IEEEXTREME PROGRAMMING COMPETITION 14.0

# Restaurant Reopening

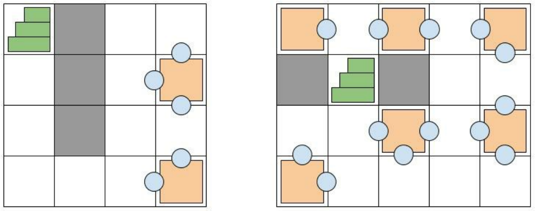
Time limit: *5000 ms*
Memory limit: *256 MB*

Many restaurants have been closed due to the global pandemic, including Cafe Xtreme. Now that the reopening is ongoing while indoor dining is still restricted, the manager of Cafe Xtreme thinks that it would be a good opportunity to redesign the restaurant floor layout to better use its space. Cafe Xtreme is big and it has multiple floors. Each floor of the restaurant can be viewed as an $R \times C$ grid with $R$ rows and $C$ columns. Two cells are adjacent if they share an edge. A cell may be empty, contain a wall, or contain the *stairs*. There is exactly one cell with the stairs per floor, which is the entrance and exit.

The manager will place dining tables in some empty cells of the floor. Each dining table occupies an entire empty cell. Dining chairs are placed in the adjacent empty cells of a dining table. A dining chair must be placed towards an edge of a dining table, and allows a customer to sit at the left, right, top, or bottom of the dining table. A dining table can therefore serve between one to four customers, depending on how many dining chairs are placed around it. A dining chair takes a negligible amount of space, so that multiple dining chairs that belong to different dining tables may be placed in a same empty cell. However, dining chairs cannot be placed in a cell that has a wall, a dining table, or the stairs.

A restaurant customer can walk through a cell as long as the cell does not contain a wall or a dining table. In particular, a customer may walk through a cell that contains his/her own dining chair or other customers' dining chairs (slipping through people's backs). All customers must be able to reach the stairs from their dining chairs so that they can enter and exit the restaurant.

The manager wants to know the maximum number of customers that each restaurant floor can accommodate.



## Standard input

The first line has a single integer $T$, the number of floors to consider. This is followed by the description of $T$ floors.

Each floor starts with two integers $R$ and $C$ on a single line.

The next $R$ lines each have a string of length $C$. Each character in the string can be a dot `.` as an empty cell, a hash `#` as a wall, or a letter `S` as the stairs.

## Standard output

For each floor, output the maximum number of customers the floor can accommodate on single line.

## Constraints and notes

- $1 \leq T \leq 10$
- $4 \leq R \cdot C \leq 100$
- Each floor map has exactly one letter `S`, and at least two empty cells.
- Before placing any dining tables and chairs, it is possible to walk to the stairs from every empty cell on a floor.

- For $37.5\%$ of the test files, $R \cdot C \leq 20$.

| Input | Output | Explanation |
|---|---|---|
| 3<br>4 4<br>S#..<br>.#..<br>.#..<br>....<br>4 5<br>.....<br>#S#..<br>.....<br>.....<br>2 3<br>#.#<br>.S. | 5<br>13<br>0 | The image illustrates an optimal layout for the first two floors in the sample test case. Each dining chair is shown as a blue dot, and each dining table is shown as an orange square. The walls are marked grey. It can be seen that every blue dot can move to the stairs via a sequence of white cells, which are either completely empty or contain only dining chairs.<br>In the first floor, no customer can be seated in the first or the second column from the left, otherwise the path to the stairs will be blocked. |