

# THE nibble ILLUSTRATOR

## The NIBBLE Illustrator

by Colin French  
2144 Iris St.  
Ottawa, Ontario  
Canada, K2C 1B3

### INTRODUCTION

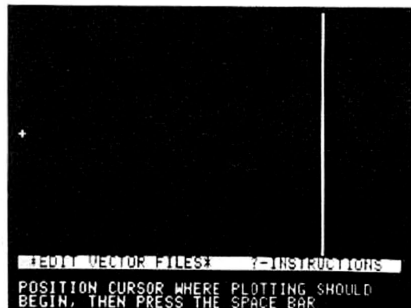
High resolution graphics shapes can easily be displayed on the Apple II Plus by using the built-in shape table commands (see the Applesoft Reference Manual). Unfortunately, constructing the tables to use with these commands is tedious and prone to error. The **NIBBLE Illustrator** allows you to quickly develop complex shape tables for use in your own programs.

The program is divided into two main parts. The first section, **EDIT VECTOR FILES**, is used to design and edit individual shapes. The set of vectors describing the shape is saved to disk, creating a disk 'library' of various shapes. The second section, **BUILD SHAPE TABLE**, combines the shapes from this library in any combination to form completed shape tables. By saving each shape separately it can be easily used in many shape tables, thereby eliminating the need to re-draw shapes for each table.

### USING THE ILLUSTRATOR

In order to explain the various functions of the Illustrator, let's develop a typical shape table. This table will be a sample NIBBLE Logo.

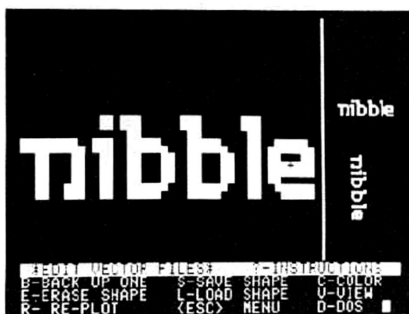
Run the NIBBLE ILLUSTRATOR program and select **EDIT VECTOR FILES** from the menu. We now have a chance to move the cursor to any starting point, so we'll move it to the left side of the screen and press the space bar.



Hold down the 'CTRL' key and press the 'I' key. The block the cursor was on is now plotted and the cursor moved up one square. The right side of the screen should show two isolated dots.

We can plot a block and move up, left, right or down by holding the CTRL key and pressing the I, J, K or M keys respectively. To move the cursor without plotting a block, press just the direction key without holding down the CTRL key. This creates a no-plot vector.

By plotting blocks and moving the cursor, sketch out the shape you wish for the logo. This shape will be seen in Hi-res on the right side of the screen both in the normal orientation and turned 90 degrees. These shapes are drawn in white; to see what they will look like drawn in the other Hi-res colors, press the C key to cycle through the four colors and white.

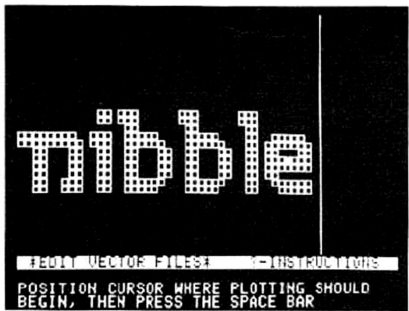


If you make an error you can back up and erase one vector each time you press the B key.

To start over again with a clean screen, press the E key to erase the entire shape and its vectors.

### RE-PLOT THE SHAPE

Once the Nibble logo shape is just right, you could immediately save the set of vectors you plotted, but chances are there are some extra, unneeded vectors in the shape. Since every two vectors use up one byte of memory, it is useful to minimize the number of vectors used to draw a shape. This will also allow the program to draw the shape as quickly as possible. For this sort of shape, plotting should begin at the shape's supposed center of gravity to ensure that it can be made to rotate in a convincing way. To do all of this we can re-plot the shape by pressing the R key. The vectors of the shape are erased, and the solid blocks of the large shape are replaced with empty ones to provide a guide to trace over.



Move the cursor using the I, J, K and M keys to the center of the shape and press the space bar. Now plot the shape in the usual manner, copying over the empty blocks in the shortest possible path.

### THE VIEW OPTION

To get a better idea of what the shape will look like, press the V key to VIEW the shape. Now besides controlling the color, you may change the scale and rotation of the shape. Because of the way Applesoft draws shapes there are fewer angles of rotation available at smaller scale factors. In fact, at a scale of 1 there are only 8 possible orientations.

The shape may also be moved about the screen by using the I, J, K and M keys. Use these and the 'REPT' key to get a rough sort of animation. Use the 'ESC' key to return to the plotting function.

### SAVE AND LOAD

Save the set of vectors describing the shape by pushing the S key. Use a name such as NIBLOGO and the file will be saved as VECTORS/NIBLOGO. To make sure the vector file was saved, press the D key for DOS and check the disk catalog by pressing the C key. You can also delete files from this point, but note that the program does not automatically supply the VECTORS/ prefix used by the SAVE shape and LOAD shape functions.

Pressing L key will load a shape from disk. Besides being shown in Hi-res, the shape will also be drawn in the enlarged size for use in the PLOTTING or RE-PLOT functions. The plotting of this large shape will begin at the cursor position. If the shape will not fit, you will be prompted to move the cursor to a different location and try again. As always, you can use ESC to escape back to the plotting function.

Return to the plotting function and erase the screen. Repeat the previous steps to design and save another shape under the name SHAPE2.

### BUILD THE SHAPE TABLE

We now have two shapes saved to disk to be combined into a shape table. Press the 'ESC' key to escape from the PLOTTING function to the main menu and select **BUILD SHAPE TABLE** by pressing the B key.

The program will prompt you for the names of the shapes to be included in the table. Type in NIBLOGO and RETURN, and that vector file will be loaded in from disk. The subsequent vector file (SHAPE2) you specify will be added to the end of the table. To check the disk catalog enter a 'C' and press 'RETURN' when prompted for a name.

When the two shapes have been loaded, type END and press RETURN and the program will construct the header of the table. The total length will be displayed, and then you will be prompted for a name to save the table under. Unless you choose to construct another table, you will then be returned to the main menu.

To check the table, select VIEW SHAPE TABLE from the main menu and enter your table's name when prompted. You can choose any shape from the table and display it with control over scale, color and rotation in the same manner as VIEW of EDIT VECTOR FILES. You can also use it to construct pictures using NIBBLE DESIGNER in this issue.

### LOADING THE ILLUSTRATOR

To use this program you must have a 48K Apple II with Applesoft in ROM or Language Card and one disk drive. Due to the size of the program, it must be loaded above the High Resolution Graphics Screen 1. This is accomplished with the first line of the Applesoft program in Listing One.

