

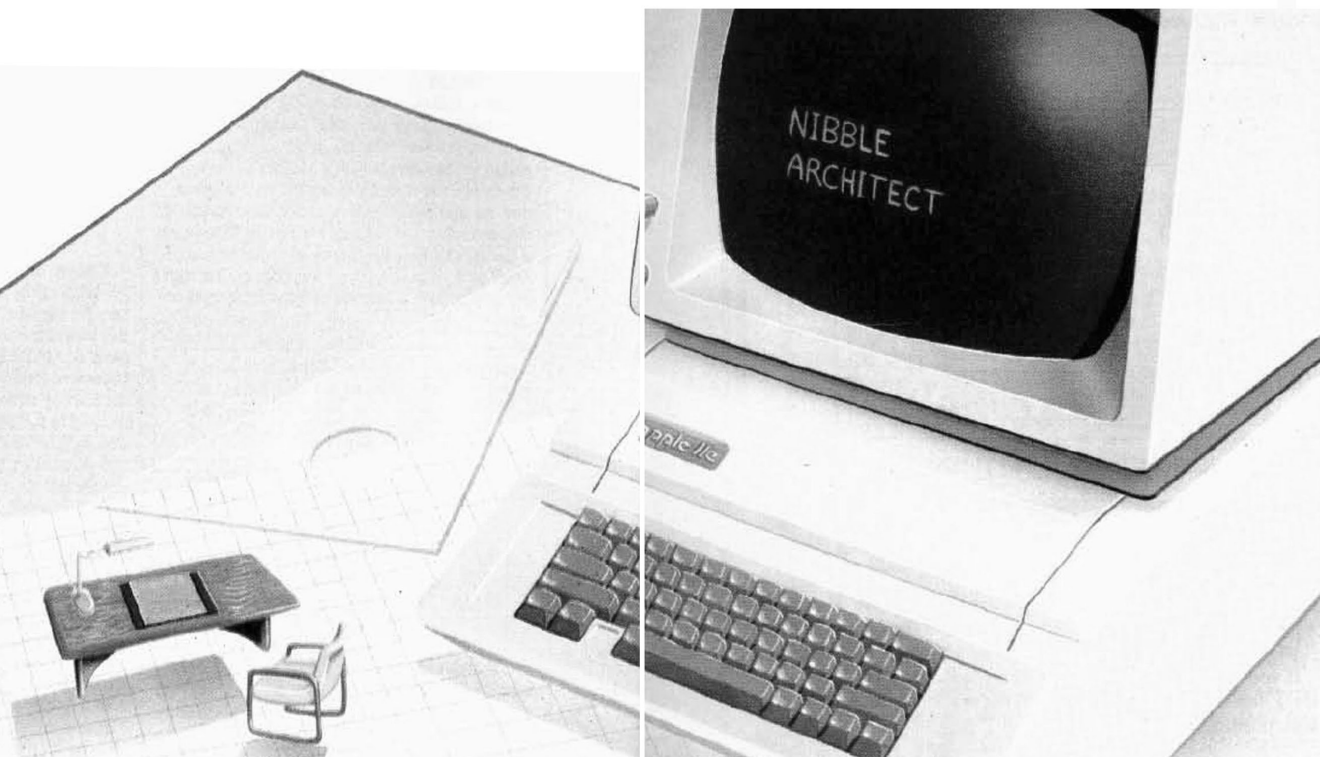
THE NIBBLE ARCHITECT

FEATURE ARTICLE

DOS 3.3



Designing the furniture layout for your living room or office is a snap with this Applesoft program that runs under DOS 3.3. Using Hi-Res graphics, you can easily visualize different layouts, save them on disk for later editing, or print them in a variety of sizes.



by I. Larry Morris, 190 Woodcliff Rd.,
Newton, MA 02161

Do you recall the first time you tried to rearrange the furniture in your bedroom? You moved all the furniture around, only to discover that one of the night tables wouldn't fit in the room. So you tried a few more variations, but then there was no room for your favorite chair. Finally, you managed to find a place for everything, but you couldn't open the closet door. In desperation you decided to sketch out a floor plan on graph paper. The first few attempts had to be discarded because you chose the wrong scale. Three or four sketches later you realized that the original arrangement was the best one after all.

The NIBBLE ARCHITECT was designed to overcome all of these problems and frustrations. Using the principles behind professional computer-aided design systems, the NIBBLE ARCHITECT allows you to develop floor plans with ease and flexibility. You can create, modify, and manipulate up to fifty items per room (including doors and windows) while maintaining accuracy to the nearest inch.

RUNNING THE NIBBLE ARCHITECT

The NIBBLE ARCHITECT performs three major functions: graphics editing, disk file management, and hardcopy generation.

When you run the program, the operations and commands associated with the graphics editor appear in menu form as shown in Figure 1. In order to get started you are prompted to select either N or O.

N — create a NEW room: Select N to display the data input screen. You can enter and modify values for the width and length of the room, the spacing between grid lines, and the title. If numerical values exceed 99 feet, two asterisks (**) appear in the relevant position on the screen. The program will proceed into the graphics editing mode only if numerical values are greater than zero and less than 99.

FIGURE 1: The Editor Commands

FUNCTIONS ASSIGNED TO THE KEYBOARD:

A - ALTER	K - RIGHT	U - UNLATCH
B - BOX	L - LATCH	V - VERIFY
C - COPY	M - DOWN	W - WINDOW
D - DOOR	N - NEW	X - EXIT W/O DOOR
E - ERASE	O - OLD	Y - YES
F - FIX	P - PLACE	GRID & TITLE
G - GET DATA	Q - QUIT	# - CIRCLE
H - HARDCOPY	R - REDRAW	< - SMALLER STEP
I - UP	S - SAVE	> - LARGER STEP
J - LEFT	T - TURN	? - HELP

PRESS 'N' OR 'O' TO CONTINUE...?

O — get an OLD room from the disk: Select O to display the disk file management screen. An old room can be obtained from the disk by selecting the GET option. If you enter a file number that is out of range, an error message will appear. Similarly, selecting the SAVE and REVISE options at this time will also produce error messages.

The file number and name will flash while a file is being accessed. After the GET is performed, the file number and name will be highlighted. At this point, the SAVE and REVISE options become valid. Press the space bar to display the old room. If you depress the space bar prior to getting an old room, the effect would be the same as selecting NEW.

GRAPHICS EDITING

The lower right-hand corner of the Hi-Res screen displays the word EDIT during edit mode. Once an editing operation has been selected, EDIT is replaced by the corresponding letter, which is displayed until the operation is complete.

Cursor Control

The program uses two cursors — a moving cursor (x) and a fixed cursor (+). The fixed cursor establishes a reference point, while the moving cursor specifies a relative displacement or orientation (initially, they are both positioned at the center of the room). The x cursor moves in steps of 1, 2, or 3 inches and cannot move outside of the outer walls of the room.

Press I, J, K, M, <, or > to control the moving cursor. Press F to control the fixed cursor. G displays the numerical values of the displacements. A description of these commands follows.

- I Move the x cursor one step up.
- J Move the x cursor one step left.
- K Move the x cursor one step right.
- M Move the x cursor one step down.
- < Choose the next smaller step size (the step size is decremented by one).
- > Choose the next larger step size (the step size is incremented by one).
- F Fix the + cursor to the current position of the x cursor.
- G Get data values associated with the cursors.

When you press G, the horizontal, vertical, and diagonal distances between the moving (x) and fixed (+) cursors are displayed. The I, J, K, M, <, and > commands are still in effect. When the x cursor enters any of the outer walls, one of the displayed values will flash.

Creating Items

Graphics objects which represent furniture, doors, and windows are created by manipulating the fixed and moving cursors and using the single letter command keys described below. Once an item has been created, it can be moved to a new location with an additional set of single letter commands.

The size of an item you create is determined by the relative position of the fixed (x) and moving (+) cursors. The type of item is determined by the commands B, *, D, W, and X. No item will be created if the x and + cursors are coincident or if the x cursor has not moved since the last item was created. Boxes and circles can be created only when the fixed cursor is inside the room. Doors and windows can be created only when the fixed cursor is located within the outer walls.

B — create a BOX: Opposite corners of the box correspond to the fixed (+) and moving (x) cursors.

*** — create a CIRCLE:** The circle is centered at the fixed (+) cursor and its perimeter passes through the moving (x) cursor.

D — create a doorway with a DOOR: The fixed (+) cursor locates the pivot point of the door, which opens into the room. The width of the door is determined by the relative displacement between the fixed (+) and moving (x) cursors.

W — create a WINDOW: The width of the window is determined by the relative displacement between the fixed (+) and moving (x) cursors.

X — create an EXIT W/O a DOOR: X performs the same function as D except that no door is drawn.

Modifying and Repositioning Items

Items to be modified or repositioned must first be selected (i.e., latched) and then later released (i.e., unlatched). The commands A, C, E, P, T, U, and V affect only the latched items. If the fixed cursor is located inside the room, then only latched boxes and circles are affected. If the fixed cursor is located within the outer walls, only latched doors and windows are affected (a distinction is also made between horizontal and vertical walls).

L — LATCH an item: An item is latched if the moving (x) cursor is within its boundary. A latch cursor (a box) will appear at the center of the item.

U — UNLATCH all LATCHed items: This command causes all of the latch cursors to disappear.

A — ALTER the LATCHed items: The latched item to be altered is identified by a plus sign that appears in the center of the latch cursor. I, J, K, M, <, >, F, and G are still in effect (thus, complete cursor control is available). Pressing A will make the old item disappear and the altered version appear. However, if the fixed and moving cursors are coincident or the moving cursor has not moved since the last item was created or altered, then A will skip to the next latched item for alteration.

C — COPY the LATCHed items: All of the latched items can be copied to a new position determined by the displacement vector from the fixed (+) cursor to the moving (x) cursor. However, movement of doors and windows is restricted to the outer walls.

E — ERASE the LATCHed items: This command lets you erase an item. You are given the opportunity to cancel the erase operation before it has been finalized.

P — disPLACE the LATCHed items: All of the latched items can be displaced to a new position determined by the displacement vector from the fixed (+) to the moving (x) cursor. Even if an item ends up completely outside the room, its latch cursor will be located within the outside wall closest to the item. The presence of the latch cursor is a reminder of the item's existence and the fact that it can still be retrieved.

T — TURN the LATCHed items: Only boxes and circles are affected by this command. If the fixed and moving cursors are coincident, then the latched items are turned 90 degrees counterclockwise about the common cursor position. Otherwise, each latched item is turned about its own center so it is parallel to the line passing through the fixed and moving cursor positions.

V — VERIFY the dimensions of the LATCHed items: This command displays a table that gives the dimensions of the room and up to five latched items.

Additional Features

R — REDRAW the room and its contents: As the various items are modified or repositioned, portions of the picture may disappear. The R command will completely redraw the picture. This command purges the room of any items that have been erased or positioned outside of the room and subsequently unlatched.

— modify the GRID size and TITLE: Both the grid size and title can be modified or turned off (i.e., not displayed) by the '#' command. The title refers to the text that appears at the bottom of the Hi-Res screen. Turning off the title removes both it and the logo on the right-hand side of the screen. If any modification is made to the grid or title, then the picture will be redrawn when returning to the graphics editing mode. The program will not return to the graphics editing mode if the grid size is zero.

