实 验 报 告

姓 名：黄俊杰 学 号：13331087

院系专业：软件学院13级教务4班

完成日期：2014 年10 月26 日

实验题目：完整计算器

◆ 需求分析：

以无歧义的陈述说明程序设计的任务，强调的是程序要做什么？并明确规定：

用户输入测试样例数，每个测试样例输入一个正确的四则混合运算表达式（可以含有负数、小数、括号）程序需要计算这个表达式的计算结果。

◆ 概要设计

说明本程序中用到的所有抽象数据类型的定义、主程序的流程以及各程序模块之间的层次(调用)关系。

抽象数据类型定义：

string str0: 每个测试样例用户输入的中缀表达式。可以含有负数、小数、括号。
string str：str0的无空格版。方便函数处理用。

queue<string> RPN: RPN即逆波兰记数法（Reverse Polish Notation）的缩写。所以该队列是用以存储对应测试样例表达式的逆波兰表达式。

stack< pair<char, int> > operators: 在把中缀表达式处理为逆波兰表达式的时候用以临时存放运算符的堆栈。仅在函数getRPN()中使用。

string str_temp: 函数getRPN()中临时存放处理中的中缀表达式的字符串。当一个数字或者运算符被提取以进行处理的时候，该数字或字符将不在存储在str_temp中。

char op: 函数getRPN()中被提取的运算符。可以是+、-、*、/、(、)。
pair<char, int> new_op: 函数getRPN()中给当前运算符赋予优先级的有序对。其中的first元素是运算符，second元素是其优先级。用以处理时和operators栈顶元素进行优先级比较。

double left、 double right： 函数Calculate()中运算的左右操作数。

函数模块：

string to_string(double number)：把浮点数处理成字符串。
double to_double(string str)：把字符串处理成浮点数。
以上两个函数由于sicily不允许使用std::to_string()和std::stod()函数，故自定义。均用于整个函数中可能存在的数字和字符串之间的相互转换。

string Prepocess(string str):
str: 用户给的中缀表达式（无空格版）
返回值：所有负数都加上了负数标记的字符串
该函数把表达式中的所有负数都加上了负数标记‘%’， 这样子给getRPN()分离数字和运算符的操作提供了方便。在函数getRPN()中被调用。

stack<string> getRPN(string str):

str: 用户给的中缀表达式（无空格版）

返回值：str对应的逆波兰表达式堆栈

该函数把str变成逆波兰表达式。

double Calculate(stack<string> RPN):

RPN: 函数getRPN()处理后得到的逆波兰表达式堆栈

返回值：RPN计算后得到的结果

主函数流程：

1） 用户输入测试样例数test_cases；
2） 判断test_cases。如果test_cases为正数，执行以下操作，否则退出主函数。
3） 用户输入一个正确的中缀表达式str；
4） 调用函数getRPN(str),得到一个逆波兰式堆栈RPN;
5） 调用函数Calculate(RPN),得到计算结果result;
6） 将result保留三位输出到屏幕中；
7） 将test_cases减一，表示已经处理了一个测试样例。
8） 重复2）- 7）。

◆ 调试分析

内容包括:

a. 调试过程遇到的问题与解决方案:

1） 当getRPN()处理str_temp的时候，由于使用了stringstream进行数字和运算符的操作，当处理到最后一个数字的时候，由于stringstream没有清理缓存使得函数重复将该数字提取和放入str_temp；而判断条件为str_temp为空时退出循环，导致进入了死循环。
解决方案：进行分离操作之前，在str_temp尾部加入结束标记'#'，并作为退出循环的判断条件。

2） 无法判定'-'是运算符减号亦或是负数符号
解决方案：预处理str_temp，在每个负数之前添加负数标记'%'。当getRPN()扫描到了一个负数标记时，分离出一个负数。那么，剩下的'-'即为减号。

3） getRPN()处理后得到的RPN含有括号，使得Calculate()函数无法正常运行而导致整个函数崩溃。
解决方案：分析代码后发现是由于运算符进入getRPN()中的operators堆栈的规则没有对左右括号进行特殊处理。修改代码，订立特殊处理规则后，程序运行正常。

b. 时间和空间复杂度:

I. 时间复杂度:

对于一个测试样例，假设有a个数，其中负数个数为b，m个四则运算符（a-1 = m），n对括号 a个数平均有x位（含小数点）

则：Prepocess复杂度：o(ax+b+m+n)

getRPN()复杂度：o(Prepocess)+(a+b)+7m+1 = o(a(x+1)+2b+8m+n) = o(a(x+8)+2b+n-8)

Calculate()复杂度：o(a+m+m*3m) = o(3(a-1)^2+2a-1) = o(3a^2-4a+2)