First enter the Applesoft program and SAVE it under the name NIBBLE ILLUSTRATOR. Then use the monitor to enter the shape table shown in Listing Two. (See the Letters section.) Cheer up! It's the last shape table you'll have to enter by hand! Save the table under the name ITABLE so the main program can load it correctly. The command is BSAVE ITABLE, A\$300, L\$29.

## NIBBLE ILLUSTRATOR

Now RUN. Test out the various subroutines to see if they work. The shape table should draw the cursor shape, a cross, and the solid block which appears when you plot a point. Exit the program and add the other functions, testing each as you complete it.

### HOW THE ILLUSTRATOR WORKS

The first line of the Illustrator program resets the high byte of the start of program pointer from \$08 to \$40, thereby causing the program to be re-loaded starting at \$4001. The memory space between the top of Text Page 1 (\$800) and the bottom of the HGR1 screen (\$2000) is used for the shape table or vector file being developed. The shape table used by the main program is located starting at \$300. Having two shape tables requires setting the shape table address pointer correctly before drawing a particular shape, for example in line 1030 of the DRAW/ERASE CURSOR subroutine.

The vectors of the first shape are poked into memory starting at location \$A01. This allows enough room for the header of a shape table containing 255 shapes and space for over 11,000 vectors.

### Memory Usage

\$FFFF	-----
	DOS, Applesoft and Monitor
\$9600	-----
	Main program, strings and variables
\$4000	-----
	High Resolution Screen 1
\$2000	-----
	Vectors
\$0A01	-----
	User Shape Table Header
\$0800	-----
	Text Page 1
\$0400	-----
	Program Shape Table
\$0300	-----
	For computer's use - pointers, etc.
\$0000	-----

### THE PROGRAM

There are a few interesting sections of the program I'd like to point out, but before discussing these it might be useful to have an overview of the program. The main program is composed of the following sections:

10 - 290 Program Lead and Initialization  
400 - 630 Main Menu

### ----- Subroutines -----

700 - 830 Multi-letter Input  
1000 - 1050 Draw/Erase Cursor  
1100 - 1170 Draw/Erase Small Shapes  
1200 - 1250 Draw/Erase Block  
1300 - 1430 Position Cursor  
1500 - 1690 Convert Byte to Binary String  
1700 - 1940 Draw Large Shape

### ----- Section 1 Edit Vector Files -----

2000 - 2870 Plotting Function  
3000 - 3230 Back Up One  
3400 - 3520 Save Vector File  
3600 - 3780 Load Vector File  
3900 - 4110 DOS Commands  
4200 - 4550 View Shape

5000 - 5160 Disk Error Handling

### ----- Section 2 Build Shape Table -----

6000 - 6530 Build Shape Table  
7000 - 7550 View Shape Table

8000 - 9460 Instructions

### MULTI-GET FOR INPUT

Using Applesoft's **GET** command for input of information has many advantages when trying to develop a bomb-proof program. However, there are times when more than a single character is required as input. This led to the development of the **Multi-letter Input Subroutine**. After the main program has prompted the user for information, it sets the variable **LMT** to the maximum length of input string that will be allowed and jumps to this routine. The input subroutine basically **GETs** one character at a time, and adds it to the input string variable **WS**. Checks can be made to trap any undesired character. For example, line 780 doesn't allow commas to be accepted. To permit changing the input string, the routine traps the Back-up-arrow key and modifies the display and string accordingly. Upon return to the main program, **WS** holds the user's input.

### PLOTTING

The **PLOTTING** function starts with a straightforward initialization of the vectors, Hi-res screen and text window, followed by

branches off to the various other functions of **EDIT VECTOR FILES**. Next comes the most important part of the function: poking the vectors into memory. I decided to have only two vectors per byte, as it is rare that three vectors can be fitted into one byte, and it simplifies the program considerably. The program determines which vector you have entered, plots the block in the large shape if it is a plot-type vector, changes the cursor coordinates to match the desired direction, and finally does the poke into the proper memory location.

The only exception to this procedure is the **up-no-plot vector**. If it is the second vector of a byte, the Applesoft Draw routine will ignore it and jump to the next byte. If it is the first vector and the second vector is anything other than another up-no-plot vector, everything works properly. But if **both vectors of the byte are up-no-plot, the Applesoft routine would interpret this as the end of the shape and stop drawing completely. The code from line 2620 to line 2660 checks for these troublesome situations and corrects them.** If the up-no-plot is the second vector it is simply moved to become the first vector of the next byte. **If you follow one up-no-plot with another, the program adds a vector to sidestep to the left, does the up-no-plot, then steps back over to the right.** All of this goes on without you knowing it, but if you back up over a couple of up-no-plot vectors, the sidestep becomes apparent.

Throughout the program the variable **LOC** is assigned different values. These indicate the location of the user in the program and therefore what function is being used. This information is used by two sections of the program, the **DISK ERROR HANDLING** and **INSTRUCTIONS** functions, to return you to the proper place in the program after error messages or instructions have been printed.

### MODIFICATIONS AND IMPROVEMENTS

As the program stands, it provides a **44 by 32 square grid** in which to plot shapes, with each square of the grid occupying 5 by 5 Hi-res points. You might want to try different sizes of grids. Using a 4 by 4 point square would allow a 55 by 40 grid, but the smaller size of the square makes it more difficult to design distinguishable shapes for the cursor and the two types of blocks. Those of you with monitors could probably get away with a 3 by 3 point square.

