**Experiment No. – 8**

**Student Name: SUMIT KUMAR UID: 20BCS8226**

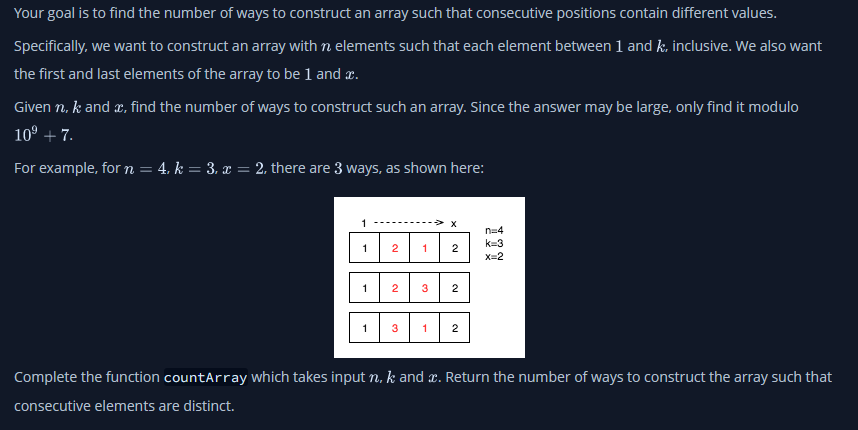
**Branch: BE-CSE Section/Group: MM 808 A**

**Semester: 5th Date of Submission: 12-11-2022**

**Subject Name: Competitive coding - I Subject Code: 20CSP-314**

# **Construct the Array**

1. **Aim/Overview of the practical:**

****

**2. Task to be done/ Which logistics used:**



1. **Hardware and Software Requirements (For programming-based labs):**

* Laptop or Desktop
* Hacker-Rank Account

1. **Steps for experiment/practical/Code:**

public static long countArray(int n, int k, int x) {

        long dp[][] = new long[n][2];

        dp[0][0] = 1;

        dp[0][1] = 0;

        for (int i=1;i<n;i++) {

            dp[i][0] = (dp[i-1][1] \* (k-1)) % 1000000007;

            dp[i][1] = (dp[i-1][0] + dp[i-1][1] \* (k-2)) % 1000000007;

        }

        if (x == 1) {

            return dp[n-1][0];