所以一个测试样例的总时间复杂度为o(a^2)

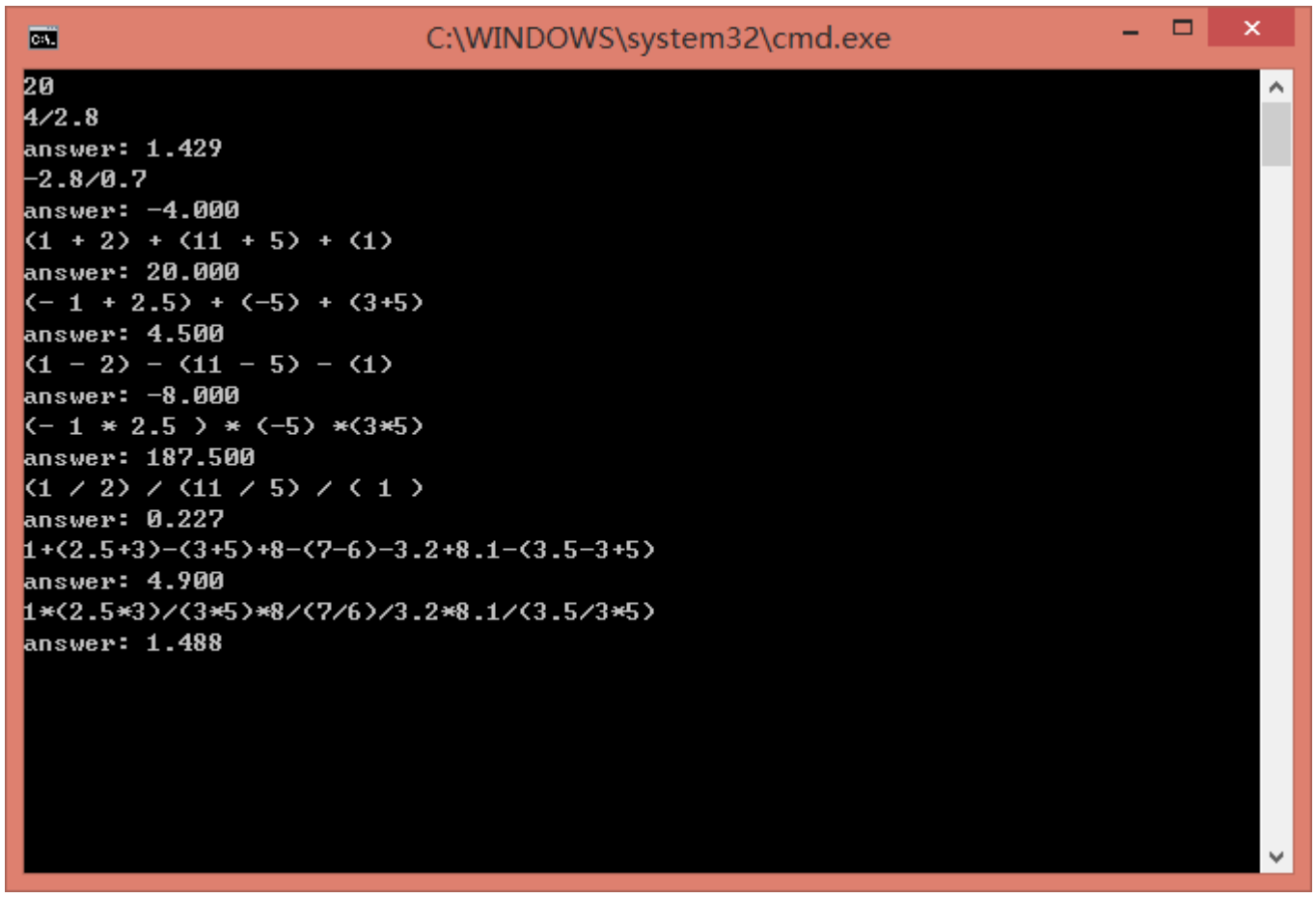
◆ 用户使用说明

说明如何使用你编写的程序，详细列出每一步的操作步骤。

1） 用户输入测试样例数test_cases， 用于计数；
2） 当一输入的测试样例总数小于等于test_cases，执行以下操作，否则退出主函数。
3） 用户输入一个正确的中缀表达式，并摁下回车；
4） 结果将保留三位输出到屏幕中。

◆ 测试结果

列出你的测试结果，包括输入和输出。这里的测试数据应该完整和严格，最好多于需求分析中所列。



◆ 实验心得

具体内容不作硬性要求

只有自己认真思考过、做过的事情，才能转化为自己的知识和经验。

◆ 附录

列出程序文件名的清单:

13331087_04.cpp