package calculator;

import java.util.Scanner;

import java.util.\*;

public class Calculator {

private static final char add = '+', minus = '-';

private static final char multiply = '\*', divide = '/';

private boolean isOperator(char i) {

return i == '+' || i == '-' || i == '\*' || i == '/' || i == '(' || i==')';

}

private boolean isEmpty(char i) {

return (i == ' ');

}

private boolean lowerPrecedence(char op1, char op2) {

switch (op1) {

case '+':

case '-':

return !(op2=='+' || op2=='-') ;

case '\*':

case '/':

return op2=='^' || op2=='(';

case '(': return true;

default:

return false;

}

}

public String convertToPostfix(String infix) {

Stack operatorStack = new Stack();

char i;

StringTokenizer a = new StringTokenizer(infix,"+-\*/^() ",true);

StringBuffer postfix = new StringBuffer(infix.length());

while (a.hasMoreTokens()) {

String token = a.nextToken();

i = token.charAt(0);

if ( (token.length() == 1) && isOperator(i) ) {

while (!operatorStack.empty() &&

!lowerPrecedence(((String)operatorStack.peek()).charAt(0), i))

postfix.append(" ").append((String)operatorStack.pop());

if (i==')') {

String operator = (String)operatorStack.pop();

while (operator.charAt(0)!='(') {

postfix.append(" ").append(operator);

operator = (String)operatorStack.pop();

}

}

else

operatorStack.push(token);

}

else if ( (token.length() == 1) && isEmpty(i) ) {

}

else {

postfix.append(" ").append(token);

}

}

while (!operatorStack.empty())

postfix.append(" ").append((String)operatorStack.pop());

return postfix.toString();

}

public int evaluate(String expr) {

Stack stack = new Stack();

int op1, op2, result = 0;

String token;

StringTokenizer tokenizer = new StringTokenizer(expr);

while (tokenizer.hasMoreTokens()) {

token = tokenizer.nextToken();

char c = token.charAt(0);

if (isOperator(c)) {

op2 = ((Integer) stack.pop()).intValue();

op1 = ((Integer) stack.pop()).intValue();

result = evalSingleOp(token.charAt(0), op1, op2);

stack.push(new Integer(result));

}

else

stack.push(new Integer(Integer.parseInt(token)));

}

result = ((Integer) stack.pop()).intValue();

return result;

}

private int evalSingleOp(char operation, int op1, int op2) {

int result = 0;

switch (operation) {

case add :

result = op1 + op2;

break;

case minus :

result = op1 - op2;

break;

case multiply :

result = op1 \* op2;

break;

case divide :

result = op1 / op2;

}

return result;

}

public static void main(String[] args) {

Scanner sn = new Scanner(System.in);

String Infix;

System.out.print("Enter Infix: ");

Infix = sn.next();

String[] testString = {Infix};

Calculator converter = new Calculator();

for (int i=0; i<testString.length; i++) {

System.out.println("Infix is : " + testString[i]);

System.out.println("The result is : " + converter.evaluate( converter.convertToPostfix(testString[i])));

}

}

}