# LISTING 1: NIBBLE ILLUSTRATOR

```

LIST
10 REM *****
11 REM * NIBBLE ILLUSTRATOR *
12 REM * BY COLIN FRENCH *
13 REM * COPYRIGHT (C) 1983 *
14 REM * BY MICROSPARC, INC *
15 REM * LINCOLN, MA. 01773 *
16 REM *****
180 IF PEEK (104) = 64 THEN 230
190 POKE 103,1: POKE 104,64: POKE 16384,0
200 PRINT CHR$ (4);"RUN NIBBLE ILLUSTRATOR"
210 REM ***** INITIALIZATION *****
220 :
230 D$ = CHR$ (4): TEXT = HOME
270 L1 = 0:H1 = 3:L2 = 253:H2 = 9
280 PRINT D$;"BLOOD ITABLE,A$300"
290 DIM L(255)
390 TEXT = HOME : UTAB 22: PRINT "*** COPYRIGHT 198
3 BY MICROSPARC, INC. ***": GOTO 430
400 :
410 REM ***** MAIN MENU *****
420 TEXT = HOME
430 UTAB 3
440 PRINT TAB( 7);"*****"
450 PRINT TAB( 7);"*"
460 PRINT TAB( 7);"* NIBBLE ILLUSTRATOR *"
470 PRINT TAB( 7);"*" : REM
24 SPACES IN QUOTES
480 PRINT TAB( 7);"*****"
490 UTAB 10
500 PRINT TAB( 9);"E - EDIT VECTOR FILES": PRINT
510 PRINT TAB( 9);"B - BUILD SHAPE TABLE": PRINT
520 PRINT TAB( 9);"V - VIEW SHAPE TABLE": PRINT
530 PRINT TAB( 9);"? - FOR INSTRUCTIONS": PRINT
540 PRINT TAB( 9);"<ESC> TO QUIT PROGRAM"
550 UTAB 24
560 PRINT TAB( 9);"WHICH WOULD YOU LIKE? ";
570 GET Z$
580 IF Z$ = "E" THEN 2000
590 IF Z$ = "B" THEN 6000
600 IF Z$ = "V" THEN 7000
610 IF Z$ = "?" THEN LOC = 1: GOTO 8000
620 IF Z$ = CHR$ (<27>) THEN HOME : END
630 GOTO 570
700 :
710 REM ***** MULTI-LETTER INPUT *****
720 :
730 W$ = ""
740 GET Z$
750 IF Z$ = CHR$ (13) THEN RETURN
760 IF Z$ = CHR$ (8) THEN 800
770 IF LEN (W$) = LMT THEN 740
780 IF Z$ = "," THEN 740
790 PRINT Z$;W$ = W$ + Z$: GOTO 740
800 IF LEN (W$) = 0 THEN 740
810 PRINT CHR$ (8);: PRINT " ";: PRINT CHR$ (8);
820 IF LEN (W$) = 1 THEN 730
830 W$ = LEFT$( W$, LEN (W$) - 1): GOTO 740
1000 :
1010 REM ***** DRAW/ERASE CURSOR *****
1020 :
1030 POKE 232,L1: POKE 233,H1: ROT= 0: SCALE= 1
1040 XDRAW 1 AT XC,YC
1050 RETURN
1100 :
1110 REM ***** DRAW/ERASE SMALL SHAPES *****
1120 :
1130 IF PEEK (2561) = 0 THEN 1170
1140 POKE 232,L2: POKE 233,H2: ROT= 0: SCALE= 1
1150 DRAW 1 AT 233 + INT (XS / 5),40 + INT (YS /
5)
1160 ROT= 16: DRAW 1 AT 266 - INT (YS / 5),90 + INT
(XS / 5)
1170 RETURN
1200 :
1210 REM ***** DRAW/ERASE BLOCK *****
1220 :
1230 POKE 232,L1: POKE 233,H1: ROT= 0: SCALE= 1
1240 DRAW 2 AT OX,OY
1250 RETURN
1300 :
1310 REM ***** POSTION CURSOR *****
1320 :
1330 UTAB 22: PRINT "
": REM 39 SPACES IN QUOTES
1340 PRINT "POSITION CURSOR WHERE PLOTTING SHOULD
"
1350 PRINT "BEGIN, THEN PRESS THE SPACE BAR
"
1360 GET Z$: GOSUB 1000
1370 IF Z$ = "1" THEN YC = YC - 5: IF YC < 0 THEN
YC = 2
1380 IF Z$ = "J" THEN XC = XC - 5: IF XC < 0 THEN
XC = 2
1390 IF Z$ = "K" THEN XC = XC + 5: IF XC > 219 THEN
XC = 217
1400 IF Z$ = "M" THEN YC = YC + 5: IF YC > 159 THEN
YC = 157
1410 IF Z$ = " " THEN RETURN
1420 IF Z$ = "?" THEN LOC = 2: GOTO 8000
1430 GOSUB 1000: GOTO 1360
1500 :
1510 REM ***** CONVERT BYTE TO BINARY STRING *****
1520 :
1530 B$ = ""; IF INT (BT / 128) THEN BT = BT - 128
:B$ = B$ + "1": GOTO 1550
1540 B$ = B$ + "0"
1550 IF INT (BT / 64) THEN BT = BT - 64:B$ = B$ +
"1": GOTO 1570
1560 B$ = B$ + "0"
1570 IF INT (BT / 32) THEN BT = BT - 32:B$ = B$ +
"1": GOTO 1590
1580 B$ = B$ + "0"
1590 IF INT (BT / 16) THEN BT = BT - 16:B$ = B$ +
"1": GOTO 1610
1600 B$ = B$ + "0"
1610 IF INT (BT / 8) THEN BT = BT - 8:B$ = B$ + "
1": GOTO 1630
1620 B$ = B$ + "0"
1630 IF INT (BT / 4) THEN BT = BT - 4:B$ = B$ + "
1": GOTO 1650
1640 B$ = B$ + "0"
1650 IF INT (BT / 2) THEN BT = BT - 2:B$ = B$ + "
1": GOTO 1670
1660 B$ = B$ + "0"
1670 IF BT THEN B$ = B$ + "1": GOTO 1690
1680 B$ = B$ + "0"
1690 RETURN
1700 :
1710 REM ***** DRAW LARGE SHAPE *****
1720 :
1730 AA = AD: ROT= 0: SCALE= 1:XC = XS:YC = YS:FER =
0
1740 BT = PEEK (AA): IF BT = 0 THEN 1910
1750 REM -CONVERT TO BINARY STRING-
1760 GOSUB 1500
1770 OX = XC:OY = YC
1780 IF MID$( B$,6,1) = "1" THEN HCOLOR= 0: GOSUB
1200: POKE 232,L1: POKE 233,H1: HCOLOR= 3: DRAW
SH AT XC,YC
1790 IF MID$( B$,7,2) = "00" THEN YC = YC - 5: IF
YC < 0 THEN FER = 1:YC = 2: GOTO 1940
1800 IF MID$( B$,7,2) = "11" THEN XC = XC - 5: IF
XC < 0 THEN FER = 1:XC = 2: GOTO 1940
1810 IF MID$( B$,7,2) = "01" THEN XC = XC + 5: IF
XC > 219 THEN FER = 1:XC = 217: GOTO 1940
1820 IF MID$( B$,7,2) = "10" THEN YC = YC + 5: IF
YC > 159 THEN FER = 1:YC = 157: GOTO 1940
1830 OX = XC:OY = YC
1840 IF MID$( B$,3,3) = "000" THEN 1900
1850 IF MID$( B$,3,1) = "1" THEN HCOLOR= 0: GOSUB
1200: POKE 232,L1: POKE 233,H1: HCOLOR= 3: DRAW
SH AT XC,YC
1860 IF MID$( B$,4,2) = "00" THEN YC = YC - 5: IF
YC < 0 THEN FER = 1:YC = 2: GOTO 1940
1870 IF MID$( B$,4,2) = "11" THEN XC = XC - 5: IF
XC < 0 THEN FER = 1:XC = 2: GOTO 1940
1880 IF MID$( B$,4,2) = "01" THEN XC = XC + 5: IF
XC > 219 THEN FER = 1:XC = 217: GOTO 1940
1890 IF MID$( B$,4,2) = "10" THEN YC = YC + 5: IF
YC > 159 THEN FER = 1:YC = 157: GOTO 1940
1900 AA = AA + 1: GOTO 1740
1910 IF AA = AD THEN 1940
1920 IF MID$( B$,3,3) = "000" THEN AA = AA - 1:V =
2: GOTO 1940
1930 V = 1
1940 RETURN
2000 :
2010 REM ***** EDIT VECTOR FILES -PLOTTING *****
2020 :
2030 HC = 3:XC = 112:YC = 82: HGR
2040 HOME : UTAB 21
2050 INVERSE : PRINT " *EDIT VECTOR FILES*
?->IN
STRUCTIONS ";: NORMAL
2060 REM -DRAW CURSOR-
2070 GOSUB 1000
2080 AD = 2561:AA = AD:V = 1
2090 REM -POKE IN HEADER-
2100 POKE AD - 1,0: POKE AD - 2,4
2110 POKE AD - 3,0: POKE AD - 4,1
2120 POKE AA,0: POKE AA + 1,0
2130 REM -DRAW DIVIDING LINE-
2140 HCOLOR= 3: HPLLOT 221,0 TO 221,158 TO 222,158 TO
222,0
2150 REM -POSITION CURSOR-
2160 GOSUB 1300

