# ECOO '16 R2 P3 - BattleShip

Time limit: 2.5s   **Memory limit:** 64M

In a popular mobile app, two friends can play a game of battleship against one another. In this version of battleship, each player has a grid of squares in which they can place their ships vertically or horizontally. Each ship takes up a contiguous segment of grid squares equal to its length. For example on the left board below, the player has placed two ships of length 2 and one each of lengths 3 and 4.

The rules of the app state that you are not allowed to place ships adjacent to one another. This means that two squares occupied by different ships must not share a side or a corner.
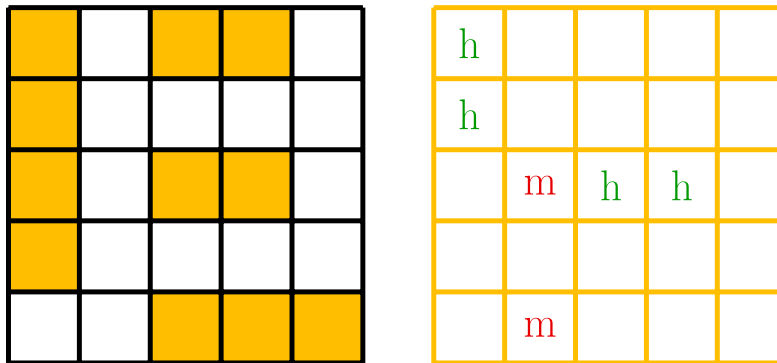
The players can't see their opponent's arrangement of ships. They take turns firing torpedoes blindly into their opponent's grid. Each torpedo hits one grid square. If there is a ship covering that square, it's a `hit`. If not, it's a `miss`. The game is over when one player has hit every grid square covered by an opponent's ship.

The board on the right above shows the other player's view of the board on the left after 6 shots have been fired. Note that all this player knows is where she has scored hits `h` and misses `m`.

In this version of the game, your opponent does not tell you when you have finished hitting a ship. In the example above, one of the length 2 ships is finished but the player firing shots won't know she's finished with it until she fires a torpedo into the grid square to the right of it and misses.

The input will contain 10 test cases. The first line of each test case will contain two integers S and L, where S is the size of the board $(1 \leq S \leq 100)$ and L is the length of a boat $(1 \leq L \leq S)$. The next S lines will contain S characters each, with lowercase characters representing the hits (`h`) and misses (`m`) to the opponent's ships. All other grid squares will be filled with a `.` character (`ASCII` 46). You must output a single line containing an integer, representing the number of possible (and known) locations for a ship of length L given the hits and misses so far.

Note that the sample input below contains only 1 test case, but the actual data files will contain 10.

Sample Input

```
5 3
.....
.hm..
.....
.....
.....
```

Sample Output

```
14
```