38. Count and Say - 22/08/24 (Medium)

38. Count and Say

Medium | Topics | Companies | Hint

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

- `countAndSay(1) = "1"`
- `countAndSay(n)` is the run-length encoding of `countAndSay(n - 1)`.

Run-length encoding (RLE) is a string compression method that works by replacing consecutive identical characters (repeated 2 or more times) with the concatenation of the character and the number marking the count of the characters (length of the run). For example, to compress the string `"3322251"` we replace `"33"` with `"23"`, replace `"222"` with `"32"`, replace `"5"` with `"15"` and replace `"1"` with `"11"`. Thus the compressed string becomes `"23321511"`.

Given a positive integer `n`, return *the* n^{th} *element of the* ***count-and-say*** *sequence.*

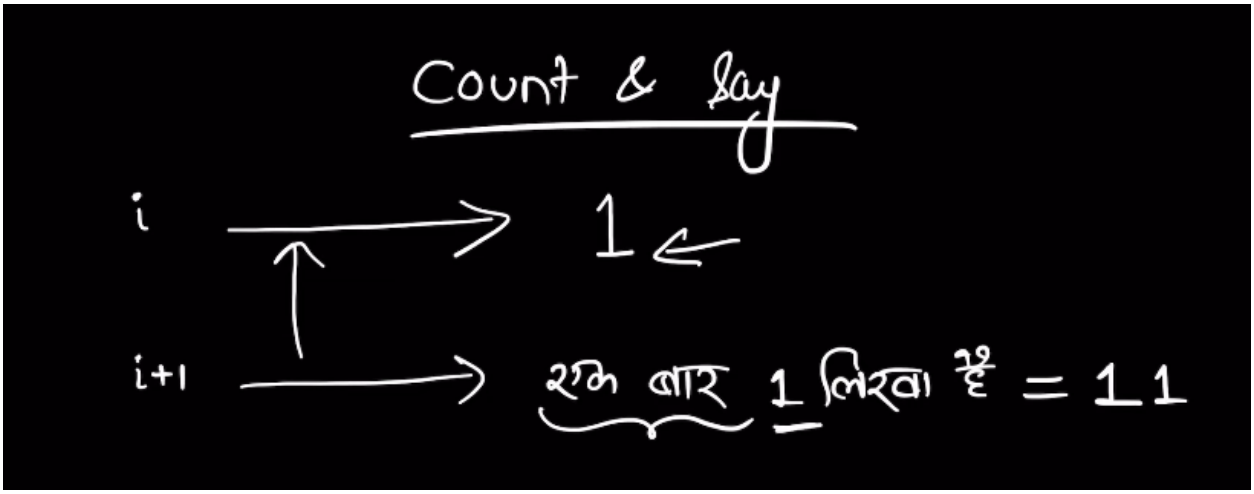
Example 1:

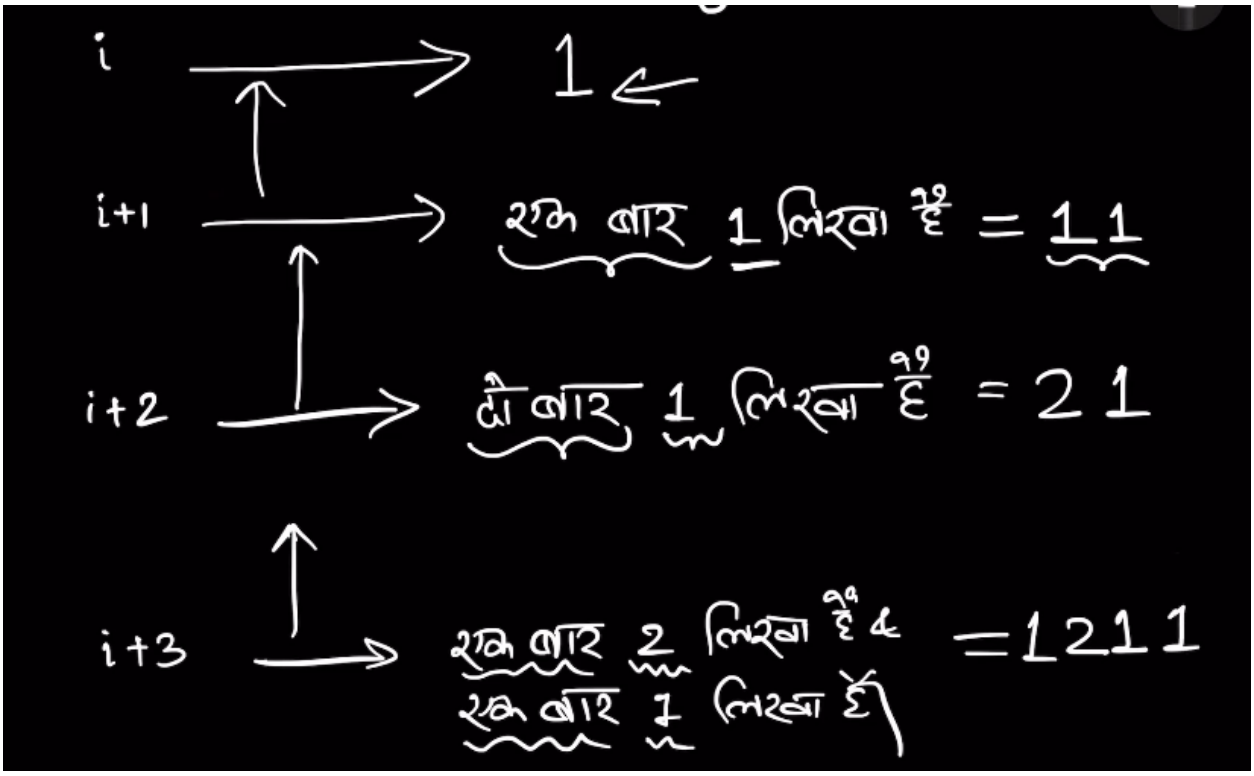
> **Input:** `n = 4`
>
> **Output:** `"1211"`
>
> **Explanation:**
>
> ```
> countAndSay(1) = "1"
> countAndSay(2) = RLE of "1" = "11"
> countAndSay(3) = RLE of "11" = "21"
> countAndSay(4) = RLE of "21" = "1211"
> ```

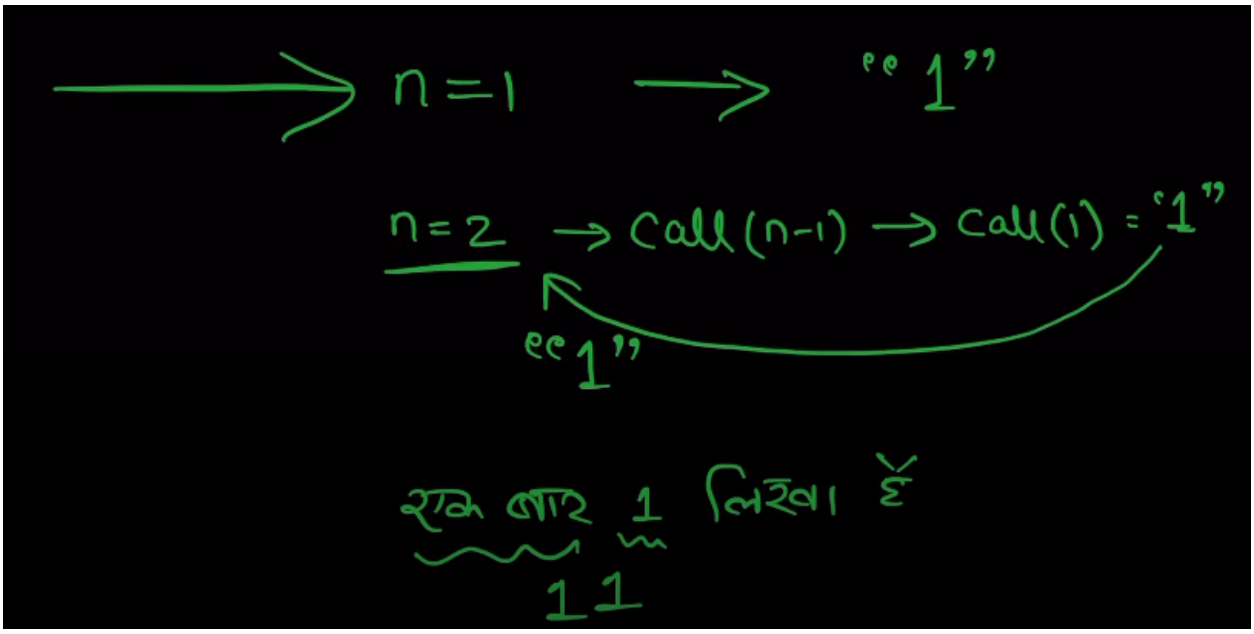
Example 2:

