Problem B
Left labyrinths

> "The instructions to turn always to the left reminded me that such was the common procedure for discovering the central courtyard of certain labyrinths."
>
> in *The garden of forking paths* by Jorge Luis Borges

Problem

A fellow librarian has exhumed in the Library of Babel a vast catalogue of labyrinths. It is our duty to classify them all and we count on your assistance. The plans of the labyrinths are already digitized and we need you to write a program to decide if the central courtyard of a given labyrinth can be reached following the simple procedure of always turning left at any intersection, and thus be called a *left labyrinth*.

The digitalization process reduces the plan of a labyrinth to a grid of cells, being each cell either a block of wall or just floor. Walls are sequences of contiguous blocks, forming either horizontal or vertical corridors between them. Each labyrinth has a single entrance, a hole in its exterior walls, and a single central courtyard. The courtyard differs from the corridors in the shape: a floor cell in a corridor has at least 2 block of wall on each side. You can assume that each plan has a single labyrinth and its outside walls can be contoured within the plan.

Input

The first two input lines are the integers, smaller than 100,

n - the number of lines in the map, and

m - the number of characters per line in the map.

The following n lines, each with m characters, have only two valid character values:

`#` - (sharp) representing a block of **wall**;

`.` - (point) representing part of the **floor**.

Output

Your program must write either `YES` if the given labyrinth is a left labyrinth or `NO` otherwise

Sample input

```
30
50
..................................................
..................................................
..................................................
.....###############.............############.....
.....#.........#..#.............#.........#.....
.....#.#######.#.##.............#.########.#.....
.....#.#.....#....#.............#.#......#.#.....
.....#.#.#####.####.............#.#.####.#.#.....
.....#.#.....#...##################.##.#...#.....
.....#.#####.###.......................#.#.#.....
.....#.........#.################.#.##########.....
.....#########.#.#.#...........#.#..#............
.............#.#.#.#...........###.##............
.............#.#.#.#...........#....#............
.............#.#...#...........#.##.#............
.............#######...........#.##.#............
.............#.................#.#..#............
.............#.##################.####............
.....#########...#.................##########.....
.....#.........###.#.#.##########.#######.#.#.....
.....#.#########.....#.#..........#.....#.#.#.....
.....#.#.........#####.##########.#.###.#.#.#.....
.....#.######.#####...........#.#.#.#.#.#.#.#.....
.....#......#.....#...........#.....#.#.#.#.#.....
.....#.##########.#...........#####.#.#.#.#.#.....
.....#............#...........#...........#.#.....
.....###############...........###############.....
..................................................
..................................................
..................................................
```

Sample output

```
YES
```