0
260 IF LEFT$(C$,1)="P" THEN X=-2
270 IF LEFT$(C$,1)="S" THEN A$=CHR$(128) ELSE
     A$=CHR$(127)
280 B$=" "
290 CLS
300 PRINT#X, B$; B$; B$; B$; A$; A$; A$; A$; A$;
     A$; A$; A$; A$; A$; A$; A$; A$; A$; A$;
     A$; A$; A$; A$; A$; A$; A$; A$
310 PRINT#X ,B$; B$; B$; B$; B$; A$; A$; A$; A$;
     B$; B$; B$; B$; B$; B$; B$; B$; B$; B$;
     B$; B$; B$; A$; A$; A$; A$
320 PRINT #X, B$; B$; B$; B$; B$; A$; A$; A$;
     A$; A$; A$; A$; A$; B$; B$; A$; A$; A$;
     A$; A$; A$; A$; A$; A$
330 PRINT #X, B$; B$; B$; B$; B$; B$; B$; A$; A$;
     A$; A$; A$; A$; A$; B$; B$; A$; A$; A$;
     A$; A$; A$; A$
340 PRINT#X, B$; B$; B$; B$; B$; B$; B$; B$; A$;
     A$; A$; A$; A$; A$; B$; B$; A$; A$; A$;
     A$; A$; A$; A$
350 PRINT#X, B$; B$; B$; B$; B$; B$; B$; B$; B$;
     A$; A$; A$; A$; A$; B$; B$; A$; A$; A$;
     A$; A$
360 PRINT#X, B$; B$; B$; B$; B$; B$; B$; B$; B$;
     B$; A$; A$; A$; A$; B$; B$; A$; A$; A$;
     A$; A$
370 PRINT#X, B$; B$; B$; B$; B$; B$; B$; B$; B$;
     B$; B$; A$; A$; A$; A$; A$; A$; A$; A$;
     A$
380 PRINT#X, B$; B$; B$; B$; B$; B$; B$; B$; B$;
     B$; B$; A$; A$; A$; A$; A$; A$; A$; A$;
     A$
390 PRINT#X, B$; B$; B$; B$; B$; B$; B$; B$; B$;
     B$; A$; B$; B$; B$; B$; B$; B$; B$; B$;
     B$; A$
400 PRINT#X, B$; B$; B$; B$; B$; B$; B$; B$; B$;
     A$; A$; B$; B$; A$; A$; A$; A$; A$; A$; B$;
     B$; A$; A$
410 PRINT#X, B$; B$; B$; B$; B$; B$; B$; B$; A$;
     A$; A$; B$; B$; A$; A$; A$; A$; A$; A$; A$;
     A$; A$; A$; A$
420 PRINT#X, B$; B$; B$; B$; B$; B$; B$; A$; A$;
     A$; A$; B$; B$; A$; A$; A$; A$; A$; A$; A$;
     A$; A$; A$; A$
430 PRINT#X, B$; B$; B$; B$; B$; B$; A$; A$; A$;
     A$; A$; B$; B$; A$; A$; A$; A$; A$; A$; B$;
     B$; A$; A$; A$; A$; A$
440 PRINT#X, B$; B$; B$; B$; B$; A$; A$; A$; A$;
     A$; A$; B$; B$; B$; B$; B$; B$; B$; B$; B$;
     B$; A$; A$; A$; A$; A$; A$
450 PRINT#X, B$; B$; B$; B$; A$; A$; A$; A$; A$;
     A$; A$; A$; A$; A$; A$; A$; A$; A$; A$; A$;
     A$; A$; A$; A$; A$; A$; A$;
460 FOR V=1 TO 4000
     : NEXT V
470 PRINT#X, " "
480 GOTO 190
```

Easy Label

James R. Demers
**77 Springfield St.
icopee, MA 01013

Use this program and your CoCo to make one label at a time or as many labels as you want of the same name and address. Multiple labels should be done first. The main feature is allowing you to print labels using the same street, city, state zip with the touch of one key or to change just one of the entries. The old entry will show on the screen so you can keep track of what the last entry is.

```
170 GOTO 650
180 CLEAR 200
    : CLS
    : PRINT
    : PRINT
190 PRINT "PUT LABELS IN PRINTER"
    : PRINT
200 PRINT "ADJUST LABELS TO PRINTER HEAD"
210 PRINT
    : PRINT "ENTER THE INFORMATION ASKED FOR"
    : PRINT
220 PRINT "TYPE (END) AT NAME PROMPT TO END"
230 SOUND 200,2
    : PRINT
    : PRINT
    : PRINT
    : LINE INPUT "ENTER NAME ";NA$
    : PRINT
240 IF NA$="END" THEN END
250 SOUND 200,2
    : LINE INPUT "HOUSE NUMBER ";HN$
    : PRINT
260 SOUND 200,2
    : LINE INPUT "STREET NAME ";SN$
    : PRINT
270 SOUND 200,2
    : LINE INPUT "CITY NAME ";CT$
    : PRINT
280 SOUND 200,2
    : LINE INPUT "STATE NAME ";ST$
    : PRINT
290 SOUND 200,2
    : LINE INPUT "ZIP CODE ";ZP$
300 PRINT "WOULD YOU LIKE MORE THEN ONE    LABEL
    LIKE THIS? Y/N"
