ACM-ICPC Live Archive

7478 Association for Cool Machineries (Part 1)

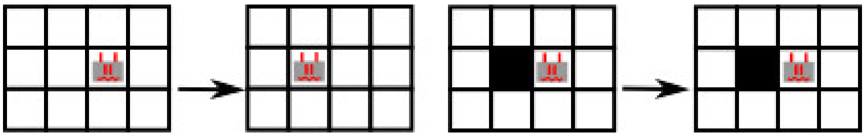
You are the boss of ACM (Association for Cool Machineries), an upstanding company with a single goal of world domination.

Like all other world domination oriented villains, you dream of having a gigantic evil robot doing evil deeds. I mean, is there anyone who can resist the urge of having your own personal gigantic laser-firing robot? Thus, it is to your ultimate joy when the scientific committee of ACM told you about their success in inventing a prototype for the GL-bot (shorthand for Giant Laser Robot).

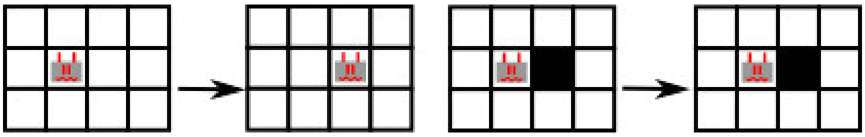
As a demonstration, they have placed GL-bot on a field modeled as an $N \times N$ grid. Each cell in the grid is either empty or impassable. All cells on the borders of the grid are impassable. GL-bot is initially located on one of these empty cells.

The GL-bot's behavior is programmed with a string of length at most N. Each character in the string is either <, >, ^, or v, denoting four directions of movement. GL-bot will continuously move on the grid according to this string. The movement works as follows. GL-bot considers the characters in the string one by one, starting from the first character until the last character, and then loop back to the first character and repeat. GL-bot performs a movement depending on the character being considered (for the images below, black cells denote impassable cells, while white cells denote empty cells):

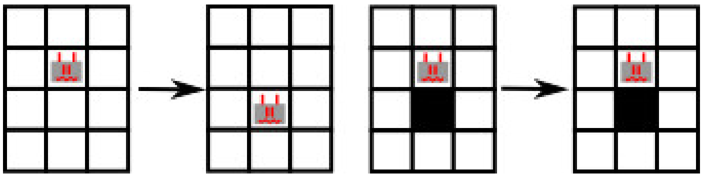
- <: GL-bot moves to the left one step, if able.



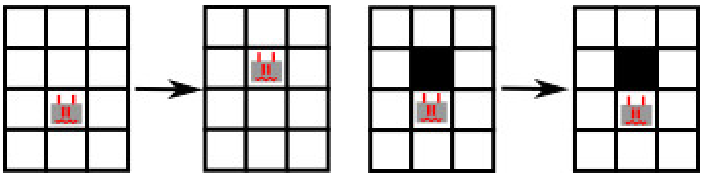
- >: GL-bot moves to the right one step, if able.



- v: GL-bot moves down one step, if able.



- ^: GL-bot moves up one step, if able.



More specifically, GL-bot moves one cell from its current location to the location as determined by the character. However, if the new location is impassable, the robot does not perform the move, and this movement is skipped. GL-bot then considers the next character in the program.

The trail of GL-bot is the (possibly infinite) sequence of (possibly non-distinct) cells of the grid that GL-bot traverses in order. The trail can be viewed to be generated as follows: initially, before GL-bot performs any movement, the trail contains one element: the initial location of GL-bot. Whenever GL-bot moves into a different cell, the new location of GL-bot is added to the trail. In particular, this implies that no two consecutive cells in the trail are the same — if the robot is unable to move in a particular direction because it was occupied by an impassable cell, the trail is not extended.

You giggle giddily as you watch GL-bot move. For some program and for some grid configuration, GL-bot will never stop moving, no matter how long time has passed. It can be proven that in this case, the trail will eventually end up repeating a continuous subsequence of itself. But you..., you don't see such a thing. What you see is a robot that moves perpetually, as if it is sapient. Your fascination with GL-bot leaves the scientific committee worried, as GL-bot is not actually sapient yet (it just needs a few more tweaks). Thus, they attempt to convince you by showing that the movement is simply follows a pattern. If the length of the trail of the robot is finite, you should output 1. Otherwise, let X be the smallest integer such that the suffix of the trail will be a repetition of a continuous subsequence of the trail of length exactly X. Find X.

Input

The input file contains several test cases, each of them as described below.

The first line contains a non-negative integer $3 \leq N \leq 200$, denoting both the dimension of the field and the maximum possible length of GL-bot's program.

The next line contains a non-empty string of length at most N, consisting only of the characters ‘<’, ‘>’, ‘^’, and ‘v’. This string denotes GL-bot's program.

Thereafter follows N lines, each consists of a string consisting of N characters. Each line describes a row of the grid — the j-th character of the i-th line denotes the content of the cell located at the i-th row from the top and j-th column from the left, and is either one of these:

- `#`: denotes an impassable cell.
- `.`: denotes an empty cell.
- `R`: denotes an empty cell on which the robot initially resides.

It is guaranteed that the grid will contain exactly one character ‘R’, and that the first and final rows and columns consist entirely of the character ‘#’.

Output

For each case, if the robot's trail is of finite length, print a single integer ‘1’, on a line by itself. Otherwise, print a single integer X as described in the problem statement.

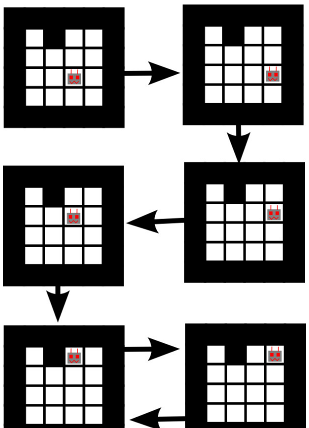
Explanation:

In the first example, the trail of the robot is illustrated as follows:

Hence, $X = 2$ in this case.

Sample Input

```
6
>^<^
######
#.#..#
```



```
#....#
#..R.#
#....#
######
4
v<^>
####
#.R#
#..#
####
4
<<<
####
#.R#
#..#
####
5
<<>>
#####
#R..#
#####
#####
#####
```

Sample Output

```
2
4
1
4
```