Experiment-3.2

Student Name: Virat Samdarshi **UID: 22BCS12648**
Branch: B.E(CSE) **Section/Group: IOT_627-B**
Semester: Fifth **Date of Performance: 17/10/24**
Subject Name: AP LAB 1 **Subject Code: 22CSP-314**

1.Aim: 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 .

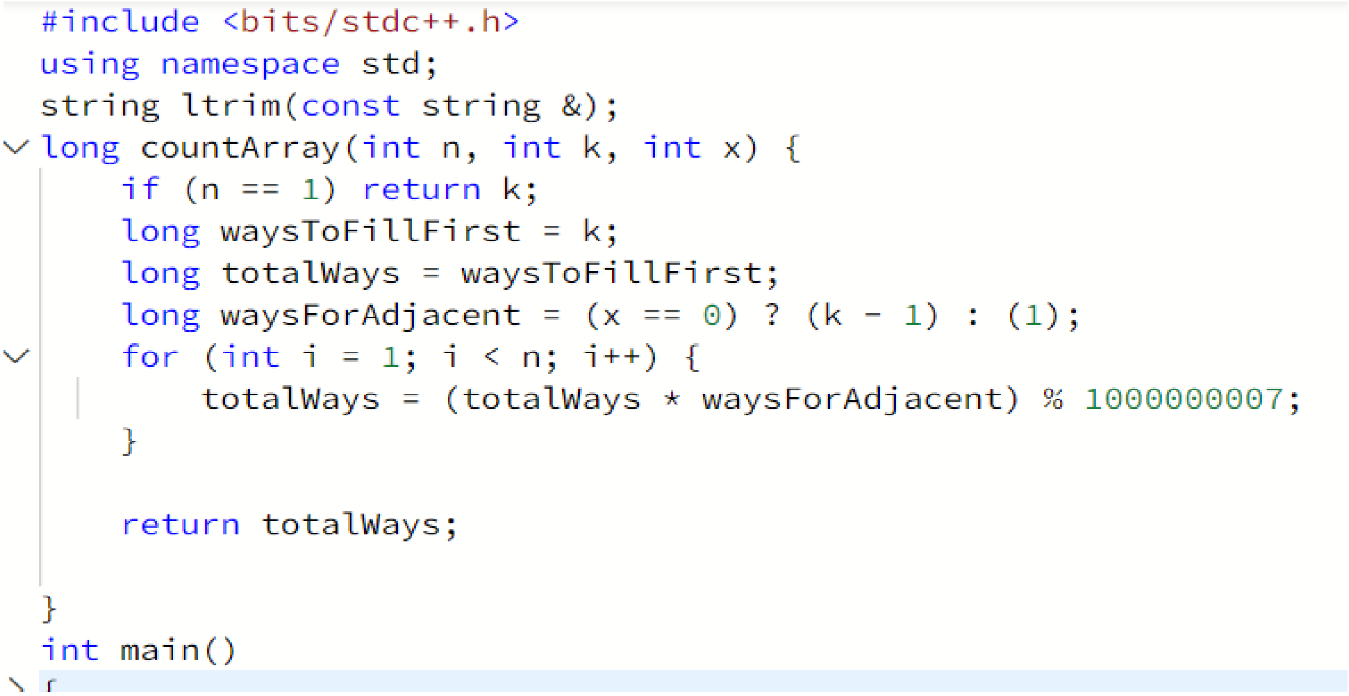
2.Objective: The objective of the Construct the Array problem on HackerRank is to determine the number of ways to fill an array of size nnn with integers ranging from 1 to k such that the XOR of every pair of adjacent elements equals a specified value x. The challenge is to find all valid configurations while adhering to these constraints.

3.Implementation/Code :-

```
long countArray(int n, int k, int x) {
if (n == 1) return k;
long waysToFillFirst = k;
long totalWays = waysToFillFirst;
long waysForAdjacent = (x == 0) ? (k - 1) : (1);
for (int i = 1; i < n; i++) {
    totalWays = (totalWays * waysForAdjacent) % 1000000007;
}

return totalWays;

}
```

