

Word squares

Another firm favourite with the children! In fact, the one in our house used to buy whole books of these problems every week until I wrote this program.

You can either enter your own choice of words for the kids to find, or they can have the computer choose words from a series of DATA statements - see lines 990 to 1070. You could put your own words in here, but memory space is rather tight and you cannot extend the number of words very much before getting a 'No room' message at run time. This is because of the graphics mode used, and a useful alteration - if you have a printer - would be to have the

```

o o q s d c p e r u h o n l d x f w m
s f u b f a r e n t m o t h e r x r k
n l c f s l c f a v a b a y s s p e x
r u i b r o l x m h g b y d k g x s q
o t e c r o j n e e r c s m q n i t y
c e j d o w e e d m o o r d e b w l m
s p e r k i t c h e n u d j q t l e h
i r j p l y q l i c n e p x m j y r j
t v i o l i n c l h m t i d b y o b v

```

recorder	kitchen	screen
wrestler	violin	flute
boy	floor	desk
mother	pencil	bedroom
		scorn

```

Word ?BEDROOM
Row of initial letter ?16
Column of initial letter ?

```

puzzle printed out on paper rather than on the screen, which would result in an enormous saving of memory.

The first screen page asks if you wish to enter your own words. Type Y or N as appropriate. If you enter your own word, you must type 18 words, each of ten letters or less, pressing RETURN after each one. This is taken care of in PROCWORDSIN, and there is no check for duplicates. If you choose to have the computer use its own words, simply type N.

The computer now displays the word ' WORKING . .' for a few seconds as it fits the words into the square - actually a rectangle, 19 characters wide by 9 high. Words are fitted in any direction - forwards, backwards, up, down or diagonally - and then the rest of the rectangle is filled with junk characters. The result is printed on the screen in white lower case, a part of the program automatically converting any upper-case letters. Beneath the white display is a ruled line in red and a list of all words that have been fitted into the square. There may be gaps in the list, representing words that have been tried 25 times without success. (Alter this if you like in line 640, but don't have it try too many times or you may sit around for a long time waiting for the puzzle.)

Beneath the list is the single prompt, ' Word? The user now sits and stares at the matrix, trying to spot one of the embedded words. When one has been seen, he or she types the whole word - spelling correctly - and presses RETURN. Now we must check that they have the right word (not just guessing) and to this end the user must enter two numbers, (a) the ROW of the initial letter (counting down from the top), and (b) the COLUMN (counting from the left).

If wrong, an appropriate message is printed, but if right, both the word in the array and the word in the list are overprinted in yellow, which is a great aid. Play continues in this way until either the user types the whole word ' QUIT' or completes the puzzle by finding every word listed. The computer then prints out the time taken in minutes. If the player has quit, the solution is highlighted; every junk letter is removed from the screen. A new game may then be restarted by typing RUN.

Turning to the listing, words to be fitted into the puzzle are entered into the array W\$(18), either from the keyboard in PROCWORDSIN or from the DATA statements, chosen in lines 150 to 180. In the latter case, duplicates are checked for and eliminated by lines 170 and 180.

PROCFIT of course fits the words into the matrix, each word taken in turn and given 25 tries before being blanked and ignored (line 640). When making a try, the computer randomly picks a starting position and direction in line 650, reading the appropriate movement vectors from the DATA line 980 into I% and J%. Then it makes a trial fit, incrementing the position according to the vectors. If the array at that point is empty or contains the same letter, fitting

can proceed (line 670), but fitting stops if the edge of the matrix is met or the other conditions are not (line 680). If the trial fit is satisfactory, lines 710 to 740 actually place the word. When all words have been dealt with, lines 780 to 800 fill up the rest of the array with junk letters.

Back in the main program, the game loop is entered. This part of the program should be self-explanatory. The user inputs his or her detected word, its row and its column. A call to PROCLCASE translates any upper-case letters and then the computer checks (lines 290 to 310) first to see if the word is contained in the puzzle. It then reads the array D%(X,Y), where the X,Y start of the word has been stored, and its direction vector. Lines 400 to 410 overprint the word in the puzzle, with line 420 overprinting the same word in the list of words to find. WIN is the number of words found and if this matches DONE - the number of words fitted - the puzzle has been solved completely. Lines 460 to 480 wipe off the junk letters, revealing the solution.

