

```

//Resolver un sistema de ecuaciones de orden 2
#include<iostream>
using namespace std;
int main()
{
    float a,b,c,d,e,f,det,x,y;
    cout<<"Introduce los coeficientes de la primera ecuacion: ";
    cin>>a>>b>>c;
    cout<<"Introduce los coeficientes de la segunda ecuacion: ";
    cin>>d>>e>>f;
    //si los terminos indep son 0 los dos
    if(c==0&&f==0)
    {
        cout<<"Sistema homogeneo"<<endl;
        x=0;
        y=0;
    }
    else
    {
        //calculo del determinante
        det=a*e-d*b;
        //si el determinante es distinto de cero
        //se calculan las soluciones
        if(det!=0)
        {
            cout<<"Sistema compatible determinado"<<endl;
            x=(c*e-b*f)/det;
            y=(a*f-d*c)/det;
        }
        else
            cout<<"\nSistema incompatible";
    }
    //si se han calculado las soluciones
    //se muestran
    if((c==0&&f==0)||det!=0)
    {
        cout<<"x="<<x<<endl<<"y="<<y<<endl;
    }
}

```

```

//mostrar la hora siguiente a una dada
#include<iostream>
using namespace std;
int main()
{
    int hora,min;
    cout<<"Introduce la hora: ";
    cin>>hora>>min;
    //si son las 11 de la noche
    if (hora==23)
        //si es el ultimo minuto
        if (min==59)
        {
            min=0;
            hora=0;
        }
    //en caso contrario
    else
        min++;
    //si no son las 11 de la noche
    else
        //si es el ultimo minuto
        if (min==59)
        {
            min=0;
            hora++;
        }
    //en caso contrario
    else
        min++;
    cout<<hora<<':'<<min;
}

```

```

//pedir tres numeros por el teclado
//y ordenarlos
#include<iostream>
using namespace std;
int main()
{
    float a,b,c,tmp;
    cout<<"Introduce tres numeros:";
    cin>>a>>b>>c;
    //se ordenan los dos primeros
    if (a<b)
    {
        tmp=a;
        a=b;
        b=tmp;
    }
    //si el tercero es mayor que a
    //es el mayor de todos
    //se mueven todos hacia abajo
    if (c>a)
    {
        tmp=c;
        c=b;
        b=a;
        a=tmp;
    }
    //en caso contrario
    else
    //si esta entre a y b
    //solo se mueve b hacia abajo
    if (c>b)
    {
        tmp=c;
        c=b;
        b=tmp;
    }
    cout<<a<<' '<<b<<' '<<c<<endl;
}

```

```

//calcular las soluciones de una ecuacion de segundo
//grado, considerar todos los casos posibles.
#include<cmath>
#include<iostream>
using namespace std;

//contante que representa "casi cero"
const float casi_cero=0.00001;
//tambien podriamos haber hecho
//#define casi_cero 0.00001

int main()
{
    float a,b,c,d;
    cout<<"Introduce los coeficientes de la ecuacion:";
    cin>>a>>b>>c;
    //si el coeficiente a es cero
    //La comparacion con cero de un real es una mala practica
    //de programacion, mejor considerar un intervalo y no hacer
    //if (a==0)
    if (abs(a)<=casi_cero)
    {
        cout<<"Ecuacion degenerada"<<endl;
        if (b!=0)
            cout<<"Solucion:"<<-c/b;
        else
            cout<<"No hay solucion";
    }
    else
    {
        //calculo del discriminante
        d=b*b-4*a*c;
        //si el discriminante es cero, igual que antes
        if (abs(d)<=casi_cero)
        {
            cout<<"Solucion real doble:"<<-b/2/a;
        }
        else
        {
            //si el discriminante es menor que cero
            if (d<0)
            {
                cout<<"Soluciones complejas conjugadas:"<<endl;
                cout<<-b/2/a<<'+ '<<sqrt(-d)/2/a<<'i'<<endl;
                cout<<-b/2/a<<'- '<<sqrt(-d)/2/a<<'i'<<endl;
            }
            //si el discriminante es mayor que cero
            else
            {
                cout<<"Dos soluciones reales:"<<endl;
                cout<<(-b+sqrt(d))/2/a<<endl;
                cout<<(-b-sqrt(d))/2/a<<endl;
            }
        }
    }
}

