import java.io.\*;

import java.util.\*;

// 2+3\*3

public class Infix{

public static void main(String[] args) throws Exception {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

String exp = br.readLine();

Stack<Integer> operand=new Stack<>();

Stack<Character> operator=new Stack<>();

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

{

char ch=exp.charAt(i);// 2+4\*7

if(ch=='(')

{

operator.push(ch);

}

else if(Character.isDigit(ch))

{

operand.push(ch-'0');//+,\*....

}

else if(ch==')')//(2\*7+3)

{

while(operator.peek()!='(')

{

char ch1=operator.pop();

int val2=operand.pop();

int val1=operand.pop();

int v=opeartion(val1,val2,ch1);

operand.push(v);

}

operator.pop();

}

else if(ch=='+'||ch=='-'||ch=='\*'||ch=='/')

{

while(operator.size()>0&&operator.peek()!='(' && precedences(ch)<=precedences(operator.peek()) )

{

char ch1=operator.pop();

int val2=operand.pop();

int val1=operand.pop();

int v=opeartion(val1,val2,ch1);

operand.push(v);

}

operator.push(ch);

}

}

while(operator.size()!=0)

{

char cch=operator.pop();

int val2=operand.pop();

int val1=operand.pop();

int opv=opeartion(val1,val2,cch);

int y=opeartion(val1,val2,cch);

operand.push(y);

}

System.out.println(operand.peek());

}

public static int precedences(char op)

{

if(op=='+')

{

return 1;

}else if(op=='-')

{

return 1;

}

else if(op=='\*')

{

return 2;

}else{

// divide

return 2;}

}

public static int opeartion(int val1, int val2, char op)

{

if(op=='+' )

{

return val1+val2;

}

else if(op=='-')

{

return val1-val2;

}else if(op=='\*')

{

return val1\*val2;

}else

{

return val1/val2;

}

}

}