

Get Out of This Maze!!!

Filename: maze

Getting out of mazes can be quite tricky! So tricky in fact, that you've decided that it's better to write a program to determine the fastest way out of a maze than to try to wing it. For the purposes of this problem, a maze will be described as a two dimensional grid, where each square is one of the following types:

Outside Border Square (~)

Illegal Square to Travel on (X)

Valid Square to Travel on, not on the Outside Border (-)

Your Initial Starting Square (S)

Here is an example of a maze with 5 rows and 7 columns:

```
~~~~~  
~XXXX~  
~X-S-X~  
---X-X~  
~~~~~
```

We consider reaching any Outside Border Square as getting out of the maze. In a single move, you may move up, down, left or right. Thus, for this maze, one of the paths that is fastest to get out would be to move right, move down and move down again, for a total of three moves. (One could also move left, down and down and get out in three moves as well.)

The Problem

Given a maze as described previously, determine whether or not there is a way to escape to the boundary, and if so, what the shortest distance to escape the maze to any of the boundary positions is. At each move, you may move up, down, left or right from your previous spot, so long as the new spot isn't forbidden.

The Input

The first line of the input file will contain a single positive integer, c ($c \leq 100$), representing the number of input cases. The input cases follow, one per line. The first line of each input case will contain two positive integers, r ($3 \leq r \leq 300$), and c ($3 \leq c \leq 300$), representing the number of rows and columns, respectively, in the maze. The following r lines will contain strings of exactly c characters, describing the contents of that particular row using the symbols described above (~, X, -, S). You are guaranteed that the first and last rows and first and last columns will only contain the border character, ~. You are guaranteed that this character will not appear anywhere else in the grid. Exactly one non border character will be an S, so there will always be exactly one starting location. Finally, the rest of the squares will either be -, to indicate that that square is valid to travel to, or X, to indicate that you may not travel to that square.

The Output

For each case, if it's possible to reach the border of the maze, output the fewest number of steps necessary to reach any border square. If it's not possible, output -1.

Sample Input

```
3
3 3
~~~
~S~
~~~
5 6
~~~~~
~XXXX~
~XS-X~
~-XX-~
~~~~~
5 7
~~~~~
~XXXXX~
~X-S-X~
~--X-X~
~~~~~
```

Sample Output

```
1
-1
3
```