Double maze   P62313_en

Quart Concurs de Programació de la UPC - Semifinal (2006-09-20)

Recently I discovered a fascinating solitary game at minijuegos.com called Doublemaze.

The game goes as follows: Two balls (call them A and B) are located at some cells in two independent mazes. The player's goal is to bring both balls to their destinations cells, marked with a flag. In order to move the balls, the player can decide to press the up, down, left or right keys, which move both balls one position in the intended direction. However, walls can obstruct the movement of a ball. The game ends if one of the balls falls out of the maze (the player loses) or if both balls reach the destination flags at the same time (the player wins and advances to a new level).
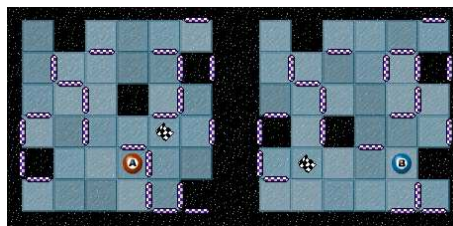
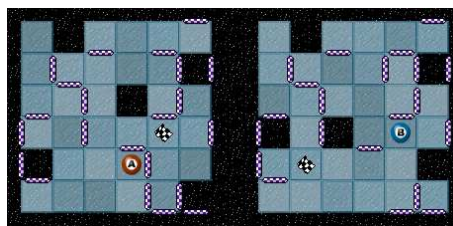For instance, the first level starts in the following position:
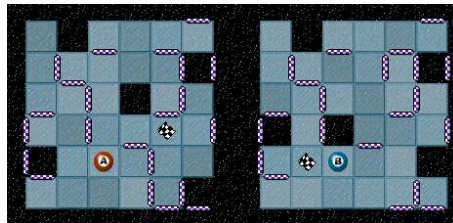
If we press the 'right' key, both balls move one cell to their right:
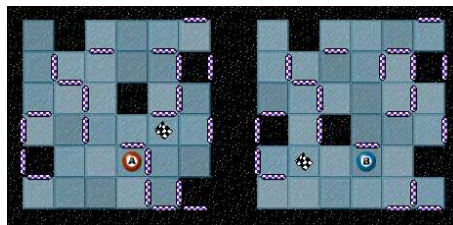
If we press the 'right' key again, ball A stays in the same place because of the wall:

If we finally press the 'up' key, ball A also stays in the same place:

If now the player would press the 'right' and 'down' keys consecutively, he or she would lose because ball B would fall out of the maze.

Please write a program to compute the minimum number of moves necessary to solve a double maze.

Input

Input consists of several double mazes with exactly the format shown in the sample input. All the fields always appear in the same order. Horizontal walls are on the top of the cells of the given coordinates. Vertical walls are at the left of the cells of the given coordinates. At boths mazes, the ball and the flag are at different positions, an never on a hole. The maximum size is 30.

Output

For every double maze, print the minimum number of movements needed to solve the game. If there is no solution, print "OOPS".

Sample input

```
SIZE: 6
LEFT MAZE:
    hole: 2,1 6,2 4,3 1,5 6,6
    h_wall: 6,1 3,2 5,2 2,3 1,4 5,4 4,5 1,6 5,7 6,7
    v_wall: 2,2 6,2 7,2 3,3 6,3 1,4 3,4 7,4 1,5 5,5 5,6 6,6
    ball: 3,5
    flag: 5,4
RIGHT MAZE:
    hole: 2,1 6,2 1,4 3,4 6,5
    h_wall: 6,1 3,2 5,2 2,3 1,4 5,4 4,5 1,6 5,7 6,7
    v_wall: 2,2 5,2 6,2 7,2 3,3 6,3 1,4 3,4 7,4 1,5 6,6
    ball: 3,5
    flag: 2,5

SIZE: 3
LEFT MAZE:
    hole:
    h_wall:
    v_wall:
    ball: 1,1
    flag: 3,3
RIGHT MAZE:
    hole:
    h_wall:
    v_wall:
    ball: 1,1
    flag: 3,3
```

Sample output

```
12
4
```

Problem information