

# WC '18 Finals S2 - Top Billing

Time limit: 1.0s Memory limit: 16M

## Woburn Challenge 2018-19 Finals Round - Senior Division

Two particularly famous actors are starring in the cows' and monkeys' production: Tom Cows and Monkey Freeman. With both of them playing equally important roles, one question is at the forefront of everybody's mind: which of them will receive top billing in the credits and get to be the main face of the movie?

Tom isn't one to stand for playing second fiddle, so he's hatching a plan to discredit Monkey in the eyes of the director, Bo Vine. He's going to sneak into the main studio building in the middle of the night and rearrange its layout, in the hopes of causing Monkey to be late for the following morning's filming session. That'll show him!

The building can be represented as a grid with  $R$  rows and  $C$  columns ( $2 \leq C \leq R \leq 100$ ). Note that the grid has at least as many rows as it has columns. Tom will choose one starting cell (leading to the parking lot) and one different ending cell (leading to the movie set). These cells may be anywhere in the grid (not necessarily on its edges). For each remaining cell in the grid, he'll choose to either leave it vacant or obstruct it with a wall. He'll make sure that it's possible to reach the ending cell from the starting one through a sequence of horizontally/vertically adjacent vacant cells.

In the morning, Tom and Monkey will begin in the starting cell and will each attempt to make their way to the ending cell, by repeatedly moving from cell to adjacent cell (up, right, down, or left) at a rate of 1 minute per cell. Tom, having arranged the building, will follow a route allowing him to reach the ending cell as quickly as possible. Monkey, on the other hand, will follow the following procedure:

1. Make an ordered list of possible next cells to move to as follows: for each of the directions up/right/down/left (in that order), if the cell adjacent to his current one in that direction is within the grid and doesn't contain a wall, include it in the list. Note that this list will always contain between 1 and 4 cells, inclusive.
2. Move to the cell in the list which is closest to the ending cell by Manhattan distance\* (ignoring any walls). If multiple cells in the list have the same minimum Manhattan distance to the ending cell, choose the one which appears earliest in the list.
3. If he's arrived at the ending cell, stop. Otherwise, return to step 1.

**Note:** The Manhattan distance between a cell in (row  $r_1$ , column  $c_1$ ) and another cell in (row  $r_2$ , column  $c_2$ ) is defined as  $|r_1 - r_2| + |c_1 - c_2|$ .

For example, consider the following two building layouts (with **S** denoting the starting cell, **E** denoting the ending cell, and walls marked with **#**):



<pre> ...S .##. .##. E... </pre>	<p>Monkey would reach the ending cell after 6 minutes (following the sequence of moves: down, down, down, left, left, left).</p>
<pre> ...S .... .... E.#. </pre>	<p>Monkey would walk around forever (following the sequence of moves: down, down, down, up, down, up, down, up, ...).</p>

Help Tom come up with any building layout which would cause Monkey to successfully arrive at the ending cell from the starting one eventually (rather than walking around forever as in the second example above), but at least  $2C - 4$  minutes after Tom does. It's guaranteed that at least one such building layout exists for any valid pair of dimensions  $R$  and  $C$  (with  $C \leq R$ ).

## Subtasks

---

In test cases worth 43% of the points,  $R \leq 4$  and  $C \leq 4$ .

## Input Specification

---

The first and only line of input consists of two space-separated integers,  $R$  and  $C$ .

## Output Specification

---

Output  $R$  lines of  $C$  characters each, a grid representing the chosen building layout. Each character must be one of the following:

- `S`: starting cell (which must appear exactly once)
- `E`: ending cell (which must appear exactly once)
- `.`: vacant cell
- `#`: wall

## Sample Input 1

---

4 2

## Sample Output 1

```
ES
..
.#
#.
```

## Sample Explanation 1

For this grid, both Tom and Monkey would reach the ending cell in 1 minute, by moving to the left. This is a difference of 0, which is at least as large as the minimum required difference of  $2C - 4 = 0$ . Therefore, this output would be judged as *correct*.

## Sample Input 2

10 9

## Sample Output 2

```
.....
.....
.....
.....
.....
.....
.....
.....
.....#.....
.S#...#E.
.....
```

## Sample Explanation 2

For this grid, Monkey would reach the ending cell in 10 minutes (following the sequence of moves: up, right, right, down, right, right, up, right, right, down), while Tom would only require 8 minutes (following the sequence of moves: down, right, right, right, right, right, right, up). This is a difference of 2, which is smaller than the minimum required

difference of  $2C - 4 = 14$ . Therefore, this output would be judged as *incorrect*. Various other grids exist which would be judged as *correct*, however they are not disclosed here.