

# 29. Hexadecimal-Binary

## *General Description*

This program is for computer scientists only in that it is a time-based tester for binary to hexadecimal and viceversa. Hexadecimal as such is on very few of the syllabuses for 'O' level and even at 'A' level one could avoid learning it. Nevertheless for machine code work its knowledge is essential. There are calculators today that do decimal/hex conversions but few that do binary/hex conversions. This runs on a Model A.

For the uninitiated this is how to do a conversion:

Consider the following binary number: 1100

Decimal for each binary position: 8421

This means the binary number is made up of  $1 * 8$  and  $1 * 4$ , which is 12.

However hexadecimal digits are as follows (the decimal equivalents are underneath):

1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

This means that '1 1 0 0' can be written as 'C' in hexadecimal counting.

As hexadecimal counting is to base 16, the decimal number 28 is '1C' in hex. Confused . . . ??? Of course not. Actually, like all counting systems it's easy when you use it regularly, and like all counting systems there are short cuts. If you use the program, you may need to find them to beat the clock!

## *Detailed Description*

**Lines 10-230** This is the main structure. Auto repeat is turned off and the cursor disconnected. The sound is defined and the three main procedures are called.

**240-380** This is the title page for the game. The brackets after the CHR\$ are not strictly necessary but are kept for lucidity.

**400-520** Only one hex digit is considered. Beware - you will find a computed restore at line 430, so DO NOT RENUMBER the program.

**530-900** The RND in line 640 gets a random number from 0-15 rather than 1-16. The value D determines the number of digits to be collected, hex or binary. The characters collected are added into a string (line 780). There is no delete facility. The result is checked at line 850 after a carriage return which you must enter after all the digits have gone in.

**910-1250** The ' rightand ' wrongprocedures. The clock is a fast digital clock in double size characters.

**1270-1650** Nothing surprising here. The cursor etc. is re-enabled before finishing. To extend the program to two hex digits would be a relatively easy extension - have a go.

### *Educational Notes*

This program cannot really be considered ' educational' except in the practice it gives against the clock. However its graphics are fun, the clock nicely displayed and it exercises my dying grey matter as it will yours.

### *Program Listing*

```
10 REM *****
20 REM **  HEXADECIMAL/BINARY.  **
30 REM **  Written by Ian Clarke.  **
40 REM **  Feb 83.  **
50 REM *****
60 ON ERROR GOTO 1430
70 MODE 7
80 *FX4,1
90 *FX11,0
100 VDU 23;8202;0;0;0
110 PRINT
120 ENVELOPE 1,2,4,4,4,3,3,3,0,0,0,-3,100,100
130 ENVELOPE 2,3,4,4,4,3,3,3,1,-1,0,-1,120,90
140 ENVELOPE 3,1,4,-4,0,8,8,1,1,-1,0,-1,120,90
150 DIM HB$(16),G$(16),E$(2)
160 E$(1)=""
170 E$(2)="s"
180 R=0
190 W=0
200 PROCxorbin
```

```

210 PROCread
220 PROCgame
230 GOTO 1320
240 REM -----
250 REM Give the user the choice
260 REM of changing binary to
270 REM hexadecimal or vice versa.
280 DEF PROCxorbin
290 FOR X=0 TO 1
300 PRINT TAB(2,6+X);CHR$(141);CHR$(129);"press '1' to
convert hexadecimal"
310 PRINT TAB(1,8+X);CHR$(141);CHR$(129);"to binary or
'2' to convert binary"
320 PRINT TAB(13,10+X);CHR$(141);CHR$(129);"to hexadeci
mal"
330 NEXT X
340 FOR X=1 TO 200:NEXT
350 *FX15
360 A=GET
370 IF A<49 OR A>50 THEN VDU 7:GOTO 340
380 IF A=50 THEN W$="hexadecimal":L$="0":L1$="F":D=1 ELSE
W$="binary":L$="0":L1$="1":D=4
390 REM -----
400 REM Read in data for the possible binary or
hex patterns
410 REM using four bits.
420 DEF PROCread
430 RESTORE (A-49)*40+1550
440 FOR X=0 TO 15
450 READ HB$(X)
460 NEXT X
470 IF A=50 THEN RESTORE 1580
480 FOR X=0 TO 15
490 READ G$(X)
500 NEXT X
510 REM .....
520 ENDPROC
530 REM -----
540 REM The computer thinks of a
550 REM random hex or binary value
560 REM and display the value. The
570 REM computer waits for the user
580 REM enter an answer.
590 DEFPROCgame
600 CLS
610 V=0
620 START=TIME
630 FOR Z=1 TO 10
640 C=RND(16)-1
650 IF V=C THEN GOTO 640
660 FOR X=0 TO 1
670 PRINT TAB(10,3+X);CHR$(141);CHR$(129);"Try no ";Z
680 PRINT TAB(3,7+X);CHR$(141);CHR$(132);"What is the
";W$
690 PRINT TAB(5,9+X);CHR$(141);CHR$(133);"equivalent
to";
700 PRINT CHR$(141);CHR$(131);G$(C)
710 PRINT TAB(14,17+X);CHR$(141);" "
720 NEXT X
730 A$=""
740 FOR X=1 TO D
750 PROCtime
760 Q$=INKEY$(0)
770 IF Q$<L$ OR Q$>L1$ OR Q$>"9" AND Q$<"A" THEN GOTO
750
780 A$=A$+Q$
790 PRINT TAB(13,17);CHR$(141);CHR$(134);A$
800 PRINT TAB(13);CHR$(141);CHR$(134);A$
810 NEXT X