```

Another improvement would be to write a machine language routine to take the place of the DRAW LARGE SHAPE subroutine. This could be much the same as the Applesoft Draw routine, but instead of plotting just a point it would draw a pre-defined shape with each vector. The increased speed of this approach could open up all sorts of other modifications such as relocating the large shape within the grid if you run out of room for plotting along one side.

## CONCLUSION

I hope you will find this program as valuable as I do. You can try many things with shape tables that you've probably been itching to do. The first project I used this program for was to create a 3000-vector shape table for use in a fancy introduction for my own copy of the program. Another useful project is to

create an alphabet, including lower case, to use on the Hi-res screen. As you work with the program you will find that more and more possibilities will present themselves.

**ED. NOTE: The NIBBLE diskette contains shape tables for Electronics and full alpha/ numerics at both standard and double sizes. It also contains vector tables for a space ship and an Apple.**

```

2170 XS = XC:YS = YC
2180 UTAB 22: HTAB 1: PRINT "B-BACK UP ONE S-SAV
E SHAPE C-COLOR "
2190 PRINT "E-ERASE SHAPE L-LOAD SHAPE V-VIEW
";
2200 PRINT "R- RE-PLOT <ESC> MENU D-DOS
";
2210 REM -DRAW CURSOR-
2220 GOSUB 1000:LOC = 3
2230 REM -DRAW SMALL SHAPES-
2240 HCOLOR= HC: GOSUB 1100
2250 REM -GET COMMAND-
2260 GET Z$: GOSUB 1000:OX = XC:OY = YC
2270 HCOLOR= 0: GOSUB 1100
2280 IF Z$ = CHR$(27) THEN 400
2290 IF Z$ = "?" THEN 8000
2300 IF Z$ = "B" THEN 3000
2310 IF Z$ = "E" THEN HGR : GOTO 2040
2320 IF Z$ = "R" THEN SH = 3: GOSUB 1700: GOTO 204
0
2330 IF Z$ = "S" THEN LOC = 4: GOTO 3400
2340 IF Z$ = "L" THEN LOC = 5: GOTO 3600
2350 IF Z$ = "C" THEN 3800
2360 IF Z$ = "V" THEN LOC = 7: GOTO 4200
2370 IF Z$ = "D" THEN LOC = 6: GOTO 3900
2380 REM -CURSOR/PLOTTING MOVES-
2390 IF Z$ = "I" THEN GOSUB 2500: GOSUB 2620:V1$ =
"0": GOTO 2700
2400 IF Z$ = "J" THEN GOSUB 2530:V1$ = "0": GOTO
2700
2410 IF Z$ = "K" THEN GOSUB 2560:V1$ = "0": GOTO
2700
2420 IF Z$ = "M" THEN GOSUB 2590:V1$ = "0": GOTO
2700
2430 IF Z$ = CHR$(9) THEN GOSUB 2500:V1$ = "1":
HCOLOR= 3: GOSUB 1200: GOTO 2700
2440 IF Z$ = CHR$(10) THEN GOSUB 2530:V1$ = "1"
: HCOLOR= 3: GOSUB 1200: GOTO 2700
2450 IF Z$ = CHR$(11) THEN GOSUB 2560:V1$ = "1"
: HCOLOR= 3: GOSUB 1200: GOTO 2700
2460 IF Z$ = CHR$(13) THEN GOSUB 2590:V1$ = "1"
: HCOLOR= 3: GOSUB 1200: GOTO 2700
2470 GOTO 2210
2500 REM -UP VECTOR-
2510 YC = YC - 5: IF YC < 0 THEN YC = 2: POP : GOTO
2210
2520 V2$ = "00": RETURN
2530 REM -LEFT VECTOR-
2540 XC = XC - 5: IF XC < 0 THEN XC = 2: POP : GOTO
2210
2550 V2$ = "11": RETURN
2560 REM -RIGHT VECTOR-
2570 XC = XC + 5: IF XC > 219 THEN XC = 217: POP : GOTO
2210
2580 V2$ = "01": RETURN
2590 REM -DOWN VECTOR-
2600 YC = YC + 5: IF YC > 159 THEN YC = 157: POP : GOTO
2210
2610 V2$ = "10": RETURN
2620 REM -UP-NO-PLOT CHECKING-
2630 IF V = 1 THEN 2660
2640 IF VK$(1) < > "000" THEN VK$ = "00000" + VK$
(1): GOSUB 2800:AA = AA + 1:V = 1: GOTO 2660
2650 VK$ = "00011" + VK$(1): GOSUB 2800:AA = AA + 1
:VK$ = "00001000": GOSUB 2800:AA = AA + 1:V =
1: POP : GOTO 2210
2660 RETURN
2700 REM -SET UP VECTORS TO BE POKED-
2710 VK$(V) = V1$ + V2$
2720 IF V = 1 THEN VK$ = "00000" + VK$(1)
2730 IF V = 2 THEN VK$ = "00" + VK$(2) + VK$(1)
2740 GOSUB 2900
2750 IF V = 1 THEN V = 2: GOTO 2210
2760 V = 1:AA = AA + 1: GOTO 2210
2800 REM -POKING OF VECTORS-
2810 BT = 0
2820 FOR N = 1 TO 8
2830 IF MID$(VK$,N,1) = "0" THEN 2850
2840 BT = BT + 2 ^ (8 - N)
2850 NEXT N
2860 POKE AA,BT: POKE AA + 1,8
2870 RETURN
3000 :
3010 REM ***** BACK UP ONE *****
3020 :
3030 IF AA = AD AND PEEK (AA) = 0 AND V = 1 THEN
3230