? — display the **HELP** menu: It is possible to recall the various editing commands while in the graphics editing mode. Type a question mark (?) to display a menu of the editing commands.

DISK FILE MANAGEMENT

The S command (as well as O) invokes the Disk File Manager. It allows you to save to disk the data for a given room for later retrieval, revision or deletion. The NIBBLE ARCHITECT can handle up to 42 rooms worth of data when no other programs reside on the same disk.

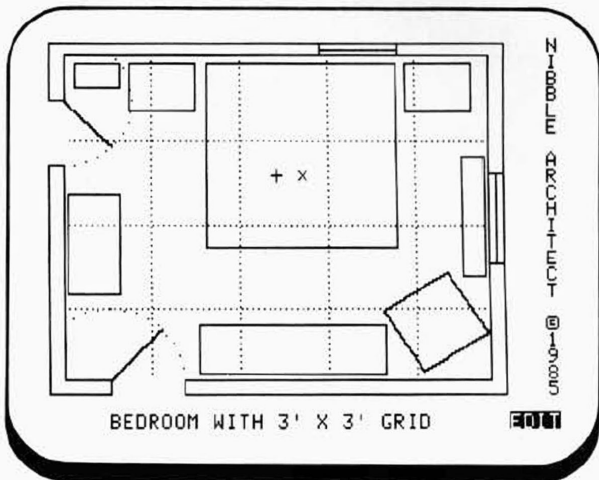
S — **SAVE the room data on the disk:** The Disk File Manager screen is displayed. The DELETE and GET options prompt the user for the file number while SAVE prompts for the file name. REVISE refers to the file number and name that is highlighted (the result of a previous GET or SAVE). After removing commas, colons, and trailing blanks, the file name for SAVE is compared to the highlighted file name and you are informed if there is a match. After deleting a file, the table of file names is automatically renumbered to close up any gaps.

NEW permits you to create a new room. QUIT terminates the program. For both of these, you are asked to confirm your action.

HARDCOPY GENERATION

The H command lets you obtain a hardcopy of the Hi-Res screen using the MX-80 printer with Graftrax.

FIGURE 2: Bedroom with 3' X 3' Grid



H — generate **HARDCOPY:** Options are available to produce either 3 in. x 4 in. or 6 in. x 8 in. pictures. You can also choose to have the picture redrawn before printing (command R). Printing can be terminated by pressing the space bar.

If you have a different printer or a special printing facility, you are given the opportunity to store the Hi-Res screen on disk for later printing.

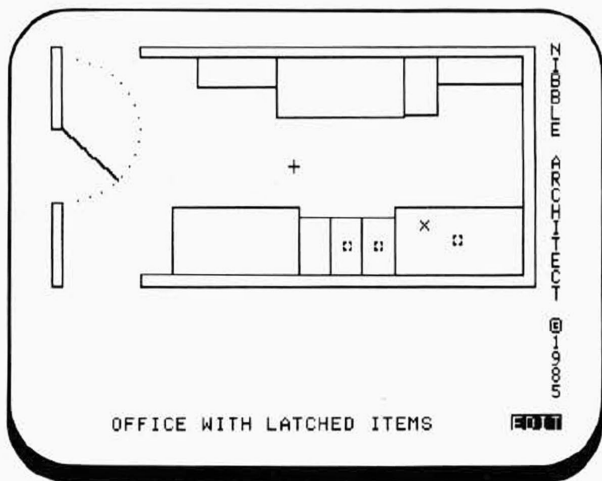
ENDING THE SESSION

Q — **QUIT NIBBLE ARCHITECT:** The NIBBLE ARCHITECT program can be terminated either from the graphics editor or the Disk File Manager by selecting Q. In either case, you are asked to confirm your action.

SOME EXAMPLES

Let us consider several examples that illustrate the capabilities of the NIBBLE ARCHITECT. Figure 2 shows a typical bedroom with the grid lines present. Note that the night table in the upper left-hand corner partially blocks the closet door.

FIGURE 3: Office with Latched Items



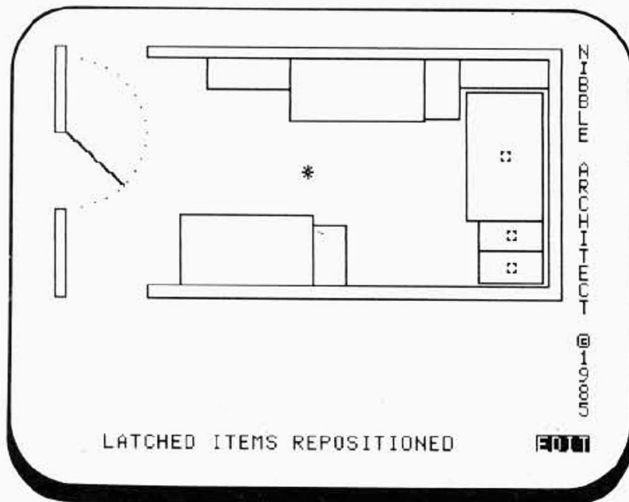
The second example illustrates the effect of moving furniture around in an office. Figure 3 shows an office with a desk and two cabinets latched for repositioning. Figure 4 is then produced by turning these items 90 degrees and placing them in the lower right-hand corner.

You can efficiently assemble an office area that contains several cubicles with the same arrangement of furniture by using the Copy function. Such a plan is shown in Figure 5, where the repetitive combination of desk, chair, cabinet, bookcase, and partition is identified by the latch cursors.

ENTERING THE PROGRAM

The NIBBLE ARCHITECT is made up of three components: an Applesoft program, a text file containing menu text and cursor shape data, and a shape table for Hi-Res characters. Since the NIBBLE ARCHITECT generates several data files that require disk space, it would be best to set up a "work disk" which originally contains just these three files. Because of the length of the program and the techniques used to store variables, the NIBBLE ARCHITECT will

FIGURE 4: Latched Items Repositioned



not run under ProDOS.

To key in the NIBBLE ARCHITECT, first enter the Applesoft program shown in Listing 1 and save it on disk with the command:

SAVE ARCHITECT

Next, enter the program shown in Listing 2 and save it on disk with the command:

SAVE MENU.CREATE

You should then run this program to create the file containing the menu text called ARCHITECT.MENU.

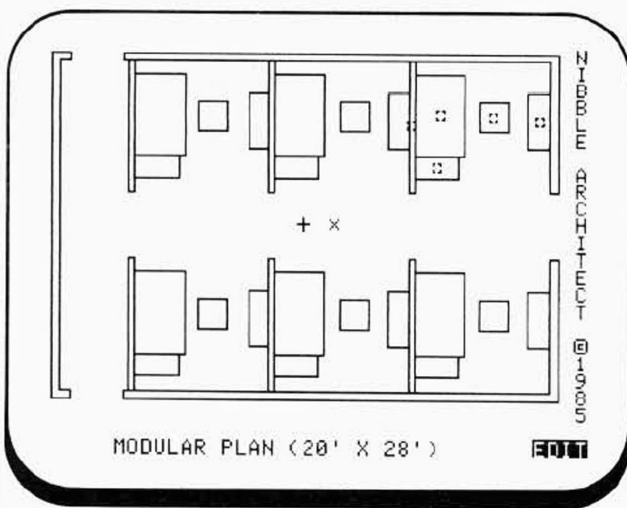
Finally, enter the program shown in Listing 3 and save it on disk with the command:

SAVE ALPH.CREATE

You should then run this program to create the shape table file called ARCHITECT.ALPH.

MENU.CREATE and ALPH.CREATE are no longer needed once you have run them to create the text file and shape table. To use the NIBBLE ARCHITECT, you need only type RUN ARCHITECT with a disk containing ARCHITECT, ARCHITECT.ALPH, and ARCHITECT.MENU in the drive.

FIGURE 5: Modular Plan



LISTING 1: ARCHITECT

```

10 REM *****
20 REM * ARCHITECT *
30 REM * BY I. LARRY MORRIS *
40 REM * COPYRIGHT (C) 1985 *
50 REM * BY MICROSPARC, INC *
60 REM * CONCORD, MA. 01742 *
70 REM *****
80 HOME : VTAB 6: HTAB 12: PRINT "NIBBLE ARC
HITECT": IF PEEK (104) < > 64 THEN POKE
103,1: POKE 104,64: POKE 16384,0: PRINT
CHR$ (4)"RUN ARCHITECT"
90 POKE 105,0: POKE 106,8: POKE 107,0: POKE
108,8: POKE 109,0: POKE 110,8
100 ST$ = "T": SA = 8192: BY = 0: TR = 0: OC = 0:
FL = 0: N1 = 1024: N1% = 255: N2% = 128: N3%
= 32: N2 = 8063: N3 = 984: ES = CHR$ (27)
110 ST = 2: MV = 0: FH = 0: FV = 0: AS = .86: HS =
3: VS = 3: C = 0: TF = 0: TI = 0
120 CS = "Z": XC = 0: YC = 0: XF = 0: YF = 0: XS =
0: YS = 0: L = 0: N = 0: CL = 3: CR = 0: F = 0
:G = 0: P5 = .5: Q$ = CHR$ (34): PR = .017
453: P4 = .7854: A = 0: WS = 6
130 GOTO 280
140 REM MOVING CURSOR
150 XDRAW 3 AT HS + HR * XC,VS + VR * YC: RETURN
160 REM FIXED CURSOR
170 XDRAW 2 AT HS + HR * XF,VS + VR * YF: RETURN
180 REM MOVES
190 CA = PEEK ( - 16384) - 128: IF CA < 0 THEN
190
200 POKE - 16368,0
210 IF CA = 73 AND YC > Y1 THEN GOSUB 140:Y
C = YC - ST: FL = 1: FV = 0: RETURN
220 IF CA = 74 AND XC > X1 THEN GOSUB 140:X
C = XC - ST: FL = 1: FH = 0: RETURN
230 IF CA = 75 AND XC < X2 THEN GOSUB 140:X
C = XC + ST: FL = 1: FH = 0: RETURN
240 IF CA = 77 AND YC < Y2 THEN GOSUB 140:Y
C = YC + ST: FL = 1: FV = 0: RETURN
250 IF CA = 74 OR CA = 75 THEN FH = 1
260 IF CA = 73 OR CA = 77 THEN FV = 1
270 FL = 0: RETURN
280 REM PREAMBLE
290 GOSUB 770
300 GET CS
310 IF CS = "N" THEN GOSUB 3190
320 IF CS = "O" THEN GOSUB 4480
330 IF N = 0 THEN 300
340 REM MONITOR
350 GOSUB 180
360 IF FL = 1 THEN MV = 1: GOSUB 140: GOTO 3
50
370 CS = CA - 64
380 IF CA = 42 THEN GOSUB 5750
390 IF CA = 35 THEN GOSUB 5830
400 IF CA = 60 THEN GOSUB 6130
410 IF CA = 62 THEN GOSUB 6160
420 IF CA = 63 THEN GOSUB 6190
430 IF CS < 0 THEN 350
440 ON CS GOSUB 1500,1730,1890,2050,2260,240
0,2420,2690,2990,2990,2990,3010,2990,319
0,4480,4050,4200,4240,4480,4920,5200,528
0,5450,5530
450 GOTO 350