310 SOUND 50,2
320 B$=INKEY$
    : IF B$="" THEN 320
330 IF B$ ="Y" THEN PRINT "HOW MANY WOULD LIKE?"
    ELSE 540
340 INPUT C
    : GOTO 580
350 SOUND 200,2
    : CLS
    : PRINT
    : PRINT
    : LINE INPUT "ENTER NAME ";NA$
    : PRINT
360 IF NA$="END" THEN END
370 SOUND 200,2
    : LINE INPUT "HOUSE NUMBER ";HN$
    : PRINT
380 PRINT "IS street NAME THE SAME AS   LAST ENTRY
    ("SN$")"
    : PRINT "Y/N"
    : PRINT
390 S$=INKEY$
    : IF S$="" THEN 390
400 IF S$="Y" THEN GOTO 420 ELSE 410
410 LINE INPUT "STREET NAME ";SN$
    : PRINT
420 PRINT "IS CITY NAME THE SAME AS LAST    ENTRY
    ("CT$")"
    : PRINT"Y/N"
    : PRINT
430 C$=INKEY$
    : IF C$="" THEN 430
440 IF C$= "Y" THEN GOTO 460 ELSE 450
450 LINE INPUT "CITY NAME ";CT$
    : PRINT
460 PRINT "IS STATE NAME THE SAME AS    LAST ENTRY
    ("ST$")"
    : PRINT "Y/N"
    : PRINT
470 S2$=INKEY$
    : IF S2$ ="" THEN 470
480 IF S2$= "Y" THEN 500 ELSE 490
490 LINE INPUT "STATE NAME ";ST$
    : PRINT
500 PRINT "IS ZIP CODE THE SAME AS LAST     ENTRY
    ("ZP$")"
    : PRINT "Y/N"
510 Z$=INKEY$
    : IF Z$="" THEN 510
520 IF Z$="Y" THEN 540 ELSE 530
530 LINE INPUT "ZIP CODE ";ZP$
540 PRINT#-2," ";NA$
550 PRINT#-2," ";HN$" ";
    : PRINT#-2,;SN$
560 PRINT#-2," ";CT$;
    : PRINT#-2,",";
    : .PRINT#-2,;" "ST$;
    : PRINT#-2," ";ZP$
570 PRINT#-2
    : PRINT#-2
    : PRINT#-2
    : GOTO 350
580 FOR D=1 TO C
590 PRINT#-2," ";NA$
600 PRINT#-2," ";HN$" ";
    : PRINT#-2,;SN$
610 PRINT#-2," ";CT$;
    : PRINT#-2,",";
    : PRINT#-2,;" "ST$;
    : PRINT#-2," ";ZP$
620 PRINT#-2
    : PRINT#-2
    : PRINT#-2
630 NEXT D
640 GOTO 230
650 PMODE 0,1
    : PCLEAR 1
660 CLS(0)
    : RESTORE
    : FOR I=1 TO 5
    : READ C,X,Y
    : FOR J=X TO Y
    : PRINT CHR$(C);
    : NEXT J
    : NEXT I
670 PRINT@42," 10/1/83 ";
    : PRINT@196,"                      ";
    : PRINT@228, " THE COLOR EASY LABEL  ";
    : PRINT@260,"                      ";
680 PRINT@386, "WRITTEN BY JAMES R. DEMERS.";
690 PRINT@416, "377 SPFLD.ST. CHICOPPE,MA 01013"
700 FOR XX=1 TO 3000
    : NEXT
    : GOTO 180
710 DATA 159, 0, 95, 255, 96, 191, 191, 192, 287,
    239, 288, 383, 175, 384, 479
```

Communications Corner

by Al and Dru Simon

Hello, and welcome back again! As promised, this month we'll be continuing our discussion of computer system security.

One method we wanted to expound upon is ENCRYPTION of data for added system security. Encryption is the encoding of data in a manner such that only a user with a "key" can decode the data.

A small sample of an extremely simple encryption program is below.

```
10 REM Peel off your data one string at a time here
      and put it in A$.
20 FOR X = 1 TO LEN(A$)
30 M$ = MID$ (A$,X,1)
40 REM N = Key number.
50 EC = ASC(M$)+N
60 IF EC>255 THEN EC = EC-255
70 R$=R$+CHR$(EC)
80 NEXT X
90 REM Your original input string contained in A$ is
      now encoded and contained in R$. Continue your
      program from here.
```

The above program encodes your data. The following program decodes it again.

```
10 REM Input your string as above in A$
20 FOR X = 1 TO LEN(A$)
30 M$ = MID$(A$,X,1)
40 DC = ASC(M$)-N 'REM N as above
50 If DC < 0 THEN DC = DC + 255
60 D$ = D$ + CHR$(DC)
70 NEXT X
80 REM Your decoded string is now contained in D$
```

The above is just one simple example of encryption. It may take many other forms based upon hash codes or random numbers or many other things. It's a simple and easy technique to ensure your data integrity and can be easily inserted in the middle of any program.

The same thing can be accomplished in Machine Language, but it is a lot easier because a character may be offset by a large hex number or contents of a location which is fed by external programs (e.g. A date-setting routine which also would input the Key either by self-generation on a daily basis or by operator input.)

That, in a nutshell, is encryption.

DIALBACKS are another effective security measure. A piece of hardware may be attached between the modem and the telephone which will intercept incoming calls and ask for a password (individually assigned to each authorized person). Upon receiving this password the callback would dial back the assigned phone number which would be, of course, the phone number of the authorized user.

If necessary, a second "black box" can be installed just to receive passwords to pass through to the modem. This would be a fixed (or system) password that would further ensure integrity and security. After passing those levels a normal security can then be maintained at login to the computer.

The key to making this system foolproof is the assignment of individual passwords to individual users and holding them fully responsible for the use of that password and encouraging them to change their passwords frequently for added security.

These are two of the most effective means we're aware of to ensure security against hackers or other deliberate attempts at breaking into publicly accessed business systems.

For bulletin boards, which may actually be a more "prime" target for hackers, since the risks of being caught are far less, there is a very effective way to maintain security, and it is the method we use on Drucom.

Our program set includes a machine language interrupt routine which resides right before the keyboard scan routine. This program vectors the bulletin board program off in order to check for any waiting interrupt. If it finds that the machine has been requested to return to any command level (i.e. BASIC ready, DOS prompt or DEBUG), it locks out all remote input. Therefore, though it is possible that the program might be broken out of, it affords the hacker no profit, since all he can do at that point is stare ineffectually at the cursor and prompt.

As soon as the hacker hangs up, the system automatically reboots when it looses the carrier and is returned to its ready state, waiting for the next caller.

We hope the above methods for added security will prove helpful in maintaining your own systems, whether business oriented or private. If you have any further questions about computer system security, please don't hesitate to write to us!

THE NEWS FROM PLUMB

The latest issue of Ric Manning's publication has some articles in it that are very interesting! Read on!

*A proposed new anti-hacker law making its way through the Wisconsin legislature could put many of the state's bulletin boards out of business.

Assembly Bill 695 was proposed last fall in reaction to the activities of the 414s, a group of Milwaukee teenagers whose members gained access to a computer at the Los Alamos National Laboratory and the Sloane-Kettering cancer-research center in New York.

