Challenge 4 - Brave Knight

« PREV NEXT »

The princess is lost in a dangerous place. The only mission the brave knight has been preparing himself for all his life is to safe her.

The terrain is represented by a grid of squares. The material of each square can be ground, trampoline or lava. Only lava squares (represented by the character '**#**') are considered invalid, the rest are valid squares.

The brave knight moves through the map in 'L'-shaped jumps like a chess knight. It can move to a square that is two squares away horizontally and one square vertically, or two squares vertically and one square horizontally. He can jump over any square but he cannot land in an invalid square or outside the terrain.

From ground squares (represented by the character '**.**') the knight moves as described above. But from trampoline squares (represented by the character '*') the knight does an super powered double jump that is four squares horizontally and two squares vertically, or four squares vertically and two squares horizontally.

The knight located on square **S** must rescue the princess located on square **P** and take her to the safe destination square **D** (**S**, **P** and **D** are ground squares).

Input

The first line has an integer **C**, which is the number of cases for the problem. Each case has a line with 2 integers separated by a space, **N** and **M**, that represent the dimensions of the map.
N lines follow, each one containing **M** characters that represent the squares of the map.

Output

For each case, there should be a line starting with "Case #x: " followed by the minimum number of jumps the knight has to make to rescue the princess and take her to the safe place. If there is no way, the output should be IMPOSSIBLE.

Limits

- 1 ≤ **C** ≤ 100
- 2 ≤ **N**, **M** ≤ 1000

Examples

```
Case 1:

4 5
P#.S#
.#...
#.#..
.D..#
```

```
Case 2:

4 5
.#.D#
.#...
#S#.#
.P..#
```

```
Case 3:

6 6
S#....
##*..D
......
######
......
P.....
```

In Case 1 the knight could follow this path: [**S**] (0,3) -> (2,4) -> (1,2) -> [**P**] (0,0) -> (1,2) -> [**D**] (3, 1) So the minimum number of jumps is 5.
In Case 2 there is no way to reach D square.
In Case 3 there is a double jump square so the knight will travel this path: [**S**] (0,0) -> (1,2) -> [**P**] (5,0) -> (4,2) -> (2,3) -> [**D**] (1,5) making a total of 5 jumps.

Sample Input

```
3
4 5
P#.S#
.#...
#.#..
.D..#
4 5
```

```
.#.D#
.#...
#S#.#
.P..#
6 6
S#....
##*..D
......
######
......
P.....
```

Sample Output

```
Case #1: 5
Case #2: IMPOSSIBLE
Case #3: 5
```

Test your code

You can test your program against both the input provided in the test phase and the input provided in the submit phase. A nice output will tell you if your program got the right solution or not. You can try as many times as you want to. Be careful with extra whitespaces, the output should be exactly as described.

Test your program against the input provided in the test phase

Download test input

Program output:

Browse... No file selected.

Submit test output

Test your program against the input provided in the submit phase

Download input

Program output:

Browse... No file selected.

Submit output

During the submit phase, in some problems, we might give your program harder inputs. As with the test token, a nice output will tell you if your program got the right solution or not. You can try as many times as you need.

In the actual contest you first need to solve the test phase before submitting the code, you must provide the source code used to solve the challenge and you can only submit once (once your solution is submitted you won't be able to amend it to fix issues or make it faster).

If you have any doubts, please check the info section.

« PREV NEXT »

Tweet about this! **Follow**

#TuentiChallenge8 **@TuentiEng**