Problem D
Icy Land

You are given a board of R rows and C columns where each cell is either a dry land ('#') or an icy land ('.'). You can move to either one of the four directions (north, south, east, or west) in the board.

You can walk or standstill on any dry land without any problem; however, an icy land is very slippery. If you stand at cell (r, c) and decide to move on a particular direction, then you will move on that direction continuously until you reach a dry land or the border of the board.

For example, consider the following board of 6 rows and 7 column.

```
#......
.#.....
..##...
..##...
.......
.#.....
```

- From $(1, 3)$, the 1st row and 3rd column, if you move to the south, then you will end up at $(3, 3)$.

- From $(4, 4)$, if you move to the north, then you will end up at $(3, 4)$.

- From $(1, 1)$, if you move to the east, then you will end up at $(1, 7)$.

- From $(6, 5)$, if you move to the west, then you will end up at $(6, 2)$.

Supposed your initial position is at cell $(3, 7)$ and your moves are west, west, west, south, east, north, west, and west, respectively. Then your positions are $(3, 7) \to (3, 4) \to (3, 3) \to (3, 1) \to (6, 1) \to (6, 2) \to (2, 2) \to (2, 1) \to (2, 1)$, thus, visiting 8 different cells. Note that cell $(2, 1)$ is visited twice. Cells in which you only passed through are also not considered as visited, e.g., in the example above, if you move south from $(3, 1)$, then you will pass through $(4, 1)$ and $(5, 1)$, and arrive at $(6, 1)$; thus, $(4, 1)$ and $(5, 1)$ are not considered as visited from that particular move, only $(3, 1)$ and $(6, 1)$ are.

You are allowed to change any icy land into a dry land, and your goal is to make sure that you can always visit **all** the cells in the board **regardless** of your initial starting position; in other words, you do not know your starting position yet, but given the board, you want to make sure that you can achieve your goal. What is the minimum number of cells you need to change to ensure that your goal can be achieved?

Input

Input begins with a line containing two integers: $R\ C$ $(1 \le R, C \le 500)$ representing the number of rows and columns of the board, respectively. The next R lines, each contains C characters representing the given board. Each character is either '#' which represents a dry land or '.' which represents an icy land.

Output

Output contains an integer in a line representing the minimum number of cells you need to change to ensure that you can visit all the cells in the board regardless of your initial starting position.

Sample Input

```
4 4
....
.###
##..
###.
```

Sample Output

```
1
```

Explanation for the sample input/output

We only need to change cell $(3, 3)$.

```
....
.###
###.
###.
```

Let 'N' be north, 'S' be south, 'W' be west, and 'E' be east.

- If you start at cell $(1, 1)$, then the movement "SSENNWENSESSEWNENWNE" will visit all the cells. The corresponding positions are: $(1, 1) \to (3, 1) \to (4, 1) \to (4, 2) \to (3, 2) \to (2, 2) \to (2, 1) \to (2, 2) \to (1, 2) \to (2, 2) \to (2, 3) \to (3, 3) \to (4, 3) \to (4, 4) \to (4, 3) \to (3, 3) \to (3, 4) \to (2, 4) \to (2, 3) \to (1, 3) \to (1, 4)$.

- If you start at cell $(1, 2)$, then the movement "W" followed by the movement in the first example above ("SSENNWENSESSEWNENWNE") will visit all the cells. The corresponding positions are: $(1, 2) \to (1, 1) \to \ldots$

- If you start at cell $(1, 3)$, then the movement "W" followed by the movement in the first example above will visit all the cells. The corresponding positions are: $(1, 3) \to (1, 1) \to \ldots$

- $\ldots$

- If you start at cell $(4, 3)$, then the movement "NNNW" followed by the movement in the first example above will visit all the cells. The corresponding positions are: $(4, 3) \to (3, 3) \to (2, 3) \to (1, 3) \to (1, 1) \to \ldots$

- If you start at cell $(4, 4)$, then the movement "NNW" followed by the movement in the first example above will visit all the cells. The corresponding positions are: $(4, 4) \to (2, 4) \to (1, 4) \to (1, 1) \to \ldots$