Sokoban

The game of Sokoban at its most basic, consists of a man pushing a box in a maze, from its source to its destination. The maze consists of cells that are empty or have obstacles. The box is heavy and a man can push it from one cell to an adjacent cell only if he is exactly behind the box. If there is no space for the man behind the box, then the box cannot be pushed in that direction. On pushing, the box is moved to an adjacent cell based on the direction in which the man has pushed. The box can only be pushed to its four-adjacent cells, and hence diagonal move of the box is not possible.

You are given a rectangular matrix representing the maze. There is one man, one box and one destination cell, in the maze. The task is to push the box to its destination. There are obstacles in the maze that hinder the movement of the box. There is a unique path for the box to move from its initial position to its destination.

Note that a man, when he is not pushing the box, can jump from a cell to any other empty cell. Hence the presence of obstacles or the box in between does not hinder the man's movement from one empty cell to another, as is necessitated in the second sample input. The destination can be considered an empty cell, as far as the man's movement is concerned.

Given a rectangular matrix representing the maze as input, you have to output the length of the path taken by the box in moving to its destination. The length of the path will always be greater than or equal to one.

Input specification:

The first line contains two integers R and C denoting the number of rows and columns of the matrix respectively. The matrix of characters then follows where:

- B indicates the initial position of the box
- M indicates the position of the man
- D indicates the destination of the box
- # indicates an obstacle
- @ indicates a free cell

  The matrix is given in the form of R lines each containing C consecutive characters.

Output specification:

The output should be the length of the path.

Sample Input and Output:

Input:

```
5 5
@@@D@
@##@@
@##@@
MB@@@
@@@@@
```

Output:

```
5
```

Input:

```
6 5
@@@@@
@@@@@
D@#@@
###@@
@B@@@
#@@M@
```

Output:

```
9
```