Experiment 7

Student Name: Aman Gokul
UID: 20BCS5449
Branch: BE CSE
Section/Group: 607/A
Semester: 6th
Date of Performance: 21/04/2023
Subject Name: CC-2 Lab
Subject Code: 20CSP-351

1. Aim/Overview of the practical:

1-bit and 2-bit Characters

We have two special characters:

- The first character can be represented by one bit 0.
- The second character can be represented by two bits (10 or 11).

Given a binary array bits that ends with 0, return true if the last character must be a one-bit character.

https://leetcode.com/problems/1-bit-and-2-bit-characters/

2. Apparatus / Simulator Used:

- Windows 7 or above
- Google Chrome

3. Objective:

- To understand the concept of Divide and Conquer.

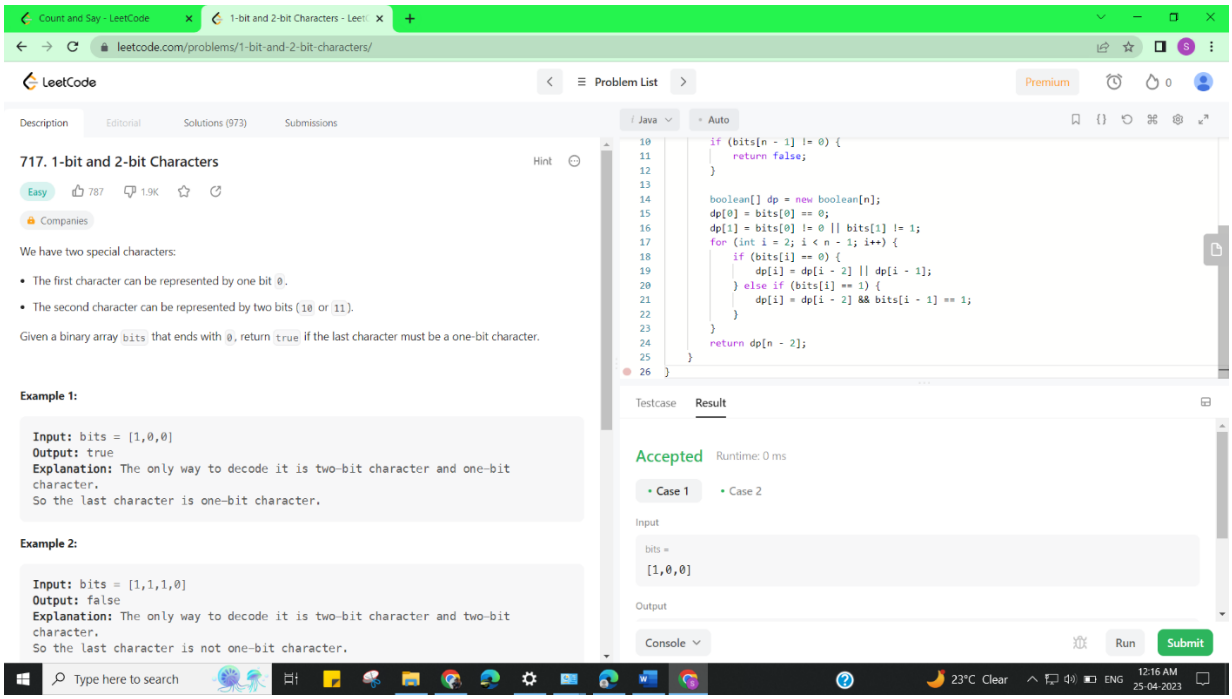
A divide-and-conquer algorithm recursively breaks down a problem into two or more sub-problems of the same or related type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem.