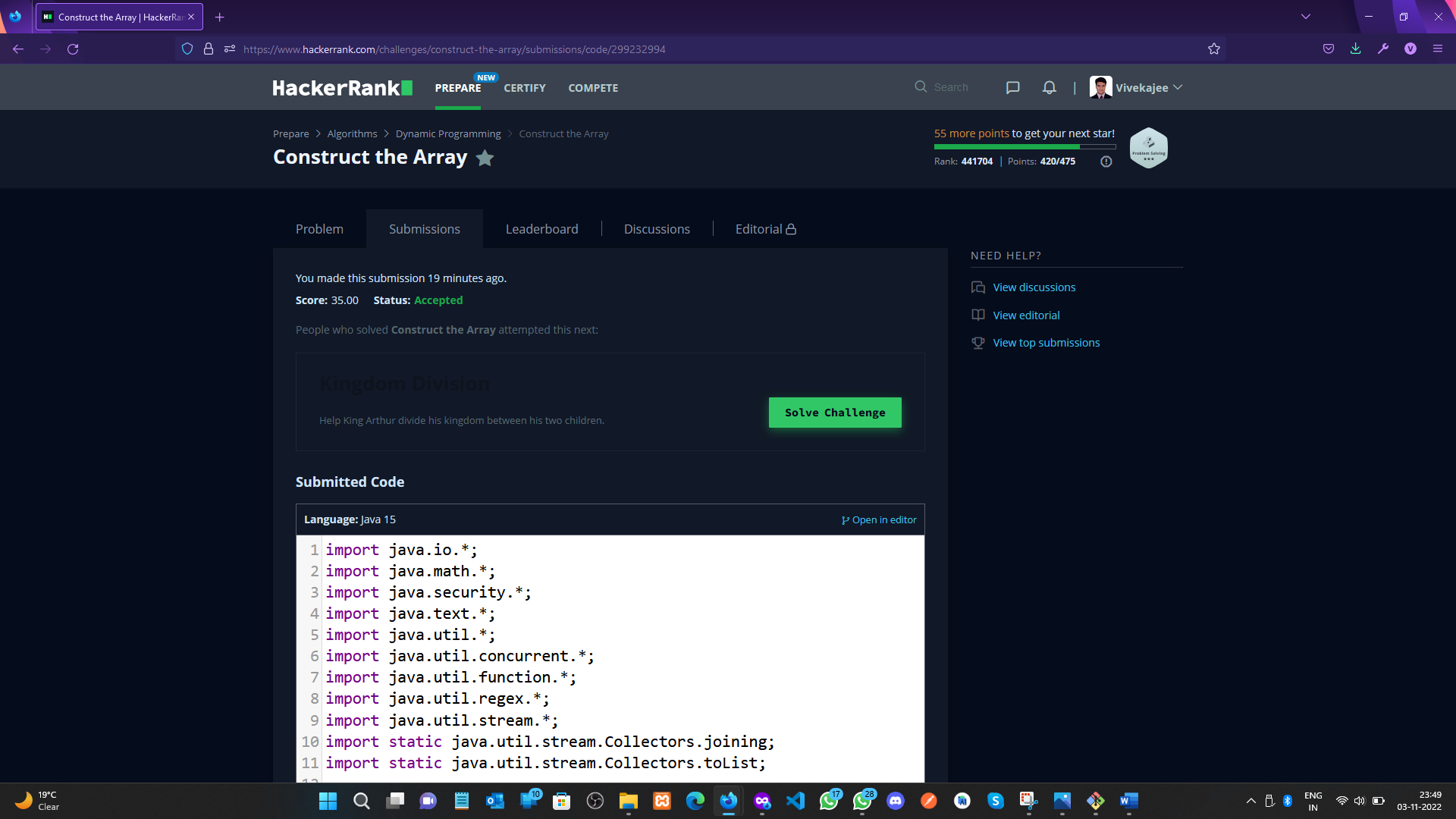
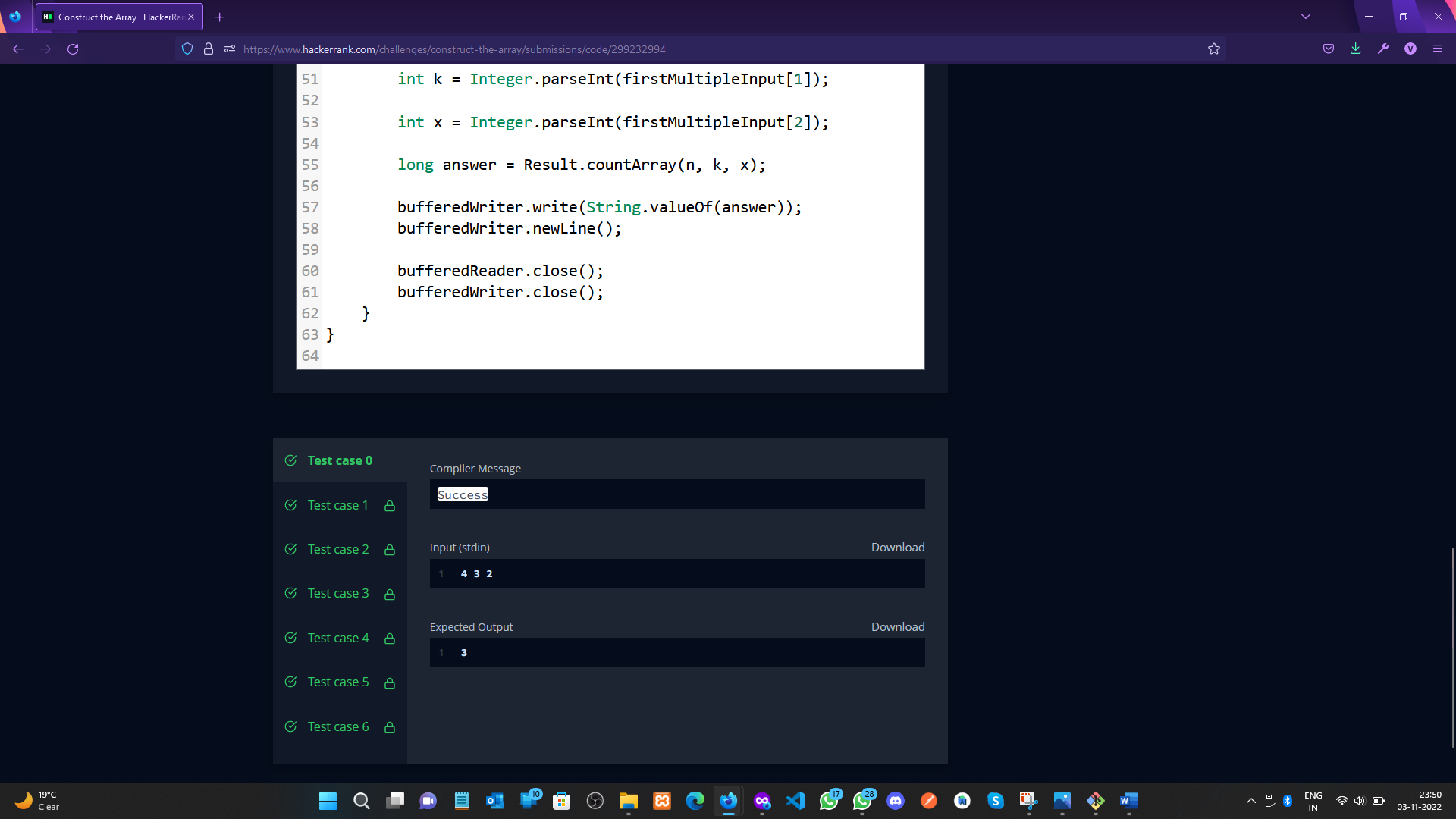
        } else {

