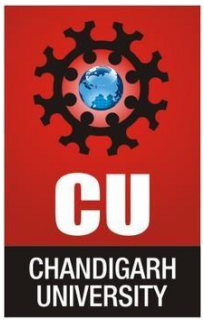
CHANDIGARH UNIVERSITY

UNIVERSITY INSTITUTE OF NGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



Submitted By: Vivek Kumar(21BCS8129)	**Submitted To:** Mamta Punia(E12337)
Subject Name	Competitive Coding - I
Subject Code	20CSP-314
Branch	Computer Science and Engineering
Semester	5^{th}

Experiment No. - 8

Student Name: Vivek Kumar
Branch: BE-CSE(LEET)
Semester: 5th
Subject Name: Competitive coding - I

UID: 21BCS8129
Section/Group: WM-20BCS-616/A
Date of Performance: 14/10/2022
Subject Code: 20CSP-314

Construct the Array

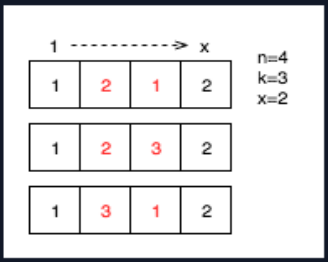
1. Aim/Overview of the practical:

Your goal is to find the number of ways to construct an array such that consecutive positions contain different values.

Specifically, we want to construct an array with n elements such that each element between 1 and k, inclusive. We also want the first and last elements of the array to be 1 and x.

Given n, k and x, find the number of ways to construct such an array. Since the answer may be large, only find it modulo $10^9 + 7$.

For example, for $n = 4, k = 3, x = 2$, there are 3 ways, as shown here:



Complete the function `countArray` which takes input n, k and x. Return the number of ways to construct the array such that consecutive elements are distinct.

2. Task to be done/ Which logistics used:

Constraints

- $3 \leq n \leq 10^5$
- $2 \leq k \leq 10^5$
- $1 \leq x \leq k$

Subtasks

- For 20% of the maximum score, $n \leq 10^3$ and $k \leq 10^2$

Sample Input

$n = 4, k = 3, x = 2$

Sample Output

3

Explanation

Refer to the diagram in the challenge statement.

