Experiment 9

Student Name: Gaganjot Singh
UID: 22BCS14843
Branch: BE-CSE
Section/Group: 22BCS-JT-802-B
Semester: 5th
Date of Performance:21/10/2024
Subject Name: Advanced Programming Lab
Subject Code: 22CSP-314

1. **Title (1a):** Recursion:Fibonacci series

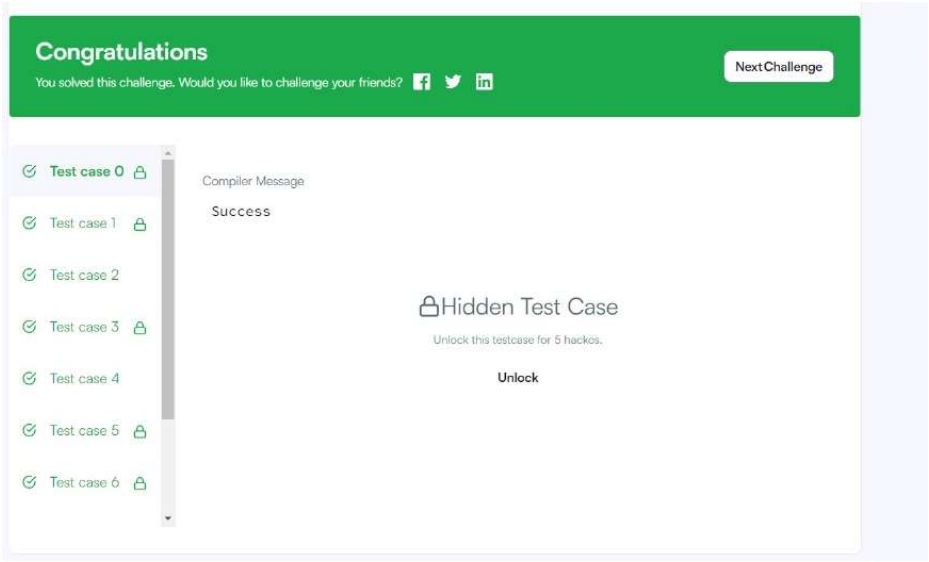
2. **Aim:** The Fibonacci sequence appears in nature all around us, in the arrangement of seeds in a sunflower and the spiral of a nautilus for example. The Fibonacci sequence begins with Fibonacci (0)=0 and Fibonacci(1)=1 as its first and second terms. After these first two elements, each subsequent element is equal to the sum of the previous two elements.

3. **Objective:** To demonstrate the concept of Back-tracking/ Dynamic Programming.

4. **Code:**

```
int fibonacci(int n) {

    int prev2 = 0;
    int prev = 1;
    for(int i = 2; i<=n; i++)
    {
        int curi = prev + prev2;
        prev2 = prev;
        prev = curi;
    }
    return prev;
}
```

5. **Output:**



6. **Complexity:**

Time Complexity: O(n)

Space Complexity: O(1)

7. **Learning Outcomes:**
   - Fibonacci Sequence: Learn how to compute the Fibonacci sequence iteratively.
   - Space Optimization: Understand how to reduce space complexity by using only two variables (prev2 and prev) instead of an array.
   - Looping Constructs: Gain experience in using for loops to iterate through a sequence of numbers.
   - Variable Updates: Learn the technique of updating variables dynamically to store the previous two values in a sequence.

1. **Title (1b):** Sam and substrings

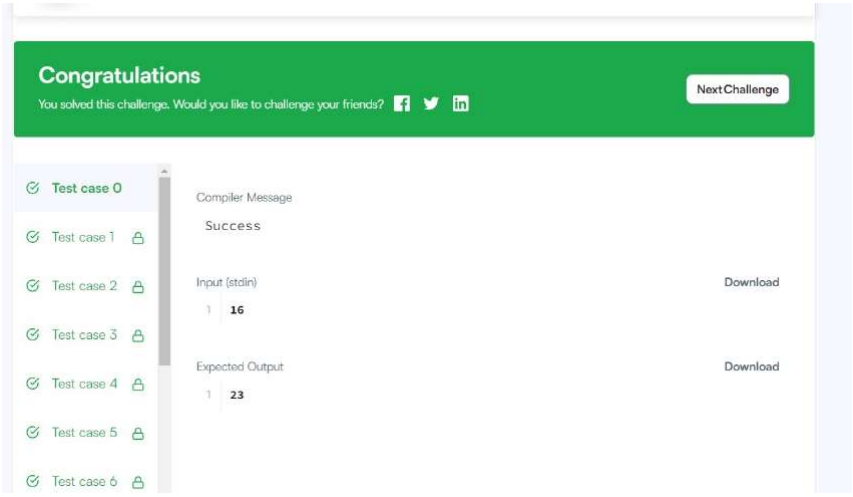
2. **Aim:** Samantha and Sam are playing a numbers game. Given a number as a string, no leading zeros, determine the sum of all integer values of substrings of the string. Given an integer as a string, sum all of its substrings cast as integers. As the number may become large, return the value modulo 10^9+7.

3. **Objective:** To demonstrate the concept of Back-tracking/ Dynamic Programming.

4. **Code:**

```
const int MOD = 1e9+7;

int substrings(string n) {
    int len = n.length();
    long long totalSum = 0;
    long long currentContribution = 0;

    // Loop through each character in the string
    for (int i = 0; i < len; i++) {
        // Convert the current character to integer
        int digit = n[i] - '0';

        // Calculate the current contribution of this digit
        currentContribution = (currentContribution * 10 + (i + 1) * digit) % MOD;

        // Add the current contribution to the total sum
        totalSum = (totalSum + currentContribution) % MOD;
    }

    return totalSum;
}
```

5. Output:



6. Complexity:

Time Complexity: O(n)

Space Complexity: O(1)

7. Learning Outcomes:

- String Manipulation: Learn how to iterate through a string and extract individual characters.
- Modular Arithmetic: Understand the use of modular arithmetic to handle large numbers and prevent overflow.
- Digit Conversion: Gain experience in converting characters to their corresponding integer values.
- Contribution Calculation: Learn to compute the contribution of each digit to the overall result based on its position.