

Apuntes de C

Programa:

Serie de ordenes al ordenador para resolver un problema. Se escribe utilizando un lenguaje de programación.

Las ordenes se denominan sentencias o instrucciones.

Variable:

Elemento de programación cuyo valor puede variar a lo largo del programa y tendrá un nombre elegido por el programador. Este nombre puede contener letras, dígitos y carácter subrayado. Nunca puede empezar por dígito.

En un programa primero tenemos que declarar todas las variables que se vayan a utilizar.

Declarar una variable consiste en indicar su nombre y el tipo de dato que va a guardar.

La forma general de una declaración es:

tipo variable;

Si existen varias variables del mismo tipo se separan por comas, ejemplo:

tipo variable, variable2, ;

Tipos de datos

int .– Para números enteros

float .– Para números reales

ejemplo: `int a,sum,c;` tres variables *a, sum, c* y son números enteros.

`float x;` variable *x* que es número real.

Órdenes

Todas deben ir escritas en letras minúsculas. El lenguaje C diferencia entre mayúsculas y minúsculas.

Escritura.– Sirve para que el ordenador escriba por pantalla lo que le indiquemos. Su forma general es:

printf(mensaje);

Existen las siguientes variantes:

printf(mensaje\n); El comando `\n` sirve para indicar salto de línea

printf(mensaje\na\n); Escribe el mensaje, salta , escribe `<<a>>` salta línea.

Printf(\n); *Deja una línea en blanco*

Todo programa de C tiene la siguiente estructura:

#include<stdio.h> *Siempre tendrán este encabezamiento. Más adelante*

void main() *se explicará el motivo*

```
{  
  
_____  
  
_____  
  
}
```

Al escribir un programa en C, se debe guardar en un archivo con extensión .C

Una vez que el programa esta limpio, hay que aplicarle un proceso de enlace (Link).

Después el programa esta listo para ser ejecutado.

Comentario: *Anotaciones para el programador siempre tiene la siguiente expresión:*

/ Se escribirá todo lo que se encuentre entre estos símbolos */*

// Se escribirá todo lo que siga a estos símbolos en cada linea

Cuando se compile siempre se debe corregir el primer error, si es que lo tenemos...

Siguiendo con la sentencia **printf**

Su forma general es:

Printf(cadena de control, expresion1, expresion2,...);

Las expresiones pueden ser variables u operaciones aritméticas. La cadena de control va entre comillas y en ella se pueden indicar los mensajes y los formatos de las expresiones que se quieren mostrar.

Ejemplos de formatos :

%d *para expresiones de tipo entero (int)*

%f *para expresiones de tipo real (float)*

Supongamos que disponemos de las siguientes variables:

int a,b;

float x;

printf(El resultado es %d\n,a); *lo que va entre comillas se escribe como plantilla y luego va el formato, el*

cual se sustituye por el valor de esa variable.

Printf(%d sumado con %d es %d\n,a,b,a+b);

Imprime en pantalla valor de <<a>> sumado con valor de <> es<< a+b>>

Printf(El doble de %f es %f\n,x,2*x); *El primer %f se sustituye por el valor de x y el segundo %f se sustituye por el valor de 2*x.*

Ejercicio 1

*/*hacer un programa que escriba por pantalla lo siguiente:*

Hola

Este es

mi primer programa

*de c */*

#include<stdio.h>

void main ()

{

printf("Hola\nEste es\nmi primer programa\nde C\n");

}

Lectura de datos

Con la orden de lectura se consigue que el ordenador lea o acepte los datos que el usuario introduce por teclado, su estructura es:

scanf(cadena de control, &vble1,&vble2,.....);

El símbolo **&** debe ponerse siempre.

La cadena de control va entre comillas, solo se indican los formatos de las variables que se quieren leer nunca mensajes ni saltos de línea.

Ejemplos:

scanf(%d,&a);

scanf(%f,&x);

scanf(%d %f,&a,&x);

Ejercicio 2

*/*Hacer un programa que pida al usuario
que introduzca un número entero
y a continuación que muestre por pantalla
el doble de dicho número y en otra línea el triple*/*

```
#include<stdio.h>
```

```
void main ( )
```

```
{
```

```
int a;
```

```
printf("Introduce un numero entero:");
```

```
scanf("%d",&a);
```

```
printf("\nel doble de %d es %d\n",a,2*a);
```

```
printf("su triple es %d\n",3*a);
```

```
}
```

Ejercicio 3

*/*Hacer un programa que solicite por teclado dos numeros enteros
y que muestre por pantalla su suma su diferencia y su producto
cada cosa en una linea*/*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int a,b;
```

```
printf("Introduce dos numeros enteros\n");
```

```
printf("primer numero:");
```

```
scanf("%d",&a);
```

```
printf("\nsegundo numero:");
```

```
scanf("%d",&b);
```

```
printf("\n%d + %d = %d\n",a,b,a+b);

printf("%d - %d = %d\n",a,b,a-b);

printf ("%d * %d = %d\n", a,b,a*b);

}
```

Ejercicio 4

*/*Solicitar un numero entero por teclado
si es positivo mostrar por pantalla su doble
en cualquier caso solicitar otro numero entero
y mostrar por pantalla la suma de ambos */*

```
#include<stdio.h>

void main ()

{

int a,b;

printf("Introduzca un numero:");

scanf("%d",&a);

if(a>0)

printf("Su doble es %d",2*a);

printf("\nIntroduzca otro numero:");

scanf("%d",&b);

printf("\nLa suma de ambos es %d\n",a+b);

}
```

SENTENCIA CONDICIONAL

If(condicion) *Observese que no lleva << ; >>*

_____;

Cuando hay varias cosas que hacer si se cumple una condición , tiene la siguiente expresión:

If(condicion)

```
{
_____; /* realizara todo lo que debe cumplir si se cumple*/
_____;
_____;
}
```

Operadores condicionales

Aritméticos:

> *mayor que*

< *menor que*

> = *mayor o igual*

< = *menor o igual*

= = *igual a*

!= *distinto de*

ejemplo: **if(a = b)**

```
_____;
```

Operadores aritméticos binarios

+

-

*

/

% (resto de la división entera entre dos números enteros)

Ejemplo: Si un numero es par: **if(a%2==0)** si el resto de la división es igual a cero

Si un numero es impar: **if(a%!=0)** si el numero es impar

Si un numero es múltiplo de 5: **if(a%5==0)**

Tenemos el planteamiento de un ejercicio y su resolución:

Solicitamos un numero por teclado, si es positivo mostramos en pantalla el doble, solicitamos otro y mostramos la suma.