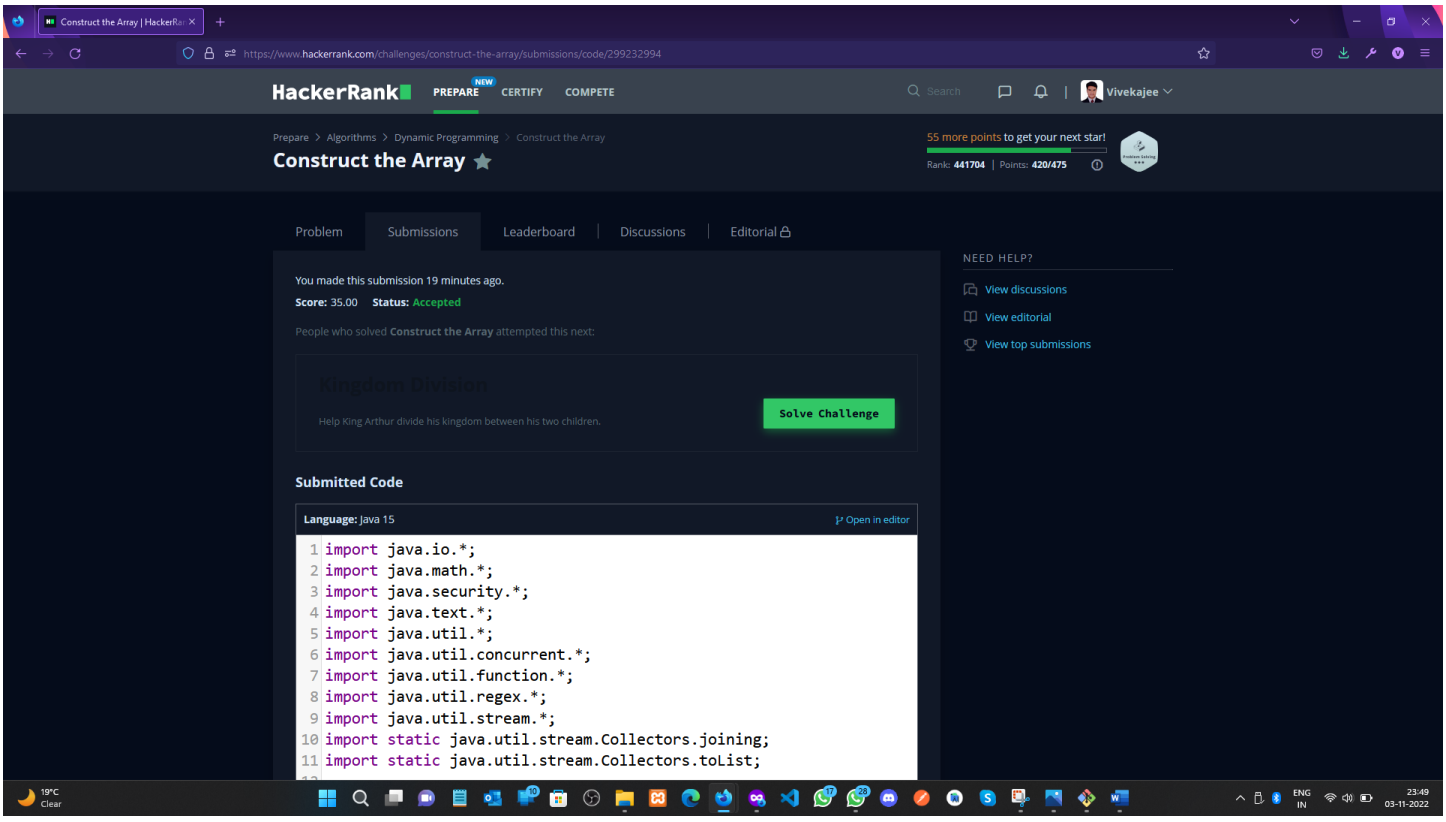
3. Hardware and Software Requirements (For programming-based labs):