The law is based on a model statute drafted by the National Association for State Information Systems are enacted by several states. It makes it illegal to access a

computer system without authorization, modify programs, or convert them to personal use.

"We were concerned mostly about people writing welfare checks to themselves or tapping into electronic fund transfer systems," said Carl Vorlander, the group's executive director. "I doubt if we even thought about the hacker when we wrote that."

Wisconsin lawmakers would expand the statute to make it a felony to create a life-threatening situation through the destruction or alteration of computer records. Violators could get 10 years in prison and be fined up to $10,000.

But the legislators' blunderbus approach would also penalize BBS sysops who grant freedom of speech to their callers. The law would allow civil action against anyone who provides "information which may enable another person to gain unauthorized access to data" — regardless of whether anyone actually uses the information. And it would legalize police wiretaps to find violators.

Dick White, executive assistant to the Wisconsin Assembly Majority Leader, said the law is targeted at "the seemingly innocuous illegal computer break-ins that potentially may create a major risk over the next decade."

But Geoffrey Goodfellow, a computer consultant for SRI International, said the provisions relating to personal computer and bulletin boards seemed rather "Orwellian." And Vorlander said he thought the legislators were "stretching things a bit."

Some Wisconsin computer hobbyists think the law is a blatant invitation to invade personal privacy and an assault on the First Amendment. By the Wisconsin Assembly's logic, the publisher of a mystery novel could be shut down because the plot could contribute to a crime.

Don Schofield, a Madison, WI computer hobbyist, is organizing a lobbying group to fight the bill. He has a package of material on the issue that he'll mail to anyone who sends $3 to cover his expenses to Box 3332, Madison, WI 53704.

The bill itself is available in a text file on the ABSURD BBS in Madison. The 24 hour board is online at 608 256 3606.

But be careful what you post — Big Brother may be listening.

*KGON-FM, a Portland, OR rock station is the sponsor of the country's first Rock BBS. The KGON Rock and Roll Computer is a Radio Shack CoCo that allows users to send messages to each other. But the real appeal of the system is its library of rock facts and figures.

The Rockfiles include such material as Portland's top LPs, new records on the charts, top selling video games and software, concert reviews, top video rentals, the worldwide top 25 records, the Portland-area concert calendar, new videos added to MTV, album reviews and the station's top five requests.

*Quick Soft, a Buenos Aires software firm, has plugged into the international modem network with a TRS-80 Model 1 and a Spanish translation of the INFOEX-80 BBS software. Actually you don't have to know much Spanish to use the system.

The system operator is Manuel J. Moguilevsky and he welcomes calls and advice. To call the BBS, dial 011 (for International access) then 54-1-46-5236.

*The Forth BBS in Hayward, CA, is devoted to users and explorers of the Forth computer language. A good example of Forth in action is the Conference-Tree software itself. It's available at 415-538-3580.

*Steve Johnson is kicking around the idea of a BBS to serve users of the Radio Shack Model 100 computer. He wants to hear from people who would use the board and perhaps pay a few dollars to support it. You can catch Steve on The Source, TCP261.

There's plenty more news where that came from! If you'd like to contact Ric Manning about PLUMB magazine, he can be reached c/o Riverside Data Inc, P.O. Box 300 Harrods Creek, KY 40027 or through CompuServe account #72715,210 or the Source account #STQ007.

THE CORNER MAILBOX

Dear Al & Dru,
. . . In the November issue . . . you mention that "there are various BBS software packages available . . ."

I am standardizing on a TRS-80, Models 12 and 16 and want very much to initiate a BBS in perhaps a rather creative way. Can you please let me have some guidance on where such packages are available? I am willing to operate under 4.2 DOS or Xenix.

What is Drucom? How does one access Drucom . . . etc?
William P. Taggart
Lafayette, NJ

Dear Bill,
Thanks for asking about Drucom. That's the BBS that we own and operate in southeast Pennsylvania. To access it, dial (215) 855-3809, using 300 baud, 8 bit words, 1 stop bit and disabled parity.

About BBS software packages, the only one that we're aware of is the one that we wrote about in a previous issue, BIG TOP in Milwaukee, WI. I suggest you contact the sysop, Neal Patrick, who is willing and eager to distribute his work for a more than reasonable fee. The software for the Model II will also work on the 12 and 16.

If you wish, you can also do a remote mail option with Xenix or write your own program to handle the input, but the communications will be handled by the Xenix itself, i.e. the remote operation.
Al & Dru

Dear Al & Dru,
I have all the equipment to use a computer bulletin board system, but I have one problem. Since I don't want to run up phone bills, I would like to call systems only in the Dallas area. . . . I was hoping that you would know of some and print them in your column.
Curt Conister
Garland, TX

Dear Curt,
Digging into my trusty issue of PLUMB I come up with a number of BBSs in your part of Texas. Most BBSs have further listings of other boards, so I suggest that you call these and peruse them for more numbers

Computalk	Fort Worth	817 595 3195
RBBS-PC	Desoto	214 223 0983
Software Exchange	Dallas	214 248 7757
Microserve	Hawkins	214 769 3036

Al & Dru

Dear Al & Dru,

I currently own a Model III with two double density drives, a tape recorder, a sound amplifier, and Modem I, a Line Printer I and many, many disks. I will soon be upgrading to a Modem II to get the AutoAnswer/Dial features in order to run my BBS. I have built up my computer needs too much not to use my computer any more, but I cannot do anything with my computer when it is online. Do you know of any way to get Multi-User for a Model III? Also, do you know of anyway to get Multi-User on the Model 4?

D.H.
Orange, CT

Dear D.H.,

We're not aware of any multi user system for the Model III or 4, however there are at least six free BBS systems available for prospective sysops, though in order to prevent mass BBSs from springing up, we will not publish the names or availability of these. If anyone is seriously interested, leave a message about it on Drucom, but first please read our articles about running a BBS. (Nov. 83 and Jan. 84)

Al & Dru

Dear Al & Dru,

Every so often I use a dialing service (Sprint, MCI, etc.) to access a BBS. On some boards I have relatively few problems. When I log onto a Network board which uses a particular modem I can't seem to get a single line unmangled. Is there a 'signal strength' for modems? Do they operate at different volume levels? Or is it just bad luck?

