



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment-7

Student Name: Anshuman Singh

UID: 20BCS2665

Branch: CSE

Section/Group: 902/A

Semester: 6th

Date of Performance: 13-04-2023

Subject Name: Competitive Coding II

Subject Code: 20CSP-351

Aim: To demonstrate the concept of Divide and conquer.

Problem1: 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.

To determine how you "say" a digit string, split it into the minimal number of substrings such that each substring contains exactly one unique digit. Then for each substring, say the number of digits, then say the digit. Finally, concatenate every said digit..

Code:-

```
class Solution {
public:
string solve(int n){
if(n==1){
return "1";
}
string s= solve(n-1);
int count=1,i=1,num=s[0]-'0';
string ans="";
while(i<s.length()){
if((s[i]-'0')==num){
count++;
}
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
else{
ans+=to_string(count);
count=1;
ans+=to_string(num);
num=s[i]-'0';
}
i++;
}
ans+=to_string(count);
ans+=to_string(num);
return ans;
}
string countAndSay(int n) {
string res= solve(n);
return res;
}
};
```

Output:-

A screenshot of a code execution environment. At the top left, the word 'Accepted' is written in green, followed by 'Runtime: 3 ms'. Below this, there are two tabs: 'Case 1' (selected) and 'Case 2'. Under the 'Case 1' tab, there is an 'Input' section with 'n =' and the value '1'. Below that is an 'Output' section with the string '"1"'. At the bottom, there is an 'Expected' section with the string '"1"'. The entire interface has a dark background with light text.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Problem2: 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.

Code:-

```
class Solution {
public:
bool isOneBitCharacter(vector<int>& bits)
{
int i=0;
while(i<bits.size())
{
if(i==bits.size()-1)
return true;

i=(bits[i]==1) ? i+2 : i+1;
}
return false;
}
};
```

Output:-