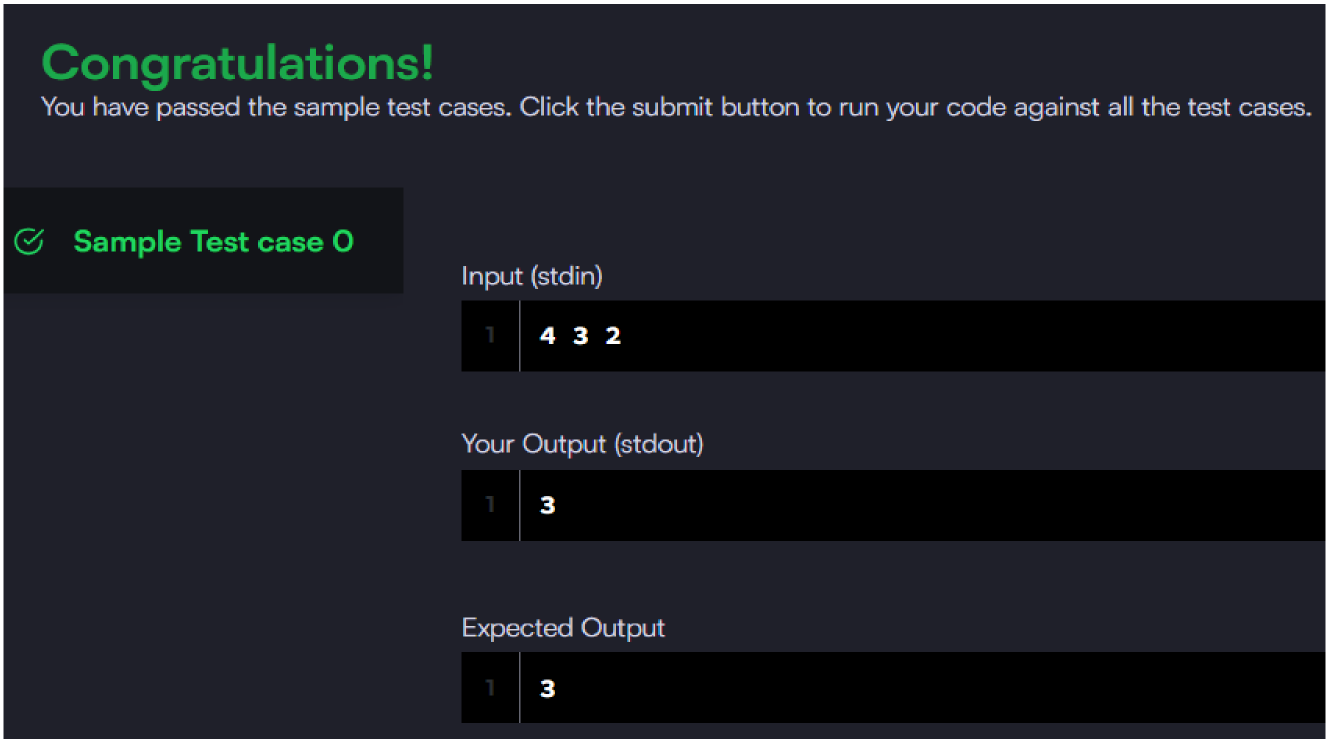


```
#include <bits/stdc++.h>
using namespace std;
string ltrim(const string &);
long countArray(int n, int k, int x) {
    if (n == 1) return k;
    long waysToFillFirst = k;
    long totalWays = waysToFillFirst;
    long waysForAdjacent = (x == 0) ? (k - 1) : (1);
    for (int i = 1; i < n; i++) {
        totalWays = (totalWays * waysForAdjacent) % 1000000007;
    }

    return totalWays;

}
int main()
{
```

4.Output :-



Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Input (stdin)

```
4 3 2
```

Your Output (stdout)

```
3
```

Expected Output

```
3
```

5.Time Complexity: O(n)

