

Postfix 2 Prefix in C++																																		
<pre>#include &lt;iostream&gt; #include &lt;stack&gt; using namespace std;  // Function to convert a postfix expression to a prefix expression. string postToPre(string exp) {     stack&lt;string&gt; op;     for (int i = 0; i &lt; exp.length(); i++) {         char ch = exp[i];         if (ch == '+'    ch == '-'    ch == '*'    ch == '/') {             string val2 = op.top();             op.pop();             string val1 = op.top();             op.pop();             string cal = ch + val1 + val2;             op.push(cal);         } else {             op.push(string(1, ch));         }     }     return op.top(); }  int main() {     string postfix1 = "ab+c*";     cout &lt;&lt; "Postfix: " &lt;&lt; postfix1 &lt;&lt; " -&gt; Prefix: " &lt;&lt;     postToPre(postfix1) &lt;&lt; endl; // Expected: "+abc"      return 0; }</pre>		<p>Input:</p> <p>Postfix Expression = "ab+c*" Expected Prefix = "+abc"</p> <p>📌 Dry Run Table:</p> <table><tr><th>i</th><th>ch</th><th>Stack Before</th><th>Action</th><th>Stack After</th></tr><tr><td>0</td><td>'a'</td><td>[]</td><td>Operand → push "a"</td><td>["a"]</td></tr><tr><td>1</td><td>'b'</td><td>["a"]</td><td>Operand → push "b"</td><td>["a", "b"]</td></tr><tr><td>2</td><td>'+'</td><td>["a", "b"]</td><td>Operator → pop "b", "a" → form +ab, push it</td><td>["+ab"]</td></tr><tr><td>3</td><td>'c'</td><td>["+ab"]</td><td>Operand → push "c"</td><td>["+ab", "c"]</td></tr><tr><td>4</td><td>'*'</td><td>["+ab", "c"]</td><td>Operator → pop "c", "+ab" → form *+abc, push it</td><td>["+*+abc"]</td></tr></table>			i	ch	Stack Before	Action	Stack After	0	'a'	[]	Operand → push "a"	["a"]	1	'b'	["a"]	Operand → push "b"	["a", "b"]	2	'+'	["a", "b"]	Operator → pop "b", "a" → form +ab, push it	["+ab"]	3	'c'	["+ab"]	Operand → push "c"	["+ab", "c"]	4	'*'	["+ab", "c"]	Operator → pop "c", "+ab" → form *+abc, push it	["+*+abc"]
i	ch	Stack Before	Action	Stack After																														
0	'a'	[]	Operand → push "a"	["a"]																														
1	'b'	["a"]	Operand → push "b"	["a", "b"]																														
2	'+'	["a", "b"]	Operator → pop "b", "a" → form +ab, push it	["+ab"]																														
3	'c'	["+ab"]	Operand → push "c"	["+ab", "c"]																														
4	'*'	["+ab", "c"]	Operator → pop "c", "+ab" → form *+abc, push it	["+*+abc"]																														
		<p>✔ <b>Final Output:</b></p> <p>Prefix: *+abc</p>																																
Postfix: ab+c* -> Prefix: *+abc																																		