John M. Morrison
Philadelphia, PA

Dear John,

When using Sprint or MCI, etc., you may experience either a low line level (caused by atmospheric conditions or sunspots etc. interfering with the microwave transmissions) or echo which is even worse for transmission, because rather than getting a dropout, you're getting an echo which causes you to receive strange characters which were really never sent. Some modems as well are very line-level sensitive. There's a certain amount of checking and requesting for resend that can be done by any modem or software and is done by most software, but if it reaches a level of intolerance, you will get garbage.

Al & Dru

Gentlemen,

I recently added a Modem I to my Model II and need a bit of help in setting up a protocol.

I have a CP/M based, non-Radio Shack computer at home and a Model II with TRSDOS 4.2.0 here at the office. Being one who takes his work home with him, I frequently will write a letter or program at home, which I intend to keep on the office hard disk. Also, I may write something at the office, then transfer it to my home computer for further editing. I can see four ways in which this can be done:

1. Carry my home computer to the office and direct connect the two machines for communications. I do this frequently, but lugging it around is not a lot of fun.

2. Have someone from my family run the home computer while I'm at the office and communicate through the modems. I can do this when there's someone home, willing to put the time into MY work.

3. The method I'd like to try: Upload from my home computer to a local BBS (the SYSOP said I was welcome to do it) then download to the office in the morning. Then reverse the process in the afternoon. While I've had perfect success in uploading and downloading from my CP/M machine, I cannot get the TRS-80 to upload or download because TERMINAL and Receive use different protocols than the XMODEM used by the BBS for file transfers. Using TERMINAL I can communicate ASCII characters interactively with the BBS but the "missing link" is the establishment of XMODEM protocol on the TRS-80.

4. A method as yet untried: Set up my home computer as an RBBS or Remote CP/M system and leave the Modem set for auto answer. Then I could call home in the morning, download the previous evenings efforts, and at the end of the day call home again to upload the planned work for the evening. To do this I would need RCP/M software for home use. Then I believe that TRSDOS' Terminal function would work for office use, OR I may need the same "missing link" as described above.

Any ideas where I can find the "missing link", that is, the programs which will send or receive from a TRSDOS based machine and be compatible with the ever-popular XMODEM protocol used by BBS's?

Or must I write my own? If such software already exists, I surely would hate to re-invent the wheel.

John Stein
St Paul, MN

Dear John,

You certainly offer a lot of options. Let's take them one by one. For mass transfers your method 1 is probably the best because it can be accomplished at 9600 baud or higher, thus cutting transfer time down to almost nothing.

Method 2 does seem somewhat impractical.

Method 3 very similar to method 4 in that XMODEM or MODEM7 or Christiansen Protocol or CPMUG or MODEM (all of these the same protocol) may be used for both, or you may tell an RCPM system to type the file, in which case it will come to you in ASCII. If it's a source file for compilation the compilation can be done on your computer or if the local BBS uses an Intel hex file (an ASCII representation of hex) you may use the two programs we published in the Feb. 84 article.

Last but not least, XMODEM (also known by any of the names we mentioned above) IS available for Z-80 (which includes the TRS-80 Models I, II, III, 4, 12, and 16).

Check out some of your local BBSs and you'll probably find it there under one of the names we used above. We will attempt to have a copy available in the Drucom download section by the time this article is published. (We have to dig through a bunch of un-catalogued discs to find it!)

We hope that this will solve your problem for you. Thanks for writing.

Pie Chart for the 2000

Earl Bollinger
Fort Worth, TX

I developed the following program to allow for the displaying of piecharts on a Model 2000. This program is written in BASIC and takes full advantage of the Model 2000's color graphics abilities. Although the program will permit you to enter up to sixteen separate items and their percentages for one chart, it is best to limit your entries to a maximum of seven. More than seven or eight items give the chart a cluttered appearance. Also you are limited to seven colors for the piechart (background is the eighth color), and more than seven items will force you to have to repeat the color sequence. The program is small and can be easily modified for inclusion in other bigger programs.

```
10 REM PIECHART GENERATOR PROGRAM -- BY E.W.BOLLINGER
      OCTOBER 1983
20 KEY OFF
30 DIM ITEM$(16), PER(16)
40 CLS
50 SCREEN 3
60 COLOR 0,1
70 ZX=.0175329
80 DIAM=150
90 X=200
   : Y=200
100 PRINT "        Pie Chart Generator Program
      Ver. 1.0"
110 PRINT
120 PRINT "Enter Chart Title Message Line: "
130 LINE INPUT "-->";TITLE$
140 PRINT
150 PRINT "    You can enter up to 16 items. Total
      percentage should add up"
160 PRINT "to 100%, but cannot exceed 100%. You are
      prompted for each item"
170 PRINT "entered. You enter an item description and
      percentage value like"
180 PRINT "so:  ITEM # 1:--> PORK BELLIES,13 <enter>"
190 PRINT "     ITEM # 2:--> SIRLOIN,23 <enter>"
200 PRINT "     ...and so on..."
210 PRINT
220 PRINT "When done enter a COMMA only then hit the
      ENTER key."
