

```

//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,revs=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
        revs=revs*10+cifra;
        //eliminar la cifra mas a la derecha
        n=n/10;
    }
    cout<<"Suma de las cifras:"<<revs<<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,revs=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
        revs=revs*10+cifra;
        //eliminar la cifra mas a la derecha
        copia=copia/10;
    }
    //si el numero del reves es igual al original
    //es capicua
    if (revs==n)
        cout<<"Es capicua"<<endl;
    else
        cout<<"No es capicua"<<endl;
}

```

```

//calcular de forma iterativa
//la raiz cuadrada de un numero real
//utilizando la formula
//x_i=.5*(x_i-1+a/x_i-1)
#include<iostream>
using namespace std;
int main()
{
    float x_i,x_i-1=.5,a,valor,precision,diff;
    cout<<"Precision:";
    cin>>precision;
    cout<<"Valor:";
    cin>>a;
    //repetir mientras la diferencia entre
    //dos terminos consecutivos sea
    //mayor que la precision
    do
    {
        //siguiente termino
        x_i=.5*(x_i-1+a/x_i-1);
        //diferencia entre dos terminos consecutivos
        diff=abs(x_i-x_i-1);
        //actualizacion del termino anterior
        x_i-1=x_i;
    }while(diff>precision);
    cout<<x_i;
}

//pedir 10 numeros por el teclado
//mostrar el tercero mayor
//ejercicio de examen
#include<iostream>
using namespace std;
int main()
{
    //a es el numero que se pide,
    //uno dos y tres son los tres mayores,
    //uno el mayor, despues dos y tres.
    int i,a,uno,dos,tres;
    //se piden los dos primeros
    cin>>uno;
    cin>>a;
    //si a es mayor que uno
    //se guarda uno en dos y a en uno
    if (a>uno)
    {
        dos=uno;
        uno=a;
    }
    //en caso contrario el lugar de a es dos
    else
        dos=a;
    //se pide el siguiente
    cin>>a;
    //si a es el mayor los otros van a tres y dos
    if (a>uno)
    {
        tres=dos;
        dos=uno;
        uno=a;
    }
    //en caso contrario
    else
        //si va en medio
        if (a>dos)
        {
            tres=dos;
            dos=a;
        }
    //si es menor que dos y tres
    else
    {
        tres=a;
    }
    //se piden los 7 restantes
    for (int i=1;i<=7;i++)
    {
        //casi identico esquema
        cin>>a;
        if (a>uno)
        {
            tres=dos;
            dos=uno;
            uno=a;
        }
        else
            if (a>dos)
            {
                tres=dos;
                dos=a;
            }
        else
            //diferencia, si a es mayor que tres
            //lo reemplaza
            if (a>tres)
            {
                tres=a;
            }
    }
    cout<<tres;
}

```

```

//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,reves,contador=0,cantidad;
    cout<<"Cuantos capicuas quieres:";
    cin>>cantidad;
    //mientras no encuentre 'cantidad' capicuas
    while(contador<cantidad)
    {
        //ver capicua.cpp
        copia=n;
        reves=0;
        //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 revés es igual al original
        //es capicua
        if (reves==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);
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 mínima
                dmin=abs((float)i/j-PI);
            }
        }
    cout<<PI<<"~="<<(float)numPI/denPI<<"="
    <<numPI<<'/'<<denPI<<endl;
}

```