

## DSA\_College\postfix\_to\_infix.cpp

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  bool isOperand(char x)
5  {
6      return (x >= 'a' && x <= 'z') ||
7             (x >= 'A' && x <= 'Z');
8  }
9
10 string getInfix(string exp)
11 {
12     stack<string> s;
13
14     for (int i=0; exp[i]!='\0'; i++)
15     {
16         // Push operands
17         if (isOperand(exp[i]))
18         {
19             string op(1, exp[i]);
20             s.push(op);
21         }
22
23         // We assume that input is
24         // a valid postfix and expect
25         // an operator.
26         else
27         {
28             string op1 = s.top();
29             s.pop();
30             string op2 = s.top();
31             s.pop();
32             s.push("(" + op2 + exp[i] +
33                   op1 + ")");
34         }
35     }
36
37     // There must be a single element
38     // in stack now which is the required
39     // infix.
40     return s.top();
41 }
42
43 int main()
44 {
45     string exp = "ab*c+";
46     cout << getInfix(exp);
47     return 0;
48 }
```