project1实验内容和要求

A SIMPLE CALCULATOR

实验目的

1. 利用Stack开发一款简单的表达式求值软件。
2. 熟练掌握Stack的逻辑结构、存储结构、基本操作
3. 灵活运用Stack的各种基本操作
4. 熟练掌握如何编辑、编译、链接和运行一个C++程序;

实验内容及要求（功能）

1. **友好的用户界面**，给出简单用户帮助
2. 通过键盘输入表达式，表达式可包含加(+)、减(-)、乘(*)、除(/)、求模（%） 、开方(&)和乘方(^) 运算，并能使用括号，最后请以“=”结束

   Ex.　-2*3^5 + 23 / (45+67) - 17%3/8&2
3. **计算表达式，并在显示器上输出结果**

程序思路（建议）

1. 建立两个栈（optr和opnd），分别用来存储运算符和操作数。
2. 通过键盘输入一个表达式，如：-2*3^5 + 23 /（45+67）- 17%3/8&2
3. 逐个判断表达式中的字符为数字或小数点(0-9, .), 运算符 (+,-,*,/,%,^,&,(,))还是其他非法字符,直到碰到”=”字符
   ① 若为数字,将其push进操作数栈(opnd), 继续处理下一个字符.
   ② 若为运算符,比较此运算符的栈外优先级与运算符栈(optr)中最顶端元素的栈内优先级的高低,
      a) 若栈内优先级低于栈外优先级,则将当前字符push入运算符栈(optr), 继续处理下一个字符
      b) 若栈内优先级高于栈外优先级, 从操作数栈(opnd)中pop出2个操作数,从运算符栈(optr)中pop出最顶端运算符进行计算,并将计算结果push进操作数栈(opnd) ,继续处理当前字符
      c) 若栈内优先级等于栈外优先级, 从运算符栈(optr)中pop出最顶端运算符, 继续处理下一个字符.
   ③ 若为其他字符, 输出提示表达式非法, 退出程序.
4. 输出最终运算结果,即操作数栈(opnd)中的栈顶元素.

栈内/栈外优先级表

运算符	栈内优先级	栈外优先级
=	0	0
+	3	2
-	3	2
*	5	4
/	5	4
%（求模）	5	4
(	1	8
)	8	1
^（乘方）	7	6
&（开方）	7	6

注：值越大，优先级越高

表达式中数字的处理（建议）

- 若读入字符为数字或小数点,用下列2种方案读入操作数：
  - 方案一：

```
cin.putback(ch);
cin>>operand;
```

  - 方案二：

① 若为数字，将其数值修正为double型并放入到栈中，再判断下一个字符是否为数字，若又为数字则取出前一个数将它们转换为一个两位数后再放入到栈中，直到出现非数字或小数点字符；

