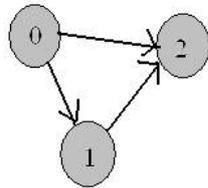


11779 Lost File

The prestigious programming contest NCPK is going to be held tomorrow at SUST. The problem setting coordinator found suddenly that, input file of one of the problem is lost. The situation is, it is difficult to reach the original problem setter to collect the input file once again. It is even more difficult to generate new input file. You all know input generation is one of the toughest job in problem setting, as you have to ensure that all corner cases have been covered, all wrong solution will get wrong answer, all inefficient solution must get time limit exceed or memory limit exceed. Only good news is the output file is still there, and coordinator thinks that it is possible to produce the original input file again using the output file. And yes, he is right. Can you help the coordinator to produce the input file?

For your information, here is the problem:

“You are given such a directed graph where there is no path from u to v if there is a path from v to u , and has no self loop. You have to calculate the number of paths exists between each pair. Note that, maximum one edge can exist between any two nodes.



Consider the image:

From 0 to 1 there is only 1 path, which is $0 \rightarrow 1$. From 1 and 2 there is 1 path, $1 \rightarrow 2$. From 0 to 2 there are two paths, $0 \rightarrow 2$ and $0 \rightarrow 1 \rightarrow 2$. There is no path from 2 to 0, from 2 to 1 and from 1 to 0.”

You are given the output file of this problem. Your job is to generate the input file from this output file.

Input

Your input file starts with an integer T ($T \leq 1000$), which denotes the number of test cases. Each test case starts with two integers N ($1 \leq N \leq 50$) and K , here N is the number of nodes in the given graph and K is the number of pair of nodes between which there exists at least one path. Each of the next K lines will contain three integers u, v and w ($0 \leq u, v < N$), representing there are w paths from node u to node v .

Output

Desired input file will start with ‘Case X : N E ’, here X is the case number, N is the number of nodes, and E is the total number of edges in the graph, which will be followed by N lines. First of these N lines will start with an integer e , number of nodes to which there is an edge from node 0 and next e integers will represent those nodes in ascending order. Next line will give the information about node 1 in the same way, and so on. See the sample output and input for exact format. Just a little friendly reminder, there should be no leading, trailing space or extra space between numbers.

Sample Input

```
1
3 3
0 2 2
0 1 1
1 2 1
```

Sample Output

Case 1: 3 3

2 1 2

1 2

0