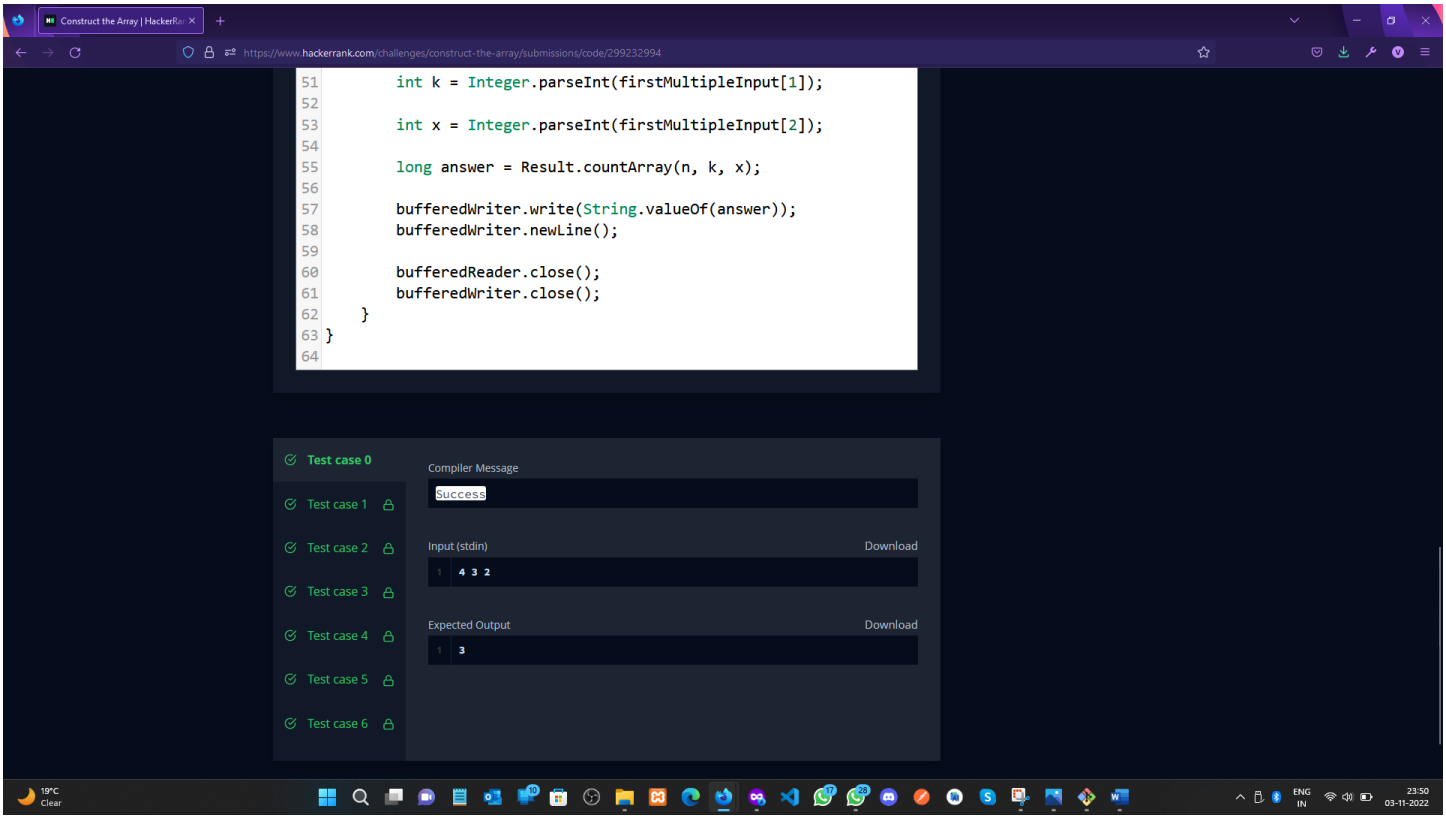
- Laptop or Desktop
- Hacker-Rank Account

4. 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];
    }
}
```

5. 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.

2. Task to be done/ Which logistics used:

Example

$arr = [1, 1, 5]$

arr represents the starting numbers of pieces for each colleague. She can give 2 pieces to the first two and the distribution is then $[3, 3, 5]$. On the next round, she gives the same two 2 pieces each, and everyone has the same number: $[5, 5, 5]$. Return the number of rounds, 2.

Function Description

Complete the equal function in the editor below.

equal has the following parameter(s):

- int arr[n]: the integers to equalize

Returns

- int: the minimum number of operations required

Input Format

The first line contains an integer t, the number of test cases.

Each test case has 2 lines.

- The first line contains an integer n, the number of colleagues and the size of arr.

- The second line contains n space-separated integers, $arr[i]$, the numbers of pieces of chocolate each colleague has at the start.

Constraints

$1 \leq t \leq 100$

$1 \leq n \leq 10000$

The number of chocolates each colleague has initially < 1000.

Sample Input

```
STDIN       Function
-----       --------
1           t = 1
4           arr[] size n = 4
2 2 3 7     arr =[2, 2, 3, 7]
```

Sample Output

```
2
```

Explanation

Start with $[2, 2, 3, 7]$

Add 1 to all but the 3rd element $\rightarrow$ $[3, 3, 3, 8]$

Add 5 to all but the 4th element $\rightarrow$ $[8, 8, 8, 8]$

Two operations were required.

Sample Input 1

