

# Friends Suggestion

---

Problem Code	hw05c_friends
Running Time Limit	1 sec
Memory Limit	16 mb

## Objective

- Be able to find neighboring vertices in a graph.

## Introduction

There exist several social network systems, for example, facebook, MySpace, hi5, etc. The central mechanism for these services is that each user can socialize with their friend. It is encouraged to reach out for your friends in these systems. To boost such activities, many systems provide tools for users to locate their friends. Automated friends suggestion is one of the popular tools. In this problem, we will consider a simplified version of friend suggestion system. The mechanism of a friend suggestion system is very simple, it assumes that if user A befriends with user B and user B befriends with user C, it is likely that A and C should befriend as well. This concept is carried through all the chain of friends. For example, if C befriends with D and D with E and so on to user Z, the system also suggests Z (and everyone else up to C) to A. However, the system does not suggest B to A since they have become friends already.

## Graph Modeling

This problem can be modeled as a graph. A user is represented by a node in the graph and there is an edge between two users if they are friends. A list of friend suggestion for a particular user is a list of nodes reachable by that user but has no edge connecting to that user.

## Task

You are to write a program that reads a list of users and their relations and also an additional list of selected users in the system. For each selected user, the program must display the list of users that should be suggested to the selected user.

## Input

The first line contains two integers  $N$  and  $M$  ( $1 < N \leq 255$ ,  $1 \leq M \leq N(N - 1)/2$ ), the number of users in the system and the number of friendship relations between pairs of users. Users are numbered from 0 to  $N - 1$ . This is followed by  $M$  lines where each line contains two numbers  $A$  and  $B$ , indicating that the user  $A$  and the user  $B$  are friends. After that, there is a line containing a number  $K$  which is followed by  $K$  lines, each containing a user  $C$  to whom the program should report a friend suggestion list.

## Output

The output must have  $K$  lines; each line gives the list of friend suggested for each user corresponding to the last  $K$  lines in the input. In each line, the list of suggested friends is sorted according to their numbering. If the suggested list contains no one, print the word "NO ONE" instead.

## Example

### Ex1

Input	Output
6 <-- number of users (nodes)	2 5
4 <-- number of relations (edges)	NO ONE
0 1	
2 5	
0 2	
3 4	
2 <-- number of friend suggestions query	
1 <-- this asks a suggestion for user 2	
3 <-- this asks a suggestion for user 4	