//

#include "pch.h"

#include<iostream>

#include<stack>

#include<string>

using namespace std;

int judge(char a); //judge函数声明，用于判断数据类型

int judge(char a) //judge函数定义

{

switch (a)

{

case '\*':

case 'x':

case 'X':

case '/': return 3;

case '-':

case '+': return 2;

case '(': return 1;

case ')': return -2;

default:return -1;

}

} //可用于优先级

int main()

{

string str;

cout << "请不要使用全角符号和中文符号" << endl;

cout << "请勿输入小数，多位数，以及不要省略乘号" << endl;

checkagain: // checkagain为goto语句标号，在每次捕获异常后都要进行重新输入数据

cout << "请输入表达式：";

cin >> str; //保存外部输入的数据，即表达式

int len = str.size(); //外部数据长度

/\*表达式初步检查\*/

try {

cout << "表达式初步检查..." << endl;

int count0 = 0;

double checkdouble = 0;

int checkbracket = 0;

char support='0';

long checktwo = 0;

float checkdiv = 0;

for (int e = 0; e < len; e++)

{

if (str[e] == '(')

{

count0++;

if (e != 0)

if (str[e - 1] >= '0'&&str[e - 1] <= '9')

throw support; //不支持省略乘号，比如2(3+5) char

}

if (str[e] == ')')

{

count0--;

if (e + 1 < len)

if (str[e + 1] >= '0'&&str[e + 1] <= '9')

throw support; //不支持省略乘号，比如(3+5)2 char

}

if (str[e] == '.')

throw checkdouble; //不支持小数，如1.2 double

if (str[e] >= '0'&&str[e] <= '9')

{

if (str[e + 1] >= '0'&&str[e + 1] <= '9')

throw checktwo; //不支持两位数及以上，如16 long

}

if (str[e] == '/')

if (str[e + 1] == '0')

throw checkdiv; //检查除数为0 float

}

if (count0 != 0)

throw checkbracket; //检查括号是否配对 int

}

catch (int)

{

cout << "括号不配对，请重新输入" << endl;

goto checkagain;

}

catch (char)

{

cout << "不支持省略乘号，请重新输入" << endl;

goto checkagain;

}

catch (double)

{

cout << "不支持小数，请重新输入" << endl;

goto checkagain;

}

catch (long)

{

cout << "不支持两位数及以上，请重新输入" << endl;

goto checkagain;

}

catch (float)

{

cout << "被除数不允许为0，请重新输入" << endl;

goto checkagain;

}

catch (...)

{

cout << "出现未知异常" << endl;

goto checkagain;

}

cout << "表达式检查完毕." << endl;

/\*表达式检查完毕\*/

//考虑到除号后面可能出现的 由int型数据作除法 除数为 表达式中int型计算得到的整型数据0(比如2/(2-5/2)会出现运算错误)，故本程序操作数的存储和计算数据均用double型数据

//本程序无法对类似 2/(2-2) 的运算式做检查

union uu { double d; char c; }; //定义共用体，用于不定型数据的存储：，操作数或者运算符

uu \*u = new uu[len]; //由外部的相关数据来确定申请的内存大小 //申请共用体的动态内存,u为指向其首地址的指针

int j = 0; //结构体组下标

stack<char> s; //临时存放运算符

stack<double> dd; //用于最后数学计算

for (int i = 0; i < len; i++) //扫描合法字符串表达式

{

if (judge(str[i]) == -2) //遇到右括号

{

for (;;) //"死"循环，但是合法表达式中一定有左括号

{ //处理括号之间的运算符

if (s.top() == '(') break;

u[j].c = s.top();

s.pop();

j++; //j为共用体单独计数，每存储一个数据，下标(指针)移动一位

}

s.pop(); //删除左括号

}

if (judge(str[i]) == -1) //遇到数字

{

u[j].d = (double)(str[i] - 48); //将字符型数字转化为10进制double数据

j++;

}

if (judge(str[i]) > 0) //遇到运算符

{

if (str[i] == '(') //优先判断是否为左括号

{

s.push('('); continue;

} //左括号入栈，立即进入下一轮循环

if (s.empty()) //栈空 则入操作符

s.push(str[i]);

else //遇到运算符但栈不空

{

while (judge(s.top()) - judge(str[i]) >= 0)//优先级判断并做相应处理。此处为出栈处理。

{

u[j].c = s.top();

s.pop();

j++;

if (s.empty()) //如果栈空，即运算符全部出栈，则退出while循环

break;

}

s.push(str[i]); //此处为入栈处理。

}

}

}

while (!s.empty()) //将栈中运算符元素全部出栈，直至空。

{

u[j].c = s.top();

s.pop();

j++;

}

double r; //中缀表达式 转 后缀表达式 已全部转换完成，进入后缀表达式运算阶段

for (int z = 0; z < j; z++) //用j(最终所使用的共用体个数，即后缀表达式数据的个数)做控制

{

if (u[z].c == '\*' || u[z].c == 'x' || u[z].c == 'X')

{

r = dd.top();

dd.pop();

dd.top() = dd.top()\*r;//先取出的数据放在后缀表达式的右边

//数栈顶是每步运算后的结果

}

else if (u[z].c == '/')

{

r = dd.top();

dd.pop();

dd.top() = dd.top() / r;

}

else if (u[z].c == '-')

{

r = dd.top();

dd.pop();

dd.top() = dd.top() - r;

}

else if (u[z].c == '+')

{

r = dd.top();

dd.pop();

dd.top() = dd.top() + r;

}

else { r = u[z].d; dd.push(r); }//将后缀表达式中的数据取出来，入数栈。由于共用体不能作为函数参数，所以用变量r来作为中间值，进行传参

}

cout << "答案是" << dd.top() << endl;

return 0;

}