

复数运算

刷新 ↻

题目描述

通过重载运算符实现复数类是一道经典的OOP入门题目。请实现一个 `Complex` 类，通过运算符重载来实现复数的以下功能：

- 加减乘除四则运算
 - 由于实数也是复数，四则运算需要支持虚数（虚部不为0的复数）和实数混合计算的情形
- 取共轭（用单目运算符 `~` 表示）
- 实现一个函数 `double abs(const Complex&)` 计算复数的模长
- 重载流运算符实现流输出

为了更加便捷地表示复数，我们可以引入新的用户自定义字面量（user-defined literals，可参考如下简单教程 <https://zhuanlan.zhihu.com/p/111369693> (<https://zhuanlan.zhihu.com/p/111369693>)）`i`，用来表示虚数单位，请你补全下面的函数并加入到 `Complex.h` 中：

```
Complex operator"" i(unsigned long long Im) {  
    //TODO: 返回一个Complex对象  
}  
  
Complex operator"" i(long double Im){  
    //TODO: 返回一个Complex对象  
}
```

然后我们便可以使用类似 `Complex z = 1 + 2i;`，`cout<<c2/c7* (-5.68i+3.865) <<endl;` 这样的代码来表示复数了。

本题需要注意一些细节，可能不像看起来那样简单，以下是几点注意与提示：

- 可以通过 `double` 类型参数的类型转换构造函数来实现 `double` 到 `Complex` 的转换，从而如果使用了合适的重载运算符的方法，则只需要实现一次四则运算符重载
- 应该使用 `double` 类型来存储复数的实部和虚部
- 需要为 `Complex` 实现单目 `-` 运算符以实现复数的取负数
- `Complex` 的流输出需要考虑各种情况（虚部、实部是否为零）

输出格式

下面是通过流输出复数的格式：

- 保留两位小数（可使用 `setprecision` 函数实现小数位数舍入）
- 实部在前虚部在后，形如 `-38.77 + 45.94i` (注意"+"两边有空格)
- 注意输出时数字前的正负号
- 实部或虚部为0时不输出，两者都为0时则输出 `0.00`，**这里实部或虚部为0的定义为绝对值小于 10^{-5}**

样例输入

可以使用下面的main.cpp文件检查 `Complex` 类实现是否正确。**该文件和提供的示例Makefile、Complex.h可以通过以下链接下载：点击下载 (/staticdata/1949.EckAydOL6bRrrOqL.pub/gGvUCWf085n6ATr3.download.zip/download.zip)**

测试使用的main.cpp文件与该文件不同。

```
#include "Complex.h"
#include <iostream>

using std::cout, std::endl;
int main() {
    //本文件用于测试Complex类，与评测所用测试点不同

    //basic complex test
    Complex c1(1.5, 2.5);
    Complex c2(2, 4);
    Complex c3 = c1 + c2;
    cout << c3 << endl;
    Complex c4 = c1 - c2;
    cout << c4 << endl;
    Complex c5 = c1 * c2;
    cout << c5 << endl;
    Complex c6 = c1 / c2;
    cout << c6 << endl;
    Complex c7 = ~c1;
    cout << c7 << endl;

    Complex c8(0, 4);
    cout << c8 << endl;

    //complex abs test
    cout << abs(c2) << endl;

    //test cout
    cout << Complex(-2, 0) << endl;
    cout << 5 + 3i << endl;
    return 0;
}
```

样例输出

```
3.50 + 6.50i
-0.50 - 1.50i
-7.00 + 11.00i
0.65 - 0.05i
1.50 - 2.50i
4.00i
4.47
-2.00
5.00 + 3.00i
```

数据限定

- 对于所有数据，保证中间运输结果使用double类型存储不会出现溢出，你不需要对中间结果进行小数位数的舍入。
- 如果你使用自己编写的Makefile文件，**需要注意加入-std=c++14编译选项，且编译出的可执行文件名必须为main**，否则在某些环境下可能出现编译错误。

提交格式

- 你需要提交Complex.h,Complex.cpp(可选),Makefile。注意main.cpp由题目提供。
- 你应该将你的文件打包成一个zip压缩包并上传。**注意：你的文件应该在压缩包的根目录下，而不是压缩包的一个子文件夹下。**

语言和编译选项

#	名称	编译器	额外参数	代码长度限制
0	oop_custom	make		1048576 B