BBC Micro

```
7 REM**************************
8 REM*      BBC      MEMPEEK 1    *
9 REM**************************
20 MODE 7
30 *TV 255
40 CLS
50 REPEAT
100 INPUT"START ADDRESS ",SA
200 INPUT"NUMBER OF BYTES (0 TO QUIT)",B
N
250 PRINT "**************************
**********"
300 FOR B%=SA TO (SA+BN-1) STEP 4
350 H$="":@%=6
400 PRINT TAB(0);B%;TAB(8);
450 @%=4
500 FOR C=0 TO 3
550 PK%=?(B%+C):PK$="."
600 PRINT PK%;
650 IF PK%=13 THEN PK$=CHR$(124)
700 IF (PK%>31) AND (PK%<128) THEN PK$=C
HR$(PK%)
750 H$=H$+PK$
800 NEXT C
850 PRINT TAB(32);H$
900 NEXT B%
950 UNTIL BN=0
1000 REM************************
```

Spectrum

```
7 REM************************
8 REM*  SPECTRUM  MEMPEEK 1    *
9 REM************************
30 DIM H$(4)
50 FOR L=0 TO 1 STEP 0
100 INPUT"START ADDRESS ";SA
200 INPUT"NUMBER OF BYTES (0 TO
QUIT)";BN
250 PRINT "**********************
*********"
300 FOR B=SA TO (SA+BN-1) STEP 4
350 LET H$="...."
400 PRINT B;TAB 7;
500 FOR C=0 TO 3
550 LET PK=PEEK(B+C)
600 PRINT PK;" ";
650 IF (PK>31) AND (PK<128) THEN
 LET H$(C+1)=CHR$ PK
700 IF PK=13 THEN LET H$(C+1)="■
"
800 NEXT C
850 PRINT TAB 26;H$
900 NEXT B
950 IF BN=0 THEN LET L=2
1000 NEXT L
1050 REM************************
```

Using Mempeek

When you enter the Mempeek program into your machine, make sure that you SAVE it and check it carefully before you RUN it, because typing errors in this sort of program can lead to unrecoverable crashes.

The program will first ask you for a start address, and then the number of bytes that you wish to examine. Both should be positive whole numbers in the range 0 to 65535. Inputting 0 as the number of bytes will cause the program to end (quit). Suppose you wish the start address to be byte 230. The screen display might look like this:

```
START ADDRESS? 230
NUMBER OF BYTES (0 TO QUIT) ? 8
**********************************

230   193   32   65   49   . A1
234   129   64   93   98   .@]b

START ADDRESS?
```

The leftmost column gives the decimal address of the first byte, the next four columns give the decimal contents of the four bytes from that address on, and the last column gives the character representation of the bytes' contents (where this is possible), and '.' otherwise.

You might like to begin by just 'wandering around' in memory with this program, noting any interesting addresses, and then try to find where in memory the Operating System stores its error messages and BASIC keywords. Your User Manual may help you with this.

Once you've found the pointers that define the boundaries of the various areas of memory, you can try adding some REM lines to the program, and see what effect that has on pointer values. Then add some lines at the start of the program to do some string manipulation, and again, see what effect that has on the pointers and on the contents of the Variable Storage Area.

For example:

```
3 DIM Z$(254)
4 LET X$=""
5 FOR M=1 TO 255.LET X$=X$+"*":NEXT M
```

Commodore 64

```
7 REM**************************
8 REM*   COMMODORE MEMPEEK 1    *
9 REM**************************
30 PRINT CHR$(147)     :REM CLEAR SCREEN
40 PRINT CHR$(142)     :REM UPPER CASE
50 FOR LP=0 TO 1 STEP 0
100 INPUT"START ADDRESS ";SA
200 INPUT"NUMBER OF BYTES (0 TO QUIT)";B
N
250 PRINT "**************************
**********"
300 FOR B=SA TO (SA+BN-1) STEP 4
350 H$=""
400 PRINT B;TAB(8);
500 FOR C=0 TO 3
550 PK=PEEK(B+C):PK$="."
600 PRINT TAB(8+5*C);PK;
650 IF PK=0 THEN PK$=CHR$(122)
700 IF (PK>31) AND (PK<128) THEN PK$=CHR
$(PK)
750 H$=H$+PK$
800 NEXT C
850 PRINT TAB(32);H$
900 NEXT B
950 IF BN=0 THEN LP=1
1000 NEXT LP
1050 REM************************
```