

# 中山大学移动信息工程学院本科生实验报告

(2015 学年秋季学期)

课程名称：软件设计 (I)

任课教师：叶士青 刘晓铭

教学助理 (TA):

年级	15 级	专业 (方向)	移动信息工程
学号	15352461	姓名	宗嘉希
电话	13676013364/653364	Email	zongjx@mail2.sysu.edu.cn
开始日期	2015. 12. 15	完成日期	2015. 12. 24

## 1、实验题目

计算器

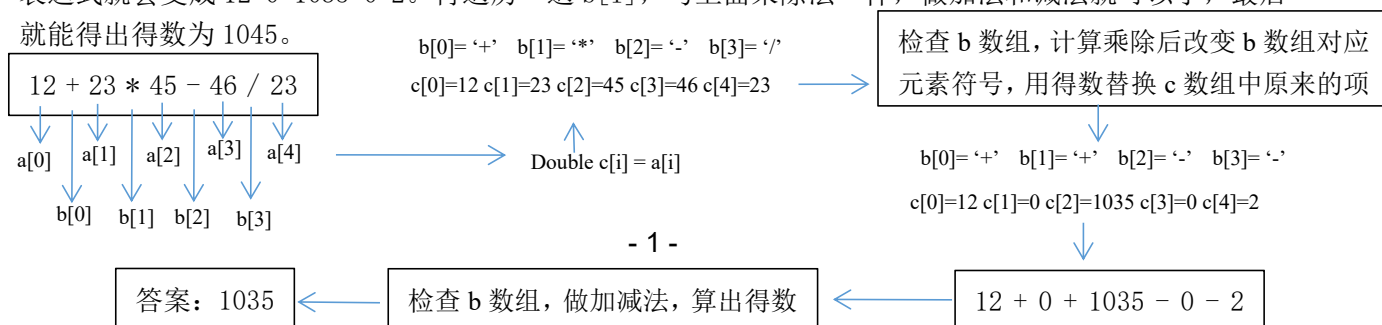
## 2、实验目的

- 1、能处理加减乘除混合运算，输入输出长度不超过 80 个字符，包括空格，不包含负数，每一步运算结果均不超出 1000。合法字符仅有 0 - 9、空格和 +、-、\*、/ 共 15 种字符。
- 2、输出能精确到小数点后两位。
- 3、能检测不是合法的运算表达式。
- 4、交互界面友好，例如有恰当的输入输出提示。

## 3、程序设计

设计思路：输入字符串（表达式）。第一步，检查输入的字符串中是否含有不合法的字符或者，如果有就直接输出“PE”，如果没有就进入第二步。第二步，把合法的表达式中多余的空格去掉，令表达式中只存在数字和四则运算符号。第三步，先用一个数组按顺序把表达式中的每一项数字以字符串形式存下来，用另一个数组把运算符按顺序存下来；再另开一个数组，把字符串转换为双精度的数。第四步，先遍历存放运算符的数组，按顺序先找出乘号和除号，然后在存放数字的数组中拿出相应的项做相应的运算，再遍历一遍再找出加号和减号，做加法和减法运算。

c 基础版流程 例：输入 12+23\*45-46/23，首先用一个数组 a 以字符串形式存下每个数，用另一个数组 b 以字符形式存下运算符，遍历一遍，当检查出是数字时，存进 a[k]，如果是运算符，就存进 b[k]，然后 k+1，再继续扫表达式。扫完以后就会有 a[0]=12, a[1]=23, a[3]=45, a[4]=46, a[5]=23, b[0]=+, b[1]=\*, b[2]=-, b[3]=/. 再把 a 数组每一项都转化为双精度型，存进 c[i]。完成后遍历 b 数组，当检查到 b[i]是\*或/是，就把 c[i]与 c[i+1]做\*或/运算，把结果放在 c[i+1]，替换掉原来的数，再把 c[i]变成 0，把 b[i]变成 b[i-1]的符号。完成这一步之后，表达式就会变成 12+0+1035-0-2。再遍历一遍 b[i]，与上面乘除法一样，做加法和减法就可以了，最后就能得出得数为 1045。



关于拓展版，和基础版相比，加多了友好交互界面，清屏功能，高精度的加法、减法和乘法。清屏功能其实并不难，只是判定是否有输入，如果没输入，按下回车后就会输出 20 多个换行符，使可视界面中的内容移走。高精度算法中，加法和乘法都比较容易。以下是伪代码，在程序中我集中成一个函数。

