

### 10.5.4 Railroads

**PC/UVa IDs:** 111004/10039, **Popularity:** C, **Success rate:** average **Level:** 3

Tomorrow morning Jill must travel from Hamburg to Darmstadt to compete in the regional programming contest. Since she is afraid of arriving 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 getting 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.

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. Fortunately, Jill is very experienced in changing trains and 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. The first part 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. All names consist only of letters.

The second part describes all the trains running during a day. It starts with a number  $T \leq 1,000$  followed by  $T$  train descriptions. Each of them consists of one line with a number  $t_i \leq 100$  and then  $t_i$  more lines, each with a time and a city name, meaning that passengers can get on or off the train at that time at that city.

The final part consists of three lines: the first containing the earliest possible starting time, the second the name of the city where she starts, and the third with the destination city. The start and destination cities are always different.

#### *Output*

For each scenario print a line containing “**Scenario  $i$** ”, where  $i$  is the scenario number starting from 1.

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

Print a blank line after each scenario.

#### *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

```