230 PRINT
240 PRINT
250 NUM=1
260 PRINT "Enter ITEM #";NUM;"--> ";
270 INPUT ITEM$(NUM),PER(NUM)
280 IF ITEM$(NUM)="" THEN GOTO 310
290 NUM=NUM+1
   : IF NUM>16 THEN GOTO 310
300 GOTO 260
310 NUM=NUM-1
320 CLS
330 '
340 '
350 REM ***** actual Chart Plotting Section *****
360 '
370 A=0
380 B=1
   : CLR=1
390 FOR N=1 TO NUM
400    A=B+1
410    B=B+(PER(N)*3.6)
420    IF N=NUM THEN GOSUB 580
430    I=X
   : J=Y
440    GOSUB 650
450    CLR=CLR+1
   : IF CLR>7 THEN CLR=1
460 NEXT N
470 LOCATE 2,5
   : PRINT TITLE$;
480 ROW=4
   : COL=50
490 CLR=1
   : PIX=16
500 FOR N=1 TO NUM
510    CIRCLE(380,ROW*PIX+6),7,CLR:
      PAINT(380,ROW*PIX+6),CLR
520    LOCATE ROW+1,COL
   : PRINT PER(N);"% ";ITEM$(N);
530    CLR=CLR+1
   : IF CLR>7 THEN CLR=1
540    ROW=ROW+2
550 NEXT N
560 CMD$=INKEY$
   : IF CMD$="" THEN GOTO 560
570 END
580 IF B>350 THEN B=359
590 RETURN
600 '
610 '
620 REM ***** Pie Chart Portion Plotting Subroutine
      *****
630 '
640 '
650 P=I+DIAM*SIN(A*ZX)
660 Q=J+DIAM*COS(A*ZX)
670 LINE(I,J)-(P,Q),CLR
680 FOR C=A TO B
690    R=I+DIAM*SIN(C*ZX)
700    S=J+DIAM*COS(C*ZX)
710    LINE(P,Q)-(R,S),CLR
720    P=R
   : Q=S
730 NEXT C
740 LINE(I,J)-(R,S),CLR
750 CX=A+((PER(N)*3.6)*.5)
760 R=I+(DIAM*.5)*SIN(CX*ZX)
770 S=J+(DIAM*.5)*COS(CX*ZX)
780 PAINT(R,S),CLR
790 RETURN
800 END
```

Radio Shack Computer Center Addresses

ALABAMA
BIRMINGHAM 2428 Green Springs Hwy.; 9233 Parkway East
HUNTSVILLE 1400 N. Memorial Pkwy.
MOBILE 405 Bel-Air Blvd.
MONTGOMERY #24 Union Square S/C

ARIZONA
PHOENIX 10233 Metro Pkwy. E.; 4301 N. 7th Ave.
SCOTTSDALE 2525 N. Scottsdale Rd.
TEMPE 83 E. Broadway
TUCSON 5622 E. Broadway; Campbell Plaza; 2830 N. Campbell Ave.

ARKANSAS
LITTLE ROCK Town & Country S/C; University & Asher

CALIFORNIA
ANAHEIM 509 Katella
BAKERSFIELD 2018 Chester Ave.
BERKELEY 1922 Grove St.
BEVERLY HILLS 8500 Wilshire Blvd.
BREA Imperial Shopping Center, 391 South State College
CANOGA PARK 8371 Topanga Canyon
CARMICHAEL 6305 Fairoaks Blvd.
CHICO 1834 Mangrove Ave.
CHULA VISTA 1201 3rd Ave.
CITRUS HEIGHTS 7405 Green Back Ln. at San Juan
DOWNEY 8031 Florence Ave.
EL TORO El Toro Rd. at San Diego Freeway
ESCONDIDO 347 W. Mission Ave.
FREMONT Fremont Hub, 39114 Fremont
FRESNO Princeton S/C, 2721 N. Blackstone Ave.
GARDEN GROVE 12821 Knott St.
GLENDALE 236 N. Brand Blvd.
HAYWARD 24784 Hesperian Blvd.
HOLLYWOOD 6922 Hollywood Blvd.
IRVINE Redhill at Main St.
LAKEWOOD 5830 Lakewood Blvd.
LA MESA 5346 Jackson Dr.
LONG BEACH 2119 Bellflower Blvd.
LOS ANGELES 740 S. Olive St.; 5240 Century Blvd. (Airport area)
MODESTO 221 McHenry Ave.
MONTCLAIR 5237 Arrow Hwy.
MONTEREY 484 Washington St.
MOUNTAIN VIEW 1933 El Camino Real W.
OAKLAND 1733 Broadway, (415) 763-3183
PASADENA 575 S. Lake Ave.
PLEASANT HILL 508 G. Contra Costa Blvd.
RIVERSIDE 3844 La Sierra Ave.
SACRAMENTO 4749 J. St.
SAN BERNARDINO Inland Center Dr.
SAN DIEGO 3062 Clairemont Dr.; 3902 El Cajon Blvd.
SAN FRANCISCO One Market Place; 2920 Geary Blvd.
SAN JOSE 1228 S. Bascom Ave.
SAN MATEO 3180 Campus Dr.
SANTA ANA 2320 S. Fairview St. (at Warner)
SANTA BARBARA 4141 State St. A-1
SANTA FE SPRINGS 14138 East Firestone Blvd.
SANTA MONICA 511 Wilshire Blvd.
SANTA ROSA 823 4th St.
SHERMAN OAKS 14936 Ventura Blvd.
STOCKTON College Sq. S/C, 963 West March Lane
TARZANA 18545 Ventura Blvd.
TORRANCE 3840 Sepulveda at Hawthorne
VENTURA 4005 E. Main St.
WEST COVINA 2516 E. Workman St.

COLORADO
AURORA Hoffman Heights Shop. Ctr.
BOULDER Arapahoe Plaza, 3550 Arapahoe
COLORADO SPRINGS 4341 N. Academy
DENVER 8000 E. Quincy, Green Briar Plaza, 7075 Pecos
LAKEWOOD 2099 Wadsworth Blvd.

CONNECTICUT
EAST HAVEN 51 Frontage Rd.
FAIRFIELD 1196 Kings Hwy. & Rt. 1
HARTFORD The Richardson Bldg., 942 Main St.
MANCHESTER 228 Spencer St.
NORWALK Rt. 7-345 Main Ave.
ORANGE Edwards Food Warehouse Shop. Ctr., 538 Boston Post Rd.
WATERBURY 105 Bank St.
WATERFORD 122 Boston Port Rd.
WEST HARTFORD 39 S. Main St.

DELAWARE
DOVER Edgehill Shop. Ctr., Rt. #13
WILMINGTON 3847 Kirkwood Hwy.

DISTRICT OF COLUMBIA
N.W. WASHINGTON 1800 M St.; 15th & L St.; Van Ness Station, 4250 Connecticut Ave.

FLORIDA
ALTAMONTE SPRINGS 766 B. East Altamonte Dr.
BOCA RATON 1662 N. Federal Hwy.
CLEARWATER 2460-D US 19 North
DAYTONA BEACH Volusia Plaza
FT. LAUDERDALE 4368 N. Federal Hwy.
