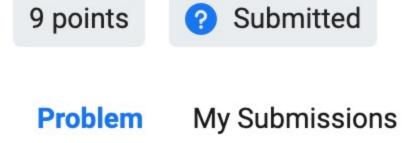


# **Problem B1: Second Friend**

Validate Solution & Submit

**Timer Expired** 



Note: The only difference between this problem and problem B2 is that here, scenes are smaller and do not contain rocks.

Boss Rob painted a beautiful scene on a 2D canvas of R rows by C columns, containing zero or more happy little trees.

To make sure none of his trees are lonely, Rob would like you to add as many trees as you'd like (possibly 0) to empty spaces so that each tree in the final painting has at least two tree *friends*, that is, two trees which are each adjacent to it (directly to its north, south, east, or west). If there are multiple solutions, you may print any one of them.

#### Constraints

```
1 \le T \le 85
1 \le R, C \le \mathbf{100}
```

## **Input Format**

Input begins with an integer T, the number of test cases. For each case, there is first a line containing two space-separated integers, R and C. Then, R lines follow, each of which contains C characters, either "." (an empty space) or "^" (a tree), representing the initial painting.

#### **Output Format**

For the ith case, print "Case #i: ", followed by "Possible" and R lines of C characters each representing the final painting (if a solution exists), otherwise "Impossible".

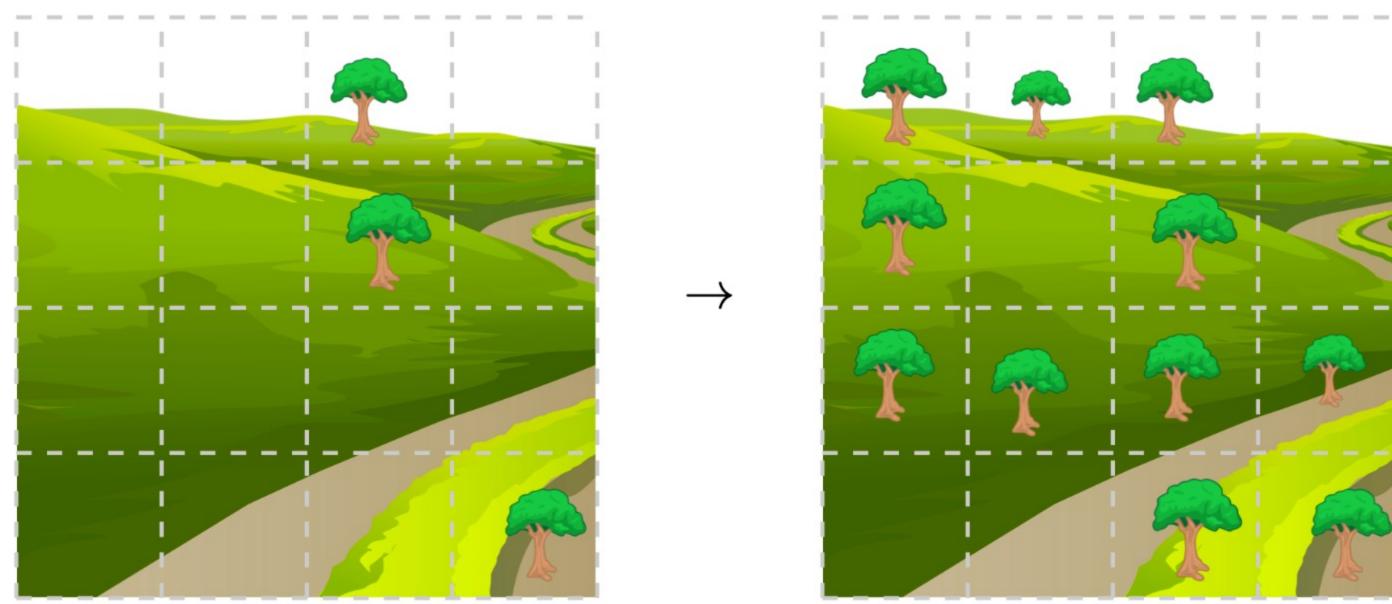
## **Sample Explanation**

In the first case (depicted below), we could add two tree friends to either side of the middle tree, but they themselves would each only have one tree friend. Therefore, it's impossible to get two friends for each tree in the final painting.



In the second case, there are no trees in the initial painting, so the condition of two friends for each tree is already satisfied.

In the third case, one possible solution is depicted below.





Validate Solution & Submit

**Timer Expired**