

```

//Escribir un programa en C que lea un número entero por teclado
//y muestre por pantalla la mayor de
//sus cifras y a continuación el número leído.
//Ejemplo: Si el número leído es el 183, el programa deberá; mostrar:
//8 es la cifra mayor de 183
#include<stdio.h>
int main()
{
int n, mayor, cifra, copia;
printf("\nIntroduce un entero:");
scanf("%d",&n);
//copia de n
copia=n;
//primera cifra inicializa mayor
mayor=copia%10;
//quitar la primera cifra
copia/=10;
//mientras queden cifras
while(copia!=0)
{
//siguiente cifra
cifra=copia%10;
//si es mayor que la mayor hasta el momento
if (cifra>mayor)
//es la nueva mayor cifra
mayor=cifra;
//eliminar cifra
copia/=10;
}
//mostrar resultados
printf("%d es la mayor cifra de %d", mayor, n);
}

```

```

//Escribir un programa que muestre los 10 primeros
//terminos de la sucesion de fibonacci
#include<stdio.h>
int main()
{
//los dos primeros son por definicion 1 y 1
int f_i, f_i_1=1,f_i_2=1,i;
//se muestran
printf("\n%d\n%d",f_i_1,f_i_2);
//se hacen los 8 restantes
for (i=1;i<=8;i++)
{
//se calcula el siguiente
f_i=f_i_1+f_i_2;
//se muestra
printf("\n%d",f_i);
//se actualizan los anteriores
f_i_2=f_i_1;
f_i_1=f_i;
}
}

```

```

//Escribir un programa que pida un numero entero por el teclado
//y muestre por la pantalla si es o no narcisista.
//Un numero es narcisista si es igual a la suma de la potencia de
//cada una de sus cifras elevado al numero total de cifras.
//Ejemplo: 153=1^3+5^3+3^3
#include<stdio.h>
int main()
{
int n,copia,potencia,cifra,suma,i,ncifras;
printf("\nIntroduce un entero:");
scanf("%d",&n);
//copia de n
copia=n;
//contar cifras
ncifras=0;
while(copia!=0)
{
ncifras++;
copia/=10;
}
//restaurar el valor de copia
copia=n;
//inicializacion de suma
suma=0;
//mientras queden cifras
while(copia!=0)
{
//inicializar la potencia
potencia=1;
//obtener la cifra que se va a elevar
cifra=copia%10;
//calcular cifra elevado a ncifras
for (i=1;i<=ncifras;i++)
potencia*=cifra;
//sumar la potencia
suma+=potencia;
copia/=10;
}
//si la suma de potencias es igual al numero
//es narcisista
if (suma==n)
printf("\nEl numero %d es narcisista",n);
else
printf("\nEl numero %d no es narcisista",n);
}

```

```

//escribir un programa que pida dos enteros a y b
//y que calcule a elevado a b
#include<stdio.h>
int main()
{
int a,b,i;
float potencia=1.0;
printf("\nIntroduce dos enteros:");
scanf("%d%d",&a,&b);
//si el exponente es mayor que cero
//se multiplica
if (b>0)
for (i=1;i<=b;i++)
potencia*=a;
//en caso contrario
else
//si la base es distinta de cero
if (a!=0)
//se divide
for (i=1;i<=-b;i++)
//para que haya decimales
//hay que convertir a a float
potencia/=(float)a;
else
//esto es discutible
//en realidad seria infinito
potencia=0;
printf("\n%f",potencia);
}

```

```

//Un numero 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<stdio.h>
int main()
{
int i,j,suma;
//para cada numero entre 1 y 99
for (i=1;i<100;i++)
{
//inicializamos la suma a 1
suma=1;
//para cada candidato a divisor
//excepto el mismo
for (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)
printf("\n%d",i);
}
}

```

```

//Determinar los numeros menores que 1000 tales que
//aparecen al final de su cuadrado, denominados
//numeros automorficos
#include<stdio.h>
int main()
{
int i,cuadrado,copia,aparece_al_final;
//analizar los numeros menores que 1000
for (i=0;i<1000;i++)
{
//elevar el numero al cuadrado
cuadrado=i*i;
copia=i;
//inicialmente suponemos que aparece al final
aparece_al_final=1;
//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=0;
//eliminamos las cifras mas a la derecha
copia/=10;
cuadrado/=10;
}
if (aparece_al_final)
printf("\n%d",i);
}
}

```