

CS205 C/ C++ Programming - Lab Assignment

Template

Name:舒子和

SID:12111727

Part 1 – Analysis

一、关于提交文件的描述：

1. main.cpp
2. BigFloat.cc
3. BigFloat.h

本次作业中的 BigFloat.cc 和 BigFloat.h 为网络上查找到的源码库，在本次作业中仅应用满足 Problem5 的需求。

二、关于 BigFloat 文件的使用说明：

程序中设置 mode2 来调用 BigFloat，将用户输入的字符串转化为可读的高精度数字，并重载了四则运算符号进行按位的加减乘除，得到高精度数字的字符串。

Conference: [Mariotti94/BigFloat: Arbitrary-Precision Floating Point Arithmetic Library \[C++\]\(github.com\)](https://github.com/Mariotti94/BigFloat)

作者:Mariotti94

三、关于 Requirements 需求的分析(困难点将在 Part4 进行分析)：

Requirement1:1、需要将用户输入进行整行读取并且分别存储数字和运算符。
2、需要符合四则运算规则的优先次序。
3、需要计算并打印出正确的结果。

Requirement2:1、增加对括号的读取。
2、对括号内的表达式进行强制优先计算。

Requirement3:1、需要对用户输入的定义自变量表达式进行识别。
2、需要存储用户定义的自变量的字母和数值并能够在后续计算中调用。
3、在需求一二的基础上进行计算。

Requirement4:1、需要将表达式中的数学函数进行识别并计算，输出结果为纯数字。
2、需要将数学函数计算值与原表达式兼容，统一计算表达。
数学函数上完成了:sqrt(),pow(),max(),min(),abs(),sin(),cos(),tan(),exp(),log(),以及完成了对常数 PI,E 的定义。

Requirement5:1、需要对用户输入的高精度数字进行读取并且不能改变精度。
2、需要对高精度的数字进行计算和打印且不能改变精度。

Requirement6:1、参照 calculator BC 增加了单行注释“#”和多行注释“/**/”的功能，

需要对输入注释符读取并使注释掉的表达式无效化。

Requirement7:本次作业都 git 上传至 github。

Part 2– Code

```
int main(){
//...
    cin.getline(line,100);
    double ans=calculateline(line);
    cout<<ans<<endl;
//...
}
```

本次作业的 main 函数功能主要是完成：

- 1、对用户输入的读取。
- 2、对用户输入表达式的合法性判断。
- 3、检测表达式中的注释并进行相应处理。
- 4、打印最终计算的值。

```
double calculateline(char line[]);
```

其中 calculateline 函数完成对用户输入整行表达式的一个计算。

Calculateline 函数包括但不限于：

```
bool is_operator(char op); //对表达式内符号是否为运算符的判断
int priority(char op); //对运算符的优先级进行赋值
double calculate(double a, double b, char op); //对两个数字一个运算符的小表达式进行计算
bool count_brackets_equal(string line); //括号输入是否合法判断

void DelSpace(char *source); //对表达式内空格的删除
void addzeros(char line[]); //补零操作 (-1)->(0-1)方便后续判断
-----
bool is_variable(char line[]); //判断用户输入是否为定义变量语句
int is_in_variable(char ch); //判断用户输入的变量是否已经被定义并赋值

bool ContainFunction(string line, string func); //判断是否含有数学函数
string SolveFunction(string line); //将数学函数进行计算并代替成相应的数值并返回字符串

string BigFloat_calculate(string strline);
对高精度数字的计算和打印。
```

Part 3 - Result & Verification

Requirement1:

```
root@DESKTOP-BC3HHPT:/mnt/c/C++pj/project2# ./a.out
Please enter the expression you want to calculate:
2+3
5
5+2*3
11
5+2*3+5
16
_
```

Requirement2:

```
root@DESKTOP-BC3HHPT:/mnt/c/C++pj/project2# ./a.out
Please enter the expression you want to calculate:
(5+2)*3
21
3*((1+2)+(3+4))
30
```

Requirement3:

```
root@DESKTOP-BC3HHPT:/mnt/c/C++pj/project2# ./a.out
Please enter the expression you want to calculate:
x=3
y=6
x+2*y
15
x*x+y*(x+x)
45
_
```

Requirement4:

```
root@DESKTOP-BC3HHPT:/mnt/c/C++pj/project2# ./a.out
Please enter the expression you want to calculate:
sqrt(3.0)
1.73205
pow(2,2)
4
cos(PI)
-1
max(1,5)
5
exp(0)
1
```

