Count - and - Say

*Description:*

The *count - and - say sequence* is the sequence of integers beginning as follows:

1, 11, 21, 1211, 111221, ...

* 1 is read off as ‘one 1’ or 11.
* 11 is read off as ‘two 1s’ or 21.
* 21 is read off as ‘one 2’, then ‘one 1’.
* 1211 is read off as ‘one 1’, ‘one two’, ‘two one’.
* 111221 is read off as ‘three 1’, two two, one one.
* 312211 is read off as ‘one 3’, ’one 1’, ‘two 2’, ‘two 1’.
* 13112221 is read off as ‘one 1’, ‘one 3’, ‘two 1’, ‘three 2’, ‘one 1’.
* 11213211 is read off as ‘two 1’, ‘one 2’, ‘one 1’, ‘one 3’, ‘one 2’, ‘two 1’.

*Given an integer n, generate nth sequence.*

*Note:*

The sequence of integer will be represented as a string.

*class Solution {*

*public:*

*string countAndSay(int n) {*

*char a[] = {'1','2','3','4','5','6','7','8','9'};*

*map<int, string> vec;*

*vec[1] = "1";*

*char num = 0;*

*int count = 0;*

*for ( int i = 2; i <= n; i ++ )*

*{*

*string origin = vec[i-1];*

*string translate;*

*num = origin[0];*

*count = 1;*

*int length = origin.length();*

*int j = 1;*

*while (j < length)*

*{*

*if (origin[j] == num) {*

*count ++;*

*}*

*else {*

*translate = translate + a[count - 1];*

*translate = translate + num;*

*num = origin[j];*

*count = 1;*

*}*

*j++;*

*}*

*translate = translate + a[count - 1];*

*translate = translate + num;*

*vec[i] = translate;*

*}*

*return vec[n];*

*}*

*};*