**实验5：运算符重载**

第一题：

有两个矩阵a和b,均为2行3列。求两个矩阵之和。重载运算符+，使之能够用于矩阵相加。

**输入**

**按行方式依次输入矩阵a和b**

**输出**

**输出求和后的矩阵**

**注意，每行最后有一个空格，漏掉或多输出将判格式错误。**

答案：

#include<iostream>

using namespace std;

class Matrix

{

    private:

    int arr[2][3];

    public:

    int i,j;

    Matrix()

    {

        for(i=0;i<2;i++)

        {

            for(j=0;j<3;j++)

            {

                arr[i][j]=0;

            }

        }

    }

    void input()

    {

        for (int i=0; i<2; i++)

        {

            for (int j=0; j<3; j++)

            {

                 cin>>arr[i][j];

            }

        }

    }

    Matrix operator +(Matrix r);

    void put()

    {

        for(i=0;i<2;i++)

        {

            cout<<arr[i][0]<<' '<<arr[i][1]<<' '<<arr[i][2]<<' '<<endl;

        }

    }

};

Matrix Matrix::operator +(Matrix r)

{

    Matrix k;

    for(i=0;i<2;i++)

    {

        for(j=0;j<3;j++)

        {

            k.arr[i][j]=arr[i][j]+r.arr[i][j];

        }

    }

    return k;

}

int main()

{

    Matrix p1,p2;

    p1.input();

    p2.input();

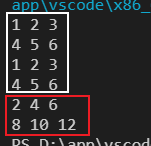
    Matrix p3=p1+p2;

    p3.put();

    return 0;

}

输入及输出：（白色框内为输入，红色框内为输出）



第二题：

定义一个复数类Complex，重载运算符+,-,\*,/，使之能用于复数的加减乘除。

运算符重载函数作为Complex类的成员函数。编程分别求两个复数的和、差、积、商。

**输入**

**输入有两行，分别代表两个复数**

**每行两个整数，分别代表实部和虚部。**

**输出**

**四行，分别对应和、差、积、商**

答案：  
#include<iostream>

using namespace std;

class Complex

{

    private:

    double a,b;

    public:

    Complex()

    {

        a=0;

        b=0;

    }

    void input()

    {

        cin>>a;

        cin>>b;

    }

    void output()

    {

        cout<<'('<<a<<','<<b<<'i'<<')'<<endl;

    }

    Complex operator +(Complex x);

    Complex operator -(Complex x);

    Complex operator \*(Complex x);

    Complex operator /(Complex x);

};

Complex Complex::operator +(Complex x)

{

    Complex y;

    y.a=a+x.a;

    y.b=b+x.b;

    return y;

}

Complex Complex::operator -(Complex x)

{

    Complex y;

    y.a=a-x.a;

    y.b=b-x.b;

    return y;

}

Complex Complex::operator\*(Complex x)

{

    Complex y;

    y.a=a\*x.a-b\*x.b;

    y.b=b\*x.a+a\*x.b;

    return y;

}

Complex Complex::operator/(Complex x)

{

    Complex y;

    y.a=(a\*x.a+b\*x.b)/((x.a)\*(x.a)+(x.b)\*(x.b));

    y.b=(b\*x.a-a\*x.b)/((x.a)\*(x.a)+(x.b)\*(x.b));

    return y;

}

int main()

{

    Complex p1,p2;

    p1.input();

    p2.input();

    Complex p3=p1+p2;

    p3.output();

    Complex p4=p1-p2;

    p4.output();

    Complex p5=p1\*p2;

    p5.output();

    Complex p6=p1/p2;

    p6.output();

    return 0;

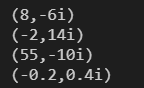
}

输入及输出：

输入：

3 4

5 -10

输出：  


第四题：

题目：  
有两个矩阵a和b,均为2行3列。求两个矩阵之和。重载运算符+，使之能够用于矩阵相加。

重载 << 和 >>使之能够用于矩阵的输入输出。

答案：

#include<iostream>

using namespace std;

class Matrix

{

    private:

    int arr[2][3];

    public:

    int i,j;

    Matrix()

    {

        for(i=0;i<2;i++)

        {

            for(j=0;j<3;j++)

            {

                arr[i][j]=0;

            }

        }

    }

    void input()

    {

        for (int i=0; i<2; i++)

        {

            for (int j=0; j<3; j++)

            {

                 cin>>arr[i][j];

            }

        }

    }

    Matrix operator +(Matrix r);

    friend ostream & operator<<( ostream & os,const Matrix & r);

    friend istream & operator>>( istream & is,Matrix & r);

};

ostream & operator <<( ostream & os,const Matrix & r)

{

    int i;

    for(i=0;i<2;i++)

    {

        os <<r.arr[i][0]<<' '<<r.arr[i][1]<<' '<<r.arr[i][2]<<' '<<endl;

    }

    return os;

}

istream & operator >>( istream & is,Matrix & r)

{

    int i,j;

    for(i=0;i<2;i++)

    {

        for(j=0;j<3;j++)

        {

            is>>r.arr[i][j];

        }

    }

    return is;

}

Matrix Matrix::operator +(Matrix r)

{

    Matrix k;

    for(i=0;i<2;i++)

    {

        for(j=0;j<3;j++)

        {

            k.arr[i][j]=arr[i][j]+r.arr[i][j];

        }

    }

    return k;

}

int main()

{

    Matrix p1,p2;

    cin>>p1;

    cin>>p2;

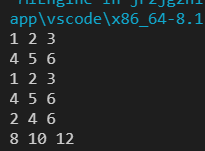
    Matrix p3=p1+p2;

    cout<<p3;

    return 0;

}

输入和输出：



第五题：

题目：

（1）．定义一boat与car两个类，两者都有weight属性，定义二者的一个友元函数totalweight()，计算二者的重量和。

C++提交的答案将被附加下面的main函数，提交时请不要包含main函数。

int main()

{

  car c1(1000);

 boat b1(2000);

 cout<<totalweight(b1,c1)<<endl;

}

答案：

#include<iostream>

using namespace std;

class car;

class boat

{

    private:

    int weight;

    public:

    boat(int w)

    {

        weight=w;

    }

    friend int totalweight(boat &b, car &c);

};

class car

{

    private:

    int weight;

    public:

    car(int w)

    {

        weight=w;

    }

    friend int totalweight(boat &b, car &c);

};

int totalweight(boat &b, car &c)

{

    return  b.weight + c.weight;

}

int main()

{

    car c1(1000);

    boat b1(2000);

    cout<<totalweight(b1,c1)<<endl;

}

输出：

