

```

#include <iostream>

#include <stack>

using namespace std;

string infixToPostfix(string infix) {
    stack<char> s;
    string postfix = "";
    for (char ch : infix) {
        if (isdigit(ch)) {
            postfix += ch;
        } else if (ch == '(') {
            s.push(ch);
        } else if (ch == ')') {
            while (s.top() != '(') {
                postfix += s.top();
                s.pop();
            }
            s.pop();
        } else {
            while (!s.empty() && (s.top() == '*' || s.top() == '/' || (ch == '+' || ch == '-') && (s.top() == '+' || s.top() == '-'))) {
                postfix += s.top();
                s.pop();
            }
            s.push(ch);
        }
    }
    while (!s.empty()) {
        postfix += s.top();
        s.pop();
    }
    return postfix;
}

int evaluatePostfix(string postfix) {
    stack<int> s;
    for (char ch : postfix) {

```

```

    if (isdigit(ch)) s.push(ch - '0');
    else {
        int b = s.top(); s.pop();
        int a = s.top(); s.pop();
        if (ch == '+') s.push(a + b);
        else if (ch == '-') s.push(a - b);
        else if (ch == '*') s.push(a * b);
        else if (ch == '/') s.push(a / b);
    }
}

return s.top();
}

int main() {
    string infix;
    cout << "Enter infix expression: ";
    cin >> infix;
    string postfix = infixToPostfix(infix);
    cout << "Postfix: " << postfix << endl;
    int result = evaluatePostfix(postfix);
    cout << "Result: " << result << endl;
    return 0;
}

```