

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## Experiment-7

**Student Name:** Kanishk Soni  
**Branch:** BE-CSE  
**Semester:** 6<sup>th</sup>  
**Subject Code:** 20CSP-351

**UID:** 20BCS9398  
**Section/Group:** 20BCS\_DM-708B  
**Subject Name:** Competitive Coding-II

**AIM:** To demonstrate the concept of Divide and conquer.

**Problem1:** Count and Say

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

**Program Code:**

```
class Solution {
public:
    string countAndSay(int n) {
        if (n == 0) return "";
        string res = "1";
        while (--n) {
            string cur = "";
            for (int i = 0; i < res.size(); i++) {
                int count = 1;
                while ((i + 1 < res.size()) && (res[i] == res[i + 1])){
                    count++;
                    i++;
                }
                cur += to_string(count) + res[i];
            }
            res = cur;
        }
    }
};
```

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

```
        res = cur;
    }

    return res;
}

};
```

## Output:

The screenshot shows a LeetCode solution page for problem 39, "Combination Sum". The status is "Accepted". The user is Kanishk, who solved it on May 02, 2023, at 12:12. The solution is written in C++. The runtime is 4 ms, and the memory is 6.7 MB. The solution beats 84.63% of other submissions for runtime and 41.53% for memory. The code is as follows:

```
class Solution {
public:
    string countAndSay(int n) {
        if (n == 0) return "";
        string res = "1";
        while (--n) {
            string cur = "";
            for (int i = 0; i < res.size(); i++) {
                int count = 1;
                while ((i + 1 < res.size()) && (res[i] == res[i + 1])){
                    count++;
                    i++;
                }
                cur += to_string(count) + res[i];
            }
            res = cur;
        }
    }
}
```

## Problem2: Water and Jug Problem

<https://leetcode.com/problems/water-and-jug-problem/>

## Program Code:

```
class Solution {
public:
    bool canMeasureWater(int jug1Capacity, int jug2Capacity, int targetCapacity) {
        if(targetCapacity == 0)
        {
            return true;
        }
    }
};
```

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

```
}  
  
if(targetCapacity > jug1Capacity + jug2Capacity)  
{  
    return false;  
}  
  
else  
{  
    int g = gcd(jug1Capacity,jug2Capacity);  
    return targetCapacity % g == 0;  
}  
}  
};
```

## Output:

The screenshot shows a LeetCode problem page for "366. Find Leaves of Binary Tree". The solution is marked as "Accepted" and was submitted by user "Kanishk" on May 02, 2023, at 12:14. The solution is written in C++ and is 100% efficient in terms of runtime and memory.

**Accepted**  
a few seconds ago

Next question  
• 366. Find Leaves of Binary Tree

More challenges  
• 2258. Escape the Spreading Fire • 753. Cracking the Safe  
• 685. Redundant Connection II

All statuses ▾ All languages ▾

Runtime 0 ms Beats 100% Memory 6 MB Beats 65.1%

Click the distribution chart to view more details

Notes  
Write your notes here

Related Tags  
Select tags 0/5

```
class Solution {  
public:  
    bool canMeasureWater(int jug1Capacity, int jug2Capacity, int targetCapacity) {  
        if(targetCapacity == 0)  
        {  
            return true;  
        }  
        if(targetCapacity > jug1Capacity + jug2Capacity)  
        {  
            return false;  
        }  
        else  
        {  
            int g = gcd(jug1Capacity,jug2Capacity);  
            return targetCapacity % g == 0;  
        }  
    }  
}
```

Console ▾ Run Submit