Back From Summer '19 P2: Straying From God's Light

Time limit: 1.0s **Memory limit:** 128M
Java: 1.8s

Marcus is stuck in a 2-dimensional grid of size $N \times N$, consisting of `.` for walkable spaces and `#` for unwalkable spaces! He is at the top left corner (position $(1, 1)$) and must travel to the bottom right corner (position (N, N)) through only walkable spaces. He can **only move** left, right, and down. In a path, let D, L, R be the number of times he moves down, left, and right, respectively. Let the *cost* of the path be $D^2 + L^2 + R^2$. Marcus wants to find a path from $(1, 1)$ to (N, N) that has the minimum *cost*.

Marcus wants to know the minimum possible cost. Please help him!

Input Specification

The first line will contain the integer N $(1 \leq N \leq 1000)$, the size of the grid.

The next N lines will each contain N characters, either `.` for a walkable space or `#` for an unwalkable space. The first character of the first line will be position $(1, 1)$ and the N^{th} character of the N^{th} line will be position (N, N).

It is guaranteed positions $(1, 1)$ and (N, N) will be walkable (`.`).

Output Specification

Output the minimum *cost* path for Marcus. If there is no path, output `-1`.

Constraints

Subtask 1 [25%]

$N \leq 25$

Subtask 2 [75%]

No additional constraints.

Sample Input

```
6
......
.#....
##.##.
......
.#####
......
```

Sample Output

```
78
```

Explanation For Sample

The minimum *cost* path consists of moving down $D = 5$ units, left $L = 2$ units, and right $R = 7$ units, for a total of $5^2 + 2^2 + 7^2 = 78$.