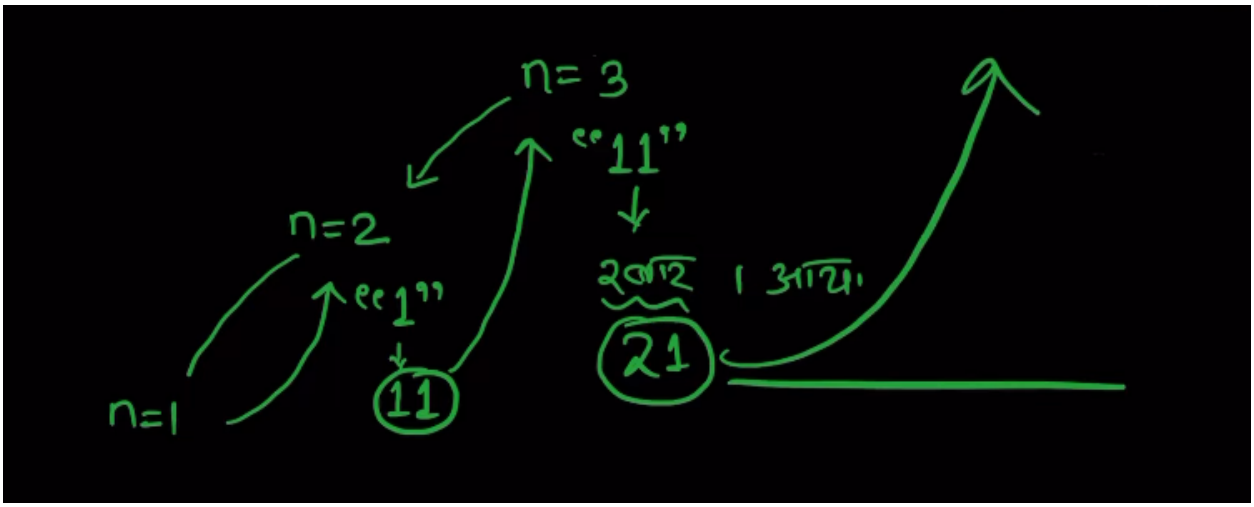
> **Input:** `n = 1`
>
> **Output:** `"1"`
>
> **Explanation:**
>
> This is the base case.

Explanation




i
1
i+1
एक बार 1 लिखा है = 11
i+2
दो बार 1 लिखा है = 21
i+3
एक बार 2 लिखा है &
एक बार 1 लिखा है
=1211


n=1
"1"
n=2
Call(n-1)
Call(1) = "1"
"1"
एक बार 1 लिखा है
11



solution

```
class Solution {
public:
    string countAndSay(int n) {
        if(n == 1)
            return "1";


        string say = countAndSay(n-1);

        string result = "";

        // Just count and store in result and return
        for(int i = 0; i<say.length(); i++) {

            int count = 1;
            char ch = say[i];

            while(i < say.length()-1 && say[i] == say[i+1]) {
                count++;
```

```
                i++;
            }

            result += to_string(count) + string(1, say[i]);

        }

        return result;

    }
};
```