Experiment-3.1

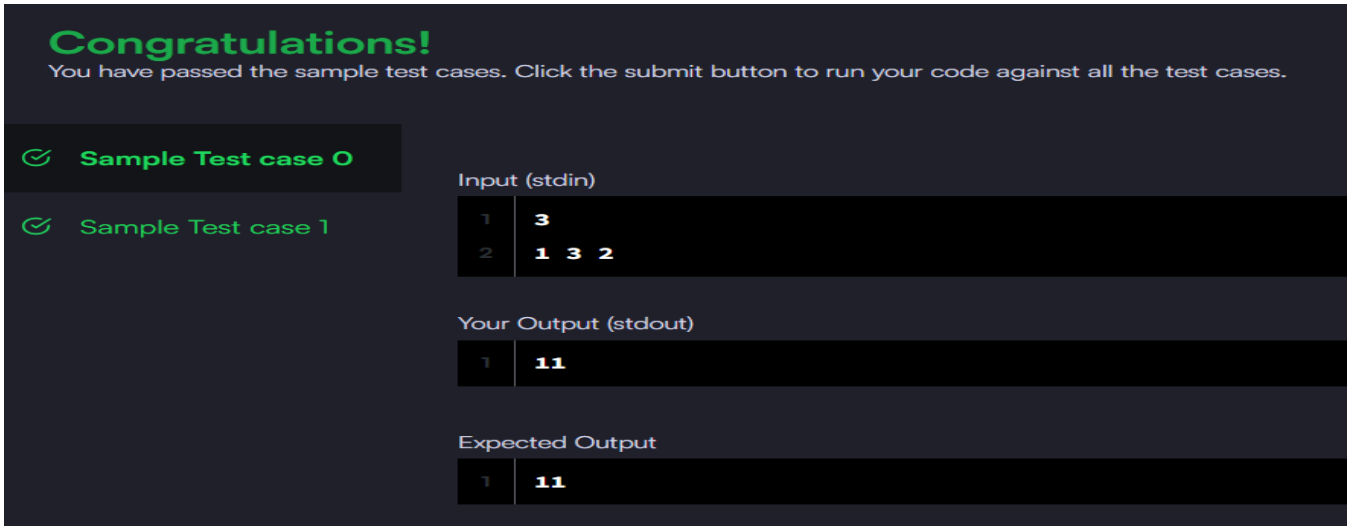
Student Name: Virat Samdarshi **UID: 22BCS12648**
Branch: B.E(CSE) **Section/Group: IOT_627-B**
Semester: Fifth **Date of Performance: 17/10/24**
Subject Name: AP LAB 1 **Subject Code: 22CSP-314**

1. **Aim:** Marc loves cupcakes, but he also likes to stay fit. Each cupcake has a calorie count, and Marc can walk a distance to expend those calories. If Marc has eaten j cupcakes so far, after eating a cupcake with ccc calories he must walk at least2j×c miles to maintain his weight.
2. **Objective:** The objective of Marc's Cakewalk problem on HackerRank is to minimize the total number of calories Marc consumes when eating a series of cupcakes, each with a different calorie count. Marc's calorie consumption is determined by a formula where the calories of each cupcake are multiplied by an exponentially increasing factor
3. **Implementation/Code :-**

```
long marcsCakewalk(vector<int> calorie) {
sort(calorie.rbegin(), calorie.rend());

 long totalCalories = 0;

 for (int i = 0; i < calorie.size(); i++) {
    totalCalories += (long)calorie[i] * (1L << i);
 }

 return totalCalories;
}
```

```
#include <bits/stdc++.h>
using namespace std;
string ltrim(const string &);
string rtrim(const string &);
vector<string> split(const string &);
long marcsCakewalk(vector<int> calorie) {
sort(calorie.rbegin(), calorie.rend());

    long totalCalories = 0;

    for (int i = 0; i < calorie.size(); i++) {
        totalCalories += (long)calorie[i] * (1L << i);
    }

    return totalCalories;
}
```

4.Output :-



5.Time Complexity: O(nlogn)

PROBLEM 2

1. Aim : Alice is a kindergarten teacher. She wants to give some candies to the children in her class. All the children sit in a line and each of them has a rating score according to his or her performance in the class. Alice wants to give at least 1 candy to each child. If two children sit next to each other, then the one with the higher rating must get more candies. Alice wants to minimize the total number of candies she must buy.

