Batch: SY-IT(B3) Experiment Number:4

Roll Number: 16010423076 Name:Ritesh Jha

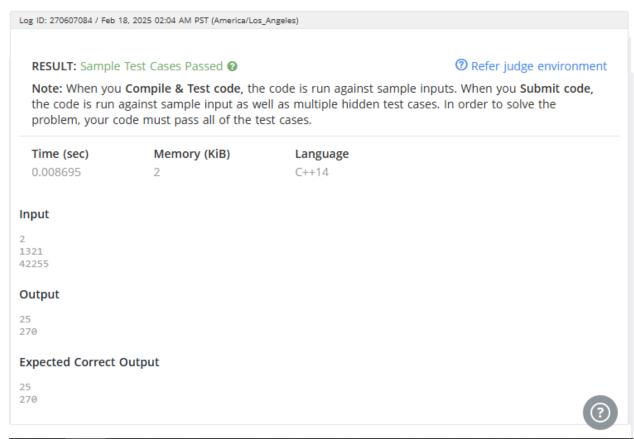
**Aim of the Experiment:** To study Greedy Programming approach for implementation of problem statement to obtain optimal solution.

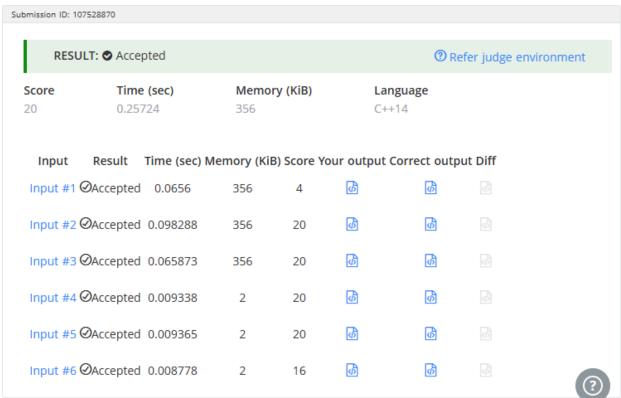
## **Program/ Steps:**

```
#include <iostream>
using namespace std;
int main() {
   int T;
    cin >> T;
   while (T--) {
        string num;
        cin >> num;
        // Step 1: Count the frequency of each digit (0 to 9)
        int freq[10] = {0};
        for (int i = 0; i < num.length(); i++) {</pre>
            freq[num[i] - '0']++; // convert character to integer
and update frequency
        }
        // Step 2: Store digits in sorted order
        char digits[20];
        int index = 0; // Index to track position in digits array
        for (int i = 0; i < 10; i++) {</pre>
            while (freq[i] > 0) { // If this digit appears in the
input
```

```
digits[index++] = '0' + i; // Store it in the
array
                freq[i]--;
            }
        }
        // Step 3: Distribute digits into two numbers alternately
        string num1 = "", num2 = ""; // Two numbers as strings
        for (int i = 0; i < index; i++) {</pre>
            if (i % 2 == 0) { // Even index goes to num1
                num1 += digits[i];
            } else { // Odd index goes to num2
                num2 += digits[i];
            }
        }
        //Convert strings to integers and calculate sum
        int sum = (num1.empty() ? 0 : stoi(num1)) + (num2.empty()
? 0 : stoi(num2));
        cout << sum << endl;</pre>
    }
    return 0;
```

**Output/Result:** 





**Outcomes**: CO2. Understand the fundamental concepts for managing the data using different data structures such as lists, queues, trees etc.

## Conclusion (based on the Results and outcomes achieved):

From this experiment, I learned how to efficiently process and manipulate numerical data using arrays and strings in C++. By implementing a digit frequency count and sorting mechanism, I understood how to systematically distribute numbers to achieve a desired outcome.

## **References:**

- 1. https://tutorialspoint.dev/algorithm/greedy-algorithms/greedy-algorithm-to-find-minim um-number-of-coins
- 2. <a href="https://www.baeldung.com/cs/min-number-of-coins-algorithm">https://www.baeldung.com/cs/min-number-of-coins-algorithm</a>