**Singapore Valley**

Little Bug has just arrived in Singapore Valley, the dreamland, as a software engineer for the government. In Singapore Valley, companies compete not by growing themselves but by buying their competitors. Since transactions happen every day, Little Bug is assigned a task to develop a program that manages all transactions.

Formally, there are companies and software engineers in Singapore Valley. These companies will be numbered from to . All engineers are also numbered from 0 to . Initially, each engineer is employed by exactly one company. There can be companies that employs no engineers.

In total, Little Bug’s program needs to handle transactions. In each transaction, company will take over company **.** After the transaction, all engineers of company will become employees of company **.** Also, companywill be closed down and therefore will not be part of any future transaction.

After all transactions, Little Bug needs to report the list of all companies that are still operating (i.e: not closed down). Also, for each such company, she needs to list down all engineers currently employed by it.

## Input

The first line contains two integers - number of engineers and companies in the beginning.

Next, lines follow, the **-th** line (where **)** contains a list of engineers employed by company initially. It begins with an integer **,** the number of engineers, follows by integers – the identity of the engineers being employed by the company.

The next line is a single integer  **()** – the number of transactions.

The next lines, each will consists of two integers indicating a transaction that the company takes over the company. It’s guaranteed that both companies and companies is still operating before the transaction.

## Output

The first line contains a single integer , number of companies that that are still operating

lines follow, each contains information about one operating company. Each line begins with two integers **,**  the company identity and the number of engineers that are currently employed by it, follows by integers – the identity of the engineers being employed by the company, **in ascending order**.

You should also output the companies in ascending order of their numerical identities.

## Examples

|  |  |
| --- | --- |
| **Input (companies1.in)** | **Output (companies1.out)** |
| 6 4  3 0 1 2  1 3  1 4  1 5  2  2 3  0 2 | 2  0 5 0 1 2 4 5  1 1 3 |

|  |  |
| --- | --- |
| **Input (companies2.in)** | **Output (companies2.out)** |
| 5 2  5 0 1 2 3 4  0  0 | 2  0 5 0 1 2 3 4  1 0 |

## Explanation for example 1

|  |  |
| --- | --- |
| **Event** | Companies’ Status |
| Initially | Company 0 employs engineers 0, 1, 2  Company 1 employs engineers 3  Company 2 employs engineers 4  Company 3 employs engineers 5 |
| After company 2 buys company 3 | Company 0 employs engineers 0, 1, 2  Company 1 employs engineers 3  Company 2 employs engineers 4, 5  Company 3 no longer exists |
| After company 0 buys company 2 | Company 0 employs engineers 0, 1, 2, 4, 5  Company 1 employs engineers 3  Company 2 no longer exists  Company 3 no longer exists |

## Important Note:

In this problem, a solution using built-in Linked-List Java API is not the intended solution (as the setters doesn’t think there exist any solution based on built-in Linked-List Java API). As a result, it is advised that you implement a customized Linked-List for the problem (which is the intended solution).

## Note:

1. A skeleton file has been given to help you. You should not create a new file or rename the file provided. You should develop your program using this skeleton file.
2. You are free to define your own helper methods and classes (or remove existing ones) if it is suitable but you must put all the new classes, if any, in the same skeleton file provided

## Skeleton File

You are given the skeleton file Companies.java. You should see the following contents when you open the file:

|  |
| --- |
| /\*\*  \* Name :  \* Matric. No :  \*/  import java.util.\*;  public class Companies {  private void run() {  }  public static void main(String args[]) {  Companies companies = new Companies();  companies.run();  }  } |