Requirement5:

2、数学逻辑计算

在 `calculateline()` 函数中，程序自定义了两个 `stack`：

```
stack<char> op;
stack<double> num;
```

分别用于存放表达式中的运算符和数字。程序对表达式进行遍历，在对数字进行读取时，若判断数字为小数点前的整数部分，则按位乘它的位数再求和，小数点后同理求和，得到一个 `double` 类型的数字存入 `num` 中；

若读取到运算符，则将其存入到 `op` 中，运用 `priority()`

```
int priority(char op);
```

对此运算符进行优先级赋值判断，若本次存入的运算符优先级大于栈顶的运算符则直接存入。

若 `priority(line[i]) <= priority(op.top())` 优先级小于栈顶运算符，则先用 `num` 栈顶的两个数字和 `op` 栈顶的运算符进行计算，将结果存入 `num` 中，再将本次运算符存入。

最终将所有的运算符和数字进行计算。

这样的操作既能保证运算符和数字数量差一直等于 1（方便后续计算）且顺序不被打乱，又能保证优先级高的运算先被计算。

3、负数情形

程序运用 `addzeros()` 函数对首位负数进行补零操作 `-1 -> 0-1` 和括号内的负数补零操作 `(-1) -> (0-1)`，以此方便后续的判断和计算。

4、除数为 0

```
Please enter the expression you want to calculate:
4/0
inf
```

单独进行判断

Difficulty2:

1、括号输入的合法性判断

```
bool count_brackets_equal(string line)
```

程序运用进行了两个判断，对左右括号个数是否相同的判断和括号次序是否符合表达式的判断。

```
root@DESKTOP-BC3HHPT:/mnt/c/C++pj/project2# ./a.out
Please enter the expression you want to calculate:
((1+1)
brackets wrong
Please enter the expression you want to calculate:
)1+1(
brackets wrong
```

2、对括号内表达式进行优先计算

在 `calculateline()` 函数中实际上对括号内的表达式进行了单独判断，先对括号内的表达式进行了运算，以此来实现优先级最高。

Difficulty3:

1、判断是否为一个赋值语句

```
bool is_variable(char line[]);
```

该函数使只有首位字母是小写，第二位字符是 '=' ,等号后是可计算的表达式为正确赋值输入，只有该判断为 true 才会进行赋值操作。

2、变量值的覆盖保存和新定义变量

```
int is_in_variable(char ch){
    for(int i=0;i<variablesiz; i++){
        if(variable[i]==ch)
            return i;
    }
    return -1;
}
```

该函数对已有的变量名进行遍历，若变量是第一次被定义返回-1，在 variable 数组中新定义一个变量存入数值，若变量是再次被定义则返回该变量在 variable 数组中的索引，后续进行再次赋值但不需要再定义变量。

```
char variable[26];
```

```
int variablesiz=0;
```

```
double variablevalue[26];
```

其中 variable 数组和对应的 variablevalue 数组为全局变量。

```
root@DESKTOP-BC3HHPT:/mnt/c/C++pj/project2# ./a.out
Please enter the expression you want to calculate:
x=1
y=1
x
1
x=2
x
2
```

Difficulty4:

1、找到表达式中的数学函数和函数内的数值并进行计算。

```
bool ContainFunction(string line,string func);//判断是否含有数学函数
string SolveFunction(string line);//将数学函数进行计算并代替成相应的数值
并返回字符串
```

运用 string.find()函数找到函数位置，并用 substr 进行精准切片，将函数前后，函数内部表达式分别切出为 strbefore,strafter,func，运用 math.h 头文件的函数计算为数字，再将字符串拼接起来计算打印

```
Please enter the expression you want to calculate:
1+sqrt(4)+1
4
```

数学函数上完成了:sqrt(),pow(),max(),min(),abs(),sin(),cos(),tan(),exp(),log(),以及完成了对常数 PI,E 的定义。

Conference: [Mariotti94/BigFloat: Arbitrary-Precision Floating Point Arithmetic Library \[C++\]\(github.com\)](#)

Difficulty5:

若要实现高精度的保留，在 `calculateline()` 函数中的运算方法则很难达到，于是运用的 `BigFloat` 库函数将数字保存为 `BigFloat` 类型，用库内重载的运算符进行按位的四则运算，其中将小数 0 进行了清除，最终打印出字符串。

[illegible]

1、对注释语句的判断

分别对应三种情况:

$$\frac{1+1\#2+2}{2}$$