

6. Validation Routines

General Description

At this point we throw in a couple of general purpose routines. The first should really be included in 'menu' driven programmes. The routine 'gets' a single character, validates it using the INSTR function and returns a value in RESULT or the flags YES and NO. The second should be used whenever there is doubt about whether a critical data entry is in upper or lower case. Any routine that you write which searches for matches between program data and keyboard entry should use such a routine at some point.

In the first routines, the delete key has not been enabled and so, for instance, should you have a menu of 10 choices (0-9), you will have to call the routine twice: firstly to get the character 0-9, secondly to ask whether the entry is correctly returning the flags YES and NO.

The first routine is called with two parameters, the first indicating what has to be validated, and the second the number of characters that **MUST** be entered from the keyboard.

Detailed Description

PROCvalidate

Lines 1000-1120 This initialises variables inside the routine and defines the various things that Check\$ can be. In the example I have chosen 'Y/N' answers, digits 0-9 and a hexadecimal entry, but if your menu only had three choices you would simply add 'Check\$' = '123':GOTO 1130.

1130-1210 This simply GET's the required characters, checks them with INSTR and leaves the result in B\$. If you wish to exit

on a Carriage Return this must be specified in Check\$.

1220-1250 If you use EVAL in conjunction with this routine, you will need a condition such as that in 1220. It does, of course, not matter that RESULT will pass a value 0 out if the routine is testing for (Y/N), and vice-versa.

The RUNS are included for your convenience. Note in line 190 that you must have RESULT > 0 if you wish to use it subsequently in an ON GOTO.

FNconvert(A\$)

Little to add here except that it is a nice example to use with computer studies pupils on how a string function can return a value. If you wish to change from upper case to lower case, 1095 and 1100 must be changed accordingly to Z=Y+32 and IF Y>64 AND Y<91 THEN Y=Z.

Program Listing

```
1000 REM =====
1010 REM   general purpose
1020 REM   validation routines
1030 REM   .....
1040 REM
1050 DEF PROCvalidate (Type, Chars)
1060 LOCAL X, Y, Check$, B$, A$
1070 Y=0: X=0: B$=""
1080 YES=FALSE: NO=FALSE
1090 ON Type GOTO 1100, 1110, 1120
1100 Check$ = "YyNn": GOTO 1130
1110 Check$ = "1234567890": GOTO 1130
1120 Check$ = "1234567890ABCDEF": GOTO 1130
1130 REPEAT
1140   REPEAT
1150     A$ = GET$
1160     X = INSTR (Check$, A$)
1170     UNTIL X<>0
1180     B$=B$+A$
1190     PRINT A$;
1200     Y=Y+1
1210   UNTIL Y=Chars
1220   IF Type = 3 THEN HEX = EVAL("&"+B$)
1230   RESULT = VAL(B$)
1240   IF RESULT=0 AND X<3 THEN YES=TRUE ELSE NO=TRUE
1245   PRINT
1250 ENDPROC
1300 REM %%%%%%%%%%%%%%%%%%%%%%%%%%
1310 REM This routine assumes use
1320 REM of the following global
1330 REM variables:
1340 REM   YES, NO, HEX, RESULT
1350 REM %%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

100 REM Example of the validation
110 REM routine in a program.
120 REM .....
125 PRINT STRING$(35,"*")
130 PRINT "Do you wish to continue ?";
140 PROCvalidate(1,1)
150 IF YES THEN PRINT "Great" ELSE PRINT "How sad"
160 PRINT STRING$(35,"*")
170 PRINT "Menu choice 0-9 ?";
180 PROCvalidate(2,1)
190 ON RESULT+1 GOTO 200,210,220 :REM etc.
200 PRINT"Oh"
210 PRINT"dear"
220 PRINT"me"
230 PRINT STRING$(35,"*")
240 PRINT "start address (4 hex digits) ?";
250 PROCvalidate(3,4)
260 PRINT;"Program starts at "HEX
270 PRINT STRING$(35,"*")
280 END

```

```

>
10 INPUT "Enter a string >> " B$
20 PRINT FNconvert(B$)
999 END
1000 REM =====
1010 REM lower to upper case
1020 REM conversion routine
1030 REM .....
1040 REM
1050 DEF FNconvert(A$)
1060 LOCAL B$,X,Y,Z
1070 B$="":X=0:Y=0:Z=0
1080 FOR X = 1 TO LEN(A$)
1090 Y = ASC(MID$(A$,X,1))
1095 Z = Y-32
1100 IF Y>96 AND Y<123 THEN Y=Z
1110 B$ = B$ + CHR$(Y)
1120 NEXT X
1130 = B$
1150 REM .....

```

```

>RUN
Enter a string >> The Cat and Fiddle !!!
THE CAT AND FIDDLE !!!

```