```

#include<stdio.h> /*El encabezamiento sera siempre el mismo */

void main( )

{

int a,b; /*Declaración de variables */

printf(Introduzca un numero:); /* El cursor espera en ese lugar a ver el teclado */

scanf(%d,&a); /*Esta instruccion permite la introduccion de datos */

if(a>0)

printf(%d,2*a); /* Se presenta en pantalla el producto 2*a */

printf(\nIntroduzca otro numero:); /*Salta una línea y espera */

scanf(%d,&b);

printf(\n%d,a+b);

}

```

Complementos con la sentencia condicional **if..... else**

Si se cumple una condición y si no la otra:

if (condicion) **if** (condicion) **if**(condicion)

_____ ; { _____ ;

else _____ ; **else**

_____ ; _____ ; {

} _____ ;

else _____ ;

_____ ; }

if(condicion)

{

_____ ;

_____ ;

}

else

```
{  
_____  
_____  
}
```

Ejercicio 5

*/*pedir un numero entero, si es un numero par pedir otro*

*y escribir el producto de ambos */*

#include<stdio.h>

void main ()

```
{
```

int a,b;

printf("Introduzca un numero entero:");

scanf("%d",&a);

if (a%2==0)

```
{
```

printf("\nVaya hombre es par");

printf("\nIntroduzca otro numero:");

scanf("%d",&b);

*printf("\nEl producto de ambos es %d\n",a*b);*

```
}
```

if (a%2!=0)

printf("como es impar se acaba el programa\n");

```
}
```

Ejercicio 6

*/*Pedir un numero entero, si es multiplo de 5 pedir otro*

y mostrar el doble de cada uno, en caso contrario

*mostrar su cuadrado */*

```
#include<stdio.h>
```

```
void main ()
```

```
{
```

```
int a,b;
```

```
printf("Introduzca un numero entero:");
```

```
scanf("%d",&a);
```

```
if(a%5==0)
```

```
{
```

```
printf("\nComo es multiplo de 5 introduzca otro numero:");
```

```
scanf("%d",&b);
```

```
printf("el doble de %d es: %d\nel doble de %d es: %d\n",a,2*a,b,2*b);
```

```
}
```

```
else
```

```
printf("El cuadrado de %d es %d \n",a,a*a);
```

```
}
```

Ejercicio 7

*/*Pedir un numero entero si es negativo pedir otro y*

en caso de que este nuevo numero tambien sea negativo

mostrar la suma de ambos y si no lo es pedir un tercer numero

*y mostrar el producto de los tres */*

```
#include<stdio.h>
```

```
void main ()
```

```
{
```

```
int a,b,c;
```

```

printf("Introduzca un numero entero negativo:");

scanf("%d",&a);

if (a<0)
{
printf("\nComo es negativo introduzca otro numero:");

scanf("%d",&b);

if(b<0)

printf("\nComo tambien es negativo\nLa suma es %d",a+b);

else

{

printf("\nComo es positivo introduzca otro numero entero:");

scanf("%d",&c);

printf("\nEl producto de los tres es %d",a*b*c);

}

}

else

printf("\nResulta que es positivo y esto se acabo\n");

}

```

Asignación

Sirve para dar o asignar valor a una variable, la forma general es:

Variable = expresión;

Cuando esta orden se ejecuta la variable que aparece a la izquierda del signo igual, toma el valor de la expresión que aparece a la derecha. Dicha expresión puede ser un valor constante una variable o una operación aritmética.

Ejemplo: a=5; /* A la izquierda solo puede ser una variable */

a=b; /* <<a>> toma el valor de <> */

Si b=a entonces <> toma el valor de <<a>> /* Lo de la izquierda toma el valor de la derecha */.

Si $a=(b+2)*5$ /*<<a>> toma el valor de <<(b+2)*5 >> */

Para las condiciones se usan == , para asignar es solo un =.

Así pues tenemos que:

Para comparar: == (dos iguales seguidos)

Para asignar: = (un solo igual)

Ejercicio 8

*/*pedir dos numeros enteros si el primero es mayor que el segundo
escribir la suma de ambos, en caso contrario si el primero es impar
pedir otro numero y escribir la suma de los tres y si no lo es escribir
el doble del primero y del segundo */*

```
#include<stdio.h>
```

```
void main ()
```

```
{
```

```
int a,b,c;
```

```
printf("Introduzca un numero entero:");
```

```
scanf("%d",&a);
```

```
printf("\nIntroduzca otro numero:");
```

```
scanf("%d",&b);
```

```
if(a>b)
```

```
printf("\nLa suma de ambos es %d\n",a+b);
```

```
else
```

```
{
```

```
if(a%2!=0)
```

```
{
```

```
printf("\nIntroduzca otro numero:");
```

```
scanf("%d",&c);
```

```

printf("\nLa suma de los tres es %d\n",a+b+c);

}

else

printf("\nEl doble del primer numero es %d y el doble del segundo es %d\n",2*a,2*b);

}

}

```

Ejercicio 9

*/*pedir numero entero si es multiplo de 5 pedir otro, y si este es positivo escribir la suma de ambos y si no lo es, su producto, pero si el primer numero no es multiplo de 5, mostrar por pantalla su cuadrado y si ademas es par mostrar su triple */*

```

#include<stdio.h>

void main ()

{

int a,b;

printf("Introduzca un numero entero:");

scanf("%d",&a);

if(a%5==0)

{

printf("\nIntroduzca otro numero:");

scanf("%d",&b);

if(b>0)

printf("\nLa suma de ambos es %d\n",a+b);

else

printf("\nEl producto de ambos es %d\n",a*b);

}

```

```

else

{

printf("\nEl cuadrado del primer numero es %d\n",a*a);

if(a%2==0)

printf("\nEl triple del primer numero es %d",3*a);

}

}

```

BUCLE WHILE

Es un conjunto de una o mas sentencias que se pueden ejecutar varias veces, en función de una determinada condición.

While (condicion)

