Look and Say Sequence

1 2 3 4 5 6 7

1, 11, 21, 1211, 111221, 312211, 13112221

Q. We will be given an 'n' we have return the n^{th} term of Look and Say Sequence.

Let Understand Look and Say Sequence:

1, 11, 21, 1211, 111221, 312211

one 1

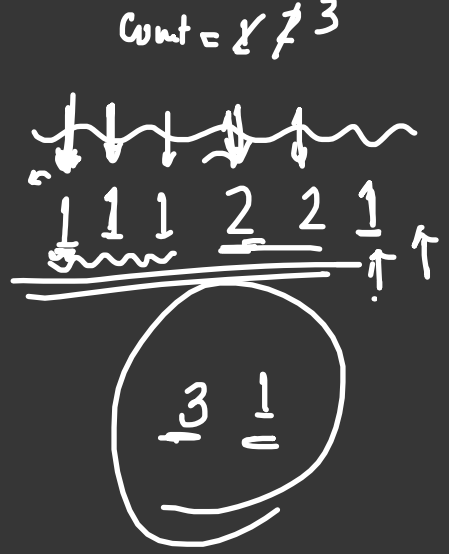
- We observing prev sequence and generating further sequence.

  We see how many time does a digit appear continuous and the digit

Approach:

if (n == 1) return "1";

- We have to the count the frequency of the prev character
- We have just append the count and the digit and reset the count



Pseudocode

```
temp = "";   count = 1;
for(i=1; i<n; i++)
if(s[i] != s[i-1]){
        char freq = count + '0';
        temp += freq;
        temp += s[i-1];   count = 1
}
cnt++;
```

0 1 2 3 4 5

prev 3 1 2 2 1 1

next =

cnt = 1

next = 1 3 1 1 2 2 2 1

for (i = 1; i < 6; i++)

if(s[i] != s[i-1])

char freq = (char)

nex

1

j = 1

1 1

2 1

1 1 #

count = 1 2

```
int count = 0;
string next = "";
for(int i=1; i<Seq.size(); i++){
    if(Seq[i] != Seq[i-1]){
        string freq = to_string(count);
        next+=freq;
        next.push_back(Seq[i-1]);
        count = 1;
    }
    count++;
}
string freq = to_string(count);
next+=freq;
next.push_back(Seq[Seq.size()-1]);
return next;
```

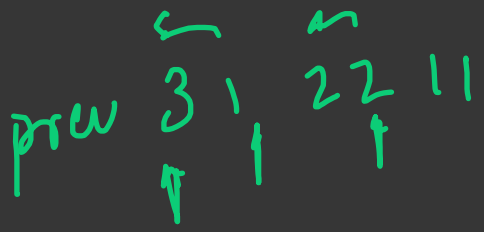
Seq = "21" next =

i = 1 count = 1

1

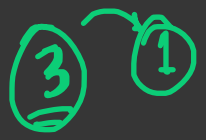
dp

vector string



curr =

1



```
string ans="";
    int count=1;
    ans.push_back(static_cast<char>(1+'0'));
    while(count<n){
        count++;
        int digit=0;
        string result="";
        for(int i=0;i<ans.length();i++){
            char num=ans[i];
            digit++;
            if(ans[i]!=ans[i+1]){
                result.push_back(static_cast<char>(digit+'0'));
                result.push_back(num);
                digit=0;
            }
            else continue;
        }
        ans=result;
    }
    return ans;
```

ans = "1" 1

count < n,

count++,

digit = 0/1 1

string res = "";

count = 1 2 cnt

1 1 1

num 1

if (ans[i] != ans[i+1]

This forward iterating

1 2 1 1

1 1