

1230 – Placing Lampposts

As a part of the mission 'Beautification of Dhaka City', the government has decided to replace all the old lampposts with new expensive ones. Since the new ones are quite expensive and the budget is not up to the requirement, the government has decided to buy the minimum number of lampposts required to light the whole city.

Dhaka city can be modeled as an undirected graph with no cycles, multi-edges or loops. There are several roads and junctions. A lamppost can only be placed on junctions. These lampposts can emit light in all the directions and that means a lamppost that is placed in a junction will light all the roads leading away from it.

The 'Dhaka City Corporation' has given you the road map of Dhaka city. You are hired to find the minimum number of lampposts that will be required to light the whole city. These lampposts can then be placed on the required junctions to provide the service. There could be many combinations of placing these lampposts that will still cover all the roads. In that case, you have to place them in such a way so that the number of roads receiving light from two lampposts is maximized. (A careful thought will reveal that all the roads will get light either from one post or two posts).

Input

Input starts with an integer **T** (≤ 30), denoting the number of test cases.

Each case starts with a blank line. The next line contains two integers **N** ($1 \leq N \leq 1000$) and **M** ($0 \leq M < N$) which indicate the number of junctions and roads respectively. The junctions are numbered from **0** to **N-1**. Each of the next **M** lines contains two integers **a** and **b** ($0 \leq a, b < N$, $a \neq b$), which denotes that there is a road from junction **a** to **b**.

Output

For each case, print the case number, the minimum number of lampposts required to light the whole city, the number of roads that are receiving lights from two lampposts and the number of roads that are receiving light from only one lamppost.

Sample Input	Output for Sample Input
2 4 3 0 1 1 2 2 3 5 4 0 1 0 2 0 3 0 4	Case 1: 2 1 2 Case 2: 1 0 4