Robots on a grid

You have recently made a grid traversing robot that can find its way from the top left corner of a grid to the bottom right corner. However, you had forgotten all your AI programming skills, so you only programmed your robot to go rightwards and downwards (that's after all where the goal is). You have placed your robot on a grid with some obstacles, and you sit and observe. However, after a while you get tired of observing it getting stuck, and ask yourself \How many paths are there from the start position to the goal position?", and \If there are none, could the robot have made it to the goal if it could walk upwards and leftwards?" So you decide to write a program that, given a grid of size n x n with some obstacles marked on it where the robot cannot walk, counts the different ways the robot could go from the top left corner s to the bottom right t, and if none, tests if it were possible if it could walk up and left as well. However, your program does not handle very large numbers, so the answer should be given modulo 2^31 - 1.

Input

On the first line is one integer, 1 < n <= 1000. Then follows n lines, each with n characters, where each character is one of '.' and '#', where '.' is to be interpreted as a walkable tile and '#' as a non-walkable tile. There will never be a wall at s, and there will never be a wall at t.

Output

Output one line with the number of different paths starting in s and ending in t (modulo 2^31 - 1) or THE GAME IS A LIE if you cannot go from s to t going only rightwards and downwards but you can if you are allowed to go left and up as well, or INCONCEIVABLE if there simply is no path from s to t.

Example

Input:

```
5
.....
#..#.
#..#.
...#.
.....
```

Output:

```
6
```