```

```

820     PROctime
830     CR=INKEY(0)
840     IF CR<>13 THEN GOTO 820
850     IF A$=HB$(C) THEN PROCright ELSE PROCwrong
860     V=C
870     NEXT Z
880     FOR X=1 TO 500:NEXT
890     REM .....
900     ENDPROC
910     REM -----
920     REM     The answer is
930     REM     correct.  Display message
940     REM     and sound envelope.
950     DEFPROCright
960     SOUND 1,1,160,2
970     R=R+1
980     PRINT TAB(23,11);CHR$(141);CHR$(130);"No. correct=";R
990     PRINT TAB(23);CHR$(141);CHR$(130);"No. correct=";R
1000    PROctime
1010    REM .....
1020    ENDPROC
1030    REM -----
1040    REM     User's answer is wrong.

1050    REM     Display message and
1060    REM     sound wrong signal.
1070    DEFPROCwrong
1080    SOUND 1,-12,90,4
1090    W=W+1
1100    PRINT TAB(23,13);CHR$(141);CHR$(131);"No. wrong=";W
1110    PRINT TAB(23);CHR$(141);CHR$(131);"No. wrong=";W
1120    PROctime
1130    REM .....
1140    ENDPROC
1150    REM -----
1160    REM     Display time, which begins
1170    REM     before the first hex/binary
1180    REM     value is displayed on the vdu.
1190    DEFPROCtime
1200    T=(TIME-START)/100
1210    PRINT TAB(12,22);CHR$(141);T
1220    PRINT TAB(19,22);CHR$(141);"seconds"
1230    PRINT TAB(12);CHR$(141);T
1240    PRINT TAB(19,23);CHR$(141);"seconds"
1250    REM -----
1260    ENDPROC
1270    REM     Game is over.  Display
1280    REM     how many correct and
1290    REM     wrong answer the user
1300    REM     got, and asks him/her
1310    REM     do you want another game.
1320    SOUND 2,2,100,80
1330    IF R=1 THEN E=1 ELSE E=2
1340    IF W=1 THEN F=1 ELSE F=2
1350    FOR X=1 TO 2000:NEXT
1360    CLS
1370    FOR Y=0 TO 1
1380        PRINT TAB(4,6+Y);CHR$(141);CHR$(133);"You got ";R;"
answer";E$(E);" correct"
1390        PRINT TAB(4,8+Y);CHR$(141);CHR$(130);"and ";W;" ans
wer";E$(F);" correct"
1400        PRINT TAB(7,10+Y);CHR$(141);CHR$(129);"in ";T;" sec
onds."
1410    NEXT Y
1420    FOR X=1 TO 12000:NEXT
1430    CLS
1440    PRINT TAB(5,15);CHR$(141);CHR$(129);"Do want another
go?"
1450    PRINT TAB(5);CHR$(141);CHR$(129);"Do want another go?"

```

```

"
1460 *FX15
1470 YN$=GET$
1480 IF YN$="Y" THEN CLS:GOTO 180
1490 CLS
1500 SOUND 3,3,120,50
1510 PRINT TAB(15,10);CHR$(141);CHR$(136);CHR$(131);"BYE!!"
!"
1520 PRINT TAB(15);CHR$(141);CHR$(136);CHR$(131);"BYE!!!"
1530 FOR X=1 TO 6000:NEXT
1540 *FX4
1550 *FX11,25
1560 MODE 7
1570 END
1580 DATA 0000,0001,0010,0011
1590 DATA 0100,0101,0110,0111
1600 DATA 1000,1001,1010,1011
1610 DATA 1100,1101,1110,1111
1620 DATA 0,1,2,3
1630 DATA 4,5,6,7
1640 DATA 8,9,A,B
1650 DATA C,D,E,F

```