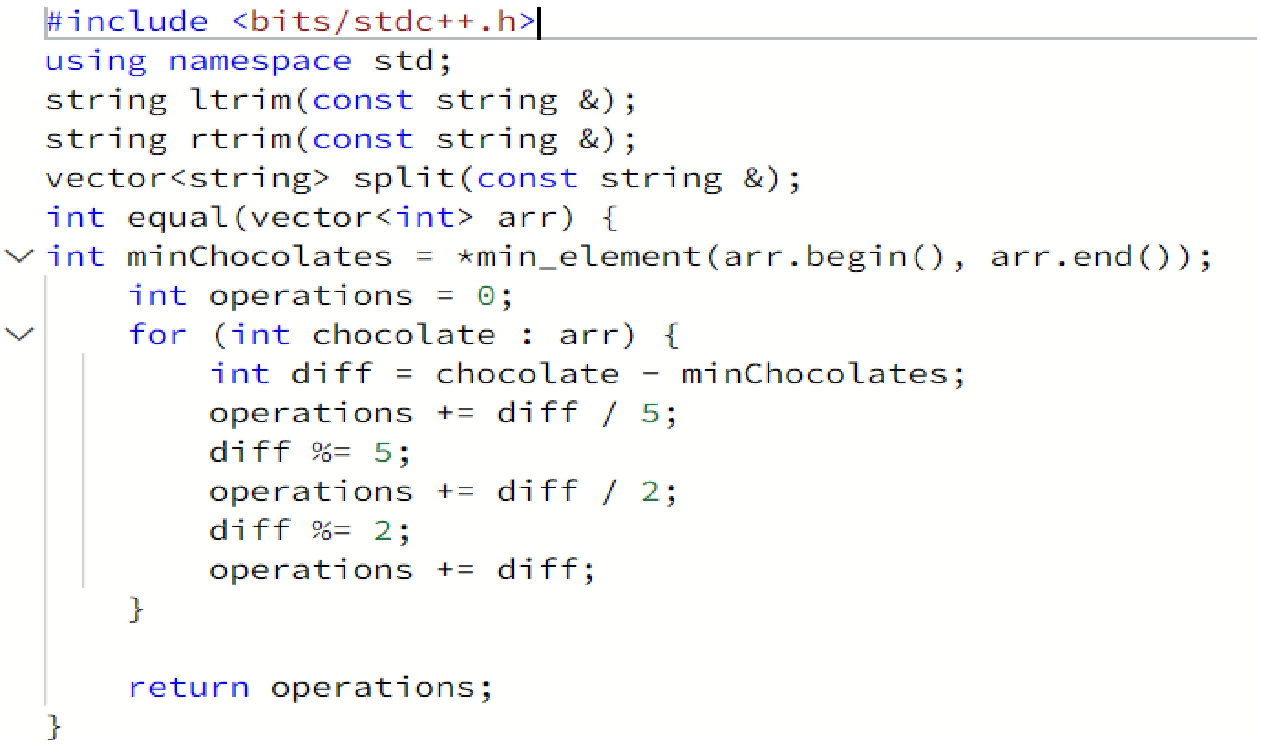
PROBLEM 2

1. Aim :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 and 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. Objective: The Equal problem requires you to determine the minimum number of operations needed to make all elements in an array (representing the distribution of chocolates among colleagues) equal.

3. Implementation/Code :-

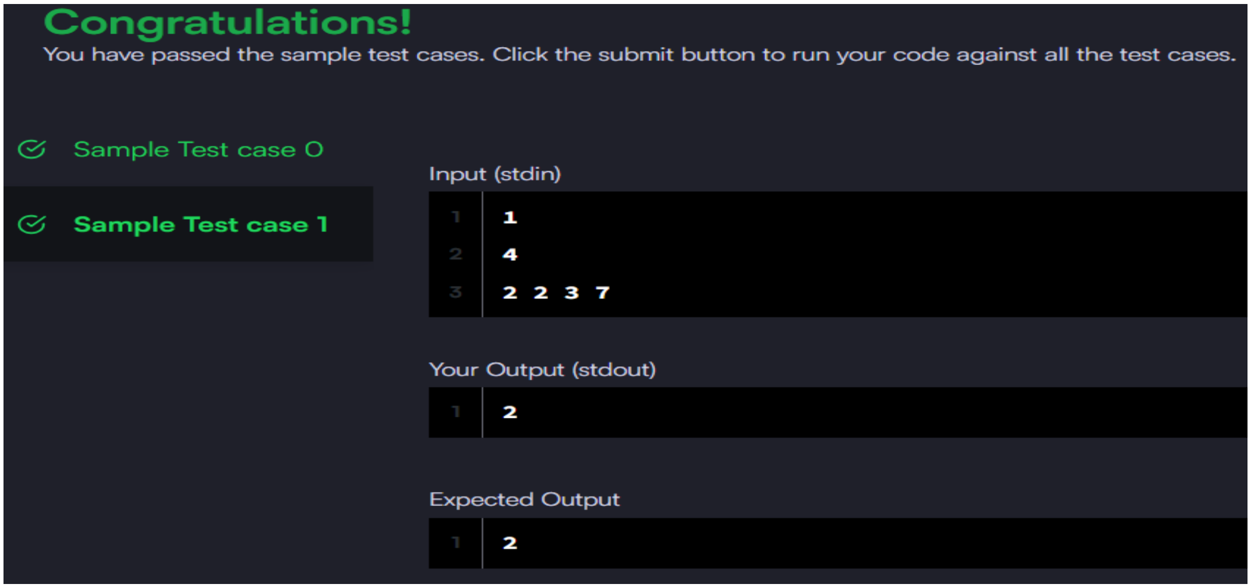
```
int equal(vector<int> arr) {
int minChocolates = *min_element(arr.begin(), arr.end());
    int operations = 0;
    for (int chocolate : arr) {
        int diff = chocolate - minChocolates;
        operations += diff / 5;
        diff %= 5;
        operations += diff / 2;
        diff %= 2;
        operations += diff;
    }
    return operations;
}
```



```
#include <bits/stdc++.h>
using namespace std;
string ltrim(const string &);
string rtrim(const string &);
vector<string> split(const string &);
int equal(vector<int> arr) {
int minChocolates = *min_element(arr.begin(), arr.end());
    int operations = 0;
    for (int chocolate : arr) {
        int diff = chocolate - minChocolates;
        operations += diff / 5;
        diff %= 5;
        operations += diff / 2;
        diff %= 2;
        operations += diff;
    }

    return operations;
}
```

4. Output:



Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Input (stdin)

```
1
4
2 2 3 7
```

Your Output (stdout)

```
2
```

Expected Output

```
2
```

5.Time Complexity: O(n)