```

```

3040 REM -ERASE SMALL SHAPES-
3050 HCOLOR= 0: GOSUB 1100
3060 IF V = 2 THEN BT = PEEK (AA)
3070 IF V = 1 THEN BT = PEEK (AA - 1)
3080 REM -CONVERT TO BINARY STRING-
3090 GOSUB 1500
3100 IF V = 1 AND MID$(BT,3,3) = "000" THEN V =
2:AA = AA - 1: GOTO 3060
3110 IF V = 2 THEN V1$ = MID$(BT,6,1):V2$ = MID$
(BT,7,2)
3120 IF V = 1 THEN V1$ = MID$(BT,3,1):V2$ = MID$
(BT,4,2):VK$(1) = MID$(BT,6,3)
3130 REM -MOVE CURSOR-
3140 IF V2$ = "00" THEN YC = YC + 5
3150 IF V2$ = "11" THEN XC = XC + 5
3160 IF V2$ = "01" THEN XC = XC - 5
3170 IF V2$ = "10" THEN YC = YC - 5
3180 REM -ERASE BLOCK-
3190 IF V1$ = "1" THEN HCOLOR= 0:OX = XC:OY = YC:
GOSUB 1200
3200 REM -ERASE VECTOR-
3210 IF V = 2 THEN V = 1: POKE AA,0: GOTO 3230
3220 V = 2:AA = AA - 1:VK$ = "00000" + MID$(BT,6,
3): GOSUB 2800
3230 GOTO 2210
3400 :
3410 REM ***** SAVE VECTOR FILE *****
3420 :
3430 IF AA = AD AND PEEK (AA) = 0 THEN 2210
3440 UTAB 22: HTAB 1: PRINT " (ESC) TO ABORT
"
3450 PRINT "NAME FOR VECTOR FILE? VECTORS/
"
3460 PRINT "
": REM 39 SPACES IN QUOTES
3470 UTAB 23: HTAB 31: LMT = 22: GOSUB 700
3480 IF W$ = "" OR W$ = CHR$(27) THEN 2180
3490 IF W$ = "?" THEN 8000
3500 UTAB 22: HTAB 39: PRINT : ONERR GOTO 5000
3510 PRINT D$;"BSAVE VECTORS/";W$;"A2561,L";AA -
AD + 2
3520 POKE 216,0: GOTO 2180
3600 :
3610 REM ***** LOAD VECTOR FILE *****
3620 :
3630 UTAB 22: HTAB 1: PRINT " (ESC) TO ABOR
T
"
3640 PRINT "NAME OF VECTOR FILE? VECTORS/
"
3650 PRINT "
": REM 39 SPACES IN QUOTES
3660 UTAB 23: HTAB 30: LMT = 22: GOSUB 700
3670 IF W$ = "" OR W$ = CHR$(27) THEN 2180
3680 IF W$ = "?" THEN 8000
3690 UTAB 22: HTAB 39: PRINT : ONERR GOTO 5000
3700 PRINT D$;"BLOAD VECTORS/";W$;"A2561": POKE 2
16,0
3710 HGR : HCOLOR= 3: HPL0T 221,0 TO 221,150 TO 22
2,150 TO 222,0
3720 SH = 2: GOSUB 1700
3730 IF FED = 0 THEN 2180
3740 HOME : GOSUB 1000: UTAB 22: PRINT "IT WON'T F
IT STARTING FROM THAT CURSOR"
3750 PRINT "POSITION. TRY ANOTHER. (PUSH <RETURN>)
"
3760 GET Z$: HOME : UTAB 21: INVERSE : PRINT " *ED
IT VECTOR FILES* ?-INSTRUCTIONS ";: NORMAL
: IF Z$ = CHR$(27) THEN 2180
3770 IF Z$ = "?" THEN 8000
3780 GOSUB 1300:XS = XC:YS = YC: GOTO 3710
3800 :
3810 REM ***** CHANGE COLOR *****
3820 :
3830 HC = HC + 1: IF HC > 6 THEN HC = 1
3840 IF HC = 4 THEN HC = 5
3850 GOTO 2210
3900 :
3910 REM ***** DOS COMMANDS *****
3920 :
3930 HOME : UTAB 21: INVERSE : PRINT " *EDIT VECTO
R FILES* ?-INSTRUCTIONS ";: NORMAL

```

```

3940 PRINT "DOS COMMANDS: C-CATALOG"
3950 PRINT TAB(16);"D-DELETE"
3960 PRINT TAB(16);"(ESC) TO RETURN ";
3970 GET Z$: IF Z$ = CHR$(27) THEN POKE -1630
4,0: GOTO 2180
3980 IF Z$ = "?" THEN 8000
3990 IF Z$ = "C" THEN 4020
4000 IF Z$ = "D" THEN 4060
4010 GOTO 3970
4020 TEXT : POKE 35,20: VTAB 1: HOME : PRINT
4030 PRINT D$;"CATALOG"
4040 POKE 35,24: VTAB 24: HTAB 33
4050 GOTO 3970
4060 VTAB 22: HTAB 1: PRINT " (ESC) TO ABORT
"
4070 PRINT "FILE TO BE DELETED?"
";: VTAB
23: HTAB 22
4080 LMT = 30: GOSUB 700: IF W$ = "" OR W$ = CHR$(
27) THEN 3930
4090 VTAB 22: HTAB 39: PRINT : ONERR GOTO 5000
4100 PRINT D$;"DELETE ";W$
4110 POKE 216,0: GOTO 3930
4200 :
4210 REM ***** VIEW SHAPE *****
4220 :
4230 HGR : VTAB 22: HTAB 1
4240 X = 140:Y = 80: POKE 232,L2: POKE 233,H2
4250 CC = 3: HCOLOR= CC
4260 RR = 0: ROT= RR
4270 SS = 1: SCALE= SS
4280 PRINT "
";: REM 39 SPACES IN QUOTES
4290 PRINT " C - COLOR (3 ) R - ROTATION (0 )
"
4300 PRINT " S - SCALE (1 ) (ESC) TO RETURN
";
4310 HCOLOR= CC: DRAW 1 AT X,Y
4320 GET Z$: HCOLOR= 0: DRAW 1 AT X,Y
4330 IF Z$ = "?" THEN 8000
4340 IF Z$ < > CHR$(27) THEN 4370
4350 HGR : HCOLOR= 3: HPL0T 221,0 TO 221,158 TO 22
2,158 TO 222,0
4360 SH = 2: GOSUB 1700: GOTO 2100
4370 IF Z$ < > "C" THEN 4430
4380 CC = CC + 1: IF CC > 6 THEN CC = 1
4390 IF CC = 4 THEN CC = 5
4400 VTAB 23: HTAB 13: PRINT " ";
4410 HTAB 13: PRINT CC;: VTAB 24: HTAB 40
4420 GOTO 4310
4430 IF Z$ < > "S" THEN 4470
4440 SS = SS + 1: IF SS > 10 THEN SS = 1
4450 SCALE= SS: VTAB 24: HTAB 13: PRINT " ";
4460 HTAB 13: PRINT SS;: HTAB 40: GOTO 4310
4470 IF Z$ < > "R" THEN 4510
4480 RR = RR + 1: IF RR > 63 THEN RR = 0
4490 ROT= RR: VTAB 23: HTAB 35: PRINT " ";
4500 HTAB 35: PRINT RR;: VTAB 24: HTAB 40: GOTO 43
10
4510 IF Z$ = "I" THEN Y = Y - 5: IF Y < 0 THEN Y =
Y + 159
4520 IF Z$ = "J" THEN X = X - 5: IF X < 0 THEN X =
X + 279
4530 IF Z$ = "K" THEN X = X + 5: IF X > 279 THEN X
= X - 279
4540 IF Z$ = "M" THEN Y = Y + 5: IF Y > 159 THEN Y
= Y - 159
4550 GOTO 4310
5000 :
5010 REM ***** DISK ERROR HANDLING *****
5020 :
5030 ER = PEEK (222): IF LOC = 8 OR LOC = 9 THEN POKE
35,24
5040 VTAB 22: HTAB 1
5050 PRINT "
";: REM 39 SPACES IN QUOTES
5060 PRINT "
"
5070 PRINT "
";: REM 39 SPACES IN QUOTES
5080 VTAB 23: HTAB 1
5090 IF ER = 6 THEN PRINT "THAT FILE IS NOT ON TH
IS DISK.": GOTO 5150
5100 IF ER = 9 THEN PRINT "THERE ISN'T ENOUGH ROO
M ON THIS DISK.": GOTO 5150
5110 IF ER = 10 THEN PRINT "THAT FILE IS LOCKED."
: GOTO 5150
5120 IF ER = 11 THEN PRINT "THAT IS NOT A LEGAL F
ILE NAME.": GOTO 5150
5130 PRINT "A SYSTEM ERROR #";ER;" HAS OCCURRED IN
"
5140 PRINT "LINE #"; PEEK (218) + PEEK (219) * 25
6: END
5150 PRINT " (PUSH (RETURN)) ";: GET Z$
5160 ON LOC GOTO 400,2040,2180,3440,3630,3930,4500
,6040,6430,7000,7190
6000 :
6010 REM ***** BUILD SHAPE TABLE *****
6020 :
6030 AD = 2561:AA = AD:SNUM = 1:TL = 0
6040 TEXT : HOME : VTAB 21
6050 INVERSE : PRINT " *VIEW SHAPE TABLE* ?-IN
STRUCTIONS ";: NORMAL
6060 PRINT "TYPE (ESC) (RETURN) WHEN FINISHED"
6070 PRINT " C (RETURN) TO DO A CATALOG"
6080 PRINT " (ESC)(RETURN) TO ABORT ";
6090 REM -SET TEXT WINDOW-
6100 POKE 35,20
6110 HOME
6120 PRINT "NAME OF SHAPE ";:SNUM;": VECTORS/";
6130 LMT = 22: GOSUB 700
6140 IF W$ = "" OR W$ = CHR$(27) THEN PRINT : GOTO
6500

