

Anagram

In this version of the well-known puzzle, you may play against another person or against the computer. When two people play, one averts his head while the other types in a word for the computer to jumble up. When you play against the computer, a word is selected from those held in memory. Scores are maintained by counting the number of tries before the original word is recovered. Two players will arrange their own rules, but when playing against the computer, you may type Q for Quit, when the program will tell you what the word was and penalise you four points. One point is subtracted for each false try, but if you get the word, you gain one. Alter these values if you wish.



The program naturally falls into two portions dealing with solo or two-player games, the split taking place on line 80. From there down to line 470 deals with the two-player version. Each player is asked for his or her name and then the first player is asked to type in a word (line 220). This should be at least five letters long, as set by line 230. PROCJUMBLE then muddles up the word. This is done very simply by having a parallel array called A%(X%), initially blank, and we 'cross off' or enter the value 1 when the corresponding letter of the word has been used. We choose a letter at random in line 910 and ensure that it has not been used. If OK, we add that letter to a growing muddled string, V\$, in line 920. When all letters have been used - as evidenced by a simple count - we end the procedure. It will be seen that V\$ is also padded out with spaces between the letters; this is for print purposes, making the anagram easier to read. The anagram is now held in V\$ and the original word is still contained in W\$.

Players are now switched by line 300. The second player is addressed and the anagram presented, printed double height for clarity. The player's guess is now accepted on line 320. If correct, a jump is made to line 370; if not, his score of errors is incremented and the display brought up to date by PROCSCORE. If preferred, another call to PROCJUMBLE on line 360 would jumble the word afresh before the user has another try.

When the word is successfully identified, a message is given and the two players change roles. Play proceeds in this way for ten rounds, the winner being the one with the lowest score.

When playing against the computer, the player is asked for a difficulty level or word length, from 4 letters to 12 - extend this if you like. There are ten rounds as before, but to ensure that the same word is not presented twice, the computer keeps a record of the word number in the array W%(X). The words are in groups of ten in lines 940 onwards, with ten four-letter words followed by ten five-letter, ten six-letter, and so on. If you like, you could increase the number of words - there must be the same number in each group - and alter the bracketed value in line 640, where the position of the word is chosen. The actual word itself is found and read in line 650, being transferred to W\$, as required by PROCJUMBLE, so if you change the number of words you will also have to change the '10' in line 650. The rest of the program is quite straightforward.

Variables

P\$(2)	Names of players
P%(2)	Scores
A%(24)	Array for check when jumbling 24 characters allowed!

W%(10) Record of words (numbers) used
 C\$ A clearing string, printed to clear part of screen
 SCORE% Record of incorrect replies
 R% Record of correct replies
 G\$ General input string
 X% General counter
 P% Current player addressed or current problem (in solo game)
 W\$ Word input, or to be jumbled
 Q\$ Player' s guess at anagram
 G Timing dummy
 ROUNDS% Record of rounds between players
 WL% Chosen word length

```

10 REM - Anagram
20 MODE7:DIM P$(2),P%(2),A%(24),W%(10)
30 C$=STRING$(30," "):SCORE%=0:R%=0
40 PROCDBL(9,5,129,"ANAGRAMS")
50 PRINT'"'"TYPE 'C' TO PLAY AGAINST THE COMPUTER"
60 PRINT"OR 'P' TO PLAY AGAINST ANOTHER PERSON.""
70 REPEAT:INPUT G$:UNTIL G$="C" OR G$="P"
80 ROUNDS%=0:IF G$="C" GOTO600
90 CLS:PRINT'"'"PLEASE TYPE YOUR FIRST NAMES..."
100 FOR X%=1 TO 2:PRINT'"PLAYER #";X%;"..."
110 INPUT P$(X%):NEXT:P%=1
120 CLS:PROCDBL(0,1,129,P$(1))
130 PROCDBL(20,1,129,P$(2))
140 PRINTTAB(0,3);CHR$129;" ";
150 PRINTSTRING$(LEN(P$(1)),"_");
160 PRINTTAB(22);STRING$(LEN(P$(2)),"_")
170
180 REM - GAME LOOP
190
200 PROCSCORE
210 PRINTTAB(0,12);P$(P%);" _"
220 PRINT"PLEASE TYPE YOUR WORD"':INPUT W$
230 IF LEN(W$)>4 GOTO250
240 PRINT"SOMETHING LONGER, PLEASE.":G$=GET$:GOTO120
250 PROCJUMBLE
260
270 REM - CLEAR SCREEN, PRESENT ANAGRAM
280
290 PRINTTAB(0,12);C$;TAB(0,14);C$;TAB(0,18);C$
300 P%=3-P%:PRINTTAB(0,7);P$(P%);" _"
310 PRINT"YOUR ANAGRAM IS":PROCDBL(2,10,131,V$)
320 PRINTTAB(0,15);C$;CHR$13;:INPUT"YOUR WORD..." ,Q$
330 IF Q$=W$ GOTO370
340 P%(P%)=P%(P%)+1:PRINTTAB(0,18);"TRY AGAIN"