```

```

460 REM BUSY
470 HCOLOR= 0: FOR I = 184 TO 191: HPLLOT 252
,I TO 279,I: NEXT : HCOLOR= 3
480 IV = 1: VL = 0: VT = 24: HT = 37: GOSUB 640:
RETURN
490 REM EDIT
500 IV = 1: VL = 0: VT = 24: HT = 37: ST$ = "EDIT
": GOSUB 640: RETURN
510 REM DRAW CIRCLE
520 XG = P5 * (A%(AI,2) + A%(AI,4)): YG = P5 *
(A%(AI,3) + A%(AI,5))
530 D = P5 * (A%(AI,4) - A%(AI,2)): D% = VR *
D: WA = 0: WB = 2 * WS: AG = 15 * PR
540 IF D% > 10 THEN AG = 12 * PR
550 IF D% > 25 THEN AG = 9 * PR
560 IF D% > 35 THEN AG = 8 * PR
570 IF D% > 45 THEN AG = 6 * PR
580 IF D% > 55 THEN AG = 5 * PR
590 FOR I = T1 TO T2 STEP AG
600 X = (XG + D * COS (I)): Y = (YG - D * SIN
(I))
610 IF X < WA OR Y < WA OR X > WB + L1 OR Y >
WB + W1 THEN 630
620 HPLLOT HS + HR * X,VS + VR * Y
630 NEXT : RETURN
640 REM ALPHABET
650 IF VL = 0 THEN BA = N2 + N2% * VT - N3 *
INT ((VT - 1) / 8) + HT
660 FOR I = 1 TO LEN (ST$): IF VL = 1 THEN
BA = N2 + N2% * VT - N3 * INT ((VT - 1)
/ 8) + HT
670 CH = ASC ( MIDS (ST$,I,1)) - N3%
680 IF CH = 0 THEN 730
690 IF IV = 1 THEN 720
700 FOR J = 0 TO 7: POKE BA + N1 * J,AL%(CH,
J): NEXT
710 GOTO 730
720 FOR J = 0 TO 7: POKE BA + N1 * J,N1% - A
L%(CH,J): NEXT
730 BA = BA + (VL = 0): VT = VT + (VL = 1): NEXT
: RETURN
740 REM MENU
750 PRINT L$(6)L$(7)L$(8)L$(9)L$(10)L$(11)L$
(12)L$(13)L$(14)L$(15)
760 RETURN
770 REM INITIALIZE
780 VTAB 12: PRINT SPC (39):: HTAB 13: FLASH
: PRINT " READING MENU ": NORMAL
790 D$ = CHR$ (4): REM CTRL-D
800 CR$ = CHR$ (13): REM C/R
810 AD = 0: LN = 1179
820 AD = PEEK (107) + 256 * PEEK (108)
830 PRINT D$"BLOAD ARCHITECT.ALPH,A"AD
840 POKE 110, INT ((AD + LN) / 256)
850 POKE 109,AD + LN - 256 * INT ((AD + LN)
/ 256)
860 NL = 67: DIM A%(52,7),L$(NL),D(5),F$(5),E
(1),G$(3),V%(383)
870 DEF FN IN(W) = XF > = WS AND XF < = W
S + L1 AND YF > = WS AND YF < = WS + W
1
880 DEF FN OT(X) = XF < = WS OR XF > = WS
+ L1 OR YF < = WS OR YF > = WS + W1
890 DEF FN HO(U) = A%(U,5) - A%(U,3) < WS +
.1: DEF FN VE(V) = A%(V,4) - A%(V,2) <
WS + .1
900 DEF FN BB(B) = MX + (B > MX) + B * (B >
= 0 AND B < = MX)
910 DEF FN BD(D) = MY * (D > MY) + D * (D >
= 0 AND D < = MY)
920 DEF FN VH(J) = A%(I,J) < WS - ST: DEF FN
RH(J) = A%(I,J) > WS + L1 + ST
930 DEF FN BV(J) = A%(I,J) > WS + W1 + ST
940 MS = "ARCHITECT.MENU": PRINT D$"OPEN "MS:
PRINT D$"READ "MS

```



```

950 FOR I = 0 TO 1:L$(I) = ""
960 GET CS: IF CS < > CR$ THEN L$(I) = L$(I
) + CS: GOTO 960
970 X = FRE (0): NEXT
980 FOR I = 2 TO NL: INPUT L$(I): NEXT
990 FOR H = 832 TO 880: INPUT X: POKE H,X: NEXT

1000 FOR I = 1 TO 2: INPUT X,Y: POKE X,Y: NEXT

1010 FOR J = 768 TO 831: INPUT X: POKE J,X: NEXT

1020 PRINT D$"CLOSE "M$: SCALE= 1
1030 ONERR GOTO 1110
1040 DIM DF$(42):M$ = "ARCHITECT.DIR": PRINT
D$"OPEN "M$: PRINT D$"READ "M$
1050 INPUT DF$(0):DF = VAL (DF$(0)): IF DF =
0 THEN 1070
1060 FOR I = 1 TO DF: INPUT DF$(I): NEXT
1070 PRINT D$"CLOSE "M$
1080 VTAB 1: PRINT L$(3)+" SPC( 38)+"L$(4)
"+" SPC( 38)+"L$(5)+" SPC( 38)+"L$(3)
1090 VTAB 8: CALL - 958: PRINT : PRINT L$(0
): PRINT : GOSUB 740: PRINT : PRINT L$(1
6):
1100 RETURN
1110 CALL - 3288: POKE 216,0: PRINT D$"CLOS
E":DF = 0: GOSUB 4810: GOTO 1080
1120 REM INPUT ANYTHING
1130 CALL 54572: FOR H = 512 TO 768: IF PEEK
(H) < > 0 THEN 1150
1140 IN$ = "":AD = VAL (IN$) + PEEK (131) +
256 * PEEK (132): POKE AD,H - 512: POKE
AD + 1,0: POKE AD + 2,2:IN$ = MID$( IN$
,1):H = 768
1150 NEXT : RETURN
1160 REM TEXT
1170 HOME : POKE - 16303,0: RETURN
1180 REM GRAPHICS
1190 POKE - 16304,0: RETURN
1200 REM FIXED & MOVING CURSORS
1210 GOSUB 140: GOSUB 160: RETURN
1220 REM RECTANGLE
1230 MX = A%(2,4):MY = A%(2,5)
1240 XL = A%(A1,2):YT = A%(A1,3):XR = A%(A1,4
):YB = A%(A1,5)
1250 IF A%(A1,7) < > 0 AND A%(A1,0) = 2 THEN
GOSUB 5080: RETURN
1260 XL = FN BB(XL):XR = FN BB(XR):YT = FN
BD(YT):YB = FN BD(YB)
1270 LX = HS + HR * XL:RX = HS + HR * XR:TY =
VS + VR * YT:BY = VS + VR * YB
1280 IF A%(A1,0) = 3 THEN GOSUB 5490: RETURN

1290 IF A%(A1,0) = 5 THEN T1 = 0:T2 = 359 *
PR: GOSUB 510: RETURN
1300 HPLLOT LX, TY TO RX, TY TO RX, BY TO LX, BY TO
LX, TY
1310 IF A%(A1,0) = 4 THEN GOSUB 5550
1320 RETURN
1330 REM GRID SPACING
1340 XM = 262:YM = 172
1350 HR = XM / (L1 + 2 * WS):VR = YM / (W1 +
2 * WS)
1360 IF VR < AS * HR THEN HR = VR / AS: GOTO
1380
1370 VR = AS * HR
1380 HD = S1 * HR:VD = S1 * VR:XZ = WS * HR:Y
Z = WS * VR
1390 XM = HR * (L1 + WS):YM = VR * (W1 + WS)
1400 X1 = WS - 1:X2 = L1 + WS + 1:Y1 = WS - 1
:Y2 = W1 + WS + 1: RETURN
1410 REM GRID LAYOUT
1420 IF YZ + VD > YM - 1 THEN 1460
1430 FOR I = VS + YZ + VD TO VS + YM - 1 STEP
VD
1440 FOR J = HS + XZ + HS TO HS + XM - 1 STEP
HS
1450 HPLLOT J, I: NEXT : NEXT
1460 IF XZ + HD > XM - 1 THEN RETURN
1470 FOR I = HS + XZ + HD TO HS + XM - 1 STEP
HD
1480 FOR J = VS + YZ + VS TO VS + YM - 1 STEP
VS
1490 HPLLOT I, J: NEXT : NEXT : RETURN

```

```

1500 REM A...ALTER
1510 IF L = 0 OR A%(0,1) = 2 THEN RETURN
1520 A = 1:ST$ = "A": GOSUB 460: FOR AI = 3 TO
A%(0,1):OB = A%(AI,0)
1530 IF A%(AI,1) = 0 THEN 1720
1540 GOSUB 3070
1550 GOSUB 180
1560 IF FL = 1 THEN MV = 1: GOSUB 140: GOTO
1550
1570 IF CA = 65 THEN 1630
1580 IF CA = 71 THEN GOSUB 2420
1590 IF CA = 70 THEN GOSUB 2400
1600 IF CA = 60 THEN GOSUB 6130
1610 IF CA = 62 THEN GOSUB 6160
1620 GOTO 1550
1630 IF MV = 0 THEN GOSUB 3070: GOTO 1720
1640 IF FN OT(XF) = 0 AND (OB = 3 OR OB = 4
) THEN 1550
1650 A = 2: GOSUB 1200: HCOLOR= 0:CR = 3:CL =
0: GOSUB 4330: HCOLOR= 3:CR = 0:CL = 3
1660 IF OB = 2 THEN GOSUB 1730
1670 IF OB = 3 THEN GOSUB 5450
1680 IF OB = 4 AND A%(A1,6) = 0 AND A%(A1,7)
= 0 THEN GOSUB 5530: GOTO 1710
1690 IF OB = 4 THEN GOSUB 2050
1700 IF OB = 5 THEN GOSUB 5750
1710 A = 0: GOSUB 3070:A = 1
1720 NEXT :A = 0: GOSUB 490: RETURN
1730 REM B...BOX
1740 IF MV = 0 THEN RETURN
1750 IF A = 0 THEN ST$ = "B": GOSUB 1820:A%(
AI,0) = 2
1760 A%(A1,2) = XC:A%(A1,4) = XF
1770 IF XC > XF THEN A%(A1,2) = XF:A%(A1,4) =
XC
1780 A%(A1,3) = YC:A%(A1,5) = YF
1790 IF YC > YF THEN A%(A1,3) = YF:A%(A1,5) =
YC
1800 IF A = 0 THEN A%(A1,6) = 1:A%(A1,7) = 0
1810 GOSUB 1220: GOSUB 1860: RETURN
1820 REM ADD
1830 IF C = 0 THEN GOSUB 460: GOSUB 1200
1840 AI = A%(0,1) + 1: IF AI > 52 THEN GOSUB
1160: VTAB 12: HTAB 10: FLASH : PRINT "
"LS(42): : NORMAL :DL = 3000: GOSUB 2380:
GOSUB 1180: GOSUB 490:C = 0: GOSUB 1200
: POP : RETURN
1850 A%(0,1) = AI:A%(AI,1) = 0: RETURN
1860 REM RETURN
1870 MV = 0: IF A = 0 THEN GOSUB 490
1880 GOSUB 1200: RETURN
1890 REM C...COPY
1900 IF L = 0 OR MV = 0 OR A%(0,1) = 2 THEN
RETURN
1910 C = 1:ST$ = "C": GOSUB 460: GOSUB 1200:X
H = XC - XF:YH = YC - YF
1920 FOR AN = 3 TO A%(0,1):OB = A%(AN,0)
1930 IF A%(AN,1) = 0 THEN 2040
1940 IF (OB = 3 OR OB = 4) AND FN OT(XF) =
0 THEN 2040
1950 IF (OB = 2 OR OB = 5) AND FN IN(XF) =
0 THEN 2040
1960 XE = XH:YE = YH
1970 IF (OB = 3 OR OB = 4) AND FN HO(AN) THEN
YE = 0: IF YF > WS AND YF < WS + W1 THEN
2040
1980 IF (OB = 3 OR OB = 4) AND FN VE(AN) THEN
XE = 0: IF XF > WS AND XF < WS + L1 THEN
2040
1990 GOSUB 1820:A%(A1,0) = A%(AN,0)
2000 A%(A1,2) = A%(AN,2) + XE:A%(A1,3) = A%(A
N,3) + YE
2010 A%(A1,4) = A%(AN,4) + XE:A%(A1,5) = A%(A
N,5) + YE
2020 A%(A1,6) = A%(AN,6):A%(A1,7) = A%(AN,7)
2030 GOSUB 4330
2040 NEXT :C = 0: GOSUB 1860: RETURN
2050 REM D...DOOR
2060 IF FN OT(XF) = 0 OR MV = 0 THEN RETURN
2070 ST$ = "D": IF F = 1 THEN ST$ = "X"
2080 IF A = 0 THEN GOSUB 1820:A%(A1,0) = 4
2090 A6 = 0:A7 = 0: GOSUB 2120
2100 IF F = 1 THEN A%(A1,6) = 0:A%(A1,7) = 0
2110 GOSUB 1220: GOSUB 1860: RETURN