```

```

6150 IF W$ = "?" THEN POKE 35,24:LOC = 8: GOTO 80
00
6160 IF W$ = "C" THEN PRINT : PRINT D$;"CATALOG":
PRINT : GOTO 6120
6170 IF W$ = "END" THEN 6270
6180 LOC = 8: ONERR GOTO 5000
6190 PRINT : PRINT D$;"LOAD VECTORS/";W$;"A";AA:
POKE 216,0
6200 L(SNUM) = PEEK ( - 21920) + PEEK ( - 21919) *
256
6210 AA = AA + L(SNUM)
6220 SNUM = SNUM + 1
6230 IF SNUM < 256 THEN 6120
6240 PRINT : PRINT "THAT'S THE MAXIMUM NUMBER OF S
HAPE YOU CAN HAVE IN ONE TABLE. END TABLE 0
R ABORT? (E/A) ";: GET Z$
6250 IF Z$ = "E" THEN 6270
6260 GOTO 6500
6270 REM -POKE HEADER OF TABLE-
6280 SNUM = SNUM - 1:TL = 0: IF AA = AD THEN 6500
6290 FOR N = 1 TO SNUM
6300 TL = TL + L(N)
6310 NEXT N
6320 TTL = TL + SNUM * 2 + 2:TL = TTL
6330 FOR N = SNUM TO 1 STEP - 1
6340 TL = TL - L(N)
6350 HB = INT (TL / 256)
6360 LB = TL - HB * 256
6370 POKE AD - (SNUM - N) * 2 - 1,HB
6380 POKE AD - (SNUM - N) * 2 - 2,LB
6390 NEXT N
6400 POKE AD - SNUM * 2 - 1,0
6410 POKE AD - SNUM * 2 - 2,SNUM
6420 PRINT : PRINT "TABLE IS ";TTL;": BYTES LONG."
6430 PRINT "NAME FOR TABLE? TABLE/";
6440 LMT = 22: GOSUB 700
6450 IF W$ = "" OR W$ = CHR$(27) THEN PRINT : GOTO
6500
6460 IF W$ = "?" THEN LOC = 9: GOTO 8000
6470 LOC = 9: ONERR GOTO 6320
6480 PRINT : PRINT D$;"SAVE TABLE/";W$;"A";AD -
SNUM * 2 - 2;"L";TTL
6490 POKE 216,0
6500 PRINT "DO YOU WANT TO DO ANOTHER? (Y/N) ";: GET
Z$
6510 IF Z$ = "Y" THEN SNUM = 1:AA = AD: GOTO 6110
6520 POKE 35,24
6530 GOTO 400
7000 :
7010 REM ***** VIEW SHAPE TABLE *****
7020 :
7030 HOME : VTAB 21: HGR :LOC = 10
7040 INVERSE : PRINT " *VIEW SHAPE TABLE* ?-IN
STRUCTIONS ";: NORMAL
7050 PRINT "TYPE (ESC) (RETURN) TO ABORT"
7060 PRINT "NAME OF TABLE? TABLE/";
7070 LMT = 22: GOSUB 700
7080 IF W$ = "" OR W$ = CHR$(27) THEN 400
7090 IF W$ = "?" THEN 8000
7100 VTAB 23: PRINT
7110 ONERR GOTO 5000
7120 PRINT D$;"LOAD TABLE/";W$;"A 2050"
7130 POKE 216,0
7140 POKE 232,2: POKE 233,0
7150 SNUM = PEEK (2050)
7160 LOC = 11
7170 CC = 3: HCOLOR= CC:RR = 0: ROT= RR:SS = 1: SCALE=
SS
7180 X = 140:Y = 80:NN = 1
7190 HOME : VTAB 21
7200 INVERSE : PRINT " *VIEW SHAPE TABLE* ?-IN
STRUCTIONS ";: NORMAL
7210 PRINT " (ESC) WHEN FINISHED"
7220 PRINT "NUMBER OF SHAPE (1 ) S-CALE (1
)"
7230 PRINT "C-HANGE COLOR (3 ) R-OTATION (0
)";
7240 HCOLOR= CC: DRAW NN AT X,Y
7250 GET Z$
7260 IF Z$ = CHR$(27) THEN 400
7270 IF Z$ = "?" THEN 8000
7280 HCOLOR= 0: DRAW NN AT X,Y
7290 IF Z$ < > "N" THEN 7340
7300 NN = NN + 1: IF NN > SNUM THEN NN = 1
7310 VTAB 23: HTAB 19: PRINT " ";
7320 HTAB 19: PRINT NN;: VTAB 24: HTAB 40
7330 GOTO 7240
7340 IF Z$ < > "S" THEN 7390
7350 SS = SS + 1: IF SS > 10 THEN SS = 1
7360 VTAB 23: HTAB 37: PRINT " ";
7370 HTAB 37: PRINT SS;: VTAB 24: HTAB 40
7380 SCALE= SS: GOTO 7240
7390 IF Z$ < > "R" THEN 7440
7400 RR = RR + 1: IF RR > 63 THEN RR = 0
7410 VTAB 24: HTAB 37: PRINT " ";
7420 HTAB 37: PRINT RR;: HTAB 40
7430 ROT= RR: GOTO 7240
7440 IF Z$ < > "C" THEN 7500
7450 CC = CC + 1: IF CC > 6 THEN CC = 1
7460 IF CC = 4 THEN CC = 5
7470 VTAB 24: HTAB 19: PRINT " ";
7480 HTAB 19: PRINT CC;: HTAB 40
7490 GOTO 7240
7500 REM -MOVE SHAPE AROUND-
7510 IF Z$ = "I" THEN Y = Y - 5: IF Y < 0 THEN Y =
159
7520 IF Z$ = "J" THEN X = X - 5: IF X < 0 THEN X =
279
7530 IF Z$ = "K" THEN X = X + 5: IF X > 279 THEN X
= 0

