#include <iostream>  
#include <stack>  
#include <string>  
#include <cctype>

using namespace std;

int precedence(char op) {  
    if (op == '\*' || op == '/') return 2;  
    if (op == '+' || op == '-') return 1;  
    return 0;  
}

int evaluate(int a, int b, char op) {  
    switch (op) {  
        case '+': return a + b;  
        case '-': return a - b;  
        case '\*': return a \* b;  
        case '/': return a / b;  
        default: return 0;  
    }  
}

int calculate(string expression) {  
    stack<int> operands;  
    stack<char> operators;

    for (int i = 0; i < expression.length(); i++) {  
        char c = expression[i];

        if (isdigit(c)) {  
            int num = 0;  
            while (isdigit(c)) {  
                num = num \* 10 + (c - '0');  
                i++;  
                c = expression[i];  
            }  
            i--;  
            operands.push(num);  
        } else if (c == '(') {  
            operators.push(c);  
        } else if (c == ')') {  
            while (operators.top() != '(') {  
                int b = operands.top(); operands.pop();  
                int a = operands.top(); operands.pop();  
                char op = operators.top(); operators.pop();  
                operands.push(evaluate(a, b, op));  
            }  
            operators.pop();  
        } else if (c == '+' || c == '-' || c == '\*' || c == '/') {  
            while (!operators.empty() && precedence(c) <= precedence(operators.top())) {  
                int b = operands.top(); operands.pop();  
                int a = operands.top(); operands.pop();  
                char op = operators.top(); operators.pop();  
                operands.push(evaluate(a, b, op));  
            }  
            operators.push(c);  
        }  
    }

    while (!operators.empty()) {  
        int b = operands.top(); operands.pop();  
        int a = operands.top(); operands.pop();  
        char op = operators.top(); operators.pop();  
        operands.push(evaluate(a, b, op));  
    }

    return operands.top();  
}

int main() {  
    string expression;  
    cout << "Enter a expression: ";  
    getline(cin, expression);

    int result = calculate(expression);  
    cout << "Result: " << result << endl;

    return 0;  
}