Qn Link : <https://www.desiqna.in/16189/backend-engineer-hackerrank-online-test>

Question Summary :

* You are given an array consists of name , action , start time and end time and an interger K.
* Your task is to find K minutes , where all the persons are free.

Observation :

* There is no need for person name & activity
* Given input is an time interval an we need to find the first time gap of K minutes

Step 1 : convert the given time into minutes array so 24 \* 60 🡪 1440

Step 2 : convert the given input into minutes and perform the range update

Step 3 : Compute the prefix sum

Step 4 : Whenever array values is zero make a counter and run till cnt value < k

Step 5 : If so , convert the current index into hrs and return

Step 6 : If soln not found then return -1.

class Solution {

    public String partyTime (String [] details) {

        int n = details.length;

        int [] rangeUpdate = new int[1440];

        for(int i = 0 ; i < n ; i++){

            //Splitting the space and now [ name , action , st , end]

            String [] str = details.split(" ");

            //taking the Start time  and split by using :

            String [] start = str[2].split(":");

            // taking the end time and aplit it by using :

            String [] end = str[3].split(":");

            // im convert it to base 10 , because 00 , 01 can be present which led to octal conversion

            //Extracting the hours and min from start time

            int Sthrs = Integer.parseInt(start[0] , 10);

            int Stmins = Integer.parseInt(start[1] , 10);

            //Extracting hours and mins from end time

            int Endhrs = Integer.parseInt(end[0] , 10);

            int Endmin = Integer.parseInt(end[1] , 10);

            // COnverting into mins

            int startPoint = Sthrs\*60 + Stmins;

            int endPoint = ENdhrs\*60 + Endmins;

            //Performing range Update opr

            if(endPoint + 1 < 1440){

                rangeUpdate[endPoint + 1] -= 1;

            }

            rangeUpdate[startPoint]+= 1;

        }

        //Calculating prefix sum

        int [] prefix  = new int[1440];

        prefix[0] = rangeUpdate[0];

        for(int i = 1 ; i < 1440 ; i++){

            prefix[i] = prefix[i - 1] + rangeUpdate[i];

        }

        //Finding the first occurence of length K

        for(int i = 0 ; i < 1440 ; i++){

            //If found increase the count and check if it is equal to K

            if(prefix[i] == 0){

                cnt++;

                //If so extract the time by i - k + 1 and convert it into hrs and min and retrun

                if(cnt == k){

                    int time = i - k + 1;

                    int hrs = time / 60;

                    int mins = time % 60;

                    return ""+hrs+":"+mins;

                }

            }

            //Else make the count as 0

            else{

                cnt = 0;

            }

        }

        return "-1";

    }

}