```

```

7540 IF Z% = 'M' THEN Y = Y + 5: IF Y > 159 THEN Y
      = 0
7550 GOTO 7240
8000 :
8010 REM ***** INSTRUCTIONS *****
8020 :
8030 TEXT : HOME
8040 IF LOC = 1 THEN 8090
8050 ON LOC GOSUB 8300,8600,8600,8990,8990,9150,91
50,9320,9320,9320,9320
8060 UTAB 24: PRINT "(PUSH <ESC> TO RETURN, 'M' FO
R MORE) ";
8070 GET Z%: IF Z% = CHR% (27) THEN 8230
8080 IF Z% < > "M" THEN 8070
8090 FOR N = 1 TO 5
8100 IF N < > 2 AND N < > 5 THEN 8150
8110 HOME : UTAB 10
8120 IF N = 2 THEN PRINT "*** SECTION ONE: EDIT
VECTOR FILES ***: GOTO 8140
8130 PRINT "*** SECTION TWO: BUILD SHAPE TABLE **
*"
8140 FOR D = 1 TO 2500: NEXT D
8150 HOME : ON N GOSUB 8300,8600,8990,9150,9320
8160 UTAB 24
8170 IF N < > 5 THEN PRINT "(PUSH <ESC> TO RETUR
N, 'M' FOR MORE) ";
8180 IF N = 5 THEN PRINT "(PUSH <ESC> TO RETURN,
'R' TO REPEAT) ";
8190 GET Z%: IF Z% = CHR% (27) THEN 8230
8200 IF N < > 5 AND Z% < > "M" THEN 8190
8210 IF N = 5 AND Z% < > "R" THEN 8190
8220 NEXT N: GOTO 8090
8230 HOME : POKE - 16384,0
8240 IF LOC = 3 OR LOC = 4 OR LOC = 5 OR LOC = 7 THEN
INVERSE : UTAB 21: PRINT "*EDIT VECTOR FILES
* ?-INSTRUCTIONS *": NORMAL
8250 ON LOC GOTO 400,2040,2180,3400,3600,3900,4200
,6040,6040,7000,7190
8300 REM -PAGE ONE-
8310 PRINT : PRINT TAB(8); "*** NIBBLE ILLUSTRATO
R ***": PRINT
8320 PRINT " THIS GRAPHICS UTILITY PROGRAM MAKES
IT";
8330 PRINT "EASY TO CONSTRUCT COMPLEX SHAPE TABLES
";
8340 PRINT "FOR USE IN YOUR OWN PROGRAMS.": PRINT
8350 PRINT " SECTION 1 'EDIT VECTOR FILES' IS USE
D";
8360 PRINT "TO DESIGN SHAPES. THE SET OF VECTORS"
8370 PRINT "DESCRIBING THE SHAPE IS SAVED TO DISK,
";
8380 PRINT "CREATING A DISK LIBRARY OF SHAPES.": PRINT
8390 PRINT " SECTION 2 'BUILD SHAPE TABLE' COMBIN
ES";
8400 PRINT "SHAPES FROM THIS LIBRARY INTO A FINISH
ED";
8410 PRINT "SHAPE TABLE.": PRINT
8420 PRINT " FOR MORE INFORMATION ON SHAPE TABLES
";
8430 PRINT "AND HOW TO USE THEM SEE PAGES 91 TO 10
0";
8440 PRINT "OF THE APPLESOFT MANUAL."
8450 UTAB 24: PRINT "(PUSH <ESC> TO RETURN, 'M' FO
R MORE) ";
8460 GET Z%: IF Z% = CHR% (27) THEN POP : GOTO 8
230
8470 IF Z% < > "M" THEN 8460
8480 HOME : PRINT " AT MOST LOCATIONS IN THE PROG
RAM YOU"
8490 PRINT "MAY ESCAPE FROM THE FUNCTION BEING USE
D";
8500 PRINT "BY PUSHING THE <ESC> KEY. AT LOCATIONS
";
8510 PRINT "WHERE YOU ARE EXPECTED TO INPUT A FULL
";
8520 PRINT "WORD, SUCH AS THE NAME FOR A FILE, TO"
8530 PRINT "ESCAPE YOU MUST ENTER <ESC> <RETURN>."
: PRINT
8540 PRINT " YOU MAY ALSO GET INSTRUCTIONS FROM"
8550 PRINT "ANYWHERE IN THE PROGRAM BY ENTERING A"
8560 PRINT "?", AND PUSHING <RETURN> IF REQUIRED.
": PRINT
8570 PRINT " WHAT FOLLOWS IS A DESCRIPTION OF EAC
H"

