CSE 4404-Algorithms Lab. Winter 2022
Date: January 31, 2023.
Target Group: A
Topic: Lab Test

Instructions:

- Task naming format: fullID_L02_T01_A.c/CPP
- **Solutions with less efficient approaches will be considered for partial marks.**

Task 1

The ex-students of an IUT class are organizing a reunion. Despite many having moved out of Bangladesh, only a few remain. They chose to host the reunion at a house instead of a restaurant for a more intimate gathering. The challenge was to determine the best house. Ultimately, they settled on the one with the lowest cost after calculating the total cost, which included the sum of distances from each participant's home to the selected location. You need to write a program that finds the house with the lowest overall cost, taking into account the distances from each member's house to the chosen location. The program should consider each member's house as a potential venue for the reunion.

Input

The first line of input consists of two integers, $N\ (\leq N\ \leq 100)$ and $M\ (1 \leq M\ \leq N(N\ -\ 1)/2)$, where N represents the total number of houses and M represents the total number of paths between the houses. There might not be a path between every pair of houses. The next M lines each consist of three integers, $a, b\ and\ c,$ which means that there is a path from house a to b with distance c. Also you can consider the edges as bidirectional.

Output

You should output the house number that has the least minimum cost, if there are multiple houses with the same minimum cost, you should output all of them.

Sample Input	**Sample Output**
Testcase #1: 4 3 1 2 10 1 3 8 1 4 6 Testcase #2: 5 6 1 2 10 1 3 8 1 4 6 4 5 8 3 5 6 2 5 10	Output #1: 1 Output #2: 1 5

Task 2

In 1976, a computer helped prove the "Four Color Map Theorem", which states that any map can be colored with only four colors so that no neighboring regions share the same color. You are now tasked with solving a similar but simpler problem: determining if a given connected graph can be colored with only two colors (from a palette of two) such that no adjacent nodes have the same color. The following assumptions are made to simplify the problem:

- No node will have an edge to itself.
- The graph is non-directed, meaning if node A is connected to node B, then node B is also connected to node A.
- The graph is strongly connected, meaning there is at least one path from any node to any other node.

Input

The input starts with a line containing two integers $n\ (1 \leq n \leq 100)$ and m, the total number of nodes and edges. After this, m lines will follow, each containing two numbers $a\ and\ b$, specifying an edge between those two nodes.

Output

You have to decide whether the input graph can be bicolored or not, and print it as shown below.

Sample Input	**Sample Output**
Testcase #1: 3 3 0 1 1 2 2 0 Testcase #2: 3 2 0 1 1 2 Testcase #3: 9 8 0 1 0 2	Output #1: NOT BICOLORABLE. Output #2: BICOLORABLE. Output #3: BICOLORABLE.

0 3 0 4 0 5 0 6 0 7 0 8	