② 若为小数点字符，则按照小数转换原则将后续出现的数字与栈中前一个数进行转换后再放入到栈中，直到出现非数字字符。

主要函数（建议）

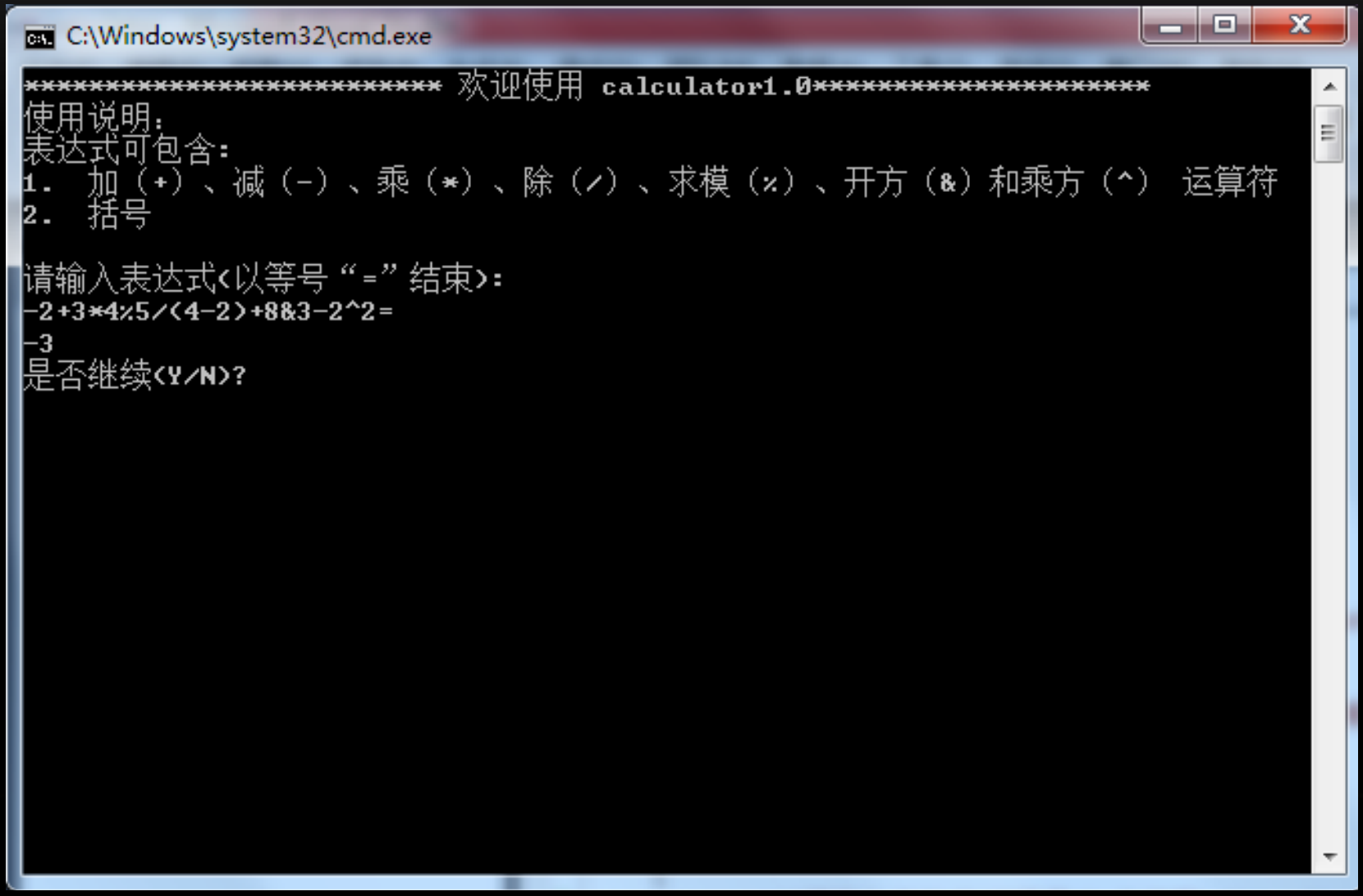
- int isp（char ch)　//获取并返回操作符 ch 的栈内优先级
- int osp（char ch)　// 获取并返回操作符 ch 的栈外优先级
- bool cal(char op, double x, double y, double & r)　// 计算r = x op y， 计算成功，返回true.
- void GetNextchar(char & ch)　// 从输入的表达式中获取一个字符
- bool isdigit(char ch)　// 判断ch是否为数字0-9
- bool IsOperator(char ch)　//判断ch是否为操作符
- bool Get2Operands(LStack & opnd，double &x, double &y)　//从操作数栈中取2个操作数