```

```

//Calcular la fecha siguiente a una dada
//Considerar el caso de Febrero bisiesto
#include<iostream>
using namespace std;
int main()
{
    int dia1,mes1,anyo1,dia2,mes2,anyo2;
    cout<<"Introduce una fecha dia/mes/anyo:";
    cin>>dia1>>mes1>>anyo1;
    //determinamos si es o no bisiesto
    //no es bisiesto si no es multiplo de cuatro o
    //si siendolo de 100 no lo es de 400
    bool bisiesto;
    if (anyo1%4!=0||((anyo1%100==0&&anyo1%400!=0))
        bisiesto=false;
    else
        bisiesto=true;
    //en mar may jul ag oct
    if (mes1==1|mes1==3|mes1==5|mes1==7|mes1==8|mes1==10)
    {
        //fin de mes
        if (dia1==31)
        {
            mes2=mes1+1;
            dia2=1;
        }
        else
        {
            dia2=dia1+1;
            mes2=mes1;
        }
        anyo2=anyo1;
    }
    //febrero y bisiesto
    if (mes1==2&&bisiesto)
    {
        if (dia1==29)
        {
            mes2=mes1+1;
            dia2=1;
        }
        else
        {
            dia2=dia1+1;
            mes2=mes1;
        }
        anyo2=anyo1;
    }
    //febrero no bisiesto
    if (mes1==2&&!bisiesto)
    {
        if (dia1==28)
        {
            mes2=mes1+1;
            dia2=1;
        }
        else
        {
            dia2=dia1+1;
            mes2=mes1;
        }
        anyo2=anyo1;
    }
}

```

```

//ab jun sep nov
if (mes1==4|mes1==6|mes1==9|mes1==11)
{
    //fin de mes
    if (dia1==30)
    {
        mes2=mes1+1;
        dia2=1;
    }
    else
    {
        dia2=dia1+1;
        mes2=mes1;
    }
    anyo2=anyo1;
}
//dic
if (mes1==12)
{
    //fin de mes
    if (dia1==31)
    {
        mes2=1;
        dia2=1;
        anyo2=anyo1+1;
    }
    else
    {
        dia2=dia1+1;
        mes2=mes1;
        anyo2=anyo1;
    }
}
cout<<dia2<<"/"<<mes2<<"/"<<anyo2<<endl;
}

```

```
//calculadora de numeros complejos
//el resultado se muestra de la forma usual en matematicas:
//1 y no 1+0i
//i y no 0+i
//0 y no 0+0i
//a-bi y no a+-bi
//...
#include<iostream>
using namespace std;
int main()
{
    float pr1,pil,pr2,pi2,pr,pi,modulo;
    char operacion;
    cout<<"Introduce el primer complejo, la operacion y el segundo complejo: ";
    cin>>pr1>>pil>>operacion>>pr2>>pi2;
    switch(operacion)
    {
        case '+':pr=pr1+pr2;
            pi=pil+pi2;
            break;
        case '-':pr=pr1-pr2;
            pi=pil-pi2;
            break;
        case '*':pr=pr1*pr2-pil*pi2;
            pi=pr1*pi2+pr2*pil;
            break;
        case '/':modulo=pr2*pr2+pi2*pi2;
            pr=(pr1*pr2+pil*pi2)/modulo;
            pi=(pil*pr2-pr1*pi2)/modulo;
            break;
        default: cout<<"Error, operacion invalida"<<endl;
    }
}
```

```
if(operacion=='+'||operacion=='-'||operacion=='*'||operacion=='/')
{
    //si la parte real es cero
    if (pr==0)
        //si la parte imaginaria es -1
        if (pi== -1)
            cout<<"-i";
        else
            //si la parte imaginaria es 1
            if (pi==1)
                cout<<'i';
            else
                //si la parte imaginaria es 0
                if (pi==0)
                    cout<<0;
        //si la parte imaginaria no es cero
        else
            cout<<pi<<'i';
    //si la parte real no es cero
    else
    {
        cout<<pr;
        //si la parte imaginaria es -1
        if (pi== -1)
            cout<<"-i";
        //si la parte imaginaria es 1
        if (pi==1)
            cout<<"+i";
        //si la parte imaginaria es distinta de -1 y de 1
        if ((pi!= -1)&&(pi!=1))
        {
            //si la parte imaginaria es menor que cero
            if (pi<0)
                cout<<pi<<'i';
            //si es mayor que cero
            if (pi>0)
                cout<<'+'<<pi<<"i";
        }
    }
}
}
```

```

//sumar las cifras de un entero
#include<iostream>
using namespace std;
int main()
{
    //notar la inicializacion de suma
    int n,cifra,suma=0;
    cout<<"Introduce un numero entero: ";
    cin>>n;
    //mientras quedan cifras
    while(n!=0)
    {
        //obtener la cifra mas a la derecha
        cifra=n%10;
        //sumarla
        suma=suma+cifra;
        //eliminar la cifra mas a la derecha
        n=n/10;
    }
    cout<<"Suma de las cifras:"<<suma<<endl;
}

```

```

//invertir el orden de las cifras
//de un numero entero
#include<iostream>
using namespace std;
int main()
{
    //notar la inicializacion de reves
    int n,cifra,reves=0;
    cout<<"Introduce un numero entero: ";
    cin>>n;
    //mientras quedan cifras
    while(n!=0)
    {
        //obtener la cifra mas a la derecha
        cifra=n%10;
        //sumarla
        reves=reves*10+cifra;
        //eliminar la cifra mas a la derecha
        n=n/10;
    }
    cout<<"Suma de las cifras:"<<reves<<endl;
}

```

```

//averiguar si un numero es o no capicua
#include<iostream>
using namespace std;
int main()
{
    //notar la inicializacion de reves
    int n,copia,cifra,reves=0;
    cout<<"Introduce un numero entero: ";
    cin>>n;
    //se hace una copia del numero
    //el proceso que lo vuelve del reves
    //es destructivo
    copia=n;
    //mientras quedan cifras
    while(copia!=0)
    {
        //obtener la cifra mas a la derecha
        cifra=copia%10;
        //sumarla
        reves=reves*10+cifra;
        //eliminar la cifra mas a la derecha
        copia=copia/10;
    }
    //si el numero del reves es igual al original
    //es capicua
    if (reves==n)
        cout<<"Es capicua"<<endl;
    else
        cout<<"No es capicua"<<endl;
}

```

```

#include<iostream>
using namespace std;
//convierte decimal en binario usando
//ndigitos como maximo
//el resultado se codifica como entero
//no comprueba si existe desbordamiento
int main()
{
    int a,copia,ndigitos;
    cout<<"introduce un decimal:";
    cin>>a;
    copia=a;
    cout<<"numero de digitos de la representacion binaria:";
    cin>>ndigitos;
    int b=0,peso=1;
    //peso del digito binario mas significativo
    for (int i=1;i<ndigitos;i++)
        peso*=2;
    //el valor de los digitos se restan de a
    while(peso!=0)
    {
        //si el peso es menor o igual
        //el digito binario es 1
        if (peso<=a)
        {
            //se resta el peso
            a=a-peso;
            //se introduce un uno a la derecha
            b=b*10+1;
        }
        else
        {
            //en caso contrario se introduce un 0
            b=b*10;
            //peso inmediatamente inferior
            peso/=2;
        }
    }
    cout<<"El decimal:"<<copia<<" en binario con "
        <<ndigitos<<" digitos es:"<<b;
}

```

```

//Obtener los 20 primeros términos de
//la sucesión de Fibonacci.

```

```

#include<iostream>
using namespace std;
int main()
{
    unsigned long int fn,fn_1=1,fn_2=1;
    //los dos primeros son por definicion
    cout<<"F(0)="<<1<<endl<<"F(1)="<<1<<endl;
    //los siguientes 18
    for (int i=2;i<=19;i++)
    {
        //siguiente
        fn=fn_1+fn_2;
        cout<<"F("<<i<<")="<<fn<<endl;
        //actualizacion de los anteriores
        fn_2=fn_1;
        fn_1=fn;
    }
}

```

```

//Aproximación de e^x por una serie,
//finaliza cuando se alcanza un máximo
//número de iteraciones o una determinada
//precisión, lo que suceda antes.
//Intentar realizar el mínimo número de
//cálculos posible.
#include<iostream>
using namespace std;
int main()
{
    //variables para el exponente y precision
    float x, precision;
    //numero de iteraciones
    int iteraciones;
    cout<<"Introduce x, precision, iteraciones maximas:";
    cin>>x>>precision>>iteraciones;
    //termino actual, anterior,
    //resultado, indice del termino
    float t_n=1,t_n_1,resultado=0,n=0;
    while(t_n>precision&& n<iteraciones)
    {
        resultado+=t_n;
        t_n_1=t_n;
        n++;
        t_n=t_n_1*x/n;
    }

    cout<<"e elevado a "<<x<<" con precision "<<precision<<
        " iteraciones="<<n<<'='<<resultado;
}

```

```

//mostrar los n primeros capicuas
//n se pide por el teclado
#include<iostream>
using namespace std;
int main()
{
    //notar la inicializacion de n, comenzamos a probar
    //por el primero, el cero
    int n=0,copia,cifra,revs,contador=0,cantidad;
    cout<<"Cuantos capicuas quieres:";
    cin>>cantidad;
    //mientras no encuentre 'cantidad' capicuas
    while(contador<cantidad)
    {
        //ver capicua.cpp
        copia=n;
        revs=0;
        //mientras quedan cifras
        while(copia!=0)
        {
            //obtener la cifra mas a la derecha
            cifra=copia%10;
            //sumarla
            revs=revs*10+cifra;
            //eliminar la cifra mas a la derecha
            copia=copia/10;
        }
        //si el numero del revs es igual al original
        //es capicua
        if (revs==n)
        {
            //lo cuento
            contador++;
            //lo muestro
            cout<<n<<endl;
        }
        //siguiente numero para probar
        n++;
    }
}

```

```

//Determinar los numeros menores que 1000 tales que
//aparecen al final de su cuadrado
#include<iostream>
using namespace std;
//numeros<1000 tales que aparecen
//al final de su cuadrado
int main()
{
    //analizar los numeros menores que 1000
    for (int i=0;i<1000;i++)
    {
        //elevar el numero al cuadrado
        int cuadrado=i*i,copia=i;
        //inicialmente suponemos que aparece al final
        bool aparece_al_final=true;
        //comparar cifras empezando por el final
        //si se encuentra una distinta
        //no hace falta seguir
        while (copia!=0&&aparece_al_final)
        {
            //si alguna es distinta,
            //no aparece al final
            if (copia%10!=cuadrado%10)
                aparece_al_final=false;
            //eliminamos las cifras mas a la derecha
            copia/=10;
            cuadrado/=10;
        }
        if (aparece_al_final)
            cout<<i<<endl;
    }
}

```

```

//Determinar cual es la fraccion que se aproxima
//mas a PI con el numerador y el denominador
//menores que 500
#include<iostream>
using namespace std;
#include<cmath>
//valor de PI
const float PI=acos(-1.0);
int main()
{
    //dmin puede inicializarse a cualquier dif
    //que sepamos se va a producir
    float dmin=PI-1;
    int numPI,denPI,i,j;
    //diferentes combinaciones de
    //denominador y denominador
    for (i=1;i<500;i++)
        for (j=1;j<i/2;j++)
        {
            //si se encuentra un cociente
            //que se aproxime mas a PI
            if (abs((float)i/j-PI)<dmin)
            {
                //guardo el numerador y el denominador
                numPI=i;
                denPI=j;
                //calculo la nueva distancia minima
                dmin=abs((float)i/j-PI);
            }
        }
    cout<<PI<<"~="<<(float)numPI/denPI<<"="
    <<numPI<<"/"<<denPI<<endl;
}

```

```

//Descomponer un número en factores primos
#include<iostream>
#include<cmath>
using namespace std;
int main()
{
    int n,divisor=2;
    cout<<"Introduce un numero entero:";
    cin>>n;
    //mientras el divisor sea menor o igual que el numero
    while(divisor<=n)
    {
        int potencia=0;
        //mientras el numero sea divisible
        //entre el divisor
        while(n%divisor==0)
        {
            //se divide
            n/=divisor;
            //se cuenta el numero de veces
            potencia++;
        }
        //si se ha contado al menos una vez
        //es un factor primo
        if (potencia>0)
            cout<<divisor<<"^"<<potencia<<endl;
        divisor++;
    }
}

```

```

//Un número perfecto es aquel que
//coincide con la suma de todos sus divisores,
//incluyendo el 1 pero no el propio número
//(por ejemplo 6 = 1+2+3, 28 = 1+2+4+7+14).
//Escribir un programa que, muestre por la
//pantalla los números perfectos menores que 100.
#include<iostream>
using namespace std;
int main()
{
    //para cada numero entre 1 y 99
    for (int i=1;i<100;i++)
    {
        //inicializamos la suma a 1
        int suma=1;
        //para cada candidato a divisor
        //excepto el mismo
        for (int j=2;j<=i/2;j++)
            //si es divisor
            if (i%j==0)
                //Se suma
                suma+=j;
        //si la suma coincide con el numero
        //cumple la condicion
        if (suma==i)
            cout<<i<<endl;
    }
}

```

```

//Buscar el mayor número entero
//(menor que 45000 por cuestiones de rango)
// cuyo cuadrado se escriba con
//dígitos distintos.
//Ejemplos de cuadrados que cumplen esa condición:
//13^2 = 169 286^2=81796 32043^2=1026753849
#include<iostream>
using namespace std;
int main()
{
    //esta variable se usa para salir del
    //bucle cuando se encuentra el primero
    bool encontrado=false;
    for (int i=44999;i>0&&!encontrado;i--)
    {
        int cuadrado=i*i;
        //inicialmente suponemos que se
        //cumple la condicion
        bool digitos_distintos=true;
        //mientras queden digitos en el numero
        while(cuadrado!=0)
        {
            //cifra menos significativa de cuadrado
            //y el resto de las cifras
            int cifra=cuadrado%10,
                resto_cifras=cuadrado/10;
            //mientras queden cifras en resto_cifras
            while(resto_cifras!=0)
            {
                //si coincide alguna con la
                //menos significativa de cuadrado
                if (cifra==resto_cifras%10)
                    //ya no son todos distintos
                    digitos_distintos=false;
                //siguiente cifra de resto_cifras
                resto_cifras/=10;
            }
            //siguiente cifra de cuadrado
            cuadrado/=10;
        }
        //si digitos_distintos sigue siendo true
        if (digitos_distintos)
        {
            //lo hemos encontrado
            encontrado=true;
            //lo mostramos
            cout<<i<<"^2="<<i*i<<endl;
        }
    }
}

```