

A Mouse in a Maze

There is a mouse stuck inside a maze. You have to write a program to help the mouse to exit the maze given the maze, starting position of the mouse and exit locations.

A maze is a two-dimensional grid having **R** rows and **C** columns, having **RxC cells** inside it. Each cell can either be a **free cell** or a **blocked cell**. The mouse cannot go into a blocked cell. Starting cell and exit cells will always be free cells so that the mouse can go.

At the start, mouse is in the starting cell. In each step, mouse can only step into 1 of 4 adjacent cells (**up, right, down, left**) from the cell, the mouse is in. The mouse can not step outside the maze and also mouse cannot step into a blocked cell. It takes **1 second** for the mouse to step into an adjacent cell.

Find the **shortest time** for the mouse to move into one of the exit cells.

Input Format

The first line of the input has **2** integers **R** and **C** respectively, separated by a space. **R** is the number of rows in the maze and **C** is the number of columns in the maze. Next **R** lines have **C** characters each representing the **RxC grid**. Each character represents a cell. Each character can be one of the following representing:

- **“.”** (dot/full stop) - An empty cell
- **“#”** - A blocked cell
- **“S”** (capital S) - The starting cell
- **“E”** - An exit cell.

Constraints

- $1 \leq R, C \leq 20$

Output Format

Output just **one integer** representing your answer, the **shortest time in seconds** for the mouse to reach from starting cell to one of the exit cells.

If the mouse can't reach an exit then output **-1**.

Sample Input 0

```
6 10
....S#E...
...###...
...#E....
#.###.###
.....#..
.....#..
```

Sample Output 0

14

Sample Input 1

```
6 10
....S#....
...###....
...#E#....
#.#.#.#.#
.....#..
.....#.E
```

Sample Output 1

```
-1
```