FORT MYERS Edison Mall
FORT PIERCE Center West Shopping Center, U.S. Hwy. 1
GAINESVILLE 3315 Archer Rd.
HOLLYWOOD 429 S. State Rd. #7
JACKSONVILLE 8252 Arlington Exprwy.; Roosevelt Mall, Roosevelt Blvd.
LAKELAND 2810 S. Florida Ave.
LAUDERDALE LAKES 4317-25 N. State Rd. 7
MERRITT ISLAND 700 E. Merritt Island Causeway
MIAMI 9459 S. Dixie Hwy.; 1601 Biscayne Blvd.; 15 SE. 2nd Ave.; 20761 S. Dixie Hwy.
N. MIAMI BEACH The Promenade, 1777 N.E. 163rd St.
ORLANDO 2810 E. Colonial Dr.
PENSACOLA Eastgate Shopping Center, 6925 N. 9th Ave.
ST. PETERSBURG 3451 66th St. N.
SARASOTA 5251 S. Tamiami Tr. (Hwy. 41)
TALLAHASSEE 2529 S. Adams
TAMPA 4555 W. Kennedy; 1825 E. Fowler Ave.
W. PALM BEACH 2271-A Palm Beach Lakes Blvd.

GEORGIA
ATLANTA 2108 Henderson Mill; 49 W. Paces Ferry, Akers Mill S/C, 2937 Cobb Parkway NW.; 113 Peachtree St.
AUGUSTA 3435 Wrightsboro Rd.
COLLEGE PARK 5309 Old National Hwy.
COLUMBUS Columbus Square Mall, 3050 Macon Rd.
DORAVILLE 5697 Buford Hwy.
MACON 1467 Eisenhower Pkwy.
SAVANNAH Chatham Plaza, 7805 Abercorn St.

HAWAII
HONOLULU Pacific Trade Center-Ground Floor, 190 South King St.

IDAHO
BOISE 691 S. Capitol Blvd.

ILLINOIS
AURORA 890 North Lake St.
BLOOMINGTON 401 N. Veterans Parkway (Eastland Commons)
CHICAGO 4355 S. Archer Ave.; CNA Plaza, 309 S. Wabash; 72 West Adams; 524 N. Michigan Ave.; 101 W. Washington St. at Clark
DEERFIELD Deerbrook Mall
ELMWOOD PARK 7212 W. Grand Ave.
FAIRVIEW HEIGHTS #4 Market Place
HOMEWOOD/GLENWOOD 329 Glenwood Lansing, 18230 S. Halsted at 183rd
JOLIET 2415 W. Jefferson St.
LaGRANGE One S. LaGrange Rd.
LIBERTYVILLE 1350 S. Milwaukee Ave.
LOMBARD 4 Yorktown Center
MOLINE 4401 44th Ave.
NILES 8349 Golf Rd.
OAK LAWN 4815 W. 95th St.
PEORIA 4125 N. Sheridan Rd.
ROCKFORD North Town S/C, 3600 N. Main St.
SCHAUMBURG 651 Mall Dr.
SPRINGFIELD Sherwood Plaza, 2482 Wabash

INDIANA
EVANSVILLE 431 Diamond Ave.
FT. WAYNE 747 Northcrest S/C
GRIFFITH 208 W. Ridge Rd.
HOBART Save-More Plaza, Rtes. #6 & 51
INDIANAPOLIS 6242 E. 82nd St., Castleton Plz.; Speedway Plaza, 6129 B Crawfordsville; 10013 E. Washington St.; Two W. Washington St.
SOUTH BEND 1827 South Bend Ave.
TERRE HAUTE 3460 U.S. Hwy. 41 S.

IOWA
CEDAR RAPIDS 111 First Ave., S.E. (Downtown)
DAVENPORT 616 E. Kimberly Rd.
DES MOINES 7660 Hickman Rd., Sherwood Forest S/M

KANSAS
OVERLAND PARK 8619 W. 95th
TOPEKA White Lakes Plaza, West Tower, 3715 Plaza Dr.
WICHITA 2732 Blvd. Plaza S/C

KENTUCKY
FLORENCE 7727 Mall Rd.
LEXINGTON 2909 Richmond Rd.
LOUISVILLE Louisville Galleria; 4133 Shelbyville Rd.

LOUISIANA
ALEXANDRIA 1213 Texas Ave.
BATON ROUGE 7007 Florida Blvd.
GRETNA 400 La Palco Blvd.
HOUMA 2348 W. Park Ave. (Hwy. 24)
LAFAYETTE University Square at Congress Blvd.
METAIRIE 3750 Veterans Hwy.
NEW ORLEANS 327 St. Charles Ave.
SHREVEPORT 1545 Line Ave.

MAINE
BANGOR Maine Square

MARYLAND
BALTIMORE 7942 Belair Rd.; Putty Hill Plaza, 115 N. Charles St. at Lexington
BETHESDA 7900 Wisconsin Ave.
CATONSVILLE One Mile West S/C, 6600 B Balt. Nat'l. Pike
FREDERICK Shoppers World, Rt. 40W
NEW CARROLLTON-LANHAM 7949 Annapolis Rd.
PASADENA 8120 Ritchie Hwy.
ROCKVILLE Congressional Plaza, 1673 Rockville Pike
SALISBURY Shoppers World S/C, Rt. 50
TEMPLE HILLS 4520 St. Barnabas Rd.
TOWSON-LUTHERVILLE Yorktown S/C York Rd. at Ridgley Rd.

MASSACHUSETTS
BOSTON 730 Commonwealth Ave.; 111 Summer St.; 372 Boylston
BOSTON 372 Boylston
BRAINTREE South Shore Plaza, 250 Granite St.
BROCKTON 675 Belmont
BURLINGTON Crossroads Plaza, Rt. 3 S.
CAMBRIDGE Harvard Square, 28 Boylston St.
CHESTNUT HILL 200 Boylston St.
FALL RIVER 90 William S. Canning Blvd.
NATICK 1400 Worcester Rd.
SAUGUS 343 Broadway
SPRINGFIELD 1985 Main St., Northgate Plz.
WORCESTER Lincoln Plaza

MICHIGAN
ANN ARBOR 2515 Jackson Rd.