```

```

2120 REM POSITION
2130 IF XF < = WS AND YF < YC THEN A2 = 0:A
4 = WS:A3 = YF:A5 = YC:A7 = 1
2140 IF XF > = WS + L1 AND YF < YC THEN A2 =
WS + L1:A4 = 2 * WS + L1:A3 = YF:A5 = YC
:A7 = 2
2150 IF XF < = WS AND YF > YC THEN A2 = 0:A
4 = WS:A3 = YC:A5 = YF:A7 = 4
2160 IF XF > = WS + L1 AND YF > YC THEN A2 =
WS + L1:A4 = 2 * WS + L1:A3 = YC:A5 = YF
:A7 = 3
2170 IF YF < = WS AND XF < XC THEN A2 = XF:
A4 = XC:A3 = 0:A5 = WS:A6 = 1
2180 IF YF > = WS + W1 AND XF < XC THEN A2 =
XF:A4 = XC:A3 = WS + W1:A5 = 2 * WS + W1
:A6 = 4
2190 IF YF < = WS AND XF > XC THEN A2 = XC:
A4 = XF:A3 = 0:A5 = WS:A6 = 2
2200 IF YF > = WS + W1 AND XF > XC THEN A2 =
XC:A4 = XF:A3 = WS + W1:A5 = 2 * WS + W1
:A6 = 3
2210 A%(AI,2) = A2:A%(AI,3) = A3:A%(AI,4) = A
4:A%(AI,5) = A5:A%(AI,6) = A6:A%(AI,7) =
A7: RETURN
2220 REM ARC
2230 CQ = CI * 2 - 1
2240 H PLOT HS + XG * HR, VS + YG * VR TO HS +
(XG + D * COS (CQ * P4)) * HR, VS + (YG +
D * SIN (CQ * P4)) * VR
2250 AG = 10 * PR:WA = WS:WB = WS: GOSUB 590:
RETURN
2260 REM E...ERASE
2270 IF L = 0 OR A%(0,1) = 2 THEN RETURN
2280 ST$ = "E": GOSUB 460: GOSUB 1200: HCOLOR=
0:CR = 3:CL = 0:E = 1: GOSUB 2320
2290 DL = 1000: GOSUB 2380: GOSUB 1160: VTAB
12: HTAB 13: PRINT L$(34);
2300 GET CS: IF CS = "Y" THEN E = 0:L = 0
2310 HCOLOR= 3:CR = 0:CL = 3: GOSUB 1180: GOSUB
2320: GOSUB 490: GOSUB 1200: RETURN
2320 REM ERASING
2330 FOR AI = 3 TO A%(0,1)
2340 IF A%(AI,1) = 0 THEN 2370
2350 A%(AI,0) = E * A%(AI,0):A%(AI,1) = E * A
%(AI,1)
2360 IF E = 1 THEN GOSUB 4330
2370 NEXT: RETURN
2380 REM DELAY
2390 FOR I = 1 TO DL: NEXT: RETURN
2400 REM F...FIX
2410 GOSUB 160:XF = XC:YF = YC: GOSUB 160:MV
= 0: RETURN
2420 REM G...GET DATA
2430 GOSUB 1160
2440 INVERSE: HTAB 5: PRINT L$(19): NORMAL
: VTAB 4: PRINT L$(1): PRINT L$(2): VTAB
9: PRINT L$(17)
2450 INVERSE: VTAB 13: HTAB 9: PRINT "V": HTAB
15: PRINT L$(37)
2460 HTAB 9: PRINT "E": HTAB 12: PRINT "D":
HTAB 9: PRINT "R": HTAB 13: PRINT "I"
2470 HTAB 9: PRINT "T": HTAB 14: PRINT "A":
HTAB 9: PRINT "I": HTAB 15: PRINT "G"
2480 HTAB 9: PRINT "C": HTAB 16: PRINT "O":
HTAB 9: PRINT "A": HTAB 17: PRINT "N"
2490 HTAB 9: PRINT "L": HTAB 18: PRINT "A":
HTAB 19: PRINT "L": NORMAL
2500 GOSUB 2570
2510 GOSUB 180: GOSUB 2570
2520 IF FL = 1 THEN MV = 1: GOSUB 140: GOTO
2510
2530 IF CA = 32 THEN GOSUB 1180: RETURN
2540 IF CA = 60 THEN GOSUB 6130
2550 IF CA = 62 THEN GOSUB 6160
2560 GOTO 2510
2570 REM DISTANCES
2580 XD = XC - XF:YD = YC - YF
2590 IF FH = 1 THEN FLASH
2600 DD = ABS (XD):VT = 13:HT = 26: GOSUB 26
50
2610 IF FV = 1 THEN FLASH
2620 DD = ABS (YD):VT = 22:HT = 7: GOSUB 265
00
2630 DD = INT ( SQR (XD * XD + YD * YD)):VT =
22:HT = 21: GOSUB 2650
2640 VTAB 9: HTAB 34: RETURN
2650 REM COUNTERS
2660 VTAB VT: HTAB HT: PRINT SPC(4): HTAB
HT: PRINT DD:Q$
2670 FT = INT (DD / 12):IN = DD - FT * 12
2680 HTAB HT: PRINT SPC(8): HTAB HT: PRINT
("FT"IN:Q$"): NORMAL: RETURN
2690 REM H...HARDCOPY
2700 GOSUB 1160: HTAB 13: INVERSE: PRINT L$
(44): NORMAL: VTAB 4: HTAB 3: PRINT L$(
46): PRINT: HTAB 3: PRINT L$(47): PRINT
: HTAB 3: PRINT L$(67)
2710 VTAB 11: HTAB 10: PRINT L$(35): PRINT:
HTAB 3: PRINT L$(17);
2720 GET CS: IF CS = " " THEN GOSUB 1180: RETURN

```

```

2730 IF CS = "1" THEN N% = 1: GOTO 2770
2740 IF CS = "2" THEN N% = 2: GOTO 2770
2750 IF CS = "3" THEN N% = 270
2760 GOTO 2720
2770 VTAB 4: HTAB 3: CALL - 958: PRINT L$(3
9):H = 8:I2 = 58:J = 6:K = 40: GOSUB 297
0
2780 VTAB 11: HTAB 3: PRINT L$(48):DL = 150
0: GOSUB 2380: IF CS = "R" THEN CALL 62
450: GOSUB 1180: GOSUB 4260
2790 GOSUB 1180: PRINT D$PR#1: FOR I = 1 TO
3 * N%: PRINT: NEXT: PRINT E$1:TM =
5 + 20 / (N% * N%)
2800 FOR BY = 39 TO 0 STEP - 1: PRINT CR$:
POKE 36, TM:K = - 1: FOR TR = 0 TO 80 STEP
40: FOR OC = 0 TO 896 STEP 128: FOR FL =
0 TO 7168 STEP 1024:K = K + 1
2810 V%(K) = PEEK (SA + TR + OC + FL + BY): IF
N% = 2 THEN POKE 768,V%(K): CALL 771:V%
(K) = PEEK (769):V%(K + 192) = PEEK (7
70)
2820 NEXT: NEXT: NEXT: FOR H = 0 TO N% *
192 - 1 STEP 192: IF H = 192 THEN PRINT
CR$: POKE 36, TM
2830 FOR I = H TO H + 191 STEP 96 / N%: PRINT
E$K" CHR$(96) CHR$(0): FOR J = I TO
I - 1 + 96 / N%: FOR I2 = 1 TO N%: POKE
49296,V%(J)
2840 IF PEEK (49601) < 0 THEN 2840
2850 NEXT: NEXT:KC = PEEK (- 16384): IF
KC = 160 THEN POKE - 16368,0: PRINT E$
"2": PRINT D$PR#0: GOTO 2700
2860 NEXT: NEXT: NEXT: PRINT E$"2": PRINT
D$PR#0: RETURN
2870 VTAB 4: HTAB 3: CALL - 958: PRINT L$(3
2)": "": GOSUB 1120:F$ = INS: GOSUB 4860
: VTAB 4: HTAB 20: PRINT F$: CALL - 86
8
2880 ONERR GOTO 2950
2890 H = 7:I2 = 41:J = 9:K = 40: GOSUB 2970:V
I = 66:VJ = 1
2900 PRINT D$: PRINT D$BSAVE PIC."F$",A$200
0,L$2000"
2910 VTAB 4: CALL - 958
2920 I2 = 60:K = 17: GOSUB 2970:VI = 65:VJ =
2: IF CS < > " " THEN 2920
2930 PRINT D$: PRINT D$VERIFY ARCHITECT.DIR
"
2940 GOSUB 1180: RETURN
2950 CALL - 3288: VTAB 4: CALL - 958: HTAB
15: FLASH: PRINT "L$(VI): NORMAL:DL
= 3000: GOSUB 2380
2960 HTAB 15: INVERSE: PRINT "L$(VI): NORMAL
: ON VJ GOTO 2890,2920
2970 REM PROMPTS
2980 VTAB H: HTAB 3: PRINT L$(I2): VTAB J: HTAB
3: PRINT L$(K): GET CS: RETURN
2990 REM I,J,K,M
3000 GOSUB 140:DL = 100: GOSUB 2380: GOSUB 1
40: RETURN
3010 REM L...LATCH
3020 IF A%(0,1) = 2 THEN RETURN
3030 ST$ = "L": GOSUB 460: FOR AI = 3 TO A%(0
,1)
3040 IF XC < A%(AI,2) OR YC < A%(AI,3) OR XC
> A%(AI,4) OR YC > A%(AI,5) OR A%(AI,0)
= 0 OR A%(AI,1) = 1 THEN 3060
3050 A%(AI,1) = 1: GOSUB 3070
3060 NEXT: GOSUB 490:L = 1: RETURN
3070 REM LATCHED
3080 XS = P5 * (A%(AI,2) + A%(AI,4)):YS = P5 *
(A%(AI,3) + A%(AI,5))
3090 GOSUB 3130: IF A = 1 THEN XDRAW 1 AT H
S + HR * XS,VS + VR * YS: GOTO 3120
3100 IF A = 2 THEN XDRAW 1 AT HS + HR * XS,
VS + VR * YS
3110 XDRAW 4 AT HS + HR * XS,VS + VR * YS
3120 RETURN
3130 REM LATCH CHECK
3140 IF XS < X1 - 2 THEN XS = X1 - 2
3150 IF XS > X2 + 2 THEN XS = X2 + 2
3160 IF YS < Y1 - 2 THEN YS = Y1 - 2
3170 IF YS > Y2 + 2 THEN YS = Y2 + 2
3180 RETURN
3190 REM N...NEW
3200 IF N = 0 THEN TI$ = "":TG = 0: FOR I =
0 TO 5:D(I) = 0:F$(I) = " ": NEXT: GOTO
3250
3210 GOSUB 1160: VTAB 12: HTAB 13: PRINT L$(
34);
3220 GET CS: IF CS < > "Y" AND TG = 1 THEN
TG = 0: GOSUB 3610: RETURN
3230 IF CS < > "Y" THEN GOSUB 1180: RETURN
3240 HTAB 12: FLASH: PRINT "L$(43): NORMAL
:DL = 1000: GOSUB 2380:II = A%(0,1): FOR
I = 0 TO II: FOR J = 0 TO 1:A%(I,J) = 0:
NEXT: NEXT:L = 0
3250 HOME: VTAB 2: HTAB 7: INVERSE: PRINT
L$(26): NORMAL: VTAB 5: HTAB 16: PRINT
L$(27): VTAB 7: HTAB 3