- 关于Stack操作，可选择AStack，也可选择LStack， 将其内容放入一个.h文件并包含在main函数所在的.cpp文件

MAIN函数框架（建议）

```
void main ( ) {
LStack<double> OPND;   // 操作数栈定义；
LStack<char>   OPTR;   // 操作符栈定义；
……
OPTR.push('='); prior_char = '=';   //prior_char 表示当前处理字符的前一个字符，如为数，则其值'0'
GetNextchar(ch); OPTR.topValue(OPTR_top);
while(OPTR_top!= '=' || ch != '=' ) {
        if(isdigit(ch)||ch=='.')  {
              cin.putback(ch);   cin>>operand;   OPND.push(operand);      prior_char='0'; GetNextChar(ch); }
        else if(!IsOperator(ch))  {  cout<<"表括达式中出现非法字符!"<<endl; return; }
        else {
            if(( prior_char=='='||prior_char=='('&&(ch=='+'||ch=='-'))   OPND.push(0);   //?????
            if (栈内优先级低于栈外优先级)   { …… }
            else if(栈内优先级高于栈外优先级)   {……}
            else  {……} }
        if ( ! OPTR.topValue(OPTR_top ) {    cout<<"表括达式有错!"<<endl; return; }   }
If (OPND.length() != 1)  {  cout<<"表括达式有错!"<<endl; return; }
输出结果;    }
```

用户界面（建议）


C:\Windows\system32\cmd.exe
********** 欢迎使用 calculator1.0**********
使用说明:
表达式可包含:
1. 加（+）、减（-）、乘（*）、除（/）、求模（%）、开方（&）和乘方（^） 运算符
2. 括号
请输入表达式(以等号“=”结束):
-2+3*4%5/(4-2)+8&3-2^2=
-3
是否继续(Y/N)?

运行演示

calculator.exe

分组，程序验收及报告提交

- 5-6人一组，自由组合，以组为单位编程，测试并完成实验报告
- 以组为单位进行程序验收： 50
  - 验收时间：第11周实验课
  - 验收步骤：1）现场编译运行程序 2）当场删掉几行程序，要求在5分钟内补齐再编译运行
- 以组为单位提交项目报告及项目源程序：20%，
  - 命名格式 ： 项目1_项目报告_第X组；
  - 项目1_源程序_第X组
- 每个同学完成一份个人总结报告： 30%（其中5%为实验课考勤）
  - 命名格式 ：项目1_个人总结_XXX
- 项目报告，源程序及个人报告由组长统一收齐后打包提交
  - 提交时间：12周实验课前
  - 提交地点：课程QQ群（以作业的形式）

关于成绩说明

- 按期完成程序验收（即运行结果正确），得满分50（组内成员统一），延期一周扣5分， 2周扣10分，以此类推。未完成（包括验收中运行结果不正确）0分
- 报告按期提交，内容及结果分析完整充分，得满分20（组内成员统一），不够充分完整酌情扣5-10分。延期1周扣2分，2周扣4分，以此类推。未提交0分
- 个人报告满分25（按个人计分），按照个人总结中内容确定。
  - 注： 若与小组报告中的分工不一致，会导致严重扣分。
  - 未提交 0分
- 实验考勤 5分（按个人计分）

关于项目报告内容

- 实验目的
- 实验内容
- 小组成员及分工
  - 详细说明每个成员做了什么，比如编写了那些函数，撰写报告中那部分内容等
  - ”各部分都是大家一起完成的” 这样笼统的说法不可以
- 程序主框架（主函数的流程图）
- 各子函数详细描述（流程图）
- 运行结果及分析

关于个人总结内容要求

- 个人信息
- 个人分工
  - 说明个人在项目中做了什么，要与小组报告中的分工保持一致。（建议直接copy小组报告中关于自己的那部分分工，不要擅自将别人的工作说成自己的）
- 个人工作中碰到的问题及解决思路
  - 详细描述在完成个人分工的时候碰到的问题，比如撰写的函数实际运行结果不是预期结果（可给出预期及你的结果截图），或者压根跑不起（可给出运行截图））等， 并详细说明自己是如何根据所学知识去分析为什么会出现这样的问题以及你是如何一步步解决这些问题的。
  - 个人总结

    从这次项目你学到了什么