CS205 C/C++ Programming - Projetct02

姓名：秦世华

SID：11812309

Github：https://github.com/Kishore4399/Project02.git

为了能够有效直观得解析命令行输入的表达式，本次project作出了如下优化。

1、不以指针定位的方式逐一获取argv[] 的每一个字符，而是使用实验课上使用的cin.getline获取char array, 再将char[] 转成std::string以方便定位

2、使用如std::string::find_first_of("+-*/") 来识别符号是否存在以及判定运算符号存在时定位符号位置。

requirement01

When you run your program and input an express in a line as follows, it can output the correct results. The operator precedence (order of operations) should be correct.

输出结果：

```
kishore@DESKTOP-9IRFFAD:/mnt/e/Ccourse/lab01/project02$ ./pro02
Please give us an expression:
2+3
5
Please give us an expression:
3+ 5*3
18
Please give us an expression:
4 + 3*5.5
20.5
```

部分说明代码：

```
double calcul(string expression)
{
    string::size_type pos;
    string::size_type lastPos;
    string::size_type fristPos;

    string snum2;
    while ((pos = expression.find_first_of("/*")) != string::npos){
  .......}  // 判断是否含有乘除法
```

分析和反思：

1、可实现小数输入和输出

2、double res = (1.0f)*num1 - (1.0f)*num2; 做加减法的时候需要注意前后都要转成flout，使得输出值错误

requirement02

Use parentheses to enforce the priorities

输出结果：可支持多括号嵌套运算

```
kishore@DESKTOP-9IRFFAD:/mnt/e/Ccourse/lab01/project02$ ./pro02
Please give us an expression:
((2+3)*2 + (4-3)/2)*3 + 2*(4+2)
43.5
```

部分说明代码：

```
string schar;
    while (expression.find_first_of("()") != string::npos)
    {
        for (int i = 0; i != expression.size(); ++i)
        {
            if (expression[i] != ')')
            {
                schar = schar + expression[i];
```

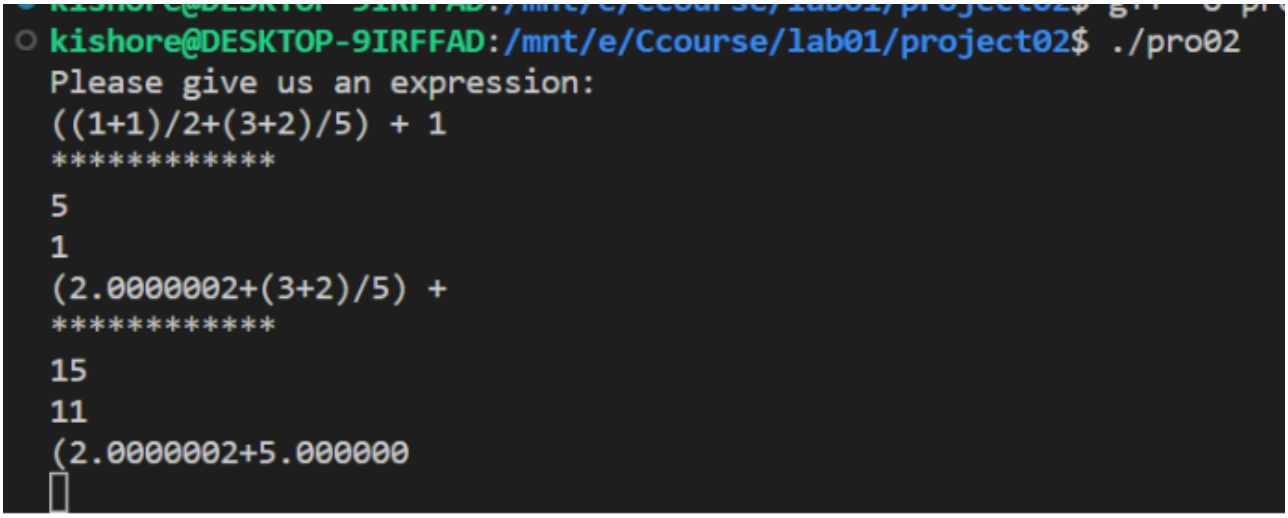
```
            }
            else
            {
                string::size_type left_pos = i;    // 用来定位"("
                string rchild;
                while (schar[schar.size()-1] != '(')
                {
                    left_pos -= 1;
                    rchild = schar[schar.size()-1] + rchild;
                    schar.erase(schar.size()-1);
                }
                left_pos -= 1;
                schar = "";   // 去除"("
                double number = calcul(rchild);
                string snum = std::to_string(number);
                // 将括号里面的数计算过后替代输入的string对应的括号里的内容
                expression.replace(left_pos, (i+1)-left_pos, snum);
                break;
            }
        }
    }
    double final_result = calcul(expression);
    return final_result;
```

