Back To School '16: Screensaver

Time limit: 1.0s **Memory limit:** 64M

On the first day of school, the computer science teacher shows a couple of impressive screensavers on the computer. The screensaver called :tim: stands out to you, mainly due to the simple rules that govern :tim:.

:tim: is a beautifully rendered three-dimensional arena with a couple of walls and the observer gets to bounce off these walls. The observer occasionally bounces out of the arena, which causes the animation to start again with a new set of walls.

The arena can be represented as a grid with N columns and M rows. Empty spaces are denoted with `.` and walls are denoted with the character `-`, `|`, `/` or `\`. The walls are similar to mirrors; the observer hits the surface of the wall and changes to the corresponding direction. The only exception to this rule is with `-` and `|` walls. The observer bounces off `-` when going vertically, and passes through when going horizontally. The same exception applies when going vertically through the `|` wall.

One notable feature of :tim: is that whenever the user bounces on a wall, that wall rotates 90°. For example, when the user bounces off the wall with direction `|`, the wall now has direction `-`.

Every tick, the observer moves into an adjacent cell in the grid. The player can travel horizontally or vertically, but **never** diagonally. The observer begins at the cell containing `O` (not occupied by a wall) and initially **moves horizontally to the right**. The animation is reset once the observer leaves the grid.

Of course, you want to implement this screensaver yourself! You have the renderer up and running, but you want the screensaver to last more than T ticks. Given the grid, can you determine whether the requirement is satisfied?

Input Specification

The first line contains the integers N, M $(1 \le N, M \le 100)$, and T $(1 \le T \le 10\,000)$.

On each of the next M lines, there are N characters. It is guaranteed that the character `O` appears exactly once.

Output Specification

If the observer stays within the grid for **more than T ticks**, output `The observer stays within the grid.`

Otherwise, output `The observer leaves the grid after X tick(s).`

Sample Input 1

```
4 4 100
O..\
....
....
.-./
```

Sample Output 1

```
The observer leaves the grid after 10 tick(s).
```

Sample Input 2

```
2 1 3
O|
```

Sample Output 2

```
The observer leaves the grid after 3 tick(s).
```

Sample Input 3

```
2 1 2
O|
```

Sample Output 3

```
The observer stays within the grid.
```