```

```

3260 PRINT L$(28): VTAB 9: HTAB 3: PRINT L$(
29): VTAB 11: HTAB 3: PRINT L$(30): VTAB
14: HTAB 3: PRINT L$(45)": VTAB 20: HTAB
10: PRINT L$(31)
3270 PRINT : HTAB 3: PRINT L$(17): HTAB 29:
PRINT "CORD": GOSUB 3380: GOSUB 3450: GOSUB
3530
3280 IF L1 < W1 THEN L1 = W1: W1 = Z
3290 A%(0,0) = 1: A%(0,1) = 2: A%(0,2) = S1
3300 A%(0,3) = 1: A%(0,4) = L1: A%(0,5) = W1
3310 A%(1,0) = 2: A%(2,0) = 2: A%(1,1) = 1: A%(2
,1) = 0
3320 A%(1,2) = WS: A%(1,3) = WS
3330 A%(1,4) = L1 + WS: A%(1,5) = W1 + WS
3340 A%(2,2) = 0: A%(2,3) = 0
3350 A%(2,4) = L1 + 2 * WS: A%(2,5) = W1 + 2 *
WS
3360 A%(0,6) = 0: A%(0,7) = 0: A%(1,6) = 1: A%(1
,7) = 0: A%(2,6) = 1: A%(2,7) = 0
3370 GOSUB 3610: RETURN
3380 REM CHECK
3390 W1 = D(1) + 12 * D(0): L1 = D(3) + 12 * D
(2): S1 = D(5) + 12 * D(4): Z = L1
3400 FM = 1: IF W1 = 0 OR L1 = 0 OR S1 = 0 THEN
FM = 0
3410 RETURN
3420 REM CONVERT
3430 FT = INT (D(K + 1) / 12): D(K) = D(K) +
FT: D(K + 1) = D(K + 1) - 12 + FT
3440 F$(K) = STR$(D(K)): F$(K + 1) = STR$(
D(K + 1)): RETURN
3450 REM DIMENSIONS
3460 FOR J = 0 TO 2: K = 2 * J
3470 IF F$(K) = "-" OR F$(K + 1) = "-" THEN
3500
3480 GOSUB 3420
3490 IF D(K) > 99 THEN F$(K) = "": F$(K + 1
) = "": FM = 0: D(K) = 0: D(K + 1) = 0
3500 VTAB 7 + K: HTAB 18: PRINT RIGHTS(" "
+ F$(K), 2):
3510 HTAB 28: PRINT RIGHTS(" " + F$(K + 1)
, 2): IF G = 1 THEN RETURN
3520 NEXT : VTAB 14: HTAB 13: PRINT TIS: VTAB
22: HTAB 36: RETURN
3530 REM MODIFY
3540 GET CS: IF CS = " " AND FM = 1 THEN RETURN
3550 IF CS = " " THEN CALL 64477
3560 CS = ASC (CS) - 65: IF CS < 0 OR CS > 6
THEN 3540
3570 IF CS = 6 THEN GOSUB 6110: GOTO 3540
3580 F$(CS) = "-": GOSUB 3450: VTAB 17: HTAB
3: PRINT L$(33)":
3590 GOSUB 1120: VTAB 17: CALL - 868
3600 D(CS) = ABS (VAL (IN$)): F$(CS) = " ": GOSUB
3380: GOSUB 3450: GOTO 3540
3610 REM OUTLINE
3620 MV = 0: FH = 0: FV = 0: FL = 0: ST = 2: N = 1
: GOSUB 1330
3630 XC = P5 * (X1 + X2): YC = P5 * (Y1 + Y2):
XF = XC: YF = YC
3640 HCOLOR = 3: POKE 230, 32: CALL 62450: POKE
- 16297, 0: POKE - 16302, 0: POKE - 163
04, 0: GOSUB 4260: RETURN
3650 REM CLEAR
3660 VTAB 22: HTAB 1: CALL - 868: VTAB 21: CALL
- 868: RETURN
3670 REM FILES
3680 POKE 34, 5: POKE 35, 20: VTAB 6: HTAB 1: FOR
I = 1 TO DF STEP 3: FOR J = I TO I + 2: IF
J > DF THEN 3730
3690 PRINT " ": IF J = TI THEN INVERSE
3700 IF J = TF THEN FLASH
3710 PRINT RIGHTS(" " + STR$(J) + " ", 3)
LEFT$(DFS(J) + " ", 9): NORMAL
: REM 9 SPACES
3720 NEXT : PRINT " ": NEXT
3730 POKE 34, 0: POKE 35, 24: VTAB 24: HTAB 36
+ (N = 0) * 2: RETURN
3740 REM PROMPT
3750 IF DF = 0 THEN K = 63: GOSUB 3950
3760 GOSUB 3650: HTAB 3: PRINT L$(33)": "S$;
: GOSUB 1120: TF = VAL (IN$)
3770 IF TF < 1 OR TF > DF THEN K = 59: GOSUB
3950
3780 GOSUB 3670: IF CS < > "D" AND CS < >
"R" THEN RETURN
3790 VTAB 22: HTAB 13: INVERSE : PRINT L$(34
): NORMAL : GET BS
3800 IF BS < > "Y" THEN POP : GOTO 4640
3810 IF CS = "R" OR CS = "Q" THEN RETURN
3820 M$ = "A." + DFS(TF): PRINT DS: PRINT DS"
DELETE "M$: RETURN
3830 REM PROMPT2
3840 GOSUB 3650: IF N = 0 AND TG = 0 THEN K =
62: GOSUB 3950
3850 IF CS = "S" THEN 3900
3860 IF TI = 0 THEN 3740
3870 TF = T1: GOSUB 3650: GOSUB 3670: VTAB 21
: HTAB 3: PRINT L$(33)": "S$T1: HTAB 13:
INVERSE : PRINT L$(34): NORMAL : GET B
S
3880 IF BS < > "Y" THEN POP : GOTO 4640
3890 RETURN

```

```

3900 GOSUB 3650: VTAB 21: HTAB 3: PRINT L$(5
7)": "S$; GOSUB 1120: F$ = LEFT$(IN$, 9
): GOSUB 4860: VTAB 21: HTAB 20: PRINT F
$: CALL - 868: PRINT
3910 IF TM < > 0 THEN HTAB 3: INVERSE : PRINT
L$(64): NORMAL : GET BS: IF BS = "Y" THEN
TF = TM: GOSUB 3670: RETURN
3920 IF TM < > 0 THEN POP : GOTO 4640
3930 IF DF = 42 THEN K = 61: GOSUB 3950
3940 DF = DF + 1: DFS(DF) = F$: TF = DF: GOSUB
3670: RETURN
3950 REM MESSAGES
3960 IF K = 66 THEN PRINT DS: PRINT DS"CLOS
E"
3970 VTAB 22: HTAB 1: CALL - 868: HT = INT
((39 - LEN (L$(K))) / 2): HTAB HT: INVERSE
: PRINT L$(K): NORMAL : TF = 0: GOSUB 36
70
3980 POP : POP : GOTO 4530
3990 REM DELETE
4000 IF TF = DF THEN 4020
4010 FOR I = TF TO DF - 1: DFS(I) = DFS(I + 1
): NEXT
4020 IF TI = TF THEN TI = 0
4030 IF TI > TF THEN TI = TI - 1
4040 DF = DF - 1: RETURN
4050 REM P...PLACE
4060 IF L = 0 OR MV = 0 OR A%(0,1) = 2 THEN
RETURN
4070 ST$ = "P": GOSUB 460: GOSUB 1200: XH = XC
- XF: YH = YC - YF
4080 FOR AI = 3 TO A%(0,1): OB = A%(AI, 0)
4090 IF A%(AI, 1) = 0 THEN 4190
4100 IF (OB = 3 OR OB = 4) AND FN OT(XF) =
0 THEN 4190
4110 IF (OB = 2 OR OB = 5) AND FN IN(XF) =
0 THEN 4190
4120 XE = XH: YE = YH
4130 IF (OB = 3 OR OB = 4) AND FN HO(AI) THEN
YE = 0: IF YF > WS AND YF < WS + W1 THEN
4190
4140 IF (OB = 3 OR OB = 4) AND FN VE(AI) THEN
XE = 0: IF XF > WS AND XF < WS + L1 THEN
4190
4150 HCOLOR = 0: CR = 3: CL = 0: GOSUB 4330
4160 A%(AI, 2) = A%(AI, 2) + XE: A%(AI, 3) = A%(A
I, 3) + YE
4170 A%(AI, 4) = A%(AI, 4) + XE: A%(AI, 5) = A%(A
I, 5) + YE
4180 HCOLOR = 3: CR = 0: CL = 3: GOSUB 4330
4190 NEXT : XF = XC: YF = YC: GOSUB 1860: RETURN
4200 REM Q...QUIT
4210 GOSUB 1160: VTAB 12: HTAB 13: PRINT L$(
34):
4220 GET CS: IF CS = "Y" THEN POKE 106, 8: POKE
105, 0: HOME : END
4230 GOSUB 1180: RETURN
4240 REM R...REDRAW
4250 CALL 62450: GOSUB 1160: VTAB 12: HTAB 1
0: FLASH : PRINT "L$(50)": NORMAL : DL =
1000: GOSUB 2380: GOSUB 4370: GOSUB 1180
4260 IF A%(0,3) = 1 THEN GOSUB 1410
4270 FOR AI = 1 TO A%(0,1)
4280 GOSUB 4330: NEXT : IF A%(1,1) = 0 THEN
4310
4290 IV = 0: VL = 1: VT = 1: HT = 40: ST$ = L$(49
): GOSUB 640
4300 VL = 0: VT = 24: IF TIS < > " " THEN HT =
1 + INT ((35 - LEN (TIS)) / 2): ST$ = T
I$: GOSUB 640
4310 GOSUB 490
4320 GOSUB 1200: RETURN
4330 REM DRAW OBJECTS
4340 IF A%(AI, 0) > = 2 AND A%(AI, 0) < = 5 THEN
GOSUB 1220
4350 IF AI > 2 AND A%(AI, 1) = 1 THEN GOSUB
3070
4360 RETURN
4370 REM HOUSKEEPING
4380 X = FRE (0)
4390 AN = A%(0,1): AI = 2: IF AN = 2 THEN RETURN
4400 FOR I = 3 TO AN
4410 IF A%(I, 0) = 0 THEN 4460
4420 IF A%(I, 1) = 0 AND (FN VH(2) * FN VH(
4) OR FN RH(2) * FN RH(4) OR FN VH(3)
* FN VH(5) OR FN BV(3) * FN BV(5)) THEN
4460
4430 AI = AI + 1: IF AI = I THEN 4460
4440 FOR J = 0 TO 7
4450 A%(AI, J) = A%(I, J): NEXT
4460 NEXT : A%(0,1) = AI: IF AI = 2 THEN AI =
3
4470 RETURN
4480 REM S...SAVE
4490 ONERR GOTO 4800
4500 GOSUB 1160: INVERSE : HTAB 12: PRINT L$(
54): NORMAL : PRINT : HTAB 3: PRINT L$(
55): HTAB 3: PRINT L$(56): VTAB 24: HTAB
3: PRINT L$(17):
4510 IF N = 0 THEN HTAB 3: PRINT "PRESS COM
MAND KEY OR SPACE BAR...";
4520 GOSUB 3670
4530 GET CS

