

Zadanie 6 (20 punktów)

Dana jest dwuwymiarowa tablica znakowa $M[R][C]$ zawierająca znaki '.' lub '#'. Znak '.' oznacza pusty element, znak '#' oznacza ścianę. Zakładamy, że labirynt jest otoczony ze wszystkich stron przez dodatkowe ściany, nie uwzględnione na wejściu programu.

Startujemy z elementu $M[0][0]$. Celem jest dotarcie do elementu $M[R - 1][C - 1]$. W każdym kroku możemy przemieścić o jedną pozycję w prawo, w lewo, w górę lub w dół, ale **nie wolno** wchodzić na ściany.

W zestawach podanych na wejściu programu dojście do celu jest niemożliwe. Jednak w **dokładnie jednym kroku** możliwe jest usunięcie jednej ściany przylegającej do bieżącej pozycji.

Proszę napisać program, który znajduje wszystkie takie ściany, że po usunięciu jednej z nich będzie możliwe dojście do celu $M[R - 1][C - 1]$. Program powinien zwrócić liczbę takich ścian. Program zwraca zero, jeżeli po usunięciu dowolnej ściany nie jest możliwe dotarcie do celu.

Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby całkowite R i C zawierająca odpowiednio liczbę wierszy i kolumn tablicy M . W kolejnych R wierszach znajduje się po dokładnie C znaków, stanowiących kolejne wiersze tablicy.

Ograniczenia

- $2 \leq R \leq 100$
- $2 \leq C \leq 100$
- Każdy znak w tablicy będzie znakiem '.' lub '#'
- $M[0][0]$ i $M[R - 1][C - 1]$ będzie zawsze '.'
- Nie jest możliwe przejście z $M[0][0]$ do $M[R - 1][C - 1]$

Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia program powinien wypisać jedną liczbę całkowitą będącą rozwiązaniem.

Przykłady

1. Dla danych wejściowych:

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

poprawną odpowiedzią jest:

3

Usunięcie dowolnej z trzech ścian umożliwia dojście do celu.

2. Dla danych wejściowych:

```
4 6
..##..
..##..
...#...
..##..
```

poprawną odpowiedzią jest:

1