            return dp[n-1][1];

        }

    }

1. **Result/Output/Writing Summary:**

**Equal**

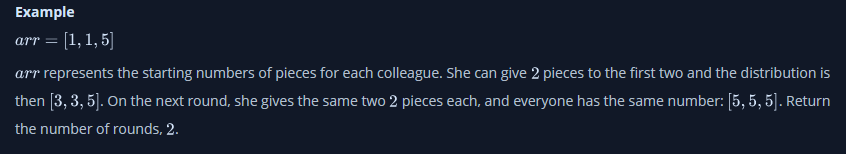
1. **Aim/Overview of the practical:**

Christy is interning at HackerRank. One day she has to distribute some chocolates to her colleagues. She is biased towards her friends and plans to give them more than the others. One of the program managers hears of this and tells her to make sure everyone gets the same number.

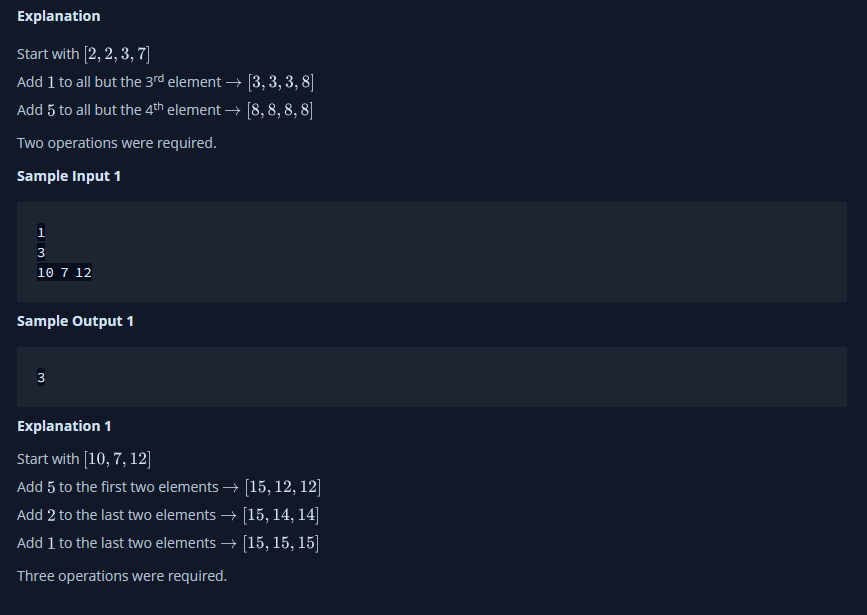
To make things difficult, she must equalize the number of chocolates in a series of operations. For each operation, she can give 1, 2 or 5 pieces to all but one colleague. Everyone who gets a piece in a round receives the same number of pieces.

