Problem #8: Who's the Boss?

Program Name: boss.java Input File: boss.dat

As a successful programmer for MegaCorp, you are constantly bombarded by requests from myriads of different people. Until recently, you spent most of your time trying to figure out whether or not these requests merited attention. Luckily, MegaCorp just published a new organization chart showing the management structure for every employee, and all you have to do is write a program that will determine if someone is your boss (or your boss's boss, or boss's boss's boss, etc.).

The basic goal for the program is this: Given an organization chart, determine if someone has to obey an order from another.

Input

The input is divided into two sections. The first section is the organization chart for MegaCorp. The second section is a list of name pairs.

MegaCorp's organization chart is given as a list of teams, and the first line of input indicates the number of teams. This is followed by one line for each team. A team line consists of the total number of minions, the boss name, and the minion names (for example, "`3 boss minionA minionB minionC`").

Note, the organization chart only lists first names, and each name uniquely identifies a person. A person will appear as a minion in any team at most once, and there will be no circular relationships (i.e., a boss will never report directly or indirectly to one of his minions).

The list of name pairs begins with a single line indicating the number of name pairs. This is followed by the name pairs, each on its own line. In each pair, the first person wants to know if they have to follow instructions given by the second person. In other words, they want to know if the second person is their boss, their boss's boss, etc.

A name pair will always contain names from the organization chart and will never contain the same name twice.

Output

For each name pair (<first> <second>) in the input, determine if the first person must obey the second and print either:

<first>: Sure <second>, I'll get right on it.

or

<first>: No <second>, I don't have time to do your work and mine.

Example Input File

```
4
2 BigBoss John Fred
3 John Wilson Stoker Bubba
4 Wilson James Marc Tim Alan
2 Fred Wilma Dino
3
Alan Dino
Alan BigBoss
Wilson James
```

Example Output To Screen

```
Alan: No Dino, I don't have time to do your work and mine.
Alan: Sure BigBoss, I'll get right on it.
Wilson: No James, I don't have time to do your work and mine.
```