```



```

4740 NEXT : RETURN
4750 REM PUT
4760 MS = "A." + DF$(TF): VTAB 22: PRINT DS:PRINT
D$*OPEN "MS: PRINT D$"CLOSE": PRINT D$"D
ELETE "MS
4770 PRINT D$"OPEN "MS: PRINT D$"WRITE "MS
4780 PRINT TIS: FOR I = 0 TO A%(0,1): FOR J =
0 TO 7: PRINT A%(1,J): NEXT : NEXT :TI =
TF
4790 PRINT D$"CLOSE "MS: RETURN
4800 CALL - 3288:K = 66: GOSUB 3950
4810 REM DIRECTORY
4820 MS = "ARCHITECT.DIR": PRINT D$"OPEN "MS:
PRINT D$"CLOSE": PRINT D$"DELETE "MS
4830 PRINT D$"OPEN "MS: PRINT D$"WRITE "MS
4840 DF$(0) = STR$(DF): FOR I = 0 TO DF: PRINT
DF$(I): NEXT
4850 PRINT D$"CLOSE "MS: RETURN
4860 REM MATCH
4870 IF RIGHT$(F$,1) = " " THEN F$ = MID$(
F$,1, LEN(F$) - 1): GOTO 4870
4880 Z$ = " ": FOR I = 1 TO LEN(F$):T$ = MID$(
F$,I,1): IF T$ < > "." AND T$ < > ":"
THEN Z$ = Z$ + T$
4890 NEXT :F$ = Z$
4900 TM = 0: FOR I = 1 TO DF: IF F$ = DF$(I) THEN
TM = I: I = DF
4910 NEXT : RETURN
4920 REM TURN
4930 IF L = 0 OR FN IN(XF) = 0 OR A%(0,1) =
2 THEN RETURN
4940 ST$ = "T": GOSUB 460: GOSUB 1200:XD = XC
- XF:YD = YC - YF:XE = XF:YE = YF
4950 FOR AI = 3 TO A%(0,1):OB = A%(AI,0)
4960 IF A%(AI,1) = 0 OR OB = 3 OR OB = 4 OR
(OB = 5 AND (XD < > 0 OR YD < > 0)) THEN
5070
4970 HCOLOR= 0: GOSUB 4330: IF XD = 0 AND YD
= 0 THEN 5000
4980 A%(AI,6) = XD:A%(AI,7) = YD: IF XD = 0 THEN
XE = P5 * (A%(AI,2) + A%(AI,4)):YE = P5 *
(A%(AI,3) + A%(AI,5)):A%(AI,6) = 1:A%(AI
,7) = 0: GOTO 5000
4990 GOTO 5060
5000 XL = XE - YE + A%(AI,3):YT = YE + XE - A
%(AI,2)
5010 XR = XE - YE + A%(AI,5):YB = YE + XE - A
%(AI,4)
5020 A%(AI,2) = XL:A%(AI,4) = XR
5030 IF XL > XR THEN A%(AI,2) = XR:A%(AI,4) =
XL
5040 A%(AI,3) = YT:A%(AI,5) = YB
5050 IF YT > YB THEN A%(AI,3) = YB:A%(AI,5) =
YT
5060 HCOLOR= 3: GOSUB 4330
5070 NEXT : GOSUB 490: GOSUB 1200: RETURN
5080 REM ROTATE RECTANGLE
5090 XN = P5 * (XL + XR):YX = P5 * (YT + YB):
CX = A%(AI,6):SI = A%(AI,7)
5100 DD = SQR(CX * CX + SI * SI):CX = CX /
DD:SI = SI / DD
5110 XX = P5 * (XR - XL):YY = P5 * (YT - YB)
5120 XA = XN + XX * CX - YY * SI:YA = YX + XX
* SI + YY * CX
5130 XB = XN + XX * CX + YY * SI:YB = YX + XX
* SI - YY * CX
5140 XU = XN - XX * CX + YY * SI:YU = YX - XX
* SI - YY * CX
5150 XV = XN - XX * CX - YY * SI:YV = YX - XX
* SI + YY * CX
5160 XA = FN BB(XA):XB = FN BB(XB):XU = FN
BB(XU):XV = FN BB(XV)
5170 YA = FN BD(YA):YB = FN BD(YB):YU = FN
BD(YU):YV = FN BD(YV)
5180 XA = HS + HR * XA:XB = HS + HR * XB:XU =
HS + HR * XU:XV = HS + HR * XV:YA = VS +
VR * YA:YB = VS + VR * YB:YU = VS + VR *
YU:YV = VS + VR * YV
5190 HPLLOT XA,YA TO XB,YB TO XU,YU TO XV,YV TO
XA,YA: RETURN
5200 REM UNLATCH
5210 IF A%(0,1) = 2 THEN RETURN
5220 ST$ = "U": GOSUB 460: FOR AI = 3 TO A%(0
,1)
5230 IF A%(AI,1) = 1 THEN GOSUB 5250:A%(AI,
1) = 0
5240 NEXT : GOSUB 490:L = 0: RETURN
5250 REM UNLATCHED
5260 XS = P5 * (A%(AI,2) + A%(AI,4)):YS = P5 *
(A%(AI,3) + A%(AI,5))
5270 GOSUB 3130: XDRAW 4 AT HS + HR * XS,VS +
VR * YS: RETURN

```

```

5280 REM V...VERIFY
5290 GOSUB 1160: INVERSE : HTAB 10: PRINT L$(
25)
5300 VTAB 3: HTAB 4: PRINT L$(36):: HTAB 14:
PRINT L$(37):: HTAB 28: PRINT L$(38): NORMAL
5310 VT = 5:VI = 20: VTAB VT: HTAB 4: PRINT L
$(VI)
5320 HT = 14:DD = L1: GOSUB 2650
5330 HT = 28:DD = W1: GOSUB 2650
5340 IF L = 0 THEN 5420
5350 VJ = 1: FOR I = 3 TO A%(0,1)
5360 IF A%(I,1) = 0 THEN 5410
5370 VT = VT + 3:VI = A%(I,0) + 19: VTAB VT: HTAB
4: PRINT L$(VI)
5380 DD = A%(I,4) - A%(I,2):HT = 14: GOSUB 26
50
5390 DD = A%(I,5) - A%(I,3):HT = 28: GOSUB 26
50
5400 VJ = VJ + 1: IF VJ > 5 THEN 5420
5410 NEXT
5420 VTAB 24: HTAB 3: PRINT L$(17):
5430 GET C$: IF C$ = " " THEN GOSUB 1180: RETURN
5440 GOTO 5430
5450 REM W...WINDOW
5460 IF FN OT(XF) = 0 OR MV = 0 THEN RETURN
5470 IF A = 0 THEN ST$ = "W": GOSUB 1820:A%(
AI,0) = 3
5480 A6 = 0:A7 = 0: GOSUB 2120: GOSUB 1220: GOSUB
1860: RETURN
5490 REM BAR
5500 IF A%(AI,6) < > 0 THEN HPLLOT LX,TY TO
LX,BY: HPLLOT RX,TY TO RX,BY:TY = P5 * (T
Y + BY):BY = TY
5510 IF A%(AI,7) < > 0 THEN HPLLOT LX,TY TO
RX,TY: HPLLOT LX,BY TO RX,BY:LX = P5 * (L
X + RX):RX = LX
5520 HPLLOT LX,TY TO RX,BY: RETURN
5530 REM X...EXIT W/O DOOR
5540 F = 1: GOSUB 2050:F = 0: RETURN
5550 REM DOORWAY
5560 HCOLOR= CR:MX = 262:MY = 172
5570 IF FN HO(AI) THEN LX = LX + 1:LX = FN
BB(LX):RX = RX - 1:RX = FN BB(RX): HPLLOT
LX,TY TO RX,TY: HPLLOT LX,BY TO RX,BY
5580 IF FN VE(AI) THEN TY = TY + 1:TY = FN
BD(TY):BY = BY - 1:BY = FN BD(BY): HPLLOT
LX,TY TO LX,BY: HPLLOT RX,TY TO RX,BY
5590 HCOLOR= CL
5600 IF A%(AI,6) = 0 THEN 5670
5610 D = XR - XL:CI = A%(AI,6):T1 = 180 * PR:
T2 = 360 * PR
5620 IF CI = 1 THEN XG = XL:YG = YB: GOSUB 2
220
5630 IF CI = 2 THEN XG = XR:YG = YB: GOSUB 2
220
5640 T2 = T1:T1 = 0
5650 IF CI = 3 THEN XG = XR:YG = YT: GOSUB 2
220
5660 IF CI = 4 THEN XG = XL:YG = YT: GOSUB 2
220
5670 IF A%(AI,7) = 0 THEN RETURN
5680 D = YB - YT:CI = A%(AI,7):T1 = - 90 * P
R:T2 = 90 * PR
5690 IF CI = 1 THEN XG = XR:YG = YT: GOSUB 2
220
5700 IF CI = 4 THEN XG = XR:YG = YB: GOSUB 2
220
5710 T1 = T2:T2 = 270 * PR
5720 IF CI = 2 THEN XG = XL:YG = YT: GOSUB 2
220
5730 IF CI = 3 THEN XG = XL:YG = YB: GOSUB 2
220
5740 RETURN
5750 REM *...CIRCLE
5760 IF MV = 0 THEN RETURN
5770 IF A = 0 THEN ST$ = "*": GOSUB 1820:A%(
AI,0) = 5
5780 XD = XC - XF:YD = YC - YF:D = INT ( SQR
(XD * XD + YD * YD))
5790 A%(AI,2) = XF - D:A%(AI,4) = XF + D
5800 A%(AI,3) = YF - D:A%(AI,5) = YF + D
5810 A%(AI,6) = 0:A%(AI,7) = 0
5820 GOSUB 1220: GOSUB 1860: RETURN
5830 REM #...GRID & TITLE
5840 G$(0) = "ON":G$(1) = "OFF":G$(2) = "ON":
G$(3) = "OFF":FM = 0:G = 1:E(0) = A%(0,3
):E(1) = A%(1,1)
5850 GOSUB 1160: HTAB 11: INVERSE : PRINT L$(
51): NORMAL : VTAB 4: HTAB 3: PRINT L$(
52): PRINT : HTAB 3: PRINT L$(53)

