# 443. String Compression

Given an array of characters chars, compress it using the following algorithm:

Begin with an empty string s. For each group of consecutive repeating characters in chars:

If the group's length is 1, append the character to s.

Otherwise, append the character followed by the group's length.

The compressed string s should not be returned separately, but instead, be stored in the input character array chars. Note that group lengths that are 10 or longer will be split into multiple characters in chars.

After you are done modifying the input array, return the new length of the array.

You must write an algorithm that uses only constant extra space.

## SOLUTION IN C++

class Solution {

public:

int compress(vector<char>& chars) {

int ans = 0;

for (int i = 0; i < chars.size();) {

const char letter = chars[i];

int count = 0;

while (i < chars.size() && chars[i] == letter) {

++count;

++i;

}

chars[ans++] = letter;

if (count > 1)

for (const char c : to\_string(count))

chars[ans++] = c;

}

return ans;

}

};