Qn Link : <https://www.desiqna.in/16114/visa-oa-sde-intern-ctc-30-lac-27th-oct>

Question Summary :

* You are given an 2d array where lamp[i][0] represents the starting point and ending point of light covers.
* You are also given an array points where we need to return an array of how many lamps it will cover.

Step 1 : range Query for the index L 🡪 R

Step 2 : Run the prefix Sum

Step 3 : Create an array of size n & return only the points

class Solution {

    public int[] lampPoint(int [][] lamps , int [] points) {

        int max = -1;

        int n = points.length;

        //Finding max to create an prefix and dummy array

        for(int i =0 ; i < n ; i++){

            max = Math.max(max , points[i]);

        }

        int [] rangeUpdate = new int [max + 1];

        int [] prefix = new int [max + 1];

        //Performing Range Sum

        for(int i = 0 ; i < lamps.length ; i++){

            int l = lamps[i][0];

            int r = lamps[i][1];

            if(r + 1 < n)

                rangeUpdate[r + 1] -= 1;

            rangeUpdate[l] += 1;

        }

       //Calculating prefix Sum

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

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

        }

        // Creating array to return

        int [] ans = new int[points.length];

        //Store the asked points

        for(int i = 0 ; i < points.length ; i++){

            ans[i] = prefix[points[i]];

        }

        return ans;

    }

}