1279 – Graph Coloring

Given an undirected graph **G**, you want to color each vertex of the graph using **K** colors. The colors are numbered from **0** to **K-1**. But there is one restriction. Let **v** be any vertex of the graph and $\mathbf{u_1}$, $\mathbf{u_2}$ **...** $\mathbf{u_m}$ be the adjacent vertices of **v**, then

color(v) = color(u_1) + color(u_2) + ... + color(u_m) (modulo K)

Now you have to find the number of ways you can color the graph maintaining this restriction.

Input

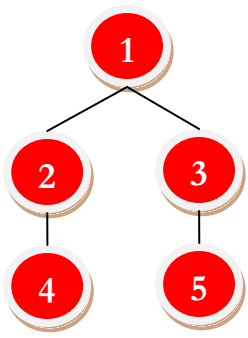
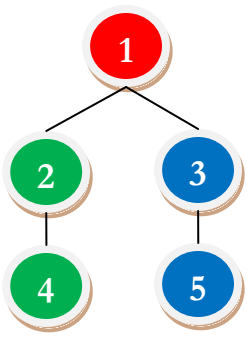
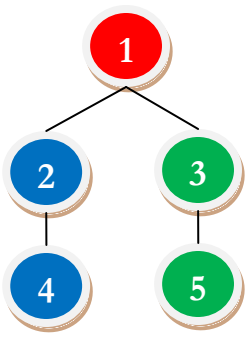
Input starts with an integer **T (≤ 200)**, denoting the number of test cases.

Each case starts with a line containing three integers **N**, **M** and **K (1 ≤ N ≤ 100, 2 ≤ K ≤ 10^9 and K is a prime)** where **N** denotes the number of vertices and **M** denotes the number of edges of the graph. Each of the next **M** lines contains two integers **u v (1 ≤ u, v ≤ N, u ≠ v)** denoting that there is an edge between vertex **u** and **v**. You can safely assume that there is at most one edge between any two vertices.

Output

For each case, print the case number and the result modulo **1000000007**.

Sample Input	Output for Sample Input
2 6 0 5 5 4 3 1 2 1 3 2 4 3 5	Case 1: 1 Case 2: 3

Note

For the second case, let the colors are red (0), green (1) and blue (2). Then the possible results are: