

```

//una operacion con complejos
//representados en cartesianas
//(pr1+pi1i)*(pr2+pi2i)/(pr3+pi3i)
//el objeto del ejemplo es mostrar lo farragoso
//de los calculos a pesar de ser una operacion sencilla

#include<iostream>
using namespace std;

int main()
{
    //parte real del 1...parte real del producto y del res final
    //parte imaginaria del 1...parte im. del producto y del res final
    //ocho variables en total, es facil confundirse
    float pr1,pr2,pr3,prp,prr,pil,pi2,pi3,pip,pir;
    cout<<"Introduce los complejos ";
    //hay que pedir cada componente por separado
    cin>>pr1>>pi1>>pr2>>pi2>>pr3>>pi3;
    //o se deduce una expresion para la parte real
    //e imaginaria del resultado, usando papel (pensando :-))
    //o se hace la operacion por partes, primero el producto
    //y despues el cociente, por ejemplo
    //(pr1+pi1i)*(pr2+pi2i)=(pr1*pr2-pi1*pi2)+(pr1*pi2+pi1*pr2)i
    prp=pr1*pr2-pi1*pi2;
    pip=pr1*pi2+pi1*pr2;
    //ademas la notacion puede resultar equivoca...:-))
    //(prp+pipi)/(pr3+pi3i)=(prp+pipi)*(pr3-pi3i)/(pr3*pr3+pi3*pi3)=
    //((prp*pr3-pip*pi3)/(pr3*pr3+pi3*pi3)+((-prp*pi3+pip*pr3)/(pr3*pr3+pi3*pi3))i
    prr=(prp*pr3+pip*pi3)/(pr3*pr3+pi3*pi3);
    pir=(-prp*pi3+pip*pr3)/(pr3*pr3+pi3*pi3);
    //se muestra el resultado
    cout<<prr<<'+ '<<pir<<'i'<<endl;
}

```

```

//una operacion con complejos
//representados en cartesianas
//(pr1+pi1i)*(pr2+pi2i)/(pr3+pi3i)
//el objeto del ejemplo es mostrar lo farragoso
//de los calculos a pesar de ser una operacion sencilla
//aun usando funciones, el codigo sigue siendo sucio,
//al representar los complejos como dos reales, las
//funciones producen dos resultados, luego hay que
//usar paso por referencia
//la unica ventaja es poder reutilizar las funciones
//en otros programas

#include<iostream>
using namespace std;

//declaracion y definicion de funciones, la vinculacion
//entre estas funciones y las operaciones que representan
//no esta expresada de forma explicita en el codigo, son
//funciones que trabajan con reales. No se define ningun
//complejo de forma explicita en el codigo.

void cociente(float pr1, float pil, float pr2, float pi2,
             float &prr, float &pir)
{
    prr=(pr1*pr2+pi1*pi2)/(pr2*pr2+pi2*pi2);
    pir=(-pr1*pi2+pi1*pr2)/(pr2*pr2+pi2*pi2);
}

void producto(float pr1, float pil, float pr2, float pi2,
             float &prr, float &pir)
{
    prr=pr1*pr2-pi1*pi2;
    pir=pr1*pi2+pi1*pr2;
}

int main()
{
    //parte real del 1...parte real del producto y del res final
    //parte imaginaria del 1...parte im. del producto y del res final
    //ocho variables en total, es facil confundirse
    float pr1,pr2,pr3,prp,prr,pil,pi2,pi3,pip,pir;
    cout<<"Introduce los complejos ";
    //hay que pedir cada componente por separado
    cin>>pr1>>pi1>>pr2>>pi2>>pr3>>pi3;
    //llamadas a las funciones
    //no siguen la notacion usual en matematicas
    //no son intuitivas.
    //primer producto
    producto(pr1,pil,pr2,pi2,prp,pip);
    //cociente entre el resultado anterior y el
    //tercer complejo
    cociente(prp,pip,pr3,pi3,prr,pir);
    //se muestra el resultado
    cout<<prr<<'+ '<<pir<<'i'<<endl;
}

```

```

//una operacion con complejos
//representados en cartesianas
//c1*c2/c3
//usando estructuras AL MODO CLASICO DE C
//se define un tipo de forma explicita, pero
//la vinculacion con sus operaciones se realiza
//mediante el tipo de los operandos (parametros de funciones)
//no se refleja de forma explicita en el codigo

#include<iostream>
using namespace std;

//definicion de la estructura complejo
//se define la representacion del tipo de forma explicita:
//dos campos de tipo real
struct complejo{
    float re,im;
};

//las operaciones que se pueden aplicar al complejo:
//operaciones aritmeticas y E/S
//se definen como funciones que toman parametros de ese tipo

//las estructuras se pueden devolver con return
complejo multiplica(const complejo &y, const complejo &x)
{
    complejo resultado;
    resultado.re=y.re*x.re-y.im*x.im;
    resultado.im=y.re*x.im+y.im*x.re;
    return resultado;
}

complejo divide(const complejo &y, const complejo &x)
{
    complejo resultado;
    resultado.re=(y.re*x.re+y.im*x.im)/(x.re*x.re+x.im*x.im);
    resultado.im=(-y.re*x.im+y.im*x.re)/(x.re*x.re+x.im*x.im);
    return resultado;
}

//se pueden definir funciones para la E/S de complejos
//se puede utilizar el paso por referencia
void lee(complejo &x)
{
    cin>>x.re>>x.im;
}

void muestra(complejo &x)
{
    cout<<x.re<<'+'<<x.im<<'i';
}

int main()
{
    //se declaran las variables de tipo complejo
    complejo c1,c2,c3,cr;
    //se piden los complejos
    cout<<"Introduce los complejos ";
    lee(c1);
    lee(c2);
    lee(c3);
    //la representacion de los calculos
    //es menos farragosa pero no es igual que en matematicas
    cr=divide(multiplica(c1,c2),c3);
    //presentacion de resultados
    muestra(cr);
}

```

```

//una operacion con complejos
//representados en cartesianas
//c1*c2/c3
//usando clases main queda mas
//limpio y la operacion se escribe
//como en matematicas

#include<iostream>
using namespace std;

//declaracion y definicion de la clase complejo
//se define el tipo de forma explicita:
//conjuntamente la representacion (dos reales)
//y las operaciones que se pueden aplicar al complejo:
//operaciones aritmeticas y E/S

class complejo{
    float re,im;
public:
    complejo(void){}
    complejo(float x,float y)
    {
        re=x;
        im=y;
    }

    //se pueden redefinir los operadores usuales
    //para representar las operaciones correspondientes
    complejo operator*(const complejo &x)const
    {
        complejo resultado;
        resultado.re=re*x.re-im*x.im;
        resultado.im=re*x.im+im*x.re;
        return resultado;
    }

    complejo operator/(const complejo &x)const
    {
        complejo resultado;
        resultado.re=(re*x.re+im*x.im)/(x.re*x.re+x.im*x.im);
        resultado.im=(-re*x.im+im*x.re)/(x.re*x.re+x.im*x.im);
        return resultado;
    }

    //se pueden definir funciones para la E/S de complejos
    //o bien redefinir >> y <<
    void lee(void)
    {
        cin>>re>>im;
    }
    void muestra(void)
    {
        cout<<re<<'+'<<im<<'i';
    }
};

int main()
{
    //se declaran las variables de tipo complejo
    complejo c1,c2,c3,cr;
    //se piden los complejos
    cout<<"Introduce los complejos ";
    c1.lee();
    c2.lee();
    c3.lee();
    //la representacion de los calculos
    //es identica a la usada en matematicas
    cr=c1*c2/c3;
    //presentacion de resultados
    cr.muestra();
}

```