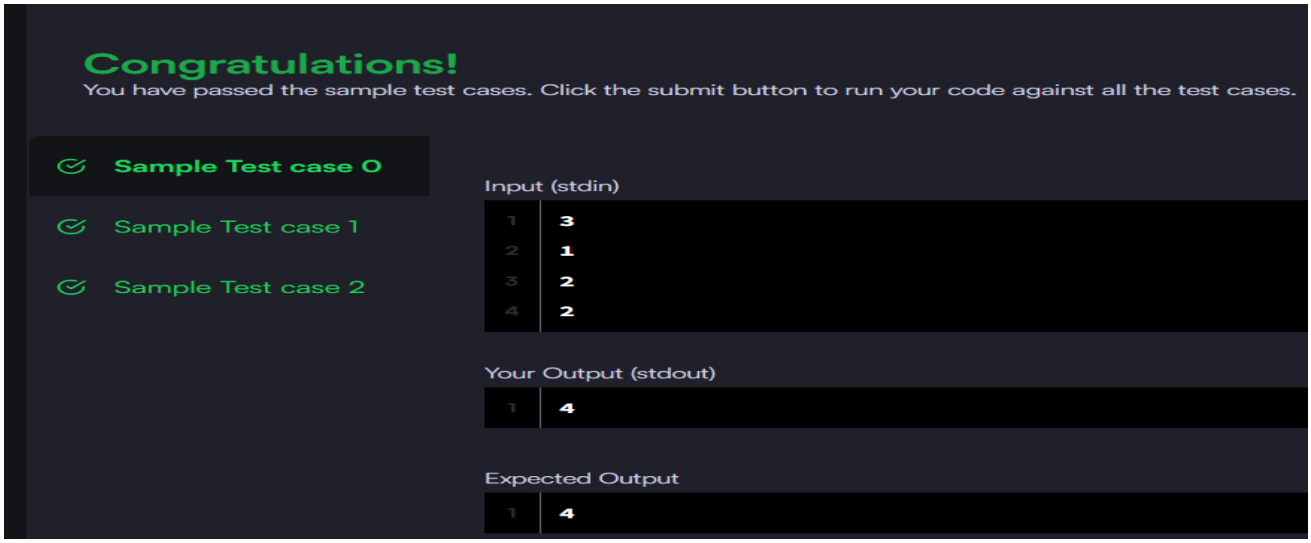
2. Objective: The objective of the Candies problem on HackerRank is to distribute a minimum number of candies among students standing in a line, where each student must receive at least one candy. If a student has a higher rating than their adjacent neighbors, they should receive more candies than those neighbors.

3. Implementation/Code :-

```
long candies(int n, vector<int> arr) {
vector<long> candies(n, 1);

  for (int i = 1; i < n; i++) {
        if (arr[i] > arr[i - 1]) {
           candies[i] = candies[i - 1] + 1;
        }
    }
    for (int i = n - 2; i >= 0; i--) {
        if (arr[i] > arr[i + 1]) {
           candies[i] = max(candies[i], candies[i + 1] + 1);
        }
    }

    return accumulate(candies.begin(), candies.end(), 0L);
}
```

```
#include <bits/stdc++.h>
using namespace std;
string ltrim(const string &);
string rtrim(const string &);
long candies(int n, vector<int> arr) {
vector<long> candies(n, 1);
    for (int i = 1; i < n; i++) {
        if (arr[i] > arr[i - 1]) {
            candies[i] = candies[i - 1] + 1;
        }
    }

    for (int i = n - 2; i >= 0; i--) {
        if (arr[i] > arr[i + 1]) {
            candies[i] = max(candies[i], candies[i + 1] + 1);
        }
    }

    return accumulate(candies.begin(), candies.end(), 0L);
}
int main()
{ ...
}
    return s;
}
```

4. Output:



5.Time Complexity: O(n)