Disjoint Paths

One of your classes has K students in it who really don't like each other. In fact, they dislike each other so much that they want to find routes to class that don't cross at any intersection so that they won't ever see each other outside of class. Can you find such routes?

Input

The input file will contain multiple cases. The first line of each case is K N, where K is the number of routes you need to find and N is the number of intersections in MIT's floor plan. The intersections are numbered 1 ... N. This is followed by N lines, one for each intersection, containing the indices of the adjacent intersections, separated by spaces. (This means that if the line for intersection 2 contains a 3, then the line for intersection 3 will contain a 2.) Every intersection is adjacent to at least one other intersection. Each case is followed immediately by the next case. The end of the input is indicated by a line containing only "0 0". You may assume that 1 ≤ K ≤ 100 and 2 ≤ N ≤ 5000. The students all start at intersection 1 and their class is at intersection 2.

Output

For each case, output the case number, in the format "Case #:", followed by a newline. If there are K non-intersecting routes from the start (1) to the end (2), then this must be followed K lines, each one giving a list of corners, in order, on one such route from 1 to 2. If not, then output one line with the word "Impossible". Output a blank line after each test case.

Example

Input:
```
3 5
3 4 5
3 4 5
1 2
1 2
1 2
4 5
3 4 5
3 4 5
1 2
1 2
1 2
0 0
```

Output:
```
Case 1:
1 3 2
1 4 2
1 5 2

Case 2:
Impossible
```