

```

//calcula el numero combinatorio
//n sobre m, sin funciones
#include<iostream>
using namespace std;
int main()
{
    int n,m;
    cout<<"Introduce n y m:";
    cin>>n>>m;
    int fn=1;
    for (int i=2;i<=n;i++)
        fn=fn*i;
    int fm=1;
    for (int i=2;i<=m;i++)
        fm=fm*i;
    int fnm=1;
    for (int i=2;i<=n-m;i++)
        fnm=fnm*i;
    cout<<n<<" sobre " <<m<<"='<<fn/fm/fnm;
}

//calcula el numero combinatorio n sobre m
//utilizando funciones
#include<iostream>
using namespace std;
//devuelve el factorial de n
int factorial(int n)
{
    int f=1;
    for (int i=2;i<=n;i++)
        f=f*i;
    return f;
}
//devuelve el numero combinatorio n sobre m
int combina(int n, int m)
{
    return factorial(n)/factorial(m)/factorial(n-m);
}
int main()
{
    int n,m;
    cout<<"Introduce n y m:";
    cin>>n>>m;
    cout<<n<<" sobre " <<m<<"='<<combina(n,m);
}

//e elevado a x con funciones
#include<iostream>
using namespace std;
//devuelve e elevado a x
//cuando se alcanza determinada precision
//o un maximo de iteraciones
//lo que ocurra antes
float ex(float x, float precision, int maxn)
{
    float t_n=1,t_n_1,resultado=0,n=0;
    while(t_n>precision&&n<maxn)
    {
        resultado+=t_n;
        t_n_1=t_n;
        n++;
        t_n=t_n_1*x/n;
    }
    return resultado;
}
int main()
{
    float x, precision;
    int iteraciones;
    cout<<"Introduce x, precision, iteraciones maximas:";
    cin>>x>>precision>>iteraciones;
    cout<<"e elevado a " <<x<<" con precision " <<precision<<
        " iteraciones max=" <<iteraciones<<"='<<ex(x,precision,iteraciones);
}

```

```

//mayor de dos numeros utilizando
//funciones
#include<iostream>
using namespace std;
//en esta primera forma se utiliza un unico
//return
float mayor1(float a, float b)
{
    float mayor;
    if (a>b)
        mayor=a;
    else
        mayor=b;
    return mayor;
}
//aquí se utilizan dos return
//si a>b es cierto, se retorna a
//y se termina la función
//si no es cierto se retorna b
float mayor2(float a, float b)
{
    if (a>b)
        return a;
    return b;
}
int main()
{
    float x,y;
    cout<<"introduce dos numeros:";
    cin>>x>>y;
    cout<<"El mayor calculado con una función es:" <<mayor1(x,y)<<endl;
    cout<<"El mayor calculado con la otra función es:" <<mayor2(x,y);
}

//mostrar los n primeros capicuas
//n se pide por el teclado
//version con funciones
#include<iostream>
using namespace std;
//devuelve un numero
//formado por las cifras del parametro
//en orden inverso
int reves(int n)
{
    //res es el resultado
    int res=0;
    while(n!=0)
    {
        //añadir la cifra mas a la derecha
        res=res*10+n%10;
        //eliminar la cifra mas a la derecha
        n=n/10;
    }
    return res;
}
//devuelve true si es capicua,
//false en caso contrario
bool capicua(int n)
{
    //la expresión es booleana
    //coincide con el tipo de retorno
    //si es cierta es capicua
    //si es falsa no lo es
    return n==reves(n);
}

```

```

int main()
{
    //notar la inicializacion de n, comenzamos a probar
    //por el primero, el cero
    int n=0,contador=0,cantidad;
    cout<<"Cuantos capicuas quieres:";
    cin>>cantidad;
    //mientras no encuentre 'cantidad' capicuas
    while(contador<cantidad)
    {
        if (capicua(n))
        {
            //lo cuento
            contador++;
            //lo muestro
            cout<<n<<endl;
        }
        //siguiente numero para probar
        n++;
    }
}

//mostrar los n primeros numeros
//que contienen la cifra x
//n y x se piden por el teclado
#include<iostream>
using namespace std;
//devuelve true si n contiene x
bool contienex(int n,int x)
{
    while(n!=0)
    {
        if (n%10==x)
            return true;
        n/=10;
    }
    return false;
}

//pide un entero entre 0 y 9
//es decir, una cifra
int pide_cifra(void)
{
    int num;
    do{
        cin>>num;
    }while(num<0||num>9);
    return num;
}

int main()
{
    int n=0,conta=0,cuantos,cifra;
    cout<<"Cuantos quieres:";
    cin>>cuantos;
    cout<<"Cifra a buscar:";
    cifra=pide_cifra();
    while(conta<cuantos)
    {
        if(contienex(n,cifra))
        {
            cout<<n<<endl;
            conta++;
        }
        n++;
    }
}

```

```

//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
//version con funciones
#include<iostream>
using namespace std;
//devuelve true si n contiene x
bool contienex(int n,int x)
{
    while(n!=0)
    {
        if (n%10==x)
            return true;
        n/=10;
    }
    return false;
}

//devuelve true si todos los digitos de n
//son distintos
bool digitos_distintos(int n)
{
    //mientras queden cifras en n
    while(n!=0)
    {
        //cifra menos significativa de n
        int cifra=n%10,
            //resto de las cifras
            resto_cifras=n/10;
        //si resto_cifras contiene a cifra,
        //cifra se repite
        if (contienex(resto_cifras,cifra))
            return false;
        //siguiente cifra de n
        n/=10;
    }
    //si llega aqui es que no se ha encontrado
    //ninguna repetida, luego se devuelve true
    return true;
}

//devuelve el mayor numero que cumple la condicion
int mayor_numero_bla_bla(int limite)
{
    for (int i=limite;i>0;i--)
    {
        int cuadrado=i*i;
        //si digitos_distintos es true
        if (digitos_distintos(cuadrado))
            //se retorna ese numero
            return i;
    }
    //si no se encuentra ninguno
    //usamos -1 para indicarlo
    //en este caso esto no se va a producir
    return -1;
}

int main()
{
    cout<<mayor_numero_bla_bla(45000);
}

```