



PROBLEM 6 – MAZE MADNESS

You have been placed somewhere in a maze and you wish to escape by the shortest possible route. Fortunately you have been given a map of the maze. Before setting off, you wish to calculate the distance you need to travel. Your task is to write a program that will calculate the shortest distance to leave the maze. Note that there may be more than one exit and the specified start position could be at any location within the maze.

The maze is set on a grid that has M columns and N rows, with $1 \leq M, N \leq 100$. Some squares of this grid have impenetrable walls of negligible thickness between them (or on their outside border). You may move from any square to a horizontally or vertically adjacent square (possibly outside the maze, thus escaping) provided that there is no wall between them. Each single move between squares adds 1 meter to the distance travelled.

INPUT FORMAT

The input involves a series of scenarios. Within each of the scenarios the first line has the size of the maze. This is given as two numbers M and N . Then the maze is drawn on $2*N+1$ lines and $2*M+1$ columns using the printable characters “-”, “|”, “+”, “.”, “ ” (space), and “s”:

- “|” is used for “vertical” walls,
- “-” is used for “horizontal” walls,
- “+” is used to indicate boundaries between rows and columns (there are always $(N+1)*(M+1)$ of these),
- “.” is used for wall openings,
- “ ” (space) is used for empty squares,
- “s” is used to show your start location (there is exactly one “s”).

A line with "0 0" indicates the end of the scenarios.



SAMPLE INPUT:

```
1 1
+-+
|s.
+-+
3 2
+-+-.+.
|.s||
+-+.+-+
|. . .
+-+--+
5 6
+-+--+--+
|. . . . |
+-+-.+-+--+
| |s. . . |
+.+-+--+-.+.
| | . . . |
+.+-+--+-.+.
| | . . . |
+.+-+.+-+.+
|. . | | |
+-+-.+.+.+
. . . | . |
+-+--+--+--+
3 2
+-+-.+.
|.s||
+-+--+
|. . .
+-+--+
0 0
```

OUTPUT FORMAT

Output a single line for each of the scenarios. This line should contain either "Maze *i*: *d*" or "Maze *i*: No escape!", where *i* is the scenario number (counting from 1) and *d* is the minimum distance (in meters) needed to escape (use single spaces for spacing).

SAMPLE OUTPUT:

```
Maze 1: 1
Maze 2: 3
Maze 3: 12
Maze 4: No escape!
```