```



```

5860 VTAB 9: HTAB 16: PRINT L$(27): PRINT : HTAB
3: PRINT L$(30): VTAB 14: HTAB 3: PRINT
L$(45)": : VTAB 20: HTAB 10: PRINT L$(31
): PRINT : HTAB 3: PRINT L$(17)
5870 GOSUB 6020
5880 GET CS: IF CS = " " AND FM = 0 THEN GOSUB
1180:G = 0: RETURN
5890 IF CS = " " AND S1 = 0 THEN CALL 64477
: GOTO 5880
5900 IF CS = " " THEN CALL 62450: GOSUB 118
0:G = 0: GOSUB 4260: RETURN
5910 CC = ASC (C$) - 65: IF CC < 0 OR CC > 6
THEN 5880
5920 FM = 1: IF CC = 0 AND A%(0,3) = 0 THEN A
%(0,3) = 1:E(0) = 1: GOTO 6010
5930 IF CC = 1 AND A%(0,3) = 1 THEN A%(0,3) =
0:E(0) = 0: GOTO 6010
5940 IF CC = 2 AND A%(1,1) = 0 THEN A%(1,1) =
1:E(1) = 1: GOTO 6010
5950 IF CC = 3 AND A%(1,1) = 1 THEN A%(1,1) =
0:E(1) = 0: GOTO 6010
5960 IF CC > = 0 AND CC < = 3 THEN 5880
5970 IF CC = 6 THEN GOSUB 6110: GOTO 5880
5980 F$(CC) = " -- " : K = 4: GOSUB 3470: VTAB 17
: HTAB 3: PRINT L$(33)": :
5990 GOSUB 1120: VTAB 17: CALL - 868:D(CC) =
ABS ( VAL ( IN$ )) : F$(CC) = " " : GOSUB 34
70:FM = 1
6000 S1 = D(5) + 12 * D(4) : HD = S1 * HR: VD =
S1 * VR: VTAB 22: HTAB 36: GOTO 5880
6010 GOSUB 6020: GOTO 5880
6020 REM ENTRIES
6030 FOR J = 0 TO 1: K = 2 * J: VTAB 4 + K: HTAB
18
6040 IF E(J) = 1 THEN INVERSE
6050 PRINT G$(K) : : HTAB 28: NORMAL
6060 IF E(J) = 0 THEN INVERSE
6070 PRINT G$(K + 1) : : NORMAL : NEXT
6080 K = 4: GOSUB 3470
6090 VTAB 14: HTAB 13: PRINT TIS
6100 VTAB 22: HTAB 36: RETURN
6110 REM TITLE
6120 VTAB 14: HTAB 13: PRINT SPC( 35) : VTAB
14: HTAB 13: GOSUB 1120:TIS = LEFT$( IN
$,35) : VTAB 14: HTAB 13: PRINT TIS: CALL
- 868: VTAB 22: HTAB 36: RETURN
6130 REM <...SMALLER STEP
6140 IF ST > 1 THEN ST = ST - 1
6150 RETURN
6160 REM >...LARGER STEP
6170 IF ST < 3 THEN ST = ST + 1
6180 RETURN
6190 REM ?...HELP
6200 GOSUB 1160: INVERSE : PRINT L$(0): NORMAL
: PRINT : GOSUB 740: PRINT : PRINT : PRINT
L$(1): PRINT L$(2): PRINT : PRINT : PRINT
L$(17):
6210 GET C$: IF C$ = " " THEN GOSUB 1180: RETURN
6220 GOTO 6210
END OF LISTING 1

```

LISTING 2: MENU.CREATE

```

10 REM *****
20 REM * MENU.CREATE *
30 REM * BY I. LARRY MORRIS *
40 REM * COPYRIGHT (C) 1985 *
50 REM * BY MICROSPARC, INC *
60 REM * CONCORD, MA. 01742 *
70 REM *****
80 HOME : VTAB 12: PRINT "CREATING MENU TEXT
FILE..."
90 NL = 67: DIM L$(NL)
100 QS = CHR$( 34)
110 REM CURSOR DEFINITION
120 DATA 4,0,10,0,18,0,29,0,39,0
130 DATA 32,141,58,63,183,9,36,0
140 DATA 32,108,145,58,63,63,183,74,33,36,0
150 DATA 12,12,223,115,22,23,77,225,7,0
160 DATA 9,32,255,55,22,46,13,37,4,0
170 DATA 232,64,233,3
180 REM HI-RES EXPAND
190 DATA 0,0,0,162,0,142,1,3,142,2,3,189,43,
3,45,0,3,240,18,189,50,3,13,1,3,141,1,3,
189,57,3,13,2,3,141,2,3,232,224,7,48,225
,96,64,32,16,8,4,2,1,96,24,6,1,0,0,0,0,0
,0,64,48,12,3
200 L$(0) = "FUNCTIONS ASSIGNED TO THE KEYBOA
RD:"
210 L$(1) = "'I', 'J', 'K', 'M', '<', AND '>'
CONTROL"
220 L$(2) = "THE MOVEMENT OF THE CURSOR."
230 L$(3) = "*****
*****"
240 L$(4) = " NIBBLE ARCHITECT
*"
250 L$(5) = "* COPYRIGHT (C) 1985 BY MICROSPA
RC INC *"
260 L$(6) = "A-ALTER K-RIGHT U-UNLATC
H
270 L$(7) = "B-BOX L-LATCH V-VERIFY
"
280 L$(8) = "C-COPY M-DOWN W-WINDOW
"
290 L$(9) = "D-DOOR N-NEW X-EXIT W
/O DOOR "
300 L$(10) = "E-ERASE O-OLD Y-YES
"
310 L$(11) = "F-FIX P-PLACE *-CIRCL
E
"
320 L$(12) = "G-GET DATA Q-QUIT #-GRID
& TITLE "
330 L$(13) = "H-HARDCOPY R-REDRAW <-SMALL
ER STEP "
340 L$(14) = "I-UP S-SAVE >-LARGE
R STEP "
350 L$(15) = "J-LEFT T-TURN ?-HELP
"
360 L$(16) = "PRESS 'N' OR 'O' TO CONTINUE...
"
370 L$(17) = "PRESS THE 'SPACEBAR' TO RETURN.
"
380 L$(18) = "TYPE ID XL YT XR YB
COS SIN"
390 L$(19) = "*** DISTANCE BETWEEN CURSORS **
*"
400 L$(20) = "ROOM"
410 L$(21) = "BOX"
420 L$(22) = "WINDOW"
430 L$(23) = "DOOR"
440 L$(24) = "CIRCLE"
450 L$(25) = "*** MEASUREMENTS ***"
460 L$(26) = "*** STARTING A NEW ROOM ***"
470 L$(27) = "FEET INCHES"
480 L$(28) = "WIDTH A) B)"
490 L$(29) = "LENGTH C) D)"
500 L$(30) = "GRID SIZE E) F)"
510 L$(31) = "SELECT (A-G) TO MODIFY"
520 L$(32) = "ENTER FILE NAME"
530 L$(33) = "ENTER NUMBER"
540 L$(34) = "ARE YOU SURE?"
550 L$(35) = "SELECT (1-3)"
560 L$(36) = "OBJECT"
570 L$(37) = "HORIZONTAL"
580 L$(38) = "VERTICAL"
590 L$(39) = "TURN PRINTER ON AND POSITION PA
PER"
600 L$(40) = "PRESS ANY KEY WHEN READY..."
610 L$(41) = "INSERT NEW DISK IN DRIVE"
620 L$(42) = "OUT OF BUFFER SPACE "
630 L$(43) = "INITIALIZING "
640 L$(44) = "*** HARDCOPY ***"
650 L$(45) = "G) TITLE"
660 L$(46) = "1) 3" + QS + " X 4" + QS
670 L$(47) = "2) 6" + QS + " X 8" + QS
680 L$(48) = "PRESS 'SPACEBAR' TO STOP PRINTI
NG..."
690 L$(49) = "NIBBLE ARCHITECT " + CHR$( 96)
+ "1985"
700 L$(50) = "REORGANIZING BUFFER "
710 L$(51) = "*** GRID & TITLE ***"
720 L$(52) = "GRID A) B)"
730 L$(53) = "TITLE C) D)"
740 L$(54) = "*** DISK FILES ***"
750 L$(55) = "D-DELETE N-NEW R-REV
ISE "
760 L$(56) = "G-GET Q-QUIT S-SAV
E"

```

KEY PERFECT 4.0 RUN ON ARCHITECT		
CODE	LINE#	LINE#
B606	10	100
8BEA	110	200
7570	210	300
3E92	310	400
8D1D	410	500
7D9C	510	600
6D85	610	700
72E3	710	800
B949	810	900
878A	910	1000
83A7	1010	1100
7987	1110	1200
9DE0	1210	1300
7136	1310	1400
52FD	1410	1500
4F81	1510	1600
667B	1610	1700
720E	1710	1800
7229	1810	1900
90B6	1910	2000
6D35	2010	2100
D3D3	2110	2200
9E61	2210	2300
5382	2310	2400
89DE	2410	2500
5020	2510	2600
8220	2610	2700
B003	2710	2800
D62A	2810	2900
8381	2910	3000
8D53	3010	3100
522A	3110	3200
BE86	3210	3300
8E7A	3310	3400
78D3	3410	3500
7D15	3510	3600

83AB	3610	3700
7999	3710	3800
80C7	3810	3900
7CAB	3910	4000
6DBB	4010	4100
93DC	4110	4200
96AA	4210	4300
49FD	4310	4400
7B32	4410	4500
855D	4510	4600
A418	4610	4700
7F6A	4710	4800
92FE	4810	4900
B2BC	4910	5000
7BAE	5010	5100
D0A1	5110	5200
8745	5210	5300
6F84	5310	5400
6784	5410	5500
A483	5510	5600
7E74	5610	5700
6F53	5710	5800
B446	5810	5900
B8B9	5910	6000
4170	6010	6100
6666	6110	6200
0D55	6210	6220

PROGRAM CHECK IS : 4582

CHECK CODE 3.0

ON: ARCHITECT

TYPE: A

LENGTH: 3DFF

CHECKSUM: 93

```

770 L$(57) = "ENTER NAME"
780 L$(58) = "( 'R' WILL REDRAW BEFORE PRINTIN
G )"
790 L$(59) = "OUT OF RANGE"
800 L$(60) = "REPLACE 'ARCHITECT' DISK IN DRI
VE"
810 L$(61) = "MAXIMUM EXCEEDED"
820 L$(62) = "NO DATA"
830 L$(63) = "NO FILES"
840 L$(64) = "FILE EXISTS - DO YOU WISH TO RE
VISE?"
850 L$(65) = "WRONG DISK "
860 L$(66) = "DISK ERROR "
870 L$(67) = "3) OUTPUT TO DISK"
880 REM CREATE ARCHITECT.MENU
890 DS = CHR$(4): REM CTRL-D
900 PRINT DS"OPEN ARCHITECT.MENU"
910 PRINT DS"CLOSE"
920 PRINT DS"DELETE ARCHITECT.MENU"
930 PRINT DS"OPEN ARCHITECT.MENU"
940 PRINT DS"WRITE ARCHITECT.MENU"
950 FOR I = 0 TO NL
960 PRINT L$(I): NEXT
970 FOR I = 1 TO 117
980 READ X: PRINT X: NEXT
990 PRINT DS"CLOSE ARCHITECT.MENU"
1000 HOME : VTAB 12: PRINT "DONE": END
END OF LISTING 2 .