加法	乘法	减法
<pre>void Plus(string s) {     int size=s.length();     int a[101],b[101];     for(int i=0;i&lt;size;i++)     {         int size1;         用循环检查出运算符的位置, 记为size1     }     把字符串分为两串, 分别表示两个加数     for(int i=0;i&lt;size1;i++)     {         把第一个加数转换成int, 倒序转换     }     for(int i=size1+1;i&lt;size;i++)     {         把第二个加数转换成int, 倒叙转换     }     int c[i];     int e=0;     for(int i=0;i&lt;100;i++)     {         c[i]=a[i]+b[i];相加 进位计算     }     for循环输出结果 }</pre>	<pre>void Multiply(string s) {     int size=s.length();     int a[101],b[101];     for(int i=0;i&lt;size;i++)     {         int size1;         用循环检查出运算符的位置, 记为size1     }     把字符串分为两串, 分别表示两个乘数     for(int i=0;i&lt;size1;i++)     {         把第一个加数转换成int, 倒序转换     }     for(int i=size1+1;i&lt;size;i++)     {         把第二个加数转换成int, 倒叙转换     }     int c[i];     int e=0;     for(int i=0;i&lt;100;i++)     {         c[i]+=a[i]*b[i];相乘 进位计算     }     for循环输出结果 }</pre>	<pre>void Minus(string s) {     int size=s.length();     int a[101],b[101];     for(int i=0;i&lt;size;i++)     {         int size1;         用循环检查出运算符的位置, 记为size1     }     把字符串分为两串, 分别表示减数和被减数     for(int i=0;i&lt;size1;i++)     {         把第一个加数转换成int, 倒序转换     }     for(int i=size1+1;i&lt;size;i++)     {         把第二个加数转换成int, 倒叙转换     }     判断减数和被减数的位数a和b, 若a&gt;b则用a-b;若b&gt;a, 则用b-a;若a=b, 则判断两个的大小, 用大数减小数.     for循环输出结果 有必要时做正负判断并加上负号再输出结果 }</pre>

## 4、程序运行与测试

```

输入样例个数: 11
输入: 3+4
输出: 7.00
输入: 3+4.6
输出: PE
输入:      2 * 3
输出:      6.00
输入: 11/3
输出: 3.67
输入: 2*(3+4)
输出: PE
输入: 3+4*2*15-8/2
输出: 119.00
输入: 3+4*2* 1 5 -8/2
输出: PE
输入: 3+4a*2* 15 -8/2
输出: PE
输入: 1111111111111+111111111111111111111111
输出: 111111111122222222222222.00
输入: 123456789123456789123456789-987654321987654321987654321
输出: -864197532864197532864197532.00
输入: 1111111111111+111111111111111111111111
输出: 1234567901234555555555554320987654321.00

```

## 5、使用说明

poj1a.15352461.cpp:这是实现基础功能,针对输入输出的程序代码。先输入测试样例的个数  $N(N>0)$ ,再输入  $N$  行,每行代表一个表达式。

poj1a.15352461.cpp:这是给用户使用，有友好交互界面的程序，并且附帶了拓展功能的程序。每次直接输入一行字符串，代表运算表达式。拓展功能：（1）两个项间的高精度加法、减法和乘法运算。（2）清屏功能，不输入任何字符直接按下回车键，能够清屏。

## 6、实验总结与心得

在本次的实验中，我发现其实一个计算器的程序编写起来也不是很容易的事。首先要判断这个表达式是否合法，这就已经是很难的，因为我在写完程序以后经常会碰到输入了错误的表达式但输出却不是“PE”，或者有时候输入了正确的表达式，却输出了“PE”。在这个编写检查错误的程序的过程中，我分清楚了 string 类和 char 类以及他们的函数的使用，以前经常分不清楚，把这两个的函数搞混乱，现在就能分清楚了。第二步就是进行计算了。我是写运算的函数的，因此主函数看起来就非常短。在些计算函数的过程中，我学到了有很多种方法，比如说栈等等，但是我没有用那些深奥的方法，我只是用数组

去把数存下来再做运算，也让我能够更熟练地使用数组、字符串还有各种循环。除此之外，我还学会了用“bool”，还有我也认识到定义变量时不要随便定义，要用自己能看得懂的英文符号去定义，否则的话是很容易出错的。另外，我认识到注释是非常重要的，因为一开始的时候我没有加注释，结果后来在找错误的时候发现自己的代码很乱，根本找不到错误在哪，后来加上注释以后就清楚地把每一部分都分开了，结构也清晰多了。最后我觉得，在本次实验中的不足就是没能写出能判断小数和括号的程序，我觉得是很困难和很复杂的，于是就没有去实现这个功能，等将来学习更多再去实现。

## 附录、提交文件清单

poj1a.15352461.cpp:完成基础功能，针对标准输入输出的，方便做功能测试的程序代码。

poj1b.15352461.cpp:给用户使用，带有友好界面，并且附带拓展功能的程序代码。

in.txt:用于测试程序的输入的数据。

out.txt:输入测试程序的数据所输出的结果。