String Compression in C++

```
#include <iostream>
#include <string>
using namespace std;

string compression1(string str) {
    if (str.empty()) return ""; // Handle edge case

    string s;
    s += str[0]; // Append the first character directly
    for (int i = 1; i < str.length(); i++) {
        char curr = str[i];
        char prev = str[i - 1];
        if (curr != prev) {
            s += curr; // Append only if current character is
different from previous
        }
    }
    return s;
}

string compression2(string str) {
    if (str.empty()) return ""; // Handle edge case

    string s;
    s += str[0]; // Append the first character directly
    int count = 1;
    for (int i = 1; i < str.length(); i++) {
        char curr = str[i];
        char prev = str[i - 1];
        if (curr == prev) {
            count++; // Increment count for consecutive
characters
        } else {
            if (count > 1) {
                s += to_string(count); // Append count if it's
greater than 1
                count = 1; // Reset count
            }
            s += curr; // Append current character
        }
    }
    if (count > 1) {
        s += to_string(count); // Append the final count if
needed
    }
    return s;
}

int main() {
    string str = "wwwwaaadexxxxxx";
    cout << compression1(str) << endl;
    cout << compression2(str) << endl;
    return 0;
}
```

Step-by-Step Dry Run: compression2("wwwwaaadexxxxxx")

i	curr	prev	count	Output so far	Action
1	w	w	2	w	same, count++
2	w	w	3	w	same, count++
3	w	w	4	w	same, count++
4	a	w	1	w4a	append 4, then a
5	a	a	2	w4a	same, count++
6	a	a	3	w4a	same, count++
7	d	a	1	w4a3d	append 3, then d
8	e	d	1	w4a3de	different, append e
9	x	e	1	w4a3dex	append x
10	x	x	2	w4a3dex	same, count++
11	x	x	3	w4a3dex	same, count++
12	x	x	4	w4a3dex	same, count++
13	x	x	5	w4a3dex	same, count++
14	x	x	6	w4a3dex	same, count++
end				w4a3dex6	append 6

🖳 **Final Output**

```
wadex
w4a3dex6
```

wadex
w4a3dex6