

Ели беше отвлечена! За щастие, тя намери плана на имението, в което бива държана като заложник. За нещастие, тя не знае къде точно в него се намира, тъй като беше със завързани очи, докато я вкарваха вътре.

За простота ще считаме имението като матрица с N реда и M колони. Всяка клетка може или да бъде празна (коридор) или блокирана (стена). В началото Ели се намира в някоя от празните клетки, но не знаем в коя от тях точно.

Ели може да се движи в четирите основни посоки – наляво, надясно, нагоре и надолу. Придвижването в съседна празна клетка не отнема време (все пак, трябва просто да се направят няколко крачки). Придвижването в блокирана клетка, обаче, отнема малко повече време – тя първо трябва да разбие стената там. Това отнема по един час на клетка. Считаме, че момичето е избягало от имението, когато достигне до неговия край – тоест до която и да е от крайните клетки на матрицата.

Ели вярва, че е затворена в някоя от празните клетки, от която ще ѝ отнеме най-много възможно време за да избяга. Тя ви е дала карта на имението под формата на матрица с N реда и M колони, всяка клетка от която е или '.' (коридор) или '#' (стена). Помогнете ѝ, като намерите броя на клетките, в които има възможност да се намира тя.

Input Format

На първия ред на стандартния вход ще съдържа две цели числа N и M – съответно броя редове и броя колони на картата. Следват N реда, всеки съдържащ по M символа от азбуката {'.', '#'}.

Constraints

$$1 \leq N, M \leq 1000$$

Output Format

На стандартния изход изведете едно цяло число – броя клетки, от които би се излязло с най-много време дори при избор на оптималния път.

Sample Input 0

```
7 14
.#.....
#####
.#.#.....
.##.##.....
.#.#.###.
.#...###.###
..#####
```

Sample Output 0

```
1
```

Sample Input 1

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

Sample Output 1

```
7
```