设计一个复数类Complex，要求满足以下条件：

有一个无参的构造函数，其初始的实部和虚部分别是0,0.

有一个带参的构造函数，通过初始化列表的方法给实部和虚部赋值。

有一个带参的默认构造函数。

成员函数setDate实现复数的设置。

成员函数display实现复数信息的获取（以2 + 6i形式输出）。

成员函数getReal和getImag实现获取复数的实部和虚部。

写一个全局函数ComplexSub(Complex c1,Complex c2,Complex &c3)，来计算两个复数的和，即将计算结果通过引用传递放到c3中。

复数c1为：2 + 1i

复数c2为：2 + 4i

复数c1和c2的和为： 4 + 5i

#include <iostream>

using namespace std;

class Complex

{

public:

Complex(int, int);

Complex(int aVal = 0);

void setData(int, int);

void display();

int getReal() { return m\_iReal;}

int getImag() { return m\_iImag;}

private:

int m\_iReal;

int m\_iImag;

};

Complex::Complex(int aR, int aI)

:m\_iReal(aR),m\_iImag(aI)

{

}

Complex::Complex(int aVal)

{

m\_iReal = aVal;

m\_iImag = aVal;

}

void Complex::setData(int aR, int aI)

{

m\_iReal = aR; m\_iImag = aI;

}

void Complex::display()

{

cout << m\_iReal << " + " << m\_iImag << "i" << endl;

}

void ComplexSub(Complex c1,Complex c2,Complex &c3)

{

c3.setData((c1.getReal()+c2.getReal()),(c1.getImag()+c2.getImag()));

}

int main(void)

{

Complex c1(2,1),c3;

cout << "复数c1为： " ;

c1.display();

Complex c2(2,4);

cout << "复数c2为： " ;

c2.display();

ComplexSub(c1,c2,c3);

cout << "复数c1和c2的和为：" ;

c3.display();

return 0;

}