

复数计算器

题目描述

实现一个复数域上的计算器。复数的形式： $a + bi$ 其中 a 表示实部， b 表示虚部，为了简单起见，对复数的形式作如下规定：

- 复数的实部和虚部都是`int`类型，即使复数的虚部为0/1，在输入或者输出该复数时，它的形式也应该为 $a + 0i$ 或者 $a + 1i$ 。
- 当复数的虚部为负数时，在输入或输出该复数时，其形式为 $a - |b|i$, 例如你应当输出 $2 - 3i$ ，而非 $2 + -3i$ 。
- 对于0这个数而言，其输入输出格式为 $0 + 0i$ 。

需要实现的功能如下：

- 复数的基本运算：加减乘除
- 复数表达式求值，输入复数字符串，输出结果。
- 我们保证复数运算的时候不会出现溢出或者除零等问题，结果一定是可以计算的。

Complex.h

Complex.h中定义了需要实现的接口，请在Complex.cpp中实现。

```
#ifndef COMPLEX_H
#define COMPLEX_H
#include <string>
#include <vector>
#include <iostream>
#include <sstream>
#include <fstream>
using namespace std;
class Complex
{
private:
    int real; // 实部
    int imag; // 虚部
}
```

```

public:
    Complex() = default;
    Complex(int real, int imag) {
        this->real = real;
        this->imag = imag;
    }
    int getReal(){
        return real;
    }
    int getImag(){
        return imag;
    }
    // TODO 重载+, 实现复数的相加
    Complex operator+(const Complex& c);

    // TODO 重载-, 实现复数的相减
    Complex operator-(const Complex& c);

    // TODO 重载*, 实现复数的相乘
    Complex operator*(const Complex& c);

    // TODO 重载/, 实现复数的相乘
    Complex operator/(const Complex& c);

    // 重载<<, 输出格式为a+bi, a和b分别是复数的实部和虚部, 没有空格, 不用加换行
    // 注意:a或b为复数的输出示例, -1-3i
    friend ostream & operator << (ostream & out, const Complex &c);
    // 重载>>, 读取a+bi格式的输入, 格式同上。
    friend istream & operator >> (istream & in, Complex & c);
};
#endif

```

MathUtils.h

```

#ifndef MATHUTILS_H
#define MATHUTILS_H
#include "Complex.h"
class MathUtils {
public:
    // 接受一个表达式字符串并返回计算结果。例如:calculator("(1+2i)+(2+3i)")
    的结果为3+5i
    static Complex calculator(string s);
};

#endif

```

$(a + bi)/(c + di)$ 复数除法按以下公式计算，直接int/int即可。

$$: \frac{ac + bd}{c^2 + d^2} + \frac{bc - ad}{c^2 + d^2}i$$

调用示例

```

#include "Complex.h"
#include "MathUtils.h"
#include <vector>
using namespace std;
int main(){
    vector<string> strs ={
        "(1+2i)+(3+4i)",
        "(-1-2i)/(3-4i)",
        "(7-8i)*(-9+10i)",
        "(7-8i)-(-9+10i)"
    };
    for(size_t i=0;i<strs.size();++i){
        cout<<MathUtils::calculator(strs[i])<<endl;
    }
}

```

结果如下：

```
//output
4+6i
0+0i
17+142i
16-18i
```

注意：

- 上述输入测试样例的形式只是为了你方便测试，实际上在你提交的文件中不应该有任何cin。
- calculator的输入一定是字符串，且为 $(a + bi) + (c + di)$ 的格式，即仅有两个复数，两个复数的括号之间一定是运算符。

Tips:

在处理复数计算表达式的时候有一些函数可以使用：（当然利用for循环遍历处理也是可行的，而且有可能不容易犯错，相较于substr来说）

表 9.14: string 搜索操作	
搜索操作返回指定字符出现的下标，如果未找到则返回npos。	
s.find(args)	查找 s 中 args 第一次出现的位置
s.rfind(args)	查找 s 中 args 最后一次出现的位置
s.find_first_of(args)	在 s 中查找 args 中任何一个字符第一次出现的位置。
s.find_last_of(args)	在 s 中查找 args 中任何一个字符最后一次出现的位置
s.find_first_not_of(args)	在 s 中查找第一个不在 args 中的字符
s.find_last_not_of(args)	在 s 中查找最后一个不在 args 中的字符



;

续表

args 必须是以下形式之一	
c, pos	从 s 中位置 pos 开始查找字符 c。pos 默认为 0
s2, pos	从 s 中位置 pos 开始查找字符串 s2。pos 默认为 0
cp, pos	从 s 中位置 pos 开始查找指针 cp 指向的以空字符结尾的 C 风格字符串。 pos 默认为 0
cp, pos, n	从 s 中位置 pos 开始查找指针 cp 指向的数组的前 n 个字符。pos 和 n 无默认值

比如说 $(a + bi) + (c + di)$ ，利用 `find_first_of("(")` 可以获得右括号第一次出现的位置。

`string.substr(pos, len)` 返回从 `pos` 开始长为 `len` 的子字符串。

提交说明

1. 提交 `MathUtils.cpp`, `Complex.cpp` 两个文件，注意编码方式为 UTF-8，直接打包为 zip 格式压缩包，不要存在多一层的目录。
2. 实现代码请严格按照给定的接口名字，否则不能通过编译。（可以根据实际需要添加其他函数）
3. 提交代码中不要包含 `main()` 函数，否则不能通过编译。
4. 严格按照要求的功能实现输出，不要尝试进行其他输入输出活动，否则不能通过测试。