```

{

_____ ; /*se le denomina cuerpo del bucle */

_____ ;

_____ ;

}

```

Lo primero es comprobar la condición, si se cumple se ejecutan entonces las sentencias que se indican en el cuerpo del bucle y automáticamente se vuelve a comprobar la condición. Todo este proceso termina en el momento de que dicha condición no se cumpla.

Contador.– Es una variable cuyo valor aumenta o disminuye una cantidad fija cada vez que se ejecuta el bucle.

Acumulador.– Es una variable cuyo valor aumenta o disminuye una cantidad variable, cada vez que se ejecuta el bucle.

Tanto los contadores, como los acumuladores deben inicializarse siempre, antes de ser utilizados, esto es, debe asignárseles un valor inicial.

Ejercicio 11

```

/*Pedir 6 numeros enteros y calcular la suma de todos ellos */

#include<stdio.h>

void main ()

```

```

{
int a,sum,cont;

sum=cont=0; /* inicializamos variables a cero */

while (cont<6) /* Comenzamos el bucle */
{
printf("Introduce un numero entero:");

scanf("%d",&a);

cont=cont+1;

sum=sum+a;

}

printf("La suma de los 6 numeros es %d\n",sum);

}

```

Análisis del programa anterior:

Sum cont a

0 0 4 /*Entramos en el bucle, introducimos el 4*/

4 1 9 /* El contador pasa a valer uno y suma 4, siguiente numero introducido es 9, etc */

13 2 5

18 3 2

20 4 10

30 5 4

34 6 /*Aquí la condicion ya no se cumple */

Sin el contador, el bucle se convierte en infinito.

Operadores aritméticos monarios (un solo operador).

+ + .- Es el operador incremento y sirve para incrementar en una unidad el valor del operando.

- - .- Es el operador decremento y sirve para decrementar en una unidad el valor del operando.

Ambos pueden actuar como prefijos o como sufijos.

Ejemplo: `cont = cont + 1;` <<equivale a>> `cont ++;` <<equivale a>> `++ cont;`

`Cont = cont - 1;` <<equivale a >> `cont --;` <<equivale a>> `-- cont;`

Ejercicio 12

*/*Pedir seis numeros enteros y calcular la suma de aquellos que sean impares */*

```
#include<stdio.h>
```

```
void main ()
```

```
{
```

```
int a,sum,cont;
```

```
sum=cont=0;
```

```
while(cont<6)
```

```
{
```

```
printf("Introduce un numero entero:");
```

```
scanf("%d",&a);
```

```
cont++;
```

```
if(a%2!=0)
```

```
sum=sum+a;
```

```
}
```

```
printf("La suma de los impares es %d\n",sum);
```

```
}
```

Ejercicio 13

*/*Pedir 6 numeros enteros calcular la suma de los que sean multiplos de 5*

*y contar los que no lo sean */*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```

int a,cont,sum,nomultiplos;

sum=cont=nomultiplos=0;

while(cont<6)

{

printf("Introduzca un numero entero:");

scanf("%d",&a);

cont++;

if(a%5==0)

sum=sum+a;

else

nomultiplos++;

}

printf("La suma de los multiplos de 5 es %d\n",sum);

printf("Los numeros que no son multiplos de 5 fueron %d\n",nomultiplos);

}

```

Ejercicio 14

```

/*Pedir 6 numeros enteros contar cuantos de ellos son negativos y de estos
sumar los que sean impares, ademas calcular el producto de todos los numeros
introducidos */

#include<stdio.h>

void main()

{

int a,cont,cuantaneg,suma,producto;

cont=cuantaneg=suma=0;

producto=1;

while(cont<6)

```



```

{

printf("Introduce un numero entero:");

scanf("%d",&a);

cont++;

producto=producto*a;

if(a<0)

{

cuentaneg++;

if(a%2!=0)

suma=suma+a;

}

}

printf("\nLos numeros negativos han sido %d\n",cuentaneg);

printf("\nLa suma de los impares negativos es %d\n",suma);

printf("\nEl producto de todos los numeros introducidos es %d\n",producto);

}

```

Ejercicio 15

*/*Pedir 6 numeros enteros, contar los que sean multiplos de 3*

*y sumar los que no lo sean, ademas contar los pares */*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int a,multiplos,suma,pares,cont;
```

```
multiplos=suma=pares=cont=0;
```

```
while(cont<6)
```

```
{
```

```

printf("Introduzca un numero entero:");

scanf("%d",&a);

cont++;

if(a%3==0)

multiplos++;

else

suma=suma+a;

if(a%2==0)

pares++;

}

printf("\nSe han introducido %d multiplos de 3\n",multiplos);

printf("\nLa suma de los que no son multiplos de 3 es %d\n",suma);

printf("\nLos numeros pares introducidos han sido %d\n",pares);

}

```

Cálculo de la media

Para calcular la media debemos sumar todos y dividir por el número de ellos, para ello utilizaremos un bucle, ejemplo:

```

While ( )

{

_____ ; /* En el cuerpo del bucle contaremos y sumaremos */

_____ ;

_____ ;

}

printf( media, suma/6); /*No hace falta ninguna variable par la media */

```

Ejercicio 16

```

/*Pedir 6 numeros enteros y calcular su media */

#include<stdio.h>

```

```

void main()

{

int a,suma,cont;

suma=cont=0;

while(cont<6)

{

printf("Intoduzca un numero entero:");

scanf("%d",&a);

cont++;

suma=suma+a;

}

printf("\nLa media de los numeros es %.2f\n",suma/6.0);

}

```

Como comentario adicional al ejercicio anterior, debemos saber que en C cuando se dividen 2 números enteros, el resultado es siempre otro numero entero. Sin embargo si uno de los 2 números que se dividen o los 2 son reales, el resultado de la división es un número real.

Para escribirlo, es decir, para mostrar un número real debemos escribir %f, pero si quiero números decimales, entonces utilizaremos la siguiente expresión:

%.n° decimales f

También se puede escribir: ***%.2f\n,sum/6.0 /* al dividir por un n° real sale real */***

La suma puede ser una variable entera ya que si las variables son enteras lo lógico es que su suma también sea entera.

Ejercicio 17

*/*Pedir 6 numeros enteros, contar cuantos son pares y calcular la*

*media de los que sean impares */*

#include<stdio.h>

void main()

{