BIRMINGHAM 3620 W. Maple Rd.
DEARBORN Westborn Shop. Ctr., 23161 Michigan Ave.
DETROIT DWNTN 1559 Woodward Ave.
FLINT G3298 Miller Rd., Yorkshire Plaza
GRAND RAPIDS 3142 28th St. SE.
KALAMAZOO 25 Kalamazoo Center
LANSING 2519 S. Cedar St.
LIVONIA 33470 W. 7 Mile Rd.
PLAINFIELD North Kent Mall
PONTIAC North Oak Plaza-Pontiac Mall, 2436 Elizabeth Lake Rd.
ROSEVILLE 31873 Gratiot Ave.
SOUTHFIELD 17651 West 12 Mile Rd.
TROY Oakland Plaza. 322 John R. Rd.
WARREN 29038 Van Dyke Ave.

MINNESOTA
BLOOMINGTON 10566 France Ave. S.
FRIDLEY 7974-76 University Ave. North
GOLDEN VALLEY Golden Valley S/C, 8016 Olson Memorial Hwy.
MINNEAPOLIS 830 Marquette Ave.
ST. PAUL 6th & Wabasha

MISSISSIPPI
GULFPORT 516A Courthouse Rd.
JACKSON 979 Ellis Ave.

MISSOURI
DES-PERES 11960 Manchester Rd.
FLORISSANT 47 Florissant Oaks S/C
INDEPENDENCE 1325 S. Noland Rd.
KANSAS CITY 4025 N. Oak Trafficway
ST. ANN 10472 St. Charles Rock Rd.
ST. LOUIS 500 No. Broadway (Commerce Bank Bldg. Downtown)
SPRINGFIELD 2684 S. Glenstone

NEBRASKA
LINCOLN 4601 "O" St.
OMAHA 3006 Dodge St.; 1318 72nd St. at Pacific

NEVADA
LAS VEGAS Commercial Center, 953 E. Sahara #31-B
RENO 3328 Kietzke Lane

NEW HAMPSHIRE
MANCHESTER Hampshire Plaza, 1000 Elm St.
NASHUA 429 Amherst St., 101A

NEW JERSEY
BRIDGEWATER 1472 U.S. Highway 22 East

ILLINOIS / E. BRUNSWICK etc. (continued)
E. BRUNSWICK 595 A Rt. 18
E. HANOVER Rt. 10, Hanover Plaza
LAWRENCEVILLE Rt. 1 & Texas Ave.
NEWARK 595 Broad
NORTHFIELD 322-24 Tilton Rd.
PARAMUS 175 Rt. 17 S.
POMPTON PLAINS Plains Plaza
SPRINGFIELD Rt. #22 Center Isle
TOMS RIVER 700 Rt. 37 West
VOORHEES 35 Eagle Plaza

NEW MEXICO
ALBUQUERQUE 2108 San Mateo NE.

NEW YORK
ALBANY Shoppers Pk., Wolf Rd.
BAYSHORE 1751 Sunrise Hwy.
BETHPAGE 422 N. Wantagh Ave.
BROOKLYN 531 86th St.
BUFFALO 839 Niagara Falls Blvd.
FRESH MEADOWS 187-12 Horace Harding Exp.
GARDEN CITY 960 Franklin Ave.
JOHNSON CITY Giant Shopping Center, Harry L. Drive
KINGSTON Kings Mall, Rt. 9W
LAKE GROVE 111-113 Alexander Ave.
MANHASSET 1550 Northern Blvd.
MELVILLE TSS Mall, Rt. 110
NEWBURGH Zayre Plaza, Rt. #17K
NEW ROCHELLE 211 North Ave.
NEW YORK 385 Fifth Ave.; 139 E. 42nd St.; 19 W. 23rd St.; 347 Madison Ave.; 270 Park Ave. South; 1282 Broadway; 9 Broadway
NIAGARA FALLS Pine Plaza, 8351 Niagara Falls Blvd.
REGO PARK 97-77 Queens Blvd.
ROCHESTER 3000 Winton Rd.
SCARSDALE 365 Central Park Ave.
SCHENECTADY Woodlawn Plaza
SPRING VALLEY White House Center, 88 W. Rt. 59
STATEN ISLAND 2409 Richmond Ave.
SYRACUSE 2544 Erie Blvd.; Hotel Syracuse, 510 S. Warren St.
UTICA Riverside Mall
VALLEY STREAM Green Acres Shop. Ctr.
YONKERS Cross Country Shop. Ctr.

NORTH CAROLINA
ASHEVILLE K-Mart Shopping Center, Tunnel Rd.
CHARLOTTE 3732 Independence Blvd.; Tyvola Mall, 5401 South Blvd.
DURHAM South Square Mall
FAYETTEVILLE Eutaw Shopping Center, 815 Elm St.; Fayetteville Street Mall
GREENSBORO 3718 High Point Rd.
RALEIGH Townridge Sq., Hwy. 70 W.
WILMINGTON Independence Mall
WINSTON-SALEM 629 Peters Creek Pkwy.

OHIO
AKRON Fairlawn Plaza, 2727 W. Market St.
BEDFORD HEIGHTS 5217 Northfield Rd.
CANTON 5248 Dressler Rd. NW.; Mellet Plaza, 3826 W. Tuscarawas.
CENTERVILLE 2026 Miamisburg-Centerville Rd.
CINCINNATI 9725 Montgomery; 16-18 Convention Way (on Skyway)
CLEVELAND 419 Euclid (Dwntwn); 27561 Euclid Ave.
COLUMBUS 862 S. Hamilton, Great Eastern S/C; The Patio Shop. Ctr., 4661 Karl Rd.; 400 N. High St.
DAYTON North West Plaza, 3279 West Siebenthaler
ELYRIA 286 Midway Blvd.
FAIRFIELD 7255 Dixie Hwy. (1/4 Mi. North of I-275)
NORTH OLMSTED Great Northern S/C
PARMA 7551 W. Ridgewood Dr.
TOLEDO 5844 W. Central Ave.; Brownstone Plaza, 1724 S. Reynolds Rd.
YOUNGSTOWN Union Square Plaza, 2543 Belmont Ave.

