1 Railroads

Background

It's Friday evening and Jill hates two things which are common to all trains:

1. They are always late.
2. The schedule is always wrong.

Nevertheless, tomorrow in the early morning hours Jill will have to travel from Hamburg to Darmstadt in order to get to the regional programming contest. Since she is afraid of arriving too late and being excluded from the contest she is looking for the train which gets her to Darmstadt as early as possible. However, she dislikes to get to the station too early, so if there are several schedules with the same arrival time then she will choose the one with the latest departure time.

Problem

Jill asks you to help her with her problem. You are given a set of railroad schedules from which you must compute the train with the earliest arrival time and the fastest connection from one location to another. One good thing: Jill is very experienced in changing trains. She can do this instantaneously, i.e., in zero time!!!

Input

The very first line of the input gives the number of scenarios. Each scenario consists of three parts.

Part one lists the names of all cities connected by the railroads. It starts with a number $1 < C \leq 100$, followed by C lines containing city names. These names consist of letters.

Part two describes all the trains running during a day. It starts with a number $T \leq 1000$ followed by T train descriptions. Each of them consists of one line with a number $t_i \leq 100$ and t_i more lines with a time and a city name, meaning that passengers can get on or off the train at that time at that city.

Part three consists of three lines: Line one contains the earliest journey's starting time, line two the name of the city where she starts, and line three the destination city. The two cities are always different.

Output

For each scenario print a line containing "`Scenario i`", where i is the number of the scenario starting at 1.

If a connection exists then print the two lines containing zero padded timestamps and locations as shown in the sample. Use blanks to achieve the indentation. If no connection exists on the same day (i.e., arrival before midnight) then print a line containing "`No connection`".

After each scenario print a blank line.

(Sample input and sample output on next page)

Sample Input

```
2
3
Hamburg
Frankfurt
Darmstadt
3
2
0949 Hamburg
1006 Frankfurt
2
1325 Hamburg
1550 Darmstadt
2
1205 Frankfurt
1411 Darmstadt
0800
Hamburg
Darmstadt
2
Paris
Tokyo
1
2
0100 Paris
2300 Tokyo
0800
Paris
Tokyo
```

Sample Output

```
Scenario 1
Departure 0949 Hamburg
Arrival   1411 Darmstadt

Scenario 2
No connection
```