



Sajjad Home

[Instructions](#)[Scoreboard](#)[My score](#)[Friends](#)[Everyone](#)[My Clarifications](#)

Facebook Hacker Cup 2018 Round 1

[Request Clarification](#)

Ethan Traverses a Tree

20 points

[Download Input](#)

Ethan is doing his second programming assignment: implementing pre-order tree traversal.

Ethan has a binary tree with N nodes (numbered 1 to N), rooted at node 1. Each node i 's left child is node A_i (with $A_i = 0$ indicating no left child), and similarly its right child is B_i (with $B_i = 0$ indicating no right child). Each node i is also assigned an integral label L_i .

Given such a tree, Ethan must compute its [pre-order traversal](#) (expressed as a sequence of node labels). The pre-order traversal of a tree involves taking its root node, then concatenating the pre-order traversal of the root's left sub-tree (if any), and then concatenating the pre-order traversal of the root's right sub-tree (if any).

Ethan has attempted to solve this problem, but unfortunately he got his computer science terms mixed up, and now his algorithm finds the tree's [post-order traversal](#) instead! The post-order traversal of a tree involves taking the post-order traversal of the root's left sub-tree (if any), and then concatenating the post-order traversal of the root's right sub-tree (if any), and finally concatenating the root node at the end.

Since you were mean to Ethan on his first assignment, you'd like to cheer him up by making his algorithm work out after all. Though the tree's shape must stay as is, you can choose a set of labels $L_1..N$ for its nodes such that Ethan's algorithm will still produce the correct answer — in other words, such that the sequence of node labels in the tree's pre-order traversal is equal to the sequence of node labels in its post-order traversal. Your only two restrictions are that each node label must be between 1 and K (inclusive), and that every integer between 1 and K (inclusive) must be used as the label of at least one node. You'd like to find any way of validly labelling the nodes, or determine that no way exists.

Input

Input begins with an integer T , the number of trees. For each tree, there is first a line containing the space-separated integers N and K . Then, N lines follow. The i th of these lines contains the space-separated integers A_i and B_i .

Output

For the i th tree, print a line containing "Case # i : " followed by your chosen node labels $L_1..N$ separated by spaces, or "Impossible" if there's no valid way to label the nodes.

Constraints

$1 \leq T \leq 80$
 $1 \leq K \leq N \leq 2,000$
 $0 \leq A_i \leq B_i \leq N$

Every tree is guaranteed to be a valid binary tree rooted at node 1.

Explanation of Sample

In the first case, if $L = [1, 1]$, then both the pre-order and post-order label sequences will be $[1, 1]$.

In the second case, for each label between 1 and K to be present, you must choose either $L = [1, 2]$ or $L = [2, 1]$, both of which would result in the pre-order and post-order label sequences differing. For example, if $L = [1, 2]$, then the pre-order sequence will be $[1, 2]$ while the post-order sequence will be $[2, 1]$.

In the third case, if $L = [2, 2, 1]$, then the pre-order and post-order label sequences will both be $[2, 1, 2]$.

Note that other outputs for example cases 3 to 5 would also be accepted.

Sample input · [Download](#)

```
6
2 1
0 2
0 0
2 2
0 2
0 0
3 2
0 3
0 0
2 0
9 4
2 3
4 0
0 5
6 0
0 7
8 0
0 9
0 0
```

Sample output · [Download](#)

```
Case #1: 1 1
Case #2: Impossible
Case #3: 2 2 1
Case #4: 2 4 3 1 3 4 3 2 2
Case #5: 1 1 2 3 4 1 2 3 1 3 1 3 2 3 1
Case #6: Impossible
```

[Chat \(22\)](#)

0	0
15	4
8	0
0	0
0	9
0	0
15	6
0	0
4	0
2	13
0	0
14	12
5	0
3	0
10	11
0	0
0	7
15	3
8	11
0	0
0	9
0	0
15	6
0	0
4	0
2	13
0	0
14	12
5	0
3	0
10	0
0	0
0	7



This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](#) .

[About](#) [Create Ad](#) [Create Page](#) [Developers](#) [Careers](#) [Privacy](#) [Cookies](#) [Ad Choices](#) [Terms](#) [Help](#)

Facebook © 2018
English (US) [فارسی](#) [Español](#) [Français \(France\)](#) [中文\(简体\)](#) [العربية](#) [Português \(Brasil\)](#) [Italiano](#) [한국어](#) [Deutsch](#) [हिन्दी](#)

Chat (22)