Bài tập Order raised over times

Given a dataset about orders raised over the time, each item is under the format:

<OrderID> <TimePoint>

in which the order <OrderID> (<OrderID> is a string of length from 2 to 15) is raised at time point <TimePoint> (the <TimePoint> is a string of length 8 representing the time point hh:mm:ss, for example, 08:30:14 means 8 hour 30 minutes 14 seconds)

Perform a sequence of queries about the given dataset of 3 categories:

- ?number_orders: return the number of orders raised
- ?number_orders_in_period <FromTimePoint> <ToTimePoint>: return the number of orders raised in the period from time point <FromTimePoint> to time point <ToTimePoint>
- ?number_orders_at_time <TimePoint>: return the number of orders raised at the time point <TimePoint>

Input

- The first block is a sequence of lines (number of lines can be up to 100000), each line contains an information about an order raised with the format above. The first block is terminated with a line containsing the character #

- The second line is a sequence of lines (number of lines can be up to 100000), each line contains a query described above. The second line is terminated with a line containing ###

Output

- Write in each line, the result of the correspinding query.

Bài tập Order raised over times

- **Example**

Input	Output
ORD0001 18:48:34 ORD0002 15:53:51 ORD0003 08:07:12 ORD0004 04:06:44 ORD0005 05:11:40 ORD0006 00:18:17 ORD0007 05:09:07 ORD0008 18:48:34 ORD0009 02:31:11 ORD0010 18:48:34 # ?number_orders ?number_orders_in_period 03:00:00 06:30 :45 ?number_orders_at_time 18:48:34 ###	10 3 3

Bài tập Order raised over times

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.*;

class Item{
    String orderID;
    String timePoint;
    public Item(String orderID, String timePoint){
        this.orderID = orderID; this.timePoint = timePoint;
    }
}
public class Main {
    public static int hash(String t){
        // convert time-point t under the format hh:mm:ss to seconds
        String[] s = t.split(":");
        int h = Integer.parseInt(s[0]);
        int m = Integer.parseInt(s[1]);
        int ss = Integer.parseInt(s[2]);
        return h*3600 + m*60 + ss;
    }
```

Bài tập Order raised over times

```
public static void main(String[] args){
        try{
            BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
            String line;
            ArrayList<Item> L = new ArrayList<Item>();
            int M = 23*3600 + 59*60 + 59; // max hh:mm:ss
            int[] a = new int[M+1];
            int[] t = new int[M+1];
            for(int i = 0; i <= M; i++) a[i] = 0;
            while(true){
                line = in.readLine();
                if(line.equals("#")) break;
                String[] s = line.split(" ");
                L.add(new Item(s[0],s[1]));
                int idx = hash(s[1]);
                a[idx] += 1;
            }
            // compute prefix sum t
            t[0] = a[0];
            for(int i = 1; i <= M; i++) t[i] = t[i-1] + a[i];
```

Bài tập Order raised over times

```
            // perform queries
            while(true){
                line = in.readLine();
                if(line.equals("###")) break;
                String[] s = line.split(" ");
                if(s[0].equals("?number_orders")) System.out.println(L.size());
                else if(s[0].equals("?number_orders_in_period")){
                    // s[1] is the start time point, s[2] is the end time point
                    int i = hash(s[1]); int j = hash(s[2]);
                    int cnt = t[j];
                    if(i > 0) cnt = t[j] - t[i-1];
                    System.out.println(cnt);
                }else if(s[0].equals("?number_orders_at_time")){
                    // s[1] is the time point
                    int i = hash(s[1]);     int cnt = a[i];     System.out.println(cnt);
                }
            }
            in.close();
        }catch (Exception e){
            e.printStackTrace();
        }
    }
}
```