Variables

C\$	User' s choice, Y = own words, N = computer' s
W\$(18)	18 words to be fitted
D%(X,Y)	Details of where each word is in puzzle, where
	X = number of word
	Y = 1 = X co-ordinate
	2 = Y co-ordinate
	3 = Direction vector, 1 to 8
A\$(19,9)	Displayed array, including junk
B\$(19,9)	Solution array
L\$	20 underscores for a straight line rule
W\$	Current input word
Y%	Random number to pick a word from DATA
Z\$	Chosen word
FLAG	Flag for duplicates
T%	Counter when searching for duplicate
TIME	Computer' s internal elapsed-time clock
WIN	Number of words correctly identified
R%	User' s input for row
C%	User' s input for column
WN%	Word number
X%	General counter
I%, J%	Movement vectors
Y\$	Lower-case version of W\$
DONE	Number of words fitted
X\$	Message string

TRY Number of attempts at fitting
 X1%, Y1% Co-ordinates while attempting fit
 L% Letters successfully tried

```

10 REM - Word square
20 MODE7:PROCDBL(5,3,131,"WORD SQUARE")
30 PRINT'"Do you want to put in your own words (Y-N)?"
40 REPEAT:C$=GET$:UNTILC$="Y"ORC$="N"
50 DIMW$(18),D$(18,3),A$(19,9),B$(19,9)
60 L$=STRING$(20," ")
70 IFC$<>"Y"GOTO130
80 PRINT'"Please type 18 words of not more than"
90 PRINT"10 letters each.":PROCWORDSIN
100
110 REM - Fit words into rectangle
120
130 CLS:PROCDBL(5,5,131,"WORKING...")
140 IFC$="Y"GOTO190
150 FORW%=1TO18
155 RESTORE 990
160 Y%=RND(65):FORZ%=1TOY%:READZ$:NEXT
170 FLAG=0:FORT%=1TOW%-1:IFZ$=W$(T%)FLAG=1
180 NEXTT%:IFFLAG=1GOTO155ELSEW$(W%)=Z$:NEXTW%
190 PROCFIT:MODEL:PROCDISPLAY:TIME=0:WIN=0
200
210 REM - Game loop *****
220
230 COLOUR3:FX15,0
240 VDU28,0,31,39,26,12
250 INPUT"Word ",W$:IFW$="QUIT"GOTO450
260 INPUT"Row of initial letter ",R%
270 INPUT"Column of initial letter ",C%
280 PROCLCASE(W$)
290 WN%=1
300 IFY$=W$(WN%)GOTO330
310 WN%=WN%+1:IFWN%<=18GOTO300
320 PRINT"NOT IN THE PUZZLE":GOTO510
330 RESTORE:FORX%=1TOD%(WN%,3):READI%,J%:NEXT:X%=1
340 IFAS(C%,R%)<>MID$(Y$,X%,1):GOTO380
350 R%=R%+J%:C%=C%+I%
360 IFR%=0 OR R%=10 OR C%=20 OR C%=0 GOTO380
370 X%=X%+1:IFX%<=LEN(Y$)GOTO340ELSE390
380 PRINT"Word not in that position.":GOTO510
390 VDU26:COLOUR2:C%=D%(WN%,1):R%=D%(WN%,2)
400 FORX%=1TOLENY$:PRINTTAB(C%*2-1,R%*2-2);AS(C%,R%)
410 R%=R%+J%:C%=C%+I%:NEXT
420 PRINTTAB((WN%-1)MOD3*15,(WN%-1)DIV3+18);Y$
430 WIN=WIN+1:IFWIN<DONE GOTO230
440 X$="Finished at ":GOTO490
450 VDU26:COLOUR2

