

After much hard work Ivancho finally made enough money to start his own software company! Now he is trying to connect his office in such a way that all his computers have internet. Unfortunately, the cable system he bought is very bad. It works in a peculiar way – each computer is connected to each other computer with a cable. Information flows only one way through a cable. Ivancho wants one of the computers to be a “hub” i.e. such that all of its connections are outgoing. His office is already set up, but unfortunately it lacks a hub. Now Ivancho has been tasked with reconfiguring the connections, in order to make a hub. To do this he can flip the connections of the various computers. When he flips the connections of a computer all incoming connections become outgoing and vice versa. Help him by writing a program, which, after a series of flips, tells him whether which computer is the hub (or that a hub doesn’t exist).

Note: We will assign each computer a unique index in the range [0, N-1]

Input

The first row of the file `hub.in` contains two positive integers **N** and **K** - the number of computers in the network and the number of times he will be flipping connections.

N*(N-1)/2 lines follow. Each line contains two positive integers **u** and **v** which denote a one way connection from **u** to **v**.

K series of flips follow. Each series is given in the following way:

A single integer **P** – the number of computers, which are going to be flipped

P unique positive integers follow – the indexes of the computers to be flipped.

Output

In the output file `hub.out` print the index of the hub after each series of flips. If there is no hub - print „-1“.

Constraints

$$3 \leq N \leq 850$$

$$0 \leq K \leq 600$$

$$1 \leq P \leq N-1$$

Time limit: 1.2 sec

Memory limit: 256 MB

Example test

Input (hub.in)	Output (hub.out)
4 2 3 1 2 3 0 2 2 1 0 3 1 0 2 2 1 2 2 0	-1 1
5 3 4 1 3 1 3 0 4 3 4 0 1 2 2 0 2 3 2 4 0 1 3 2 0 3 4 1 3 2 0 3 1 2 4	-1 2 3

Clarifications (example 1)

After the first series of flips all connections of the computers 2 and 1 are flipped. There is no hub in the network so we print -1. After the second series of flips all connections of the computers 2 and 0 are flipped. The new network contains a hub – the computer with index 1.