```

int a,cont,suma,pares,contimpar;

cont=suma=pares=contimpar=0;

while(cont<6)

{

printf("Introduzca un numero entero:");

scanf("%d",&a);

cont++;

if(a%2==0)

pares++;

else

{

suma=suma+a;

contimpar++;

}

}

printf("\nLos numeros pares son %d\n",pares);

if(contimpar!=0)

printf("\nLa media de los que son impares es %.2f\n",(float)suma/contimpar);

else

printf("\nNo se han introducido impares por lo que no hay media posible\n");

}

```

Comentario al ejercicio 17

El ordenador no es capaz de dividir entre cero, (caso de que no se introduzcan impares, lo que hace que `contimpar = 0`), introduciremos otro **if** que contemple el hecho de que esa cuenta sea cero.

Cuando no se contempla lo anterior se dice que el programa rompe (ventana con error....) y hay que evitarlo.

Ejercicio 18

*/*Pedir seis numeros enteros calcular la suma de los que sean multiplos*

de 3 y de estos calcular la media de los negativos, calcular el producto

de los que no sean multiplos de 3 */

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int a,suma,cont,contneg,producto,sumaneg;
```

```
cont=suma=cont=contneg=sumaneg=0;
```

```
producto=1;
```

```
while(cont<6)
```

```
{
```

```
printf("Introduzca un nuemro entero:");
```

```
scanf("%d",&a);
```

```
cont++;
```

```
if(a%3==0)
```

```
{
```

```
suma=suma+a;
```

```
if(a<0)
```

```
{
```

```
sumaneg=sumaneg+a;
```

```
contneg++;
```

```
}
```

```
}
```

```
else
```

```
producto=producto*a;
```

```
}
```

```
printf("\nLa suma de los que son multiplos de 3 es %d\n",suma);
```

```

if(contneg!=0)

printf("\nLa media de los multiplos de 3 negativos es %.2f\n",(float)sumaneg/contneg);

else

printf("\nComo no se hay numeros negativos no hay media\n");

printf("\nEl producto de los que no son multiplos de 3 es %d\n",producto);

}

```

Ahormado

Ahormar una variable consiste en cambiar momentáneamente su tipo, se realiza colocando delante de la variable entre paréntesis el nuevo tipo. El ahormado es provisional, queda:

Printf(media impares %f\n,(float)suma/contimpar);

Esto es ahormar provisional

Ejercicio 19

```

/*Pedir un numero entero a continuación, pedir seis numeros enteros y
calcular la media de los que sean menores que el primer numero leido y
de los que no lo sean mostrar un mensaje con cada uno indicando si son
pares o impares */

#include<stdio.h>

void main()

{

int x,a,cont,suma,contmenor;

cont=suma=contmenor=0;

printf("Introduce un numero entero:");

scanf("%d",&a);

while(cont<6)

{

printf("\nIntroduce otro numero entero:");

```

```

scanf("%d",&x);

cont++;

if(x<a)
{
suma=suma+x;

contmenor++;
}

else
{

if(x%2==0)

printf("\nEste numero es par\n");

else

printf("\nEste numero es impar\n");

}

}

if(contmenor!=0)

printf("\nLa media de los que son menores que %d es %.2f\n",a,(float)suma/contmenor);

else

printf("\nNo hay media pues no hay numeros menores que el 1º\n");

}

```

Ejercicio 20

*/*Pedir un numero entero y mostrar su tabla de multiplicar del 1 al 10 */*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int a,cont;
```

```

cont=1;

printf("Introduzca un numero entero:");

scanf("%d",&a);

printf("\nLa tabla del numero %d es:\n",a);

while(cont<=10)

{

printf("\n%d * %d = %d\n",a,cont,a*cont);

cont++;

}

}

/*el cont++ no puede ir antes de printf porque empezaria en 2.

tambien se puede inicializar con cont =0*/

```

Ejercicio 21

```

/*Pedir un numero entero si es impar pedir a continuacion 6 numeros enteros

y contar cuantos son mayores que él, y si no es impar mostrar su tabla de

multiplicar */

#include<stdio.h>

void main()

{

int a,b,cont,mayores;

printf("Introduzca un numero entero:");

scanf("%d",&a);

if(a%2!=0)

{

cont=mayores=0;

while(cont<6)

```



```

{
printf("\nIntroduzca otro numero entero:");
scanf("%d",&b);
cont++;
if(b>a)
mayores++;
}

printf("\nLos numeros mayores que el primero han sido %d\n",mayores);
}

else
{
printf("\nLa tabla de %d es:\n",a);
cont=1;
while(cont<=10)
{
printf("\n%d * %d = %d\n",a,cont,a*cont);
cont++;
}
}
}

```

Operadores lógicos

&& <<equivale a>> **Y**

|| <<equivale a>> **O**

! <<equivale a>> **NO**

Su estructura es: **if (____ && ____)**

Ejemplo: **if (c>a && c<b)**

A partir de ahora inicializaremos las variables justo antes de la sentencia while

Ejercicio 22

*/*inicializar las variables antes justo de while */*

*/*Pedir dos numeros enteros si el primero es mayor o igual que el segundo*

mostrar la tabla de multiplicar del primero y en caso contrario pedir 6

*numeros y calcular la media de los que sean mayores que el primero y menores
que el segundo */*

#include<stdio.h>

void main()

{

int a,b,c,cont,suma,cantidad;

printf("Introduzca dos numeros enteros:\n");

scanf("%d",&a);

scanf("%d",&b);

if(a>=b)

{

printf("\nLa tabla de %d es\n",a);

cont=0;

while(cont<=10)

{

*printf("\n%d * %d = %d\n",a,cont,a*cont);*

cont++;

}

}

else

{

```

printf("\nComo %d > %d\n",b,a);

cont=cantidad=suma=0;

while(cont<6)

{

printf("\nIntroduzca otro numero entero:\n");

scanf("%d",&c);

cont++;

if(c>a&& c<b)

{

suma=suma+c;

cantidad++;

}

}

if(cantidad!=0)

printf("\nLa media de mayor que %d y menor que %d es %.2f\n",a,b,(float)suma/cantidad);

else

printf("\nComo no hay mayor que %d ni menor que %d no hay media\n",a,b);