分析和反思：

本题关于确认括号的思路在于，在找到第一个“）”的index，再往左移找到里第一个“）”最近的“（”，由此可以获得最小括号里的内容。计算出括号内运算的值后将原来输入的expression中对应括号部分的内容替换为计算结果，由此不断循环，知道括号被全部运行。得出最终结果。

编写过程中遇到如下问题：在调用Std::string.replace(left_pos, (i+1)-left_pos, snum)时，没有按照指定区间替换。查明原因后发现replace有多种用法：
用法一：用str替换指定字符串从起始位置pos开始长度为len的字符
用法二：用str替换 迭代器起始位置 和 结束位置 的字符
用法三：用substr的指定子串（给定起始位置和长度）替换从指定位置上的字符串　等九种用法

而我开始用的是第二种，后来成了第三种 即第二个参数为从起始位置pos开始长度为len的字符后，运行结果正确



requirement03

ariables can be defined as follows

输出结果：

```
kishore@DESKTOP-9IRFFAD:/mnt/e/Ccourse/lab01/project02$ ./pro02
Please give us an expression:
x=3.5
y=45
x*10 + (x+y)*5
487.5
```

部分说明代码：

```
if((pos_input = str.find("=")) != string::npos){ // 当发现有“=”输入时进入模式二
        var01=str.substr(0,pos_input);
        var01_value = str.substr(pos_input+1);
        // 输入第二变量
        char ch02[50];
        cin.getline(ch02, 50);
        string str02 = ch02;
        var02=str02.substr(0,pos_input);
        var02_value = str02.substr(pos_input+1);
        // 输入待求的表达式
        char ch03[50];
        cin.getline(ch03, 50);
        string str03 = ch03;
```

```
        int pos_var01 = str03.find(var01);
        int pos_var02 = str03.find(var02);
        // 用数值代替表达式中的变量
        while (pos_var01!= string::npos){
            str03.replace(pos_var01, 1, var01_value);
            pos_var01 = str03.find(var01);
        }

        while (pos_var02 != string::npos){
            str03.replace(pos_var02, 1, var02_value);
            pos_var02 = str03.find(var02);
        }
        str = str03;
    };
    cout << result(str) << endl;
```

分析和反思：

可实现对变量进行加减乘除，以及识别括号。

requirement04

ome math functions can be supported

输出结果(支持输入为表达式)

```
kishore@DESKTOP-9IRFFAD:/mnt/e/Ccourse/lab01/project02$ ./pro02
Please give us an expression:
sqrt(36)
6
kishore@DESKTOP-9IRFFAD:/mnt/e/Ccourse/lab01/project02$ ./pro02
Please give us an expression:
sqrt((1+2)*3 + 3*9)
6
kishore@DESKTOP-9IRFFAD:/mnt/e/Ccourse/lab01/project02$ ./pro02
Please give us an expression:
pow(32,2)
1024
kishore@DESKTOP-9IRFFAD:/mnt/e/Ccourse/lab01/project02$ ./pro02
Please give us an expression:
```

```
pow(12+10,2)
484
```

部分说明代码

```
// 第三题 一些数学公式
// 求平方根
    if (str.find("sqrt")!=string::npos){
        int pos_left = str.find_first_of('(');
        int pos_right = str.find_last_of(')');
        string str_out = str.substr(pos_left+1, pos_right-pos_left-
1);
        cout<<sqrt(result(str_out))<<endl;
        return 0;
    }
// 求指数
    if (str.find("pow")!=string::npos){
        int pos_left = str.find_first_of('(');
        int pos_right = str.find_last_of(')');
        int pos_middle = str.find(',');
        string str_out = str.substr(pos_left+1, pos_right-pos_left-
1);
        double number1 = result(str.substr(pos_left+1, pos_middle-
pos_left-1));
        double number2 = result(str.substr(pos_middle+1, pos_right-
pos_middle-1));
        cout<<number1<<endl;
        cout<<number2<<endl;
        cout<<pow(number1, number2)<<endl;
        return 0;
    }
```

分析与反思：

使用substr同样需要注意参数问题！