```

KEY PERFECT 4.0 RUN ON MENU.CREATE		CHECK CODE 3.0
CODE	LINE# - LINE#	
6C5E	10 - 100	ON: MENU.CREATE TYPE: A LENGTH: 0A3B CHECKSUM: 38
D317	110 - 200	
B2C7	210 - 300	
A9EC	310 - 400	
68F1	410 - 500	
7C4C	510 - 600	
8EEC	610 - 700	
9219	710 - 800	
71E5	810 - 900	
5649	910 - 1000	
PROGRAM CHECK IS : 0B76		

LISTING 3: ALPH.CREATE

```

10 REM *****
20 REM * ALPH.CREATE *
30 REM * (ALPHABET BIT PATTERN) *
40 REM * MICROCOMPUTER GRAPHICS *
50 REM * BY ROY E. MYERS *
60 REM *
70 REM * PRINTED WITH AUTHOR'S *
80 REM * PERMISSION *
90 REM *****
100 HOME : VTAB 12: PRINT "CREATING HI-RES C
HARACTER SHAPE TABLE..."
110 ADDR = 0: LN = 0
120 DIM AL%(64,8)
130 FOR I = 0 TO 64
140 FOR J = 0 TO 7
150 READ AL%(I,J)
160 NEXT J,I
170 ADDR = PEEK (107) + 256 * PEEK (108)
180 LN = PEEK (109) + 256 * PEEK (110) - AD
DR
190 PRINT CHR$(4):"BSAVE ARCHITECT.ALPH. A
";ADDR;"L";LN
200 HOME : VTAB 12: PRINT "DONE": END
210 DATA 0,0,0,0,0,0,0,0: REM SPACE
220 DATA 0,8,8,8,8,8,0,8: REM !
230 DATA 0,20,20,20,0,0,0,0: REM "
240 DATA 0,20,20,62,20,62,20,20: REM #
250 DATA 0,8,60,10,28,40,30,8: REM $
260 DATA 0,6,38,16,8,4,50,48: REM %
270 DATA 0,4,10,10,4,42,18,44: REM &
280 DATA 0,8,8,8,0,0,0,0: REM '
290 DATA 0,8,4,2,2,2,4,8: REM (
300 DATA 0,8,16,32,32,32,16,8: REM )
310 DATA 0,8,42,28,8,28,42,8: REM *
320 DATA 0,0,8,8,62,8,8,0: REM +
330 DATA 0,0,0,0,0,8,8,4: REM ,
340 DATA 0,0,0,0,62,0,0,0: REM -
350 DATA 0,0,0,0,0,0,0,8: REM .
360 DATA 0,0,32,16,8,4,2,0: REM /
370 DATA 0,28,34,50,42,38,34,28: REM 0
380 DATA 0,8,12,8,8,8,8,28: REM 1
390 DATA 0,28,34,32,24,4,2,62: REM 2
400 DATA 0,62,32,16,24,32,34,28: REM 3
410 DATA 0,16,24,20,18,62,16,16: REM 4
420 DATA 0,62,2,30,32,32,34,28: REM 5
430 DATA 0,56,4,2,30,34,34,28: REM 6
440 DATA 0,62,32,16,8,4,4,4: REM 7
450 DATA 0,28,34,34,28,34,34,28: REM 8
460 DATA 0,28,34,34,60,32,16,14: REM 9
470 DATA 0,0,0,8,0,8,0,0: REM :
480 DATA 0,0,0,8,0,8,8,4: REM ;
490 DATA 0,16,8,4,2,4,8,16: REM <
500 DATA 0,0,0,62,0,62,0,0: REM =
510 DATA 0,4,8,16,32,16,8,4: REM >
520 DATA 0,28,34,16,8,8,0,8: REM ?
530 DATA 0,28,34,42,58,26,2,60: REM @
540 DATA 0,8,20,34,34,62,34,34: REM A
550 DATA 0,30,34,34,30,34,34,30: REM B
560 DATA 0,28,34,2,2,2,34,28: REM C
570 DATA 0,30,34,34,34,34,34,30: REM D
580 DATA 0,62,2,2,30,2,2,62: REM E
590 DATA 0,62,2,2,30,2,2,2: REM F
600 DATA 0,60,2,2,2,50,34,60: REM G
610 DATA 0,34,34,34,62,34,34,34: REM H
620 DATA 0,28,8,8,8,8,8,28: REM I
630 DATA 0,32,32,32,32,32,34,28: REM J
640 DATA 0,34,18,10,6,10,18,34: REM K
650 DATA 0,2,2,2,2,2,2,62: REM L
660 DATA 0,34,54,42,34,34,34,34: REM M
670 DATA 0,34,34,38,42,50,34,34: REM N

```

```

680 DATA 0,28,34,34,34,34,28: REM O
690 DATA 0,30,34,34,30,2,2,2: REM P
700 DATA 0,28,34,34,34,42,18,44: REM Q
710 DATA 0,30,34,34,30,10,18,34: REM R
720 DATA 0,28,34,2,28,32,34,28: REM S
730 DATA 0,62,8,8,8,8,8,8: REM T
740 DATA 0,34,34,34,34,34,34,28: REM U
750 DATA 0,34,34,34,34,34,20,8: REM V
760 DATA 0,34,34,34,42,42,54,34: REM W
770 DATA 0,34,34,20,8,20,34,34: REM X
780 DATA 0,34,34,20,8,8,8,8: REM Y
790 DATA 0,62,32,16,8,4,2,62: REM Z
800 DATA 0,62,6,6,6,6,6,62: REM LEFT BRACKE
T
810 DATA 0,0,2,4,8,16,32,0: REM BACK SLASH
820 DATA 0,62,48,48,48,48,62: REM PROMPT
830 DATA 0,0,8,20,34,0,0,0: REM ^
840 DATA 0,0,0,0,0,0,0,62: REM UNDERLINE
850 DATA 0,62,65,93,69,93,65,62: REM COPY
RIGHT

```

END OF LISTING 3

KEY PERFECT 4.0 RUN ON ALPH.CREATE		8BF4 710 - 800 4BF1 810 - 850 PROGRAM CHECK IS : 08CB
CODE	LINE# - LINE#	CHECK CODE 3.0
75D5	10 - 100	ON: ALPH.CREATE TYPE: A LENGTH: 7BD CHECKSUM: 0D <i>continued on next page</i>
5BCD	110 - 200	
8126	210 - 300	
7BB0	310 - 400	
7720	410 - 500	
8550	510 - 600	
8631	610 - 700	

TABLE 1
Internal Data Buffer

	A%(I,0)	A%(I,1)	A%(I,2)	A%(I,3)	A%(I,4)	A%(I,5)	A%(I,6)	A%(I,7)
A%(0,J)	1	AN	S1	GF	L1	W1	0	0
A%(1,J)	2	1	WS	WS	L1+WS	W1+WS	1	0
A%(2,J)	2	0	0	0	L1+2*WS	W1+2*WS	1	0
A%(3,J)	IT	LF	XL	YT	XR	YB	XD	YD

HOW THE NIBBLE ARCHITECT WORKS
Memory Utilization

Several techniques are employed to reserve as much memory as possible for the NIBBLE ARCHITECT. First, disk storage is used as a staging area for menu strings, cursor shapes and subroutines. For example, 68 strings of characters (L\$(0) through L\$(67)) are required to produce the various menus, prompts, and instructions. MENU.CREATE writes these strings to the file ARCHITECT.MENU, which is subsequently read into memory by ARCHITECT. As a result, only the strings themselves — and not their definitions — take up space in memory. The same staging technique is used by MENU.CREATE for the cursor shapes and to provide Hi-Res expansion for double-size hardcopy images.

A variation of this approach is utilized by ALPH.CREATE (adapted from *Microcomputer Graphics* by Roy E. Myers). This program fills a two-dimensional integer array AL%(I,J) with bit patterns for the Hi-Res characters. When this is done, it is saved on disk as the binary file ARCHITECT.ALPH. The starting address (ADDR) of this array is obtained by examining memory locations 107 and 108. Memory locations 109 and 110 contain the ending address (ADDR + LN). For this to work properly, it is important

that no additional variables be defined between the time ADDR and LN are defined (line 110) and the array is BSAVED (line 190).

A third technique is to organize memory in order to provide a large, contiguous block for the program. This means using Hi-Res page 1 and having ARCHITECT reload itself just above it. In addition, simple variables and array variables are stored upward from 2048, while strings are stored downward from HIMEM (line 90 of ARCHITECT).

INTERNAL DATA BUFFER

The information that uniquely describes a given room and its contents is stored in an internal data buffer A%(52,7). The specific assignment for each member of this integer array is shown in Table 1. The second and third rows correspond to the inner and outer walls, respectively.

where

- AN = pointer to last row of A%(I,J)
- S1 = grid spacing in inches
- GF = grid flag (0 = off, 1 = on)
- L1 = length of room in inches
- W1 = width of room in inches
- WS = thickness of outer wall (6 inches)
- IT = item type (0 = null, 1 = grid, 2 = box, 3 = window, 4 = door, 5 = circle)
- LF = latch flag (0 = unlatched, 1 = latched)
- XL = left X-coordinate in inches
- YT = top Y-coordinate in inches
- XR = right X-coordinate in inches
- YB = bottom Y-coordinate in inches

The assignment of values to XD and YD is more complicated. Table 2 describes the relationship to the item type.

The images on the screen are constructed directly from the internal data buffer A%(I,J). A buffer that contains the maximum of 50 items will require only 6 sectors of storage space on the disk. A Hi-Res dump, on the other hand, requires 34 sectors.

PROGRAM DESCRIPTION

Line 80 relocates the program above Hi-Res page 1.

Lines 90-120 establish LOMEM and initialize some variables.

Line 130 performs a GOTO 280 to start the input process.

Lines 140-170 XDRAW the moving (x) and fixed (+) cursors.

Lines 180-270 control the movement of the moving cursor. Flags are set if the cursor is prevented from moving vertically (FV) or horizontally (FH) across the outer wall.

Line 280 GOSUBs to line 770 to read in the disk files and display the graphics editing menu.

TABLE 2
Assignment of XD and YD Values

Type	Condition	XD	YD
Box	No rotation	1	0
Box	Rotation	X displacement between cursors	Y-displacement between cursors
Door	Exit without door	0	0
Door in vertical wall	Down to the right	0	1
	Down to the left	0	2
	Up to the left	0	3
	Up to the right	0	4
Door in horizontal wall	Down to the right	1	0
	Down to the left	2	0
	Up to the left	3	0
	Up to the right	4	0