

Xiao Ming, a clever and lazy boy, is good at programming and sleeping. He hated traveling by train very much. Nevertheless, tomorrow in the early morning hours he will travel from Liuzhou to Xi'an to in order to get to the National Olympiad of Informatics in 2001 (NOI2001). Since he is afraid of arriving too late and being excluded from the contest, he is looking for the train which gets her to Xi'an as early as possible. However he dislikes getting to the station too early, so if there are several schedules with the same arrival time, he will choose the one with the latest departure time.

Xiao Ming asks you to help him with his problem, so that she can sleep a bit longer tomorrow. You are given a set of railroad schedules from which you have to compute the fastest connection among those with the earliest arrival time for going from one location to another. One good thing: Xiao Ming can switch trains in zero time.

Input

The input file contains several scenarios. Each of them consists of 3 parts:

Part one lists the names of all cities connected by the railroads. It starts with a line containing an integer C ($1 \leq C \leq 100$) followed by C lines containing city names. These names consist of at most 20 letters.

Part two describes all the trains running **during the day**. It start with a number $T \leq 100$ followed by T train descriptions. Each train description consists of one line with a number $t_i < 100$ 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. The times are given in the 24-hour format 'hhmm'.

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

The end of the input file is marked by a line containing only a zero (instead of C). Do not process this line.

Output

For each scenario print one line solution. If a connection exists the print 2 four-digit numbers, meaning 'hhmm', the starting time and the arriving time. If not connection exists print a line containing 'No connection'.

Sample Input

```
3
Liuzhou
Guilin
Xian
3
2
0900 Liuzhou
1200 Guilin
2
1200 Guilin
2200 Xian
3
0900 Liuzhou
1200 Guilin
2300 Xian
0800
Liuzhou
Xian
3
Liuzhou
Guilin
Xian
1
3
0900 Liuzhou
1200 Guilin
2300 Xian
1000
Liuzhou
Xian
0
```

Sample Output

```
0900 2200
No connection
```