}

}

```

Orden en que se realizan las operaciones aritméticas:

1º << * >> y << / >> de izquierda a derecha

2º << + >> y << - >> de izquierda a derecha

Este orden siempre puede alterarse mediante la utilización de los paréntesis ().

Ejemplo:

4 / 2 + 5 * 3

17

El uso del if y el <<;>> . Casos

Caso nº1:

if (condicion); /* Al tener << ; >> no compila */

_____;

else

_____;

Caso nº 2:

a = 3; equivale a a = 3;

if(a>5); if(a>5) /* el << ; >> equivale a {} */

printf(Hola\n); {

}

printf(Hola\n);

Caso nº 3:

a = 3;

if(a>5); /* Da igual que valor tiene << a >> y si cumple la

{ condicion o no siempre escribirá <<Hola>> y <<Adios>>

printf(Hola\n);

printf(Adios\n);

}

El uso del while y el <<;>> . Casos

Caso nº1:

While (condicion); equivale a while(condicion)

{ { /*El cuerpo del bucle

```
_____ ; } estaria siempre fuera */
```

```
_____ ; _____ ;
```

```
} _____ ;
```

Caso nº 2:

```
x = 0;
```

```
While(x<6) /*escribira 6 veces << Hola>>*/
```

```
{
```

```
printf(Hola\n);
```

```
x ++;
```

```
}
```

Si ponemos << ; >>

```
X = 0; x = 0;
```

While(x<6); equivale a while(x<6) /*No escribe nada, el

```
{ { programa se queda
```

```
printf(Hola\n); } atrapado en el bucle*/
```

```
x ++; printf(Hola\n);
```

```
} x ++;
```

Caso nº 3:

```
x = 0;
```

```
While(x!=0); /*si no tenemos << ; >> no haría nada */
```

```
{
```

```
printf(Hola\n);
```

```
x ++;
```

```
}
```

como equivale a:

```
x = 0;
```

```
While(x!=0) /*Escribe <<Hola>> una vez y aumenta << x >> una unidad */
```

```
{
```

```
}
```

```
printf(Hola\n);
```

```
x ++;
```

Realizamos test I y II

Realizar un ejercicio de un producto como suma abreviada y otro de potencia como producto abreviado.

Ejercicio 23

*/*Pedir dos numeros enteros si el segundo es mayor o igual que cero,*

calcular el producto de ambos, a base de sumar el primero tantas veces

*como indique el segundo */*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int a,b,suma,cont;
```

```
printf("Introduzca dos numeros enteros:");
```

```
scanf("%d",&a);
```

```
scanf("%d",&b);
```

```
if(b>=0)
```

```
{
```

```
cont=suma=0;
```

```
while(cont<b)
```

```
{
```

```
suma=suma+a;
```

```
cont++;
```

```
}
```

```
printf("\nEl producto de %d por %d es %d\n",a,b,suma);
```

```
}
```

```
}
```

Ejercicio 24

*/*Pedir dos numeros enteros, si el primero es distinto de cero y el segundo mayor o igual que cero, calcular la potencia en la que el primero sea la base y el segundo el exponente */*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int a,b,cont,producto;
```

```
printf("Introduzca dos numeros enteros:");
```

```
scanf("%d",&a);
```

```
scanf("%d",&b);
```

```
if(a!=0&&b>=0)
```

```
{
```

```
cont=0;
```

```
producto=1;
```

```
while(cont<b)
```

```
{
```

```
producto=producto*a;
```

```
cont++;
```

```
}
```

```
printf("\n%d elevado a %d es %d\n",a,b,producto);
```

```
}
```

```
}
```

Ejercicio 25

*/*pedir dos numeros enteros, si el segundo es menor que cero, pedir entonces seis numeros enteros y contar los que sean mayores que el primero y contar tambien los que sean multiplos del segundo, pero si el segundo numero no es menor que cero calcular el producto de ambos a base de sumar el primero las veces que indique el segundo */*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int a,b,c,cont,mayores,multiplos,suma;
```

```
printf("Introduzca dos numeros enteros:");
```

```
scanf("%d",&a);
```

```
scanf("%d",&b);
```

```
if(b<0)
```

```
{
```

```
cont=mayores=multiplos=0;
```

```
while(cont<6)
```

```
{
```

```
printf("\nIntroduzca otro numero entero:\n");
```

```
scanf("%d",&c);
```

```
cont++;
```

```
if(c>a)
```

```
mayores++;
```

```
if(c%b==0)
```

```
multiplos++;
```

```
}
```



```

printf("\nLos multiplos del segundo han sido: %d\n",multiplos);

printf("\nLos mayores que el primero han sido: %d\n",mayores);

}

else

{

cont=suma=0;

while(cont<b)

{

suma=suma+a;

cont++;

}

printf("\nEl producto de %d por %d es %d\n",a,b,suma);

}

}

```

Ejercicio 26

*/*Pedir un entero si es distinto de cero, pedir otro, si este es menor que cero
mostrar su tabla de multiplicar y si no lo es calcular la potencia en la que el
primer numero sea la base y el segundo el exponente */*

```

#include<stdio.h>

void main()

{

int a,b,cont,producto,potencia;

printf("introduzca un numero entero:");

scanf("%d",&a);

if(a!=0)

{

```

```

printf("\nIntroduzca otro numero entero:\n");

scanf("%d",&b);

if(b<0)

{

cont=0;

producto=0;

printf("\nLa tabla de %d es:\n",b);

while(cont<10)

{

cont++;

printf("\n%d * %d = %d\n",b,cont,producto+b);

}

}

else

{

cont=0;

potencia=1;

while(cont<b)

cont++;

printf("\n%d elevado a %d es %d\n",a,b,potencia*a);

}

}

}

```

Factorial

Los números menores que cero no tienen factorial, entonces normalmente se comenzara con un **if**.

El factorial consiste en decrementar un numero dado hasta que sea 1 e ir multiplicándolos.