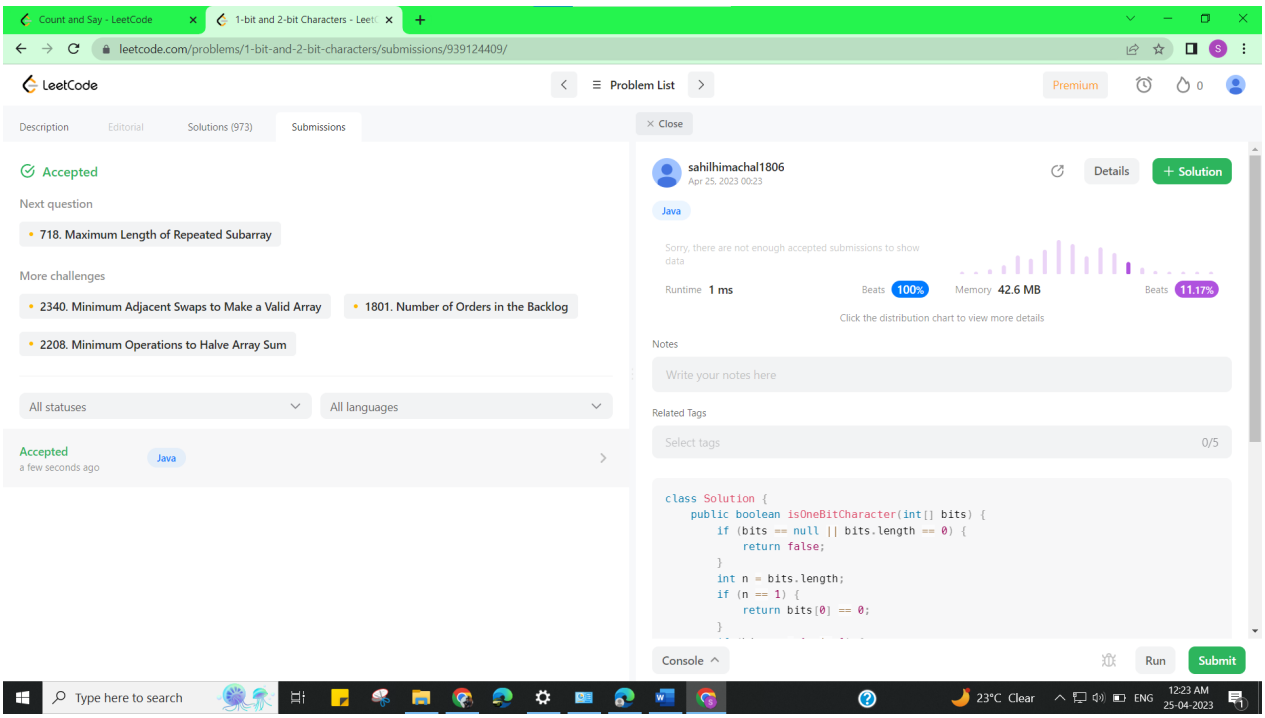
4. Code:

```
class Solution {
   public boolean isOneBitCharacter(int[] bits) {
      if (bits == null || bits.length == 0) {
         return false;
      }
      int n = bits.length;
      if (n == 1) {
         return bits[0] == 0;
      }
      if (bits[n - 1] != 0) {
         return false;
      }
      boolean[] dp = new boolean[n];
      dp[0] = bits[0] == 0;
      dp[1] = bits[0] != 0 || bits[1] != 1;
      for (int i = 2; i < n - 1; i++) {
         if (bits[i] == 0) {
```

```
            dp[i] = dp[i - 2] || dp[i - 1];
        } else if (bits[i] == 1) {
            dp[i] = dp[i - 2] && bits[i - 1] == 1;
        }
    }
    return dp[n - 2];
  }
}
```

4. Result/Output/Writing Summary:





Experiment 7.2

1. Aim/Overview of the practical:

Count and Say

The **count-and-say** sequence is a sequence of digit strings defined by the recursive formula:

- countAndSay(1) = "1"
- countAndSay(n) is the way you would "say" the digit string from countAndSay(n-1), which is then converted into a different digit string.

https://leetcode.com/problems/count-and-say/

2. Apparatus / Simulator Used:

- Windows 7 or above
- Google Chrome

3. Objective:

- To understand the concept of Divide and Conquer.

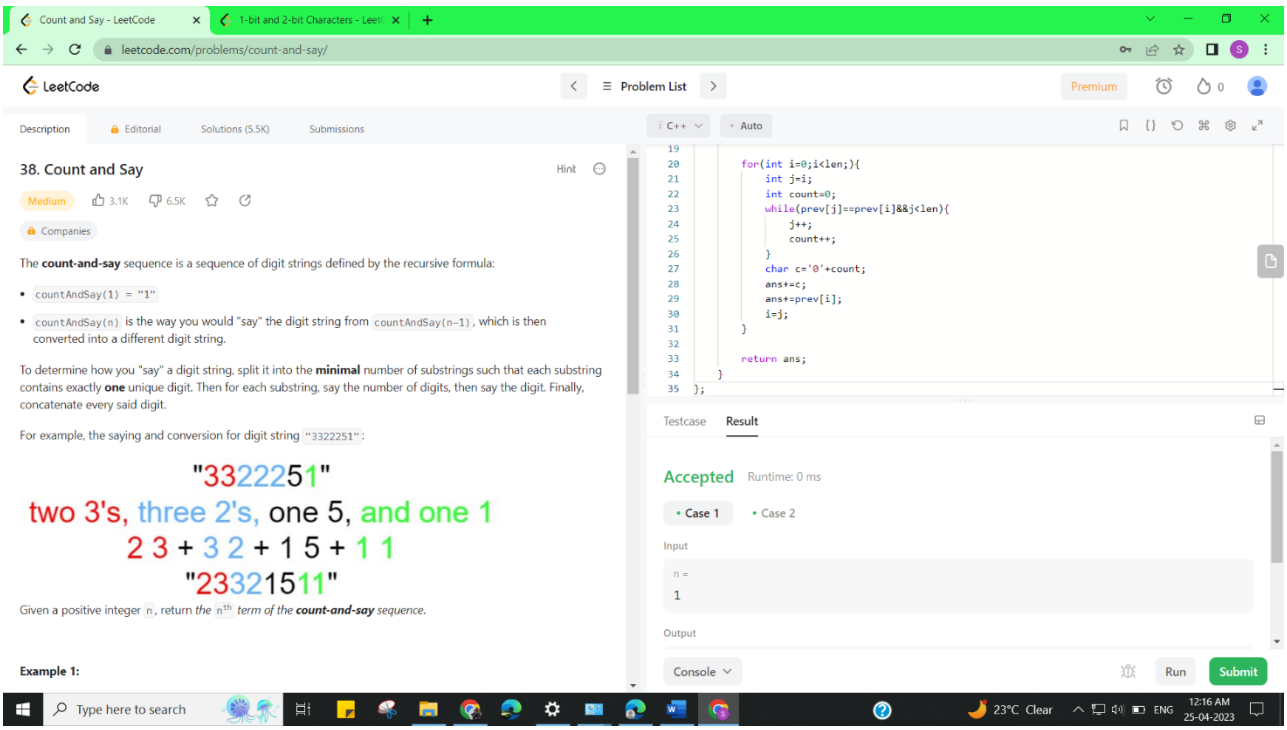
A divide-and-conquer algorithm recursively breaks down a problem into two or more sub-problems of the same or related type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem.

