


Remove adjacent duplicate in C++																															
<pre>#include <iostream> #include <stack> #include <string> using namespace std; string removeAdjacentDuplicates(string s) { stack<char> st; for (char ch : s) { if (!st.empty() && st.top() == ch) { st.pop(); } else { st.push(ch); } } string result = ""; while (!st.empty()) { result = st.top() + result; st.pop(); } return result; } int main() { string s = "abbaca"; cout << removeAdjacentDuplicates(s) << endl; // Output: "ca" return 0; }</pre>		<p>Input:</p> <p>string s = "abbaca";</p> <p> Step-by-Step Stack Trace:</p> <table><tr><th>Step</th><th>Char</th><th>Stack (top to bottom)</th><th>Action</th></tr><tr><td>1</td><td>'a'</td><td>[a]</td><td>Push</td></tr><tr><td>2</td><td>'b'</td><td>[a, b]</td><td>Push</td></tr><tr><td>3</td><td>'b'</td><td>[a]</td><td>'b' == top → Pop</td></tr><tr><td>4</td><td>'a'</td><td>[]</td><td>'a' == top → Pop</td></tr><tr><td>5</td><td>'c'</td><td>[c]</td><td>Push</td></tr><tr><td>6</td><td>'a'</td><td>[c, a]</td><td>Push</td></tr></table> <p>✔ Final Stack (bottom to top): c a</p> <p>So result = "ca".</p>		Step	Char	Stack (top to bottom)	Action	1	'a'	[a]	Push	2	'b'	[a, b]	Push	3	'b'	[a]	'b' == top → Pop	4	'a'	[]	'a' == top → Pop	5	'c'	[c]	Push	6	'a'	[c, a]	Push
Step	Char	Stack (top to bottom)	Action																												
1	'a'	[a]	Push																												
2	'b'	[a, b]	Push																												
3	'b'	[a]	'b' == top → Pop																												
4	'a'	[]	'a' == top → Pop																												
5	'c'	[c]	Push																												
6	'a'	[c, a]	Push																												
ca																															