```

```

460 FORX%=1TO9:FORY%=1TO19
470 PRINT" ";:IF B$(Y%,X%)="PRINT" ";ELSEPRINTB$(Y%,X%);
480 NEXT:COLOUR3:PRINT':NEXT:X$="Quit at "
490 VDU28,0,31,39,26,12
500 PRINT:X$;STR$(TIME/6000);" minutes.":END
510 INPUT"Press RETURN...",Q$:GOTO240
520
530 DEFPROCDBL(X%,Y%,C%,X$)
540 PRINTTAB(X%,Y%);CHR$141;CHR$C%;X$
550 PRINTTAB(X%,Y%+1);CHR$141;CHR$C%;X$:ENDPROC
560
570 DEFPROCWORDSIN
580 FORW%=1TO18:PRINT';"Word #";W%;
590 INPUTX$:IFLENX$>10X$=LEFT$(X$,10)
600 PROCLCASE(X$):W$(W%)=Y$:NEXT
610 ENDPROC
620
630 DEFPROCFIT:DONE=0:FORW%=1TO18:TRY=0
640 TRY=TRY+1:IFTRY>25W$(W%)="" :GOTO750
650 X%=RND(18):Y%=RND(9):D%=RND(8):X1%=X%:Y1%=Y%
660 RESTORE:FORV%=1TOD%:READI%,J%:NEXT:L%=1
670 IFA$(X%,Y%)<>"ANDMID$(W$(W%),L%,1)<>A$(X%,Y%)GOTO640
680 X%=X%+I%:Y%=Y%+J%:IFX%=0ORX%=19ORY%=0ORY%=10GOTO640
690 L%=L%+1:IFL%<=LEN(W$(W%))GOTO670
700 D%(W%,1)=X1%D%(W%,2)=Y1%D%(W%,3)=D%:DONE=DONE+1
710 FORL%=1TOLEN(W$(W%))
720 A$(X1%,Y1%)=MID$(W$(W%),L%,1)
730 X1%=X1%+I%:Y1%=Y1%+J%
740 NEXT L%
750 NEXTW%
760 FORX%=1TO19:FORY%=1TO9
770 B$(X%,Y%)=A$(X%,Y%):NEXT:NEXT
780 FORX%=1TO19:FORY%=1TO9
790 IFA$(X%,Y%)="THENAS$(X%,Y%)=CHR$(RND(26)+96)
800 NEXT:NEXT:ENDPROC
810
820 DEFPROCDISPLAY
830 FORY%=1TO9:FORX%=1TO19
835 PROCLCASE(A$(X%,Y%))
840 PRINT" ";Y$;:NEXT
850 PRINT':NEXT:COLOUR1
860 PRINTCHR$11;L$;L$
870 FORX%=0TO5:FORY%=0TO2
880 PRINTTAB(Y%*15,X%+18);W$(X%*3+Y%+1)
890 NEXT:NEXT
900 PRINTTAB(0,24);L$;L$:COLOUR2:ENDPROC
910
920 DEFPROCLCASE(X$)
930 Y$="":FOR Z%=1 TO LENX$
940 IF ASC(MID$(X$,Z%,1))>=97 Y$=Y$+MID$(X$,Z%,1):GOTO960
950 Y$=Y$+CHR$(ASC(MID$(X$,Z%,1))+32)
960 NEXT:ENDPROC
970
980 DATA1,-1,1,0,1,1,0,1,-1,1,-1,0,-1,-1,0,-1

```

990 DATAcomputer, disk, screen, television, star, moon, book, volume
1000 DATAgirl, boy, father, mother, teacher, desk, school, pencil
1010 DATArubber, ruler, boxer, wrestler, terminal, backup, pear, apple
1020 DATAcorn, fruit, orange, banana, pomegranate, personal
1030 DATAspectacles, bed, chair, table, carpet, sideboard, bench
1040 DATAwindow, wall, door, ceiling, floor, kitchen, bathroom
1050 DATAbedroom, recorder, piano, flute, violin, accordion, cello
1060 DATAtrumpet, drum, trombone, viola, garden, grass, trees
1070 DATAflower, petal, leaf, daisy, carnation, cooker, padlock