Ejercicio 27

*/*Pedir un numero entero calcular y mostrar su factorial */*

#include<stdio.h>

void main()

{

int a,fact,cont;

printf("Introduzca un numero entero:");

scanf("%d",&a);

if(a<0)

printf("\nPor ser un numero negativo no tiene factorial");

else

{

cont=1;

fact=1;

while(cont<=a)

{

*fact=fact*cont;*

cont++;

}

printf("\nEl factorial de %d es %d\n",a,fact);

}

}

Ejercicio 27 bis

*/*Pedir un numero entero calcular y mostrar su factorial */*

*/*se empieza siempre por un if para evitar que sea menor que cero*/*

#include<stdio.h>

```

void main()

{

int a,fact,cont;

printf("Introduzca un numero entero:");

scanf("%d",&a);

if(a<0)

printf("\nPor ser un numero negativo no tiene factorial");

else

{

cont=a;

fact=1;

while(cont!=1)

{

fact=fact*cont;

printf("%d",fact);

cont--;

}

printf("\nEl factorial de %d es %d\n",a,fact);

}

}

```

Ejercicio 28

```

/*Pedir un entero, si es mayor o igual que cero calcular y escribir su
factorial, y si no lo es, pedir 6 numeros y calcular la media de los que sean
menores que él */

#include<stdio.h>

void main()

```

```

{
    int a,b,media,fact,cont,menores;

    printf("Introduzca un numero entero:");

    scanf("%d",&a);

    if(a>=0)
    {
        cont=a;

        fact=1;

        while(cont>0)
        {
            fact=fact*cont;

            cont--;
        }

        printf("\nEl factorial de %d es %d\n",a,fact);
    }

    else
    {
        cont=media=menores=0;

        while(cont<6)
        {
            printf("\nIntroduzca otro numero entero:\n");

            scanf("%d",&b);

            if(b<a)
            {
                media=media+b;

                menores++;
            }
        }
    }
}

```

```

}

cont++;

}

if(menores!=0)

printf("\nLa media de los numeros menores que %d es %.2f\n",a,(float)media/menores);

else

printf("\nNo se puede calcular la media pues no hay menores que %d\n",a);

}

}

```

Cómo calcular los divisores de un número

Los divisores de un número son aquellos que dividen al número y su resto es cero. El primer divisor es siempre 1, entonces necesitamos un contador inicializado a 1, posteriormente, el resto de dividir el número entre el contador debe ser cero.

Ejercicio 29

*/*Pedir un numero entero si es menor o igual que cero tabla de multiplicar,
y si no lo es mostrar todos sus divisores */*

```

#include<stdio.h>

void main()

{

int a,cont,producto;

printf("Introduzca un numero entero:");

scanf("%d",&a);

if(a<=0)

{

cont=0;

printf("\nLa tabla de multiplicar de %d es:\n",a);

while(cont<10)

```

```

{
    cont++;

    printf("%d * %d = %d\n",a,cont,producto);
}

}

else

{

    printf("\nLos divisores de %d son:\n",a);

    cont=1;

    while(cont<=a)

    {

        if(a%cont==0)

            printf("%d\t",cont);

        cont++;

    }

    printf("\n");

}

}

```

Comentario al ejercicio 29.

Cuando se quiere regular la posición del cursor y separarlo de la última posición, usaremos el comando `\t` e imprimiremos una línea en blanco para separar del texto presione una tecla para continuar.

Ejercicio 30

```

/*Pedir un numero entero si es mayor que cero calcular y escribir la suma de sus
divisores y si esta suma es mayor que 30, mostrar su tabla de multiplicar */

#include<stdio.h>

void main()

{

```

```

int a,cont,suma;

printf("Introduzca un numero entero:");

scanf("%d",&a);

if(a>0)
{
    cont=1;
    suma=0;
    while(cont<=a)
    {
        if(a%cont==0)
            suma=suma+cont;
        cont++;
    }
    printf("\nLa suma de los divisores de %d es: %d\n",a,suma);

    if(suma>30)
    {
        printf("\nLa tabla de multiplicar de %d es:",a);
        cont=0;
        while(cont<10)
        {
            cont++;
            printf("\n%d * %d = %d\n",a,cont,a*cont);
        }
    }
}

```


BUCLE FOR

Es el mas utilizado en C, su estructura es:

```
for(expresion1;expresion2;expresion3) /* Observese que no tiene << ; >> */  
{  
    _____;  
    _____; /* Cuerpo del bucle */  
    _____;  
}
```

Expresion1 .– Consiste en una o mas inicializaciones de variables

Expresion2 .– Es una condición, mientras esta se cumpla, se ejecutará el cuerpo del bucle.

Expresión3 .– Consiste en la sentencia o sentencias necesarias para que el bucle se vuelva a ejecutar sin ser infinito.

Funcionamiento de FOR

Paso 1 .– Se ejecutan las inicializaciones indicadas en la expresion1

Paso 2 .– Se comprueba la condición indicada en la expresión2. En caso de que no se cumpla, el bucle termina. Si se cumple pasamos al paso 3.

Paso 3 .– Se ejecutan las sentencias indicadas en el cuerpo del bucle.

Paso 4 .– Se ejecutan las sentencias indicadas en la expresión3 y se vuelve al paso 2.

Ejemplo de programa:

Pedir 6 números y calcular su suma.

```
for(sum=cont=0;cont<6;cont+ +) /*Inicializamos 1 vez al principio */  
{  
    printf(Introduce un numero:);  
    scanf(%d,&a);  
    suma=suma+a;  
}  
printf(Los 6 numeros suman %d\n,sum);
```

Tambien se puede encontrar, para el test, con la siguiente forma:

```
Sum=cont=0; /* se inicializan antes como con el while */
```

```
for( ;cont<6; ) /*Cualquiera de las expresiones 1 ó 3 se puede quitar pero se conservan los << ; >> */
```

```
/* Las inicializaciones como sum=cont=0; se pueden escribir como sum=0,cont=0; es decir separados por comas */
```

Ejemplo:

```
For( ; ; ) /* Esto es para crear un bucle infinito
```

```
{
```

```
_____ ; /*Como no hay nada expresado, lo que hay en el cuerpo del bucle se repetira infinitamente */
```

```
_____;
```

```
}
```

Todos los ejercicios realizados con la sentencia **while** se pueden realizar con **for**

Ejercicio 31

```
/*Pedir un entero y mostrar su tabla de multiplicar */
```

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int a,cont;
```

```
printf("Introduzca un numero entero:");
```

```
scanf("%d",&a);
```

```
printf("\nLa tabla de multiplicar de %d es\n",a);
```

```
for(cont=1;cont<=10;cont++)
```

```
printf("%d * %d = %d\n",a,cont,a*cont);
```

```
}
```

Ejercicio 32

```
/*Pedir un numero entero, si es menor que cero, pedir 6 numeros y calcular la media
```

