Trains are cool. You can get pretty much anywhere in Europe on a train, including little towns. In Canada, train service is not quite as good. Sometimes you have to change trains many times, and it's often hard to know the quickest route and where to switch. Depending on the time of day, the route may even pass through completely different cities. Fortunately, you can write a program to help you. Given the schedules of trains running between different places, your task is to find all the shortest connections between two places.

A shortest connection is one for which there is no other connection that would allow you to leave later and arrive at the same time or earlier, or leave at the same time and arrive earlier. Time to change trains need not be considered, as it is already built into the schedule.

Input

The first line of input contains a positive integer N, the number of test cases. The first line of each test case contains $T \leq 20$, the number of train routes. Each train route is described by one or more lines containing:

- $S \leq 20$ The number of stations on the route including origin and terminus.
- $hh{:}mm$ The starting time of the route (in 24-hour clock notation; that is, in the range 00:00 to 23:59). You may assume that a train leaves the origin every day at this time. A station name is a string of no more than 40 alphabetic characters.
- A list of the names of the S stations, separated by the travel time between adjacent stations. Travel time is given in hours and minutes.

Finally, each test case gives the name of an origin and destination for which a schedule is to be generated. You may assume there is at least one route from the specified origin to the specified destination.

Output

The output consists of a list of shortest connection departure and travel times from the origin to the destination, ordered by departure time. Each combination of departure and travel time should be listed only once, even if there are multiple routings yielding the same connection times. The departure time should be given as $hh{:}mm$ (that is, exactly 2 digits each). Travel time on the other hand should be given as $h{:}mm$, $hh{:}mm$, $hhh{:}mm$, etc., as appropriate to avoid unnecessary leading 0's. Leave an empty line between the output for successive test cases.

For example, the first route in the sample input begins in Windsor at 08:00 a.m., arrives in London at 09:55, in Kitchener at 11:30, Guelph at 12:25, Toronto at 13:30 and Montreal at 18:20. To get from Waterloo to Toronto we can leave at 07:00 and travel direct for a total time of 1:45. Or we can leave at 08:00 and travel to Kitchener, then to Guelph and Toronto, arriving at 13:30 for a travel time of 5:30. Or we can depart Waterloo at 09:00 for Hamilton, Niagara, and Toronto, arriving at 14:00. Finally, we can depart Waterloo at 23:00 and arrive in Toronto at 07:05 the next morning, for a travel time of 8:05.

Sample Input

```
1
7
6 08:00 Windsor 1:55 London 1:35 Kitchener 0:55 Guelph 1:05 Toronto 4:50 Montreal
2 08:00 Waterloo 0:45 Kitchener
3 09:00 Waterloo 1:45 Hamilton 1:05 Niagara
2 12:00 Niagara 2:00 Toronto
2 07:00 Waterloo 1:45 Toronto
2 23:00 Waterloo 0:55 Guelph
2 06:00 Guelph 1:05 Toronto
Waterloo Toronto
```

Sample Output

```
07:00 1:45
08:00 5:30
09:00 5:00
23:00 8:05
```