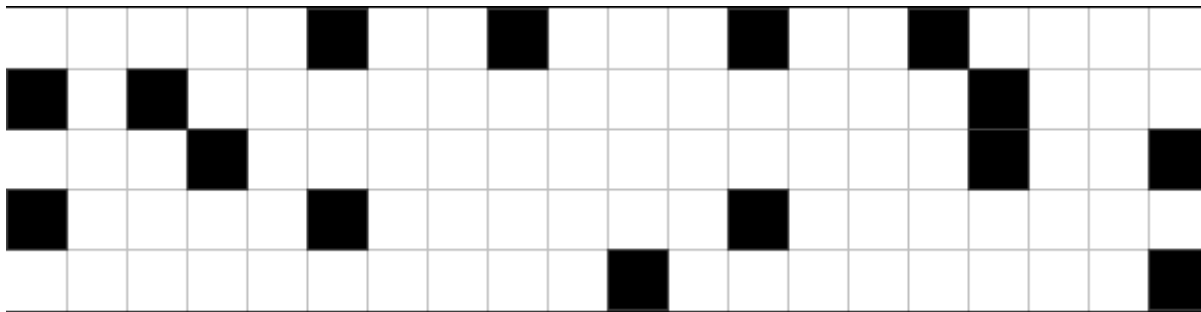


DMPG '16 S6 - Black and White II

Time limit: 1.0s **Memory limit:** 64M

Ruby and her sister Yang are playing with a mechanized board game.

The board consists of $M \times N$ square cells of unit dimensions on a plane, with the origin defined as the topmost left tile. Some of these cells are originally colored black, signifying an impassable block.



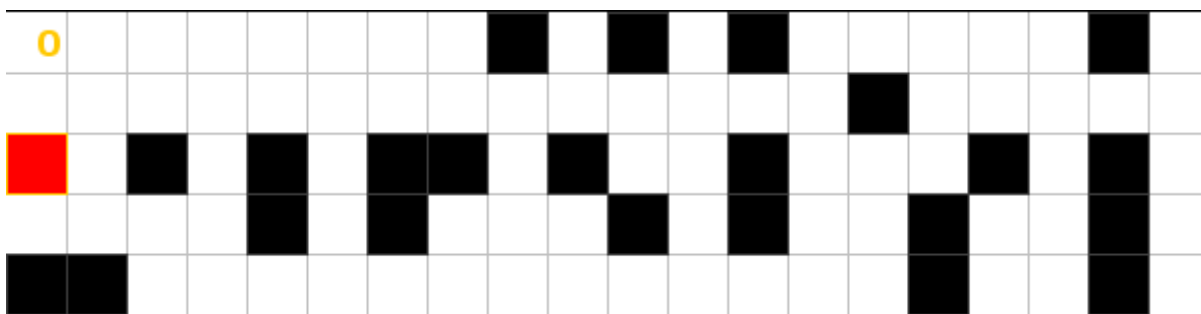
A game board.

Every second, the board moves every block up 1 unit, and wraps them around to $N - 1$. That is, a tile located at $(0, 1)$ at $t = 0$ will be at $(0, 0)$ when $t = 1$, and $(0, N - 1)$ when $t = 2$.

A player may start the game by placing a red piece on any clear tile where $x = 0$. A player may move their piece to a tile X at time t if and only if X is clear (i.e., does not contain a wall) at t . Likewise, the starting position must be clear at $t = 0$.

Every second, the player may move their red piece by **at most** 1 unit every second in cardinal directions, with the player's piece **wrapping around the board** like wall blocks do. A player may keep their piece stationary at time t so long as their current position is not covered by a wall at $t + 1$.

The goal of the player is to have their block reach $x = M - 1$.



Please visit <https://www.dmoj.ca/problem/dmpg16s6> for an animation of a game.

Ruby is playing Yang. Since they recently argued over something petty, Ruby wants to get back at Yang by making the game impossible for Yang to finish. To this end, she will be placing — while Yang is not looking — a number of new wall blocks to cut off Yang's movement. How many blocks does she need, if she wants to use as few as possible to make the game unwinnable?

Input Specification

The first line of input will contain two space-separated integers M and N ($1 \leq M, N \leq 100$). The next N lines will each contain M characters, with `.` representing a blank space and `#` representing a block at $t = 0$.

Output Specification

A single integer; the number of blocks Ruby needs to place.

Sample Input 1

```
5 2
..#..
.....
```

Sample Output 1

```
1
```

Explanation for Sample Output 1

Ruby may place a block at $(2, 1)$ to block all of Yang's possible routes.

Sample Input 2

```
5 1
.....
```

Sample Output 2

```
1
```

Explanation for Sample Output 2

Placing any block will make the game unwinnable.

Sample Input 3

```
3 3
.#.
.##
..#
```

Sample Output 3

```
1
```

Sample Input 4

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

Sample Output 4

```
1
```

Explanation for Sample Output 4

A block must be placed at (1, 0) to prevent Yang from ever being able to finish.