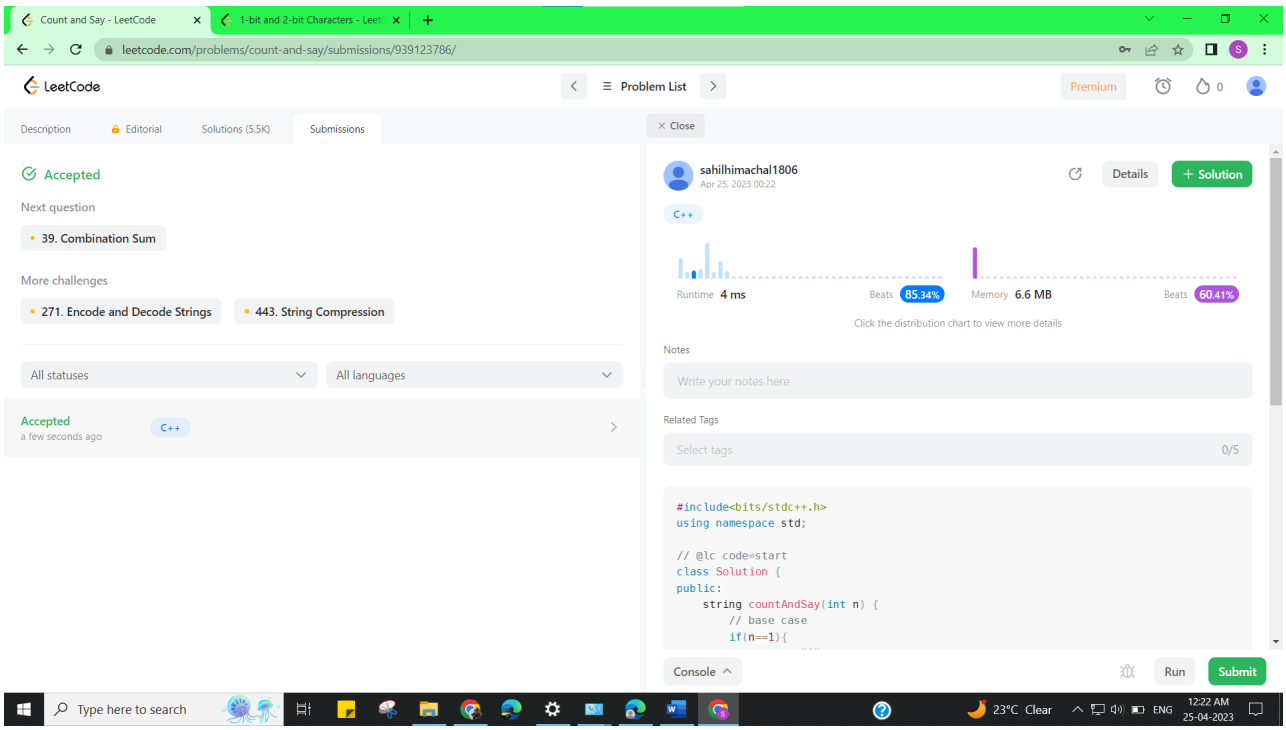
4. Code:

```
#include<bits/stdc++.h>
using namespace std;
class Solution {
public:
   string countAndSay(int n) {
      if(n==1){
```

```
        return "1";
    }
    string prev=countAndSay(n-1);
    cout<<n<<" "<<prev<<"\n";

    string ans;
    int len=prev.length();

    for(int i=0;i<len;){
        int j=i;
        int count=0;
        while(prev[j]==prev[i]&&j<len){
            j++;
            count++;
        }
        char c='0'+count;
        ans+=c;
        ans+=prev[i];
        i=j;
    }

    return ans;
  }
};
```

5. Result/Output/Writing Summary:





Learning outcomes (What I have learnt):

- Learned the concept of Divide and Conquer.