```
1
3
10 7 12
```

Sample Output 1

```
3
```

Explanation 1

Start with $[10, 7, 12]$

Add 5 to the first two elements $\rightarrow [15, 12, 12]$

Add 2 to the last two elements $\rightarrow [15, 14, 14]$

Add 1 to the last two elements $\rightarrow [15, 15, 15]$

Three operations were required.

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

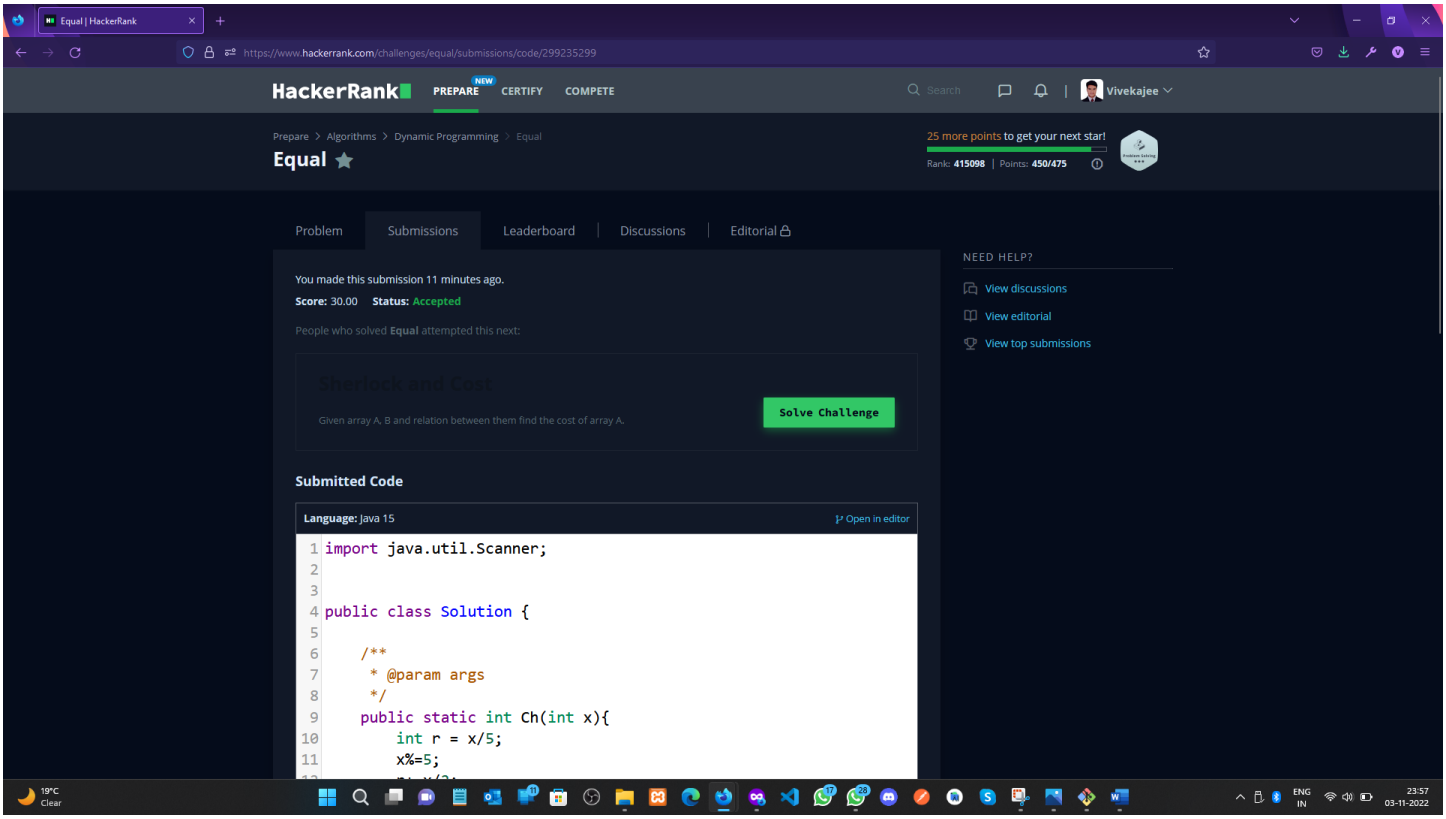
- Laptop or Desktop
- Hacker-Rank Account

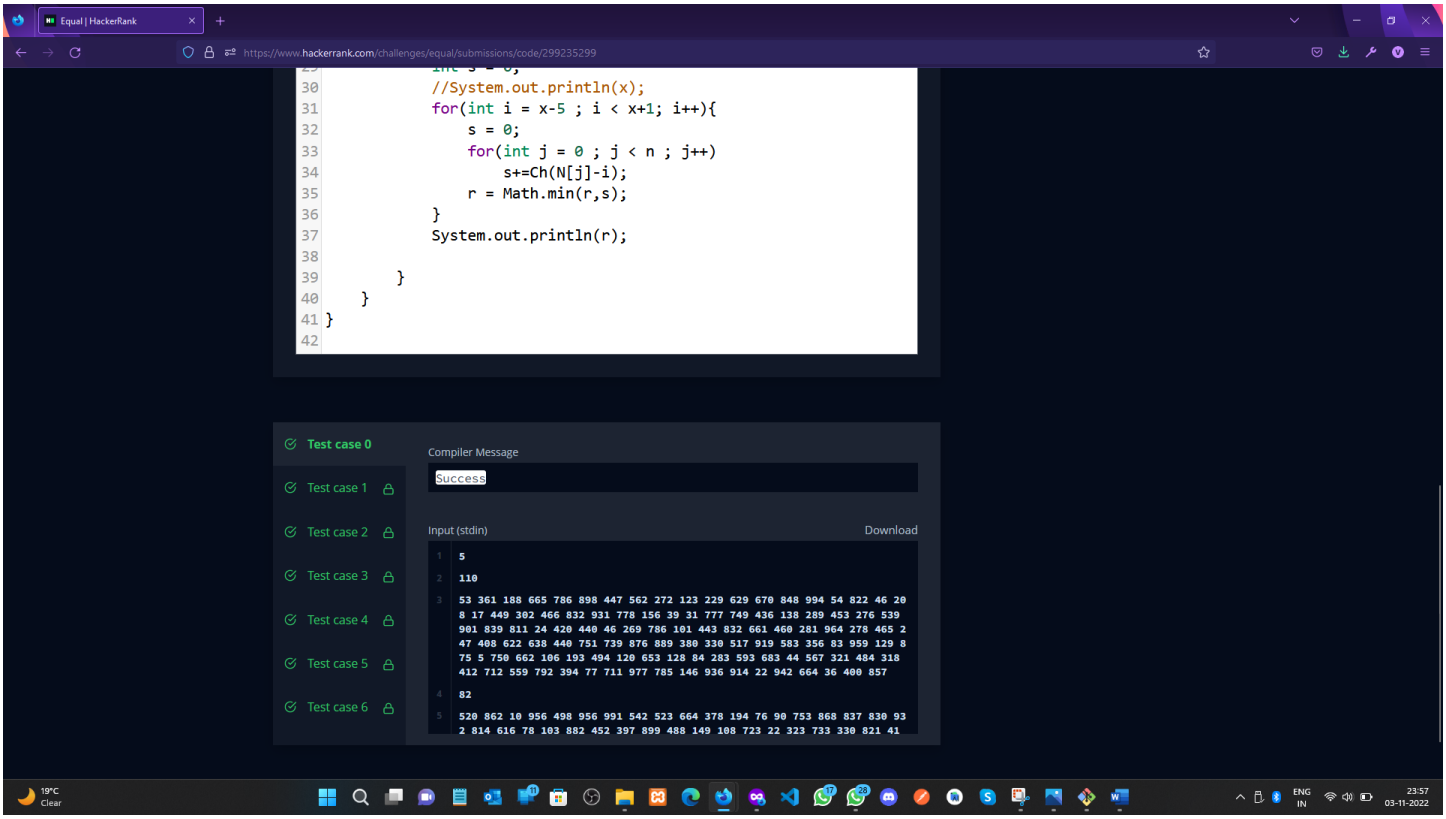
4. Steps for experiment/practical/Code:

```
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.

b. Learnt about Array in Dynamic Programming.

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:	