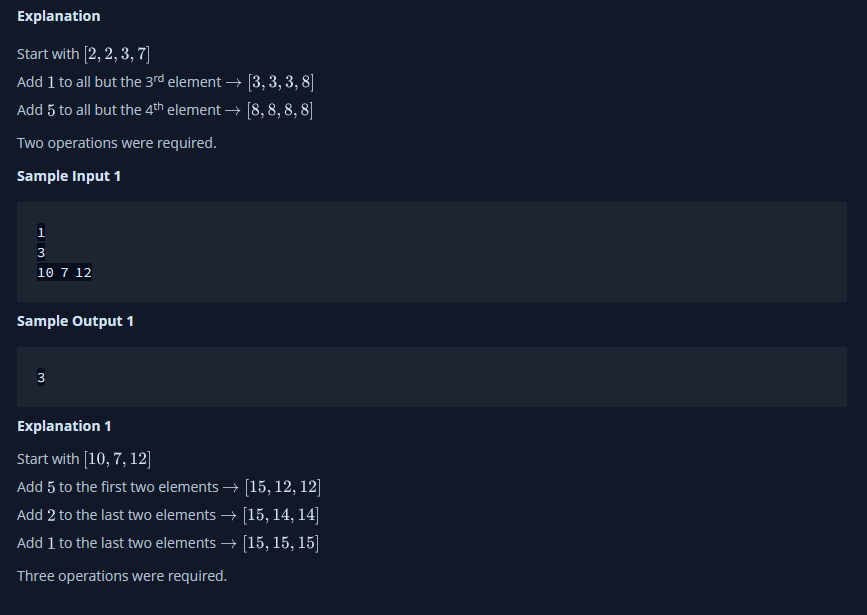
Given a starting distribution, calculate the minimum number of operations needed so that every colleague has the same number of pieces.

1. **Task to be done/ Which logistics used:**









1. **Hardware and Software Requirements (For programming-based labs):**

* Laptop or Desktop
* Hacker-Rank Account

1. **Steps for experiment/practical/Code:**

import java.util.Scanner;

public class Solution {

    public static int Ch(int x){

        int r = x/5;

        x%=5;

        r+=x/2;

        x%=2;

        return r+x;

    }

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        Scanner cin = new Scanner(System.in);

        int t = cin.nextInt();

        while(t-- != 0){

            int n = cin.nextInt();

            int [] N = new int[n];

            int x = 10000;

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

                N[i] = cin.nextInt()+5;

                x = Math.min(x, N[i]);

            }

            int r = 10000000;

            int s = 0;

            //System.out.println(x);

            for(int i = x-5 ; i < x+1; i++){

                s = 0;

                for(int j = 0 ; j < n ; j++)

                    s+=Ch(N[j]-i);

                r = Math.min(r,s);

