

Problem C. WallRepair

Problem Statement

The front of your house is covered with decorative square bricks, and you're worried that some of the bricks aren't properly supported and may fall down. You want to add enough bricks so that the wall is stable. In the below diagram, bricks are marked with 'X' and empty spaces are marked with '.'.

```
..X...X.....
....X.X...XX
.....X
..X.....X.
```

The wall is stable if each brick is supported by bricks all the way to the ground. To fix the wall above, you would need to add 8 bricks in the positions marked with '#' below:

```
..X...X.....
..#.X.X...XX
..#.#.#...#X
..X.#.#...X#
```

Each element of **wallRows** contains one row of the wall - in order from top to bottom - with 'X' for bricks and '.' for open spaces. Return the minimum number of bricks that must be added in order to make the wall stable (see example 0 for more info).

Definition

Class: WallRepair

Method: bricksRequired

Parameters: String[]

Returns: int

Method signature: int bricksRequired(String[] wallRows)

(be sure your method is public)

Constraints

- **wallRows** will contain between 1 and 50 elements, inclusive.
- Each element of **wallRows** will contain between 1 and 50 characters, inclusive.
- Each element of **wallRows** will contain the same number of characters.
- Each element of **wallRows** will contain only '.' and 'X' characters.

Examples

0)

```
{  
  "..X...X.....",  
  "....X.X...XX",  
  ".....X",  
  "..X.....X."}
```

Returns: 8

The example from the problem statement. Basically you find a hole with a brick above it, and put a brick there. You keep doing this until there are no holes with bricks above them (and therefore no bricks with holes below).

1)

```
{"..X...X..X"}
```

Returns: 0

Bricks on the lowest level do not need to be supported.

2)

```
{  
  ".....X.....",  
  ".....X..X..X..X.....",  
  "X.....X..X.....X",  
  "....XXXXX..X.XXXXX..X",  
  ".....X..X..X.....X",  
  "...X....X..X.....X",  
  ".X...XX..X..X..X.....",  
  ".....X...X...XX.....X"}  
}
```

Returns: 53

This problem statement is the exclusive and proprietary property of TopCoder, Inc. Any unauthorized use or reproduction of this information without the prior written consent of TopCoder, Inc. is strictly prohibited. (c)2022, TopCoder, Inc. All rights reserved.