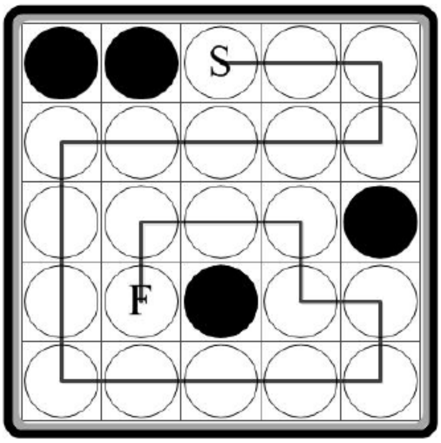
6567 Full Board

Full Board is an interactive one-person puzzle/game by LightForce played on a rectangular grid.

A game starts with an $M \times N$ board with some squares marked as "obstacles" (drawn as black dots in the figure below). The player chooses a starting position for a ball (marked S in Figure below) and chooses a direction (up, down, left, or right) to advance. Once the direction is chosen, the ball will advance in that direction until it hits an obstacle, the boundary of the board, or its own trajectory. When it hits one of these, the ball stops. Then the player is allowed to choose another direction, and the ball will advance in the same manner. The game ends when no legal move can be made. The player wins if and only if the trajectory includes all the empty squares on the board. The figure on the right traces a trajectory that shows one way to win the first game of the Sample Input in 10 steps.



S: Position at Start
F: Position at Finish

Given an initial setting of a board, write a program to calculate the minimum number of steps to win the game.

Input

Each case on the input file begins with two integers m and n ($1 \leq m, n \leq 30$), indicating the size of board. The next m lines describe the initial setting of the board. Each line contains n characters, either '*' or '.', indicating if the corresponding square is an obstacle or empty, respectively. It is guaranteed that the initial board is not fully covered by obstacles. Process until end-of-file; there is no end of data flag.

Output

Output the minimum number of steps to cover the board. Follow the format exactly: 'Case', one space, the case number, a colon and one space, and one integer indicating the minimum number of steps to win the game, or if there is no way to win, the number '-1'. Do not print any trailing spaces.

Sample Input

```
5 5
**...
.....
....*
..*..
.....
3 4
****
*...
*..*
```

Sample Output

```
Case 1: 10
Case 2: 3
```