

```

#include<iostream>
using namespace std;

class Complex {
public:
    Complex() : real(0), imag(0) {}
    Complex(int a, int b) : real(a), imag(b) {}

    Complex complex_add(const Complex& b) {
        Complex d;
        d.real = real + b.real;
        d.imag = imag + b.imag;
        return d;
    }

    Complex complex_add(int a) {
        Complex b;
        b.real = a + real;
        b.imag = imag;
        return b;
    }

    friend Complex add(const Complex& a, const Complex& b);

    void display() {
        if (real == 0 && imag != 0)
            cout << imag << "i" << endl;
        else if (imag == 0)
            cout << real << endl;
        else if (imag > 0)
            cout << real << "+" << imag << "i" << endl;
        else
            cout << real << imag << "i" << endl;
    }

private:
    double real;
    double imag;
};

Complex add(const Complex& a, const Complex& b) {
    Complex c;
    c.real = a.real + b.real;
    c.imag = a.imag + b.imag;
}

```

```

    return c;
}

int main() {
    int realPart, imagPart;

    // Input for complex number a
    cout << "分别输入给定的第 1 个复数 a 的实部和虚部: ";
    cin >> realPart >> imagPart;
    Complex a(realPart, imagPart);

    // Input for complex number b
    cout << "分别输入给定的第 2 个复数 b 的实部和虚部: ";
    cin >> realPart >> imagPart;
    Complex b(realPart, imagPart);

    Complex c;
    cout << " 【1】 测试双目友元函数 add"<<endl;
    cout << "复数 a + b 的结果: ";
    c = add(a, b);
    c.display();
    cout << "将 a + b 的结果设为复数 c"<<endl;

    cout << "\n 【2】 测试重载函数 complex_add"<<endl;
    cout << "输入要与复数 c 相加的整数: ";
    int num;
    cin >> num;
    cout << "复数 c + " << num << "的结果: ";
    c = c.complex_add(num);
    c.display();

    cout << "\n 【3】 测试单目函数 complex_add, 其中自身 Complex 对象为 c, 传入为
b"<<endl;
    cout << "复数 c + b 的结果: ";
    c = c.complex_add(b);
    c.display();

    system("pause");
    return 0;
}

```