When pop stars make their international tours they usually prefer to spend as little time as possible for travel and to save more time for rehearsal, shows and for their private life. Therefore they travel only by airplanes and constantly search for the fastest route to destination. However, nowadays there are so many possibilities for traveling that finding the fastest way for a long distant trip is not an easy task. That is why a program capable to solve this problem would be prized on the market and you as an independent software developer are strongly encouraged to write such a program.

Since pop stars travel only by airplanes your task is greatly simplified. You need to take into account only international airports and flights connecting them.

There are some facts about a journey by airplanes that you should know:

- international airports are located in different time zones,
- each airport has flight schedule which defines destination, departure time and travel time for each flight; this schedule works on daily basis,
- boarding the airplane (as well as changing from one airplane to another) takes time, which differs from one airport to another.

To formalize the problem, we make the following steps:

- all international airport names are coded as identifiers represented by sequences of no more than 20 alphanumeric characters or underline characters (i.e. 'a'...'z', 'A'...'Z', '0'...'9' or '_'); all identifiers are unique,
- all flight identifiers are combination of company code and flight number with a total length of no more than 5 alphanumeric characters; all such identifiers are unique,
- all identifiers are case sensitive,
- all data which represent time have a format of '*hh*:*mm*', where '*hh*' and '*mm*' are digits from '0' to '9' representing hours and minutes respectively; if not specified otherwise the local time is used.

Using these assumptions, you should write a program to find the fastest route from the airport of origin to the destination one.

Input

The input will contain several test cases, each of them as described below. Consecutive test cases are separated by a single blank line.

The first line of the input contains identifiers of the airports of origin and of destination and starting time of the journey separated by spaces. The starting time is the time when pop star arrives to the airport of origin.

The second line of the input contains single integer N. This integer represents the number of international airports and is not less than 2 and not greater than 100.

The rest of the input consists of descriptions of international airports. Each description starts with a headline and may contain some complementary lines.

The headline consists of the airport identifier, airport time zone, boarding time and integer M separated by spaces. Time zone is time difference between local time and Greenwich Mean Time and has a format of '*shh*:*mm*', where '*s*' is the sign of time difference and are either '+' or '-'. Boarding time is the time delay needed for boarding or airplane change in that particular airport. Integer M defines a number of flights in the airport schedule (not greater than 300), each flight is described on its separate line following the headline.

The description of the flight consists of the flight identifier, destination airport identifier, departure time and travel time, separated by spaces. The travel time is the time gap between departure and landing (arrival).

The task will always have a solution for the given data.

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

In the first line of the output file your program should print the total travel time which is counted from the moment when pop star arrives to the airport of origin till the moment of arrival to the destination airport using the format of '*d*:*hh*:*mm*', where '*d*' is the number of full days of travel (no trip can last more than 9 full days).

In the second line the program should print the local time of the arrival of the star to the destination airport. In the following lines program should print the list of flight identifiers of the best route — one flight identifier per line.

Sample input

```
Pulkovo JFK 11:15
3
Pulkovo +03:00 01:30 2
BA347 Heathrow 12:10 04:25
Z8805 Heathrow 18:25 04:30
Heathrow +00:00 00:45 3
BA160 JFK 09:20 08:10
BA346 Pulkovo 14:45 04:20
Z8804 Pulkovo 21:30 04:25
JFK -05:00 00:45 1
BA161 Heathrow 14:25 08:05
```

Sample Output

```
1:09:15
12:30
Z8805
BA160
```