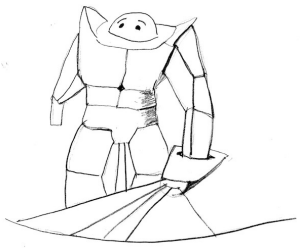
# Portals



There is a cake placed inside of a labyrinth and you desperately want to eat it. You have a map of the labyrinth, which is a grid of $R$ rows and $C$ columns. Each grid cell contains one of the following characters:

`#` (number sign) which denotes a wall,

`.` (dot) which denotes an open square,

`S` (uppercase letter s) which denotes an open square of your current position,

`C` (uppercase letter c) which denotes an open square with the cake.

You may only walk on the open squares and move from one open square to another if they share a side. Additionally, the rectangular area depicted on the map is surrounded by walls from the outside with no open squares.

In order to reach the cake faster you have acquired a portal gun from Aperture Science™, which operates as follows. At any time it can fire a portal in one of the four directions *up*, *left*, *down* and *right*. When a portal is fired in some direction, it will fly in that direction until it reaches the first wall. When this happens, a portal will be spawned on the side of the wall that faces you.

At most two portals can exist at any given time. If two portals are already placed in the labyrinth, then one of them (selected by you) will be removed immediately upon using the portal gun again. Firing a portal at a wall where there is already a portal placed will replace that portal (there may be at most one portal per side of wall). Note that there may be multiple portals placed on different sides of same wall.

Once two portals are placed on the map you can use them to teleport yourself. When standing next to one of the portals, you can walk into it and end up at the open square next to the other portal. Doing this takes as much time as moving between two adjacent squares.

You may assume that firing portals does not take time and moving between two squares (or teleporting through portals) takes one unit of time.

## Task

Given the map of the labyrinth together with your starting location and the location of the cake, calculate the minimum possible time needed for you to reach the cake.

## Input

The first line of the input contains two integer numbers: the number of rows in the map $R$, and the number of columns $C$. The next $R$ lines describe the map. Each of these lines contain $C$ characters: `#`, `.`, `S` or `C` (whose meaning is described above).

It is guaranteed that characters `S` and `C` each appear exactly once in the map.

## Output

The output should contain a single integer number — the minimum time that is needed to reach the cake from the starting position.

You may assume that it is always possible to reach the cake from your starting location.

## Example

| Input | Output | Comments |
|---|---|---|
| 4 4<br>.#.C<br>.#.#<br>....<br>S... | 4 | One quickest sequence of moves is as follows: 1) move right, 2) move right, shoot one portal up, and one portal down, 3) move through the bottom portal — you will appear at the location ($row = 0, column = 2$), 4) move one square right and reach the cake. |

## Scoring

**Subtask 1 (? points):** $0 \leq R \leq 10, 0 \leq C \leq 10$.

**Subtask 2 (? points):** $0 \leq R \leq 50, 0 \leq C \leq 50$.

**Subtask 3 (? points):** $0 \leq R \leq 200, 0 \leq C \leq 200$. Every open square has at least one wall adjacent to it.

**Subtask 4 (? points):** $0 \leq R \leq 200, 0 \leq C \leq 200$.

**Subtask 5 (? points):** $0 \leq R \leq 1000, 0 \leq C \leq 1000$.

## Constraints

**Time limit:** ? s.

**Memory limit:** ? MB.