            }

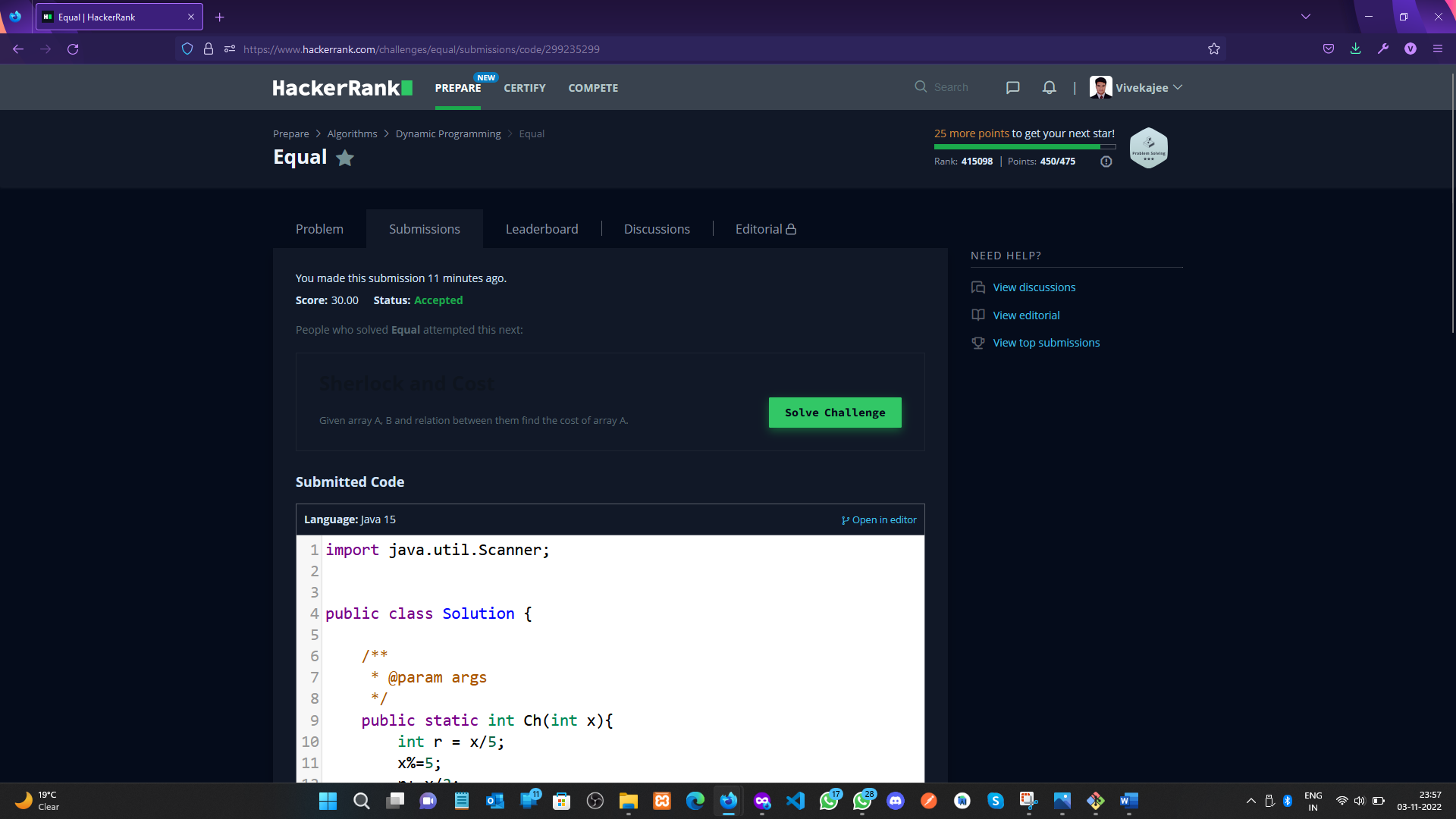
            System.out.println(r);