```
de los que sean multiplos de él, y si no es menor que cero, calcular y escribir su
```

```

factorial */

#include<stdio.h>

void main()

{

int a,b,suma,fact,cuenta,cont;

printf("Introduzca un numero entero:");

scanf("%d",&a);

if(a<0)

{

for(cont=cuenta=suma=0;cont<6;cont++)

{

printf("\nIntroduzca otro numero entero:");

scanf("%d",&b);

if(b%a==0)

{

suma=suma+b;

cuenta++;

}

}

if(cuenta!=0)

printf("\nLa media de los %d numeros multiplos de %d es %.2f\n",cuenta,a,(float)suma/cuenta);

else

printf("\nNo hay posibilidad de media");

}

else

{

```

```

for(fact=1,cont=a;cont>0;cont--)
fact=fact*cont;
printf("\nEl factorial de %d es %d\n",a,fact);
}
}

```

Ejercicio 33

*/*Pedir un entero si es mayor que cero, mostrar por pantalla sus divisores,
si no lo es mostrar su tabla de multiplicar */*

```

#include<stdio.h>

void main()
{
int a,cont;

printf("Introduzca un numero entero:");

scanf("%d",&a);

if(a>0)
{
printf("\nLos divisores de %d son:\n",a);

for(cont=1;cont<=a; cont++)

if(a%cont==0)

printf("%d\t",cont);

printf("\n");
}

else
{
printf("\nLa tabla de multiplicar de %d es:\n",a);

for(cont=1;cont<=10;cont++)

```

```
printf("\n%d * %d = %d\n",a,cont,a*cont);

}

}
```

Ejercicio 34

```
/*Mostrar por pantalla todos los numeros del 1 al 100*/

#include<stdio.h>

void main()

{

int cont;

for(cont=1;cont<=100;cont++)

printf("%d\t",cont);

}
```

Ejercicio 35

```
/* Mostrar todos los numeros impares del 1 al 100*/

#include<stdio.h>

void main ()

{

int cont;

for(cont=1;cont<100;cont=cont+2)

{

printf(" %d\t",cont);

}

}
```

SENTENCIA SWITCH

Es la sentencia de selección múltiple, se utiliza para expresar varias condiciones de igualdad excluyentes, es equivalente a optimizar varios if.....else encadenados, no se puede utilizar para examinar una variable real, solo variables enteras.

Su estructura es del tipo:

switch(variable) /* No lleva < ; > , si lo ponemos no compila */

{ /* La variable no puede ser float pues no compila*/

case valor 1:

_____;

_____;

_____;

break;

case valor 2:

_____;

_____;

break;

case valor 3:

_____;

_____;

break;

case valor 4:

_____;

_____;

break;

case valor 5:

_____ /* Se ejecuta en caso de que la variable tome

_____ cualquiera de esos 3 valores

case valor 6:

_____;

default:

_____ ; /* Se ejecuta en caso de que la variable no tome

_____ ; ninguno de los valores anteriores */

Ejercicio 36

/ Pedir un numero entero y mostrar por pantalla el dia de la semana correspondiente, tanto si es 6 como si es 7 escribir fin de semana.*

En caso de que el numero no corresponda a ningun dia mostrar un mensaje de error/*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int a;
```

```
printf("introduce un num:");
```

```
scanf("%d",&a);
```

```
switch(a)
```

```
{
```

```
case 1:
```

```
printf("Lunes\n");
```

```
break;
```

```
case 2:
```

```
printf("Martes\n");
```

```
break;
```

```
case 3:
```

```
printf("Miercoles\n");
```

```
break;
```

```
case 4:
```

```
printf("Jueves\n");
```

```
break;
```

case 5:

```
printf("Viernes\n");
```

```
break;
```

case 6:

case 7:

```
printf("Fin de semana\n");
```

```
break;
```

default:

```
printf("La cagaste\n");
```

```
}
```

```
}
```

La sentencia <break> lo que hace es que una vez que sea introducido el número que solo salga lo que queremos. En el último del todo no hace falta el break

Vamos haciendo por parte, primero el switch sin nada, compilamos, luego el caso 1 y compilamos, etc.

Ejercicio 37

/ Pedir un numero entero si dicho numero vale*

1: Pedir un entero y escribir su factorial.

2: pedir un entero y escribir su tabla de multiplicar.

3: Pedir un entero y si es mayor que cero escribir sus divisores.

*4: Pedir 6 enteros y calcular la media de los impares. */*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int x,a,cont,f,cim;
```

```
printf("Introduce una opcion:");
```

```
scanf("%d",&x);
```

```
switch(x)
```



```

{
case 1:

printf("Introduce un numero:");

scanf("%d",&a);

if(a>0)
{
cont=1;

f=1;

while(cont<=a)
{
f= f * cont;

cont++;

}

printf("El factorial de %d es %d\n",a,f);

}

else

printf("Los numero negativos no tienen factorial\n");

break;

case 2:

printf("Introduce un numero:");

scanf("%d",&a);

for(cont=0;cont<=10;cont++)

printf("%d x %d = %d\n",a,cont,a*cont);

break;

case 3:

printf("Introduce Un Numero:");

```

```

scanf("%d",&a);

if(a>0)

{

printf("Los divisores de %d son:\n",a);

for(cont=1;cont<=a;cont++)

{

if(a%cont==0)

printf("%d\t",cont);

}

printf("\n");

}

break;

case 4:

cim=0;

cont=0;

f=0;

while(cont<6)

{

printf("Introduce un numero:");

scanf("%d",&a);

if(a%2!=0)

{

f=f+a;

cim++;

}

cont++;

```

```

}

printf("La media de los numeros impares es %.2f\n", (float)f/cim);

break;

}

}

```

BUCLE DO....WHILE

El bucle < **do.....while**>, preferentemente este tipo de bucle se evitará.

Su estructura es:

do

```

{
_____;
_____;
_____;

```

} **while**(condicion); /* Se coloca a la altura de la llave final para poder identificarla, pero no tiene obligación de ser así */

/* Al final lleva < ; > */

Diferencia con el bucle <while>

Bucle: Bucle:

do while(condicion)

