

Slide

This program is a computer implementation of those little plastic puzzles that you can buy, in which the user slides the pieces one at a time into the only available space. The object of the exercise is to arrange them in their natural order:

```
A B C D
E F G H
I J K L
M N O
```

The advantage of this implementation over the plastic type is that every start condition is totally random and it is amazingly fast. Also, as a special dispensation, but only once per game, the user may swap any two letters, which is useful of course if you get into a situation you cannot see your way out of. Finally, the computer counts your moves and gives you a grading when - or if - you complete the puzzle.

Lines 30 and 40 determine the start condition, then lines 60 to 200 prepare the screen and print the puzzle, which appears as black letters on a white background, with a large red border. The instructions of line 80 appear in black on this border. Notice that there is no need to press RETURN after each move.

The game loop starts at line 120, so the computer is actually reprinting the puzzle after every move, but it is so fast that no delay is experienced and you can rattle the keys as fast as you are able. Notice too that although the puzzle is presented on a 4 x 4 square - a matrix, in the jargon - it is actually stored as a linear array, B%(1) to B%(16). After each move, the computer makes a swift check to see if you have won. It does this on lines 180-200. If the puzzle is not finished, on lines 210-250 it accepts a legal command only, then checks to see if you can move the letter that you say that you wish to



move. The rest of the program is fairly well annotated with REM statements, and it will be seen that the player grading takes place on lines 600-690.

Variables

B%(16)	The array for the letter and space positions
SWAP%	If 0, swap option has not yet been taken
M%	The count of moves
I%	General counter
J%	General counter
SPACE%	The position of the space
X%	General counter
A\$	Command letter input by user
L%	ASCII value of letter to be moved
S\$, T\$	Two letters to be swapped

```

10 REM - Slide
20 DIM B%(16):SWAP%=0:M%=0:MODE7
30 FOR I%=65 TO 80
40 J%=RND(16):IF B%(J%)<>0 GOTO40 ELSE B%(J%)=I%
50 NEXT:SPACE%=J%
60 MODE5:GCOL0,129:CLG:COLOUR 131:GCOL0,2
70 VDU5:MOVE 0,400
80 PRINT"Letter to be moved, S for swap or Q for quit.":VDU4
90 REM - Define windows
100 VDU28,5,15,13,6,12,24,0;0;400;80;14
110 REM - Print puzzle
120 VDU4:COLOUR 0:PRINTTAB(1,1);
130 FOR X%=1 TO 16
140 IF B%(X%)=80 PRINT " "; ELSE PRINT CHR$B%(X%);" ";
150 IF X% MOD 4=0 PRINT'" ";
160 NEXT:VDU5
170 REM - Check for solution
180 X%=1
190 IF B%(X%)<>X%+64 GOTO220
200 IF X%=15 GOTO570 ELSE X%=X%+1:GOTO190
210 REM - Get move
220 REPEAT:A$=GET$
230 UNTIL (A$>="A" AND A$<="O")OR A$="Q" OR A$="S"
240 IF A$="Q" THEN 520
250 IF A$="S" THEN 400
260 L%=ASC(A$):X%=1
270 IF B%(X%)<>L% X%=X%+1:GOTO270
280 REM - Check legality
290 IF ABS(SPACE%-X%)=4 GOTO370
300 IF ABS(SPACE%-X%)<>1 GOTO350
310 IF X%/4<>INT(X%/4) GOTO330
320 IF SPACE%=X%+1 GOTO350
330 IF (X%-1)/4<>INT((X%-1)/4) GOTO370
340 IF SPACE%<>X%-1 GOTO370
350 VDU7:GOTO220
360 REM - OK; Swap
370 M%=M%+1:B%(X%)=80:B%(SPACE%)=L%:SPACE%=X%
380 VDU16:MOVE 0,50:PRINT;M%:GOTO120
390 REM - Swap option
400 IF SWAP%=1 VDU7:GOTO120
410 VDU24,0;0;1279;270;16:MOVE0,220
420 PRINT "Which 2 letters do you want to swap?"
430 PRINT"(Press RETURN after each) "
440 INPUT S$,T$:I%=ASC(S$):J%=ASC(T$)
450 IF I%<65ORJ%>79ORJ%<65ORJ%>79ORJ%=J%VDU7,16:GOTO420
460 REM - Find and swap
470 X%=1:Y%=1
480 IF B%(X%)<>I% X%=X%+1:GOTO480
490 IF B%(Y%)<>J% Y%=Y%+1:GOTO490
500 B%(X%)=J%:B%(Y%)=I%
510 SWAP%=1:GCOL0,129:VDU16,24;0;0;400;80;:GOTO120
520 MODE7
521 PRINTTAB(5,10);"You quit the game after ";M%;" moves."
530 PRINT'"Do you want another game (Y-N)?"

```

```
540 REPEAT:G$=GET$:UNTIL G$="Y" OR G$="N"
550 IF G$="N"PRINT'''' G O O D B Y E":END
560 M%=0:SWAP%=0:FORX%=1 TO 16:B%(X%)=0:NEXT:GOTO30
570 MODE7:PRINTTAB(3,5);"C O N G R A T U L A T I O N S !"
580 FOR X%=1 TO 10:FOR Y%=50 TO 150
582 SOUND 17,-15,Y%,2:NEXT:NEXT
590 PRINT''''You solved the problem in ";M%;" moves!"
600 IF M%<75 PRINT"You're a genius!":GOTO530
610 IF M%<100 PRINT"Absolutely brilliant!":GOTO530
620 IF M%<125 PRINT"Tremendous":GOTO530
630 IF M%<150 PRINT"Excellent!":GOTO530
640 IF M%<175 PRINT"Very good!":GOTO530
650 IF M%<200 PRINT"Good!":GOTO530
660 IF M%<300 PRINT"Pretty fair.":GOTO530
670 IF M%<400 PRINT"Not very good, I'm afraid.":GOTO530
680 PRINT"Why don't you take up knitting?":GOTO530
```