```

```

8580 PRINT "FUNCTION IN BOTH OF THE SECTIONS."
8590 RETURN
8600 REM -PAGE TWO-
8610 PRINT "PLOTING": PRINT
8620 PRINT " THE CURSOR MAY BE MOVED ANYWHERE IN"
8630 PRINT "THE 32 X 44 SQUARE PLOTTING AREA BEFOR
E"
8640 PRINT "STARTING. USE THE I,J,K & M KEYS TO MO
VE";
8650 PRINT "UP, LEFT, RIGHT AND DOWN RESPECTIVELY.
";
8660 PRINT "PUSH THE SPACE BAR WHEN YOU HAVE REACH
ED";
8670 PRINT "YOUR STARTING POINT.": PRINT
8680 PRINT " PLOT POINTS BY USING THE I,J,K & M "
8690 PRINT "KEYS AGAIN TO MOVE THE CURSOR. PUSHIN
G"
8700 PRINT "JUST THE KEY WILL MOVE THE CURSOR IN T
HE";
8710 PRINT "PROPER DIRECTION, CREATING A 'NO-PLOT'
";
8720 PRINT "VECTOR. HOLDING DOWN THE 'CTRL' KEY AN
D"
8730 PRINT "PUSHING I,J,K OR M WILL PLOT THE BLOCK
";
8740 PRINT "THE CURSOR IS ON BEFORE MOVING IN THE"
8750 PRINT "APPROPRIATE DIRECTION. THIS CREATES A
";
8760 PRINT "'PLOT' TYPE VECTOR."
8770 UTAB 24: PRINT "(PUSH <ESC> TO RETURN, 'M' FO
R MORE) ";
8780 GET Z%: IF Z% = CHR% (27) THEN POP : GOTO 8
230
8790 IF Z% < > "M" THEN 8780
8800 HOME : PRINT " ON THE RIGHT HAND SIDE OF THE
SCREEN"
8810 PRINT "THE SHAPE YOU ARE PLOTTING IS DISPLAYE
D"
8820 PRINT "WITH ROTATION SET TO ZERO AND SCALE SE
T"
8830 PRINT "TO ONE. BELOW THIS IS THE SAME SHAPE"
8840 PRINT "DISPLAYED WITH ROTATION SET TO 16. THE
";
8850 PRINT "COLOR OF THESE SHAPES CAN BE CHANGED B
Y"
8860 PRINT "PUSHING THE 'C' KEY.": PRINT
8870 PRINT " IF YOU MAKE AN ERROR YOU CAN BACK UP
";
8880 PRINT "AND 'UN-PLOT' ONE VECTOR AT A TIME BY"
8890 PRINT "PUSHING THE 'B' KEY.": PRINT
8900 PRINT " YOU MAY ALSO RE-PLOT THE ENTIRE SHAP
E"
8910 PRINT "BY PUSHING THE 'R' KEY. THE CURSOR IS
";
8920 PRINT "THEN POSITIONED WHERE YOU WANT TO BEGI
N"
8930 PRINT "PLOTTING AND THE BLOCKS PLOTTED IN THE
";
8940 PRINT "USUAL WAY. THE OLD SHAPE REMAINS ON TH
E"
8950 PRINT "SCREEN AS A GUIDE.": PRINT
8960 PRINT " TO START ALL OVER AGAIN PUSH THE 'E'
";
8970 PRINT "KEY TO ERASE THE SHAPE AND ITS VECTORS
";
8980 RETURN
8990 REM -PAGE THREE-
9000 PRINT "SAVE SHAPE.": PRINT
9010 PRINT " THIS SAVES TO DISK THE FILE OF VECTO
RS";
9020 PRINT "THAT DESCRIBES THE SHAPE. PUSHING <ESC
>"
9030 PRINT "<RETURN> ESCAPES BACK TO THE PLOTTING"
";
9040 PRINT "FUNCTION.": PRINT : PRINT : PRINT
9050 PRINT "LOAD SHAPE.": PRINT
9060 PRINT " WHEN LOADING A VECTOR FILE FROM DISK
";
9070 PRINT "THE SHAPE IS PLOTTED OUT FOR USE IN TH
E"
9080 PRINT "RE-PLU+I FUNCTION, ETC. IT IS DRAWN FRO
M"
9090 PRINT "THE CURSOR LOCATION. IF IT WILL NOT FI
T"
9100 PRINT "ON THE SCREEN YOU WILL BE PROMPTED TO"
9110 PRINT "MOVE THE CURSOR AND TRY AGAIN."
9120 PRINT " USE THE <ESC> KEY TO ESCAPE BACK TO"

```

```
9130 PRINT "THE PLOTTING FUNCTION."  
9140 RETURN  
9150 REM -PAGE FOUR-  
9160 PRINT "DOS COMMANDS:"; PRINT  
9170 PRINT " TWO COMMANDS ARE AVAILABLE: 'CATALOG  
"  
9180 PRINT "AND 'DELETE'. NOTE THAT DELETE DOES NO  
T"  
9190 PRINT "SUPPLY A PREFIX SUCH AS 'VECTORS/' OR"  
9200 PRINT "'TABLE/' FOR THE FILENAME OF THE FILE"  
9210 PRINT "TO BE DELETED."  
9220 PRINT " USE <ESC> TO RETURN TO THE PLOTTING"  
9230 PRINT "FUNCTION.": PRINT ; PRINT  
9240 PRINT "VIEW SHAPE:"; PRINT  
9250 PRINT " SHAPES MAY BE VIEWED WITH CONTROL OV  
ER";  
9260 PRINT "THEIR COLOR, SCALE AND ROTATION. NOTE  
"  
9270 PRINT "THAT AT SMALLER SCALE FACTORS THERE AR  
E"  
9280 PRINT "FEWER ANGLES OF ROTATION AVAILABLE."  
9290 PRINT " USE <ESC> TO RETURN TO THE PLOTTING"  
9300 PRINT "FUNCTION."  
9310 RETURN  
9320 REM -PAGE FIVE-  
9330 PRINT "BUILD TABLE:"; PRINT  
9340 PRINT " WHEN PROMPTED, TYPE IN THE NAMES OF"
```

```
9350 PRINT "THE VECTOR FILES YOU WANT COMBINED INT  
O"  
9360 PRINT "A SHAPE TABLE. TO INDICATE THE END OF  
A"  
9370 PRINT "TABLE ENTER 'END' <RETURN>. YOU CAN DO  
"  
9380 PRINT "A CATALOG OF THE DISK BY TYPING 'C' AN  
D"  
9390 PRINT "<RETURN>; AND TO ABORT BUILDING A TABL  
E"  
9400 PRINT "ENTER <ESC> <RETURN>."; PRINT ; PRINT  
; PRINT  
9410 PRINT "VIEW TABLE:"; PRINT  
9420 PRINT " YOU MAY LOAD A SHAPE TABLE AND VIEW"  
9430 PRINT "ANY OF ITS SHAPES WITH CONTROL OVER TH  
E"  
9440 PRINT "COLOR, SCALE AND ROTATION OF THE SHAPE  
."  
9450 PRINT "NOTE THAT AT SMALLER SCALE FACTORS THE  
RE";  
9460 PRINT "ARE FEWER ANGLES OF ROTATION AVAILABLE  
."  
9470 PRINT " USE <ESC> TO RETURN TO THE MAIN MENU  
."  
9480 RETURN
```

