

1390 – Weight Comparison

Rimi has n different objects having distinct weights and identified by x_1, x_2, \dots, x_n . All she wanted to do is to sort them in ascending order according to their weights. That's why she asked her robot-helper to do all the weight comparisons. She went for another task, and the robot continued doing the comparisons. This robot was not that intelligent, that's why it was picking two objects randomly, compared them and wrote down the object names as $x_i x_j$ meaning that x_i is heavier than x_j . After a while, Rimi came back and found this foolishness of the robot. So, she wants to find the comparisons that are really necessary. For example, the robot compared and found the following:

1. $x_1 > x_2$
2. $x_2 > x_5$
3. $x_1 > x_5$

Clearly the third one is unnecessary, as from the first two relations, it's clear that x_1 is heavier than x_5 . But the second one is necessary since from first and third relation, it's impossible to determine the relation between x_2 and x_5 . The first relation is also necessary. Now Rimi wants to keep minimum number of necessary relations such that they are enough to find all the comparisons.

Input

Input starts with an integer T (≤ 10), denoting the number of test cases.

Each case starts with a blank line. Next line contains two integers n, m ($1 \leq n \leq 5000, 0 \leq m \leq 10^5$), where m denotes the number of comparisons done by the robot. Each of the next m lines contains two integers $i j$ ($1 \leq i, j \leq n, i \neq j$) meaning that x_i is heavier than x_j . There is no duplicate transition. And assume that the data is valid.

Output

For each case, print the case number and the number of necessary comparisons. Then print the comparisons in $i j$ (x_i is heavier than x_j) form, first sorted by i then by j in ascending order.

Sample Input	Output for Sample Input
2 4 4 1 2 3 4 2 3 1 4 4 1 4 2	Case 1: 3 1 2 2 3 3 4 Case 2: 1 4 2

Note

Dataset is huge; use faster I/O methods.