OKLAHOMA
OKLAHOMA CITY 4732 SE 29th St.; Springdale S/C, 4469 NW 50th; 1101 SW 59th St.
TULSA 7218 & 7220 E. 41st St.

OREGON
EUGENE 390 Coburg Rd.
PORTLAND 7463 SW Barbur Blvd.; 9131 SE Powell; 3rd and Washington Sts. (Downtown)
SALEM Salem Plaza, 403 Center

PENNSYLVANIA
ALLENTOWN Crest Plaza S/C, Cedar Crest Blvd., US 22
BALA CYNWYD 67 E. City Line Ave.
EASTON 25th St. Shopping Center
ELKINS PARK Elkins Park Square, 8080 Old York Rd.
ERIE 5755 Peach St.
HARRISBURG Union Deposit Mall, Union Deposit Rd. #17
LANCASTER Park City Plaza, US 30
MONROEVILLE 3828 Wm. Penn. Hwy.
MONTGOMERYVILLE Airport Sq., Rt. 309
PHILADELPHIA 7542 Castor Ave.; 1002 Chestnut St.; 1801 Market St.; 10 Penn Center

PENNSYLVANIA (continued)
PITTSBURGH 5775 Baptist Rd.; Hills Plaza; 303 Smithfield St.; 4643 Baum Blvd.; 4768 McKnight Rd.
SCRANTON 206 Meadow Ave.
WILKES BARRE, 23 W. Market St.
WYOMISSING Berkshire Mall West, 1101 Woodland Rd.,
YORK York County Shopping Center

PUERTO RICO
HATO REY 243 Franklin D. Roosevelt Ave.

RHODE ISLAND
E. PROVIDENCE 850 Waterman Ave.
PROVIDENCE 177 Union St.

SOUTH CAROLINA
COLUMBIA Old Sears Bldg., 1001 Harden St.
GREENVILLE N. Hills S/C
N. CHARLESTON 5900 Rivers Ave.
SPARTANBURG Hillcrest Shop. Ctr.

SOUTH DAKOTA
SIOUX FALLS 1700 S. Minnesota at 25th

TENNESSEE
CHATTANOOGA Northgate Mall
JOHNSON CITY Peerless Center
KNOXVILLE Cedar Bluff S/C, 9123 Executive Park Dr.
MEMPHIS 4665 American Way; 1997 Union Ave.
NASHVILLE 2115 Franklin Pike; Rivergate Plaza

TEXAS
AMARILLO Wellington Sq. S/C, 1619 S. Kentucky
ARLINGTON 2500 E. Randol Mill, Suite 113
AUSTIN 8764 E. Research Blvd.; Southwood Mall, White Blvd.
BROWNSVILLE 1639 Price Rd. (Hwy. 77)
BEAUMONT 5330 Eastex Frwy.
COLLEGE STATION 2414 Texas Ave., South
CORPUS CHRISTI 1711 S. Staple St.
DALLAS 15340 Dallas Pkwy., Suite 1100; 2930 W. Northwest Hwy.; 1517 Main St.; 2588 Royal Ln.
EL PASO 9515 Gateway West, Kern Plaza Shopping Center, 3100 N. Mesa
FT. WORTH 231 One Tandy Center, 2801 Alta Mere
GALVESTON 5924 Broadway
HARLINGEN 1514 S. Hwy 77, Sunshine Strip
HOUSTON 211C-FM 1960, 10543 Gulf Fwy.; 5900 North Fwy.; 6813 SW Fwy.; 809 Dallas St., Holland Square Center, 10920 East Freeway, (713) 453-0600; Champion Forest Plaza, Champion Forest Dr. and F.M. 1960 West; 1018 Gessner; 3278 South Loop West (So. Main at 610)
HUMBLE 19300 "B" Hwy. 59 (at FM 1960)
HURST Northeast Mall
IRVING 2011 West Airport Fwy.
LAREDO 102 East Calton Rd.
LUBBOCK 3625 34th St.
MESQUITE Town East Crossing Shop. Ctr.
ODESSA 1613 "A" East 8th Street
RICHARDSON Fleetwood Sq. S/C, 202 W. Campbell Rd., 320 S. Central Expr.
SAN ANTONIO 6018 West Ave.; 4249 Centergate; Riverbend Parking Garage, 211 W. Market St (Downtown)
WICHITA FALLS 1720-A 9th St.

UTAH
OGDEN K-Mart Shopping Center, 3672 Wall Ave.
OREM Grand Central Plaza, 384 East & 1300 South
MURRAY 6051 S. State Ave.
SALT LAKE CITY 301 South State St.

VIRGINIA
ALEXANDRIA 3425 King St. at Quaker Ln.
ARLINGTON Crystal City, 2301 So. Jefferson Davis Hwy.
FAIRFAX Westfair Center, 11027 Lee Hwy.
LYNCHBURG Hill's Plaza, Ward's Rd.
NEWPORT NEWS Newmarket South Shop. Ctr.
NORFOLK 5731 Poplar Hall Dr.; Wards Corner, 122 E. Little Creek Rd.
RICHMOND Willow Lawn S/C, 1617 Willow Lawn Dr.; 7728 Midlothian Turnpike
ROANOKE Franklin Bldg., 3561 Franklin Rd. S.W.
ROSSLYN 1911 N. Ft. Myer Dr. at Rt. 29

WASHINGTON
BELLEVUE Crossroads Mall, North East 8th & 156 St.
BELLINGHAM 111 Cornwall Ave., Suite B & C
FEDERAL WAY 33505 Pacific Hwy. South
OLYMPIA 106 N. Wilson
SEATTLE 18405 Aurora Ave. N.; 1521 3rd Ave.; 5030 Roosevelt Way NE
SPOKANE 7702 N. Division; E. 12412 Sprague
TACOMA 7030 S. Sprague
TUKWILA 15425 53rd Ave. S.
YAKIMA 1111 N. First St.

WEST VIRGINIA
DUNBAR, Dunbar Village Shop. Ctr.
HUNTINGTON 2701 1/2 5th Ave.

WISCONSIN
APPLETON 2310 West College Ave.
MADISON 57 West Towne Mall
MILWAUKEE 6450 N. 76th St.; 729 N. Milwaukee (Downtown)
WEST ALLIS 2717 South 108th St.