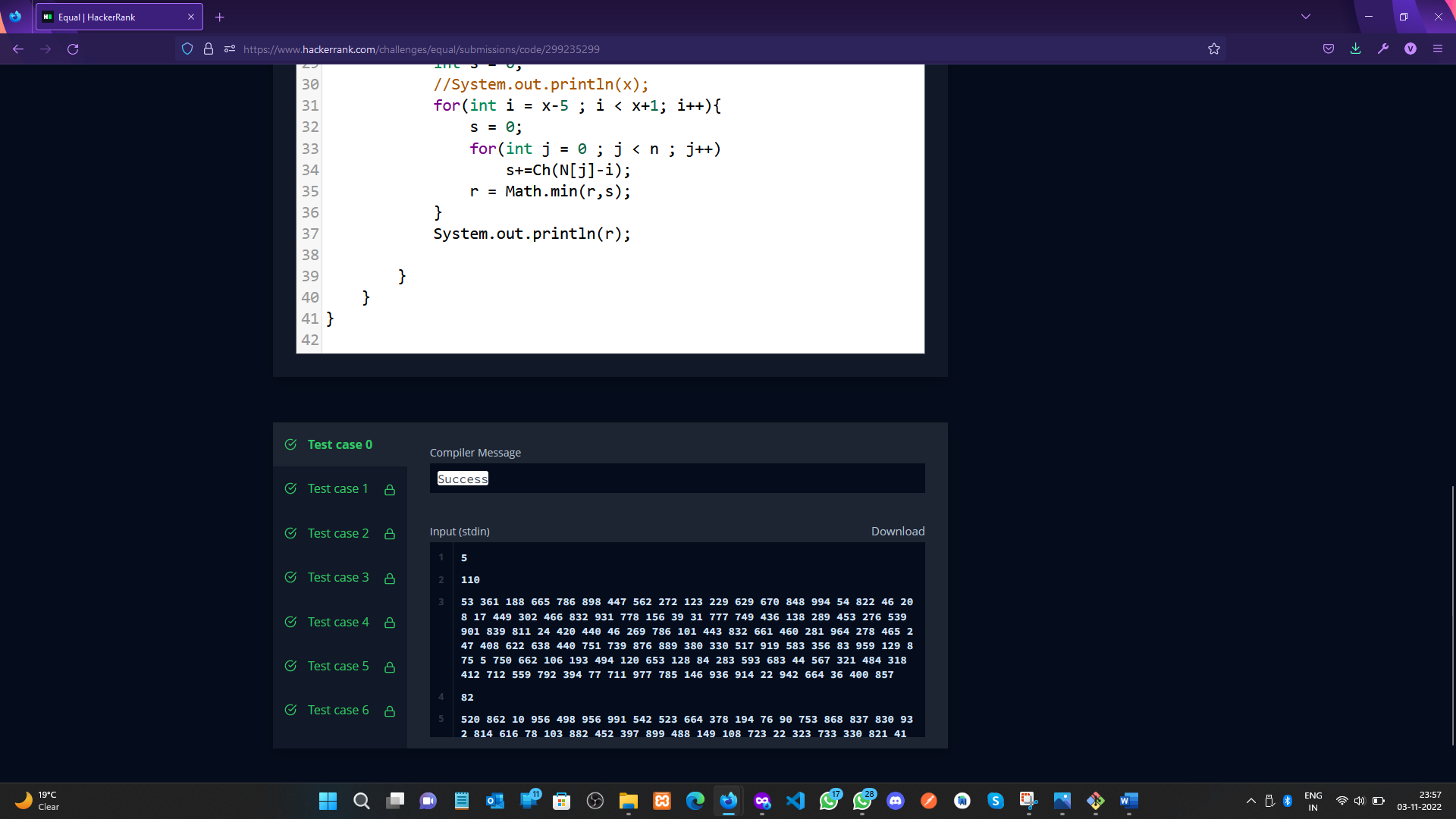
        }

    }

}

**5. Result/Output/Writing Summary:**





**Learning outcomes (What I have learnt):**

a. Learnt the concepts of Dynamic programming.

1. b. Learnt about Array in Dynamic Programming.
2. c. Learn about the countArray and Equal concept.

**Evaluation Grid (To be created per the faculty's SOP and Assessment guidelines):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. | Worksheet completion including writing learning objectives/Outcomes.  (To be submitted at the end of the day). |  |  |
| 2. | Post-Lab Quiz Result. |  |  |
| 3. | Student Engagement in  Simulation/Demonstration/Performance and Controls/Pre-Lab Questions. |  |  |
|  | Signature of Faculty (with Date): | Total Marks Obtained: |  |