```

```

350 G=INKEY(100):PRINTAB(0,18);C$
360 PROCSCORE:GOTO320
370 PROCDBL(6,18,134,"RIGHT!"):ROUNDS%=ROUNDS%+1
380 IF ROUNDS%>10 GOTO450
390 PRINT'"NOW YOU CHANGE OVER..."
400 PRINT"PRESS RETURN WHEN READY.....":G$=GET$
410 FOR X%=1 TO 24:A$(X%)=0:NEXT:GOTO120
420
430 REM - END OF CONTEST *****
440
450 CLS:PROCDBL(5,6,129,"FINAL SCORES")
460 FORX%=1 TO 2:PRINT'P$(X%);TAB(20);P%(X%):NEXT
470 END
480
490 DEFPROCDBL(X%,Y%,C%,X$)
500 PRINTTAB(X%,Y%);CHR$(C%);CHR$(141);X$
510 PRINTTAB(X%,Y%+1);CHR$(C%);CHR$(141);X$
520 ENDPROC
530
540 DEFPROCSCORE
550 PROCDBL(2,4,129,STR$(P%(1)))
560 PROCDBL(22,4,129,STR$(P%(2))):ENDPROC
570
580 REM - PLAY AGAINST COMPUTER
590
600 INPUT'WHAT WORD LENGTH (4-12)",WL%
610 IF WL%<4 OR WL%>12 GOTO600 ELSE WL%=WL%-4
620 FOR X=1 TO 10:W%(X)=0:NEXT
630 FOR P%=1 TO 10:REM -TEN PROBLEMS
640 X%=RND(10):IF W%(X%)<>0 GOTO640 ELSE W%(X%)=99
650 RESTORE:FOR Y%=1 TO X%+10*WL%:READ W$:NEXT
660 CLS:PROCDBL(12,1,131,"S C O R E")
670 PRINTTAB(13);CHR$(131);"_____
680 PROCDBL(5,4,130,"RIGHT"):PROCDBL(23,4,129,"WRONG")
690 PROCJUMBLE:PROCDBL(7,6,130,STR$(R%))
700 PROCDBL(25,6,129,STR$(SCORE%))
710 PRINTAB(0,11);"YOUR ANAGRAM IS -"
720 PROCDBL(7,13,131,V$)
730 FORX=1TO9:PRINTSTRING$(30," ") :NEXT
740 PRINTAB(0,17);"YOUR GUESS.....":INPUT Q$
750 IF Q$=W$ GOTO810 ELSE SCORE%=SCORE%+1
760 IF Q$<>"Q" GOTO800
770 PRINT'CHR$(131);"THE ANSWER IS"
780 PROCDBL(14,19,129,CHR$(136+W$))
790 G=INKEY(1000):SCORE%=SCORE%+4:GOTO830
800 PRINT'"TRY AGAIN....":GOTO690
810 VDU7:PROCDBL(13,19,130,CHR$(136+"RIGHT!")):R%=R%+1
820 PRINT'"PRESS RETURN WHEN READY...":G$=GET$
830 NEXT P$:CLS:PRINTAB(0,5);
840 PRINT"YOUR SCORE WAS ";R%;" OUT OF 10"
850 INPUT"PLAY AGAIN (Y-N)",Q$:IF Q$<>"N"R%=0:GOTO600
860 END
870

```

```
880 DEFPROCJUMBLE
890 FOR X%=1 TO 20:A%(X%)=0:NEXT
900 V$="":X$="":FOR X%=1 TO LEN(W$)
910 Y%=RND(LEN(W$)):IF A%(Y%)<>0 GOTO910
920 A%(Y%)=1:V$=V$+MID$(W$,Y%,1)+" "
930 NEXT:ENDPROC
935
940 DATAGODS,FOAM,PIPE,LADY,ZOOM,CANE,DESK,BOOK,CELL,CHIP
950 DATAMESSY,GLOVE,CHESS,CIGAR,GLASS,CHUTE,BOOTS,MATCH
960 DATAGHOST,LOCKS,DONKEY,CONTENT,VOLUME,BIGGER,SADDLE
965 DATAAREPEAT,CHANGE,CRUNCH,HAMMER,CITIES,CHIMNEY,DRAWING
970 DATAMANHUNT,TOBACCO,CABINET,SCARLET,COMMAND,PROPOSE
975 DATACURTAIN,CHESTNUT,MINISTER,LIFEBELT,COMPUTER,IDENTITY
980 DATATELETEXT,CHASTISE,FILIGREE,PARAPETS,RUNABOUT
985 DATAINFLATED,ELONGATED,REDUCTION,VANISHING,CARNATION
990 DATAJACOBITES,JUDGEMENT,PARTITION,STROLLING,SWINEHERD
998 DATAJACOBITES,JUDGEMENT,PARTITION,STROLLING,SWINEHERD
999 DATACOMPANY,FAVOURABLE,UNDEFENDED,OPERATIONS
1000 DATAEVENTUALLY,REPUTATION,CHANDELIER,LOCOMOTIVE
1005 DATASEAMANSHIP,ZOOLOGICAL,WRISTBANDS,PROGRESSION
1010 DATABENEVOLENCE,MATERIALIZE,WEIGHBRIDGE,CONSOLIDATE
1015 DATADELINQUENTS,REGURGITATE,TEMPESTUOUS,ASSASSINATE
1020 DATAAGRICULTURE,INCINERATORS,EXTINGUISHER,ABSTRACTIONS
1025 DATASURVEILLANCE,DISENCHANTED,MISCALCULATE,UNQUESTIONED
1030 DATAACQUAINTANCE,RELINQUISHED,OBSTRUCTIONS
```