```

{ {
_____; _____;
_____; _____;

}while (condicion); }

```

La condición se comprueba al final La condición se comprueba al inicio

Lo que supone que el cuerpo del bucle lo que supone que el cuerpo del

Se ejecuta de una a n veces (al menos bucle se ejecuta de cero a n veces.

una vez). Puede que desde el principio no se

Solo se utilizará pues cuando se sepa a cumpla la condición.

ciencia cierta que se va a cumplir.

Ejemplo:

Se tienen dos números $a > 0$ y $b > 0$ ($a \neq 0$ y $b \neq 0$) calcular $a > b$

```
cont = 0; cont = 0;
```

```
p = 1; p = 1;
```

```
while(cont<b) do
```

```
{ {
```

```
p=p*a; p=p*a;
```

```
cont++; cont ++;
```

```
} }while (cont<b);
```

comprobación 1

a b cont p

3 2 0 1

- 3
- 9

comprobación 2 comprobación 2

si funciona, el exponente es 0 a b cont p

entonces $p = 1$ (no entra en while) 3 0 0 1

1 3 /* No se cumple*/

El do.....while sirve para hacer un menú

El ejercicio anterior se mejorara con la introducción de un menu mediante el bucle do...while

Ejercicio 38

/ Pedir un numero entero si dicho numero vale*

1: Pedir un entero y escribir su factorial.

2: pedir un entero y escribir su tabla de multiplicar.

3: Pedir un entero y si es mayor que cero escribir sus divisores.

4: Pedir 6 enteros y calcular la media de los impares. */

/*Vamos a incluir un menu que nos indique que es lo que queremos que haga mediante el do while y escribiendo mensajes */

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int x,a,cont,f,cim;
```

```
do{
```

```
printf("1.Factorial\n");
```

```
printf("2.Tabla de multiplicar\n");
```

```
printf("3.Divisores\n");
```

```
printf("4.Media impares\n");
```

```
printf("5.Salir\n");
```

```
printf("\n");
```

```
printf("Introduce una opcion:");
```

```
scanf("%d",&x);
```

```
switch(x)
```

```
{
```

```
case 1:
```

```
printf("Introduce un numero:");
```

```
scanf("%d",&a);
```

```
if(a>0)
```

```
{
```

```
cont=1;
```

```
f=1;
```

```
while(cont<=a)
```

```

{
f= f * cont;

cont++;

}

printf("El factorial de %d es %d\n",a,f);

}

else

printf("Los numero negativos no tienen factorial\n");

break;

case 2:

printf("Introduce un numero:");

scanf("%d",&a);

for(cont=0;cont<=10;cont++)

printf("%d x %d = %d\n",a,cont,a*cont);

break;

case 3:

printf("Introduce Un Numero:");

scanf("%d",&a);

if(a>0)

{

printf("Los divisores de %d son:\n",a);

for(cont=1;cont<=a;cont++)

{

if(a%cont==0)

printf("%d\t",cont);

}

}

```

```

printf("\n");

}

break;

case 4:

cim=0;

cont=0;

f=0;

while(cont<6)

{

printf("Introduce un numero:");

scanf("%d",&a);

if(a%2!=0)

{

f=f+a;

cim++;

}

cont++;

}

printf("La media de los numeros impares es %.2f\n",(float)f/cim);

break;

}

}while(x!=5);

}

```

Utilizaremos el bucle do....while para hacer menús y/o para pedir un dato que cumpla unas determinadas condiciones

Ejemplos:

Pedir un número mayor que cero

```

do

{

printf(-----);

scanf(%d,&a);

}while (a<=0); /* La condición de while es la opuesta a lo que se pide */

Pedir un número par

do

{

printf(-----);

scanf(-----);

}while(a%2!=0);

```

Ejercicio 39

/*Pedir un numero entero mayor que cero e indicar cuantas cifras tiene*/

```

#include<stdio.h>

void main()

{

int a,cont,aux;

do{

printf("Introduzca un numero entero mayor que cero:");

scanf("%d",&a);

}while(a<=0);

cont=0;

aux=a;

while(aux!=0)

{

cont++;

```



```

aux=a/10;

}

printf("El numero introducido tiene %d cifras\n",cont);

}

```

Ejercicio 40

*/*Pedir un numero entero mayor que cero, si tiene una cantidad de cifras impar
mostrar sus divisores y en caso contrario, mostrar su tabla de multiplicar*/*

```

#include<stdio.h>

void main()

{

int a,cont,c,aux;

do{

printf("Introduzca un numero entero mayor que cero:");

scanf("%d",&a);

}while(a<=0);

cont=0;

aux=a;

while(aux!=0)

{

cont++;

aux=aux/10;

}

printf("El numero %d tiene %d cifras\n",a,cont);

if(cont%2!=0)

{

printf("Los divisores de %d son:\n",a);

```

```

for(c=1;c<=a;c++)
{
    if(a%c==0)
        printf("%d\t",c);
}
printf("\n");
}
else
{
    printf("la tabla del %d\n",a);
    for(cont=1;cont<=10;cont++)
        printf("%d x %d = %d\n",a,cont,a*cont);
}
}

```

Ejercicio 41

*/*Pedir un entero mayor que cero, calcular y escribir la suma de sus cifras*/*

```

#include<stdio.h>

void main()
{
    int a,suma,aux,cifra;

    do{
        printf("Introduzca un numero entero:");

        scanf("%d",&a);

    }while(a<0);

    for(suma=0,aux=a;aux!=0; )
    {

```

```

cifra=aux%10;

suma=suma+cifra;

aux=aux/10; /*el orden en que se hace es importante*/

}

printf("\nLa suma de las cifras de %d es %d\n",a,suma);

}

```

Ejercicio 42

*/*Pedir un entero mayor que cero, calcular la suma de sus cifras impares
y contar sus cifras pares*/*

```

#include<stdio.h>

void main()

{

int suma,a,aux,cifra,cont;

do{

printf("Introduzca un numero entero:");

scanf("%d",&a);

}while(a<0);

for(suma=cont=0,aux=a;aux!=0;aux=aux/10)

{

cifra=aux%10;

if(cifra%2!=0)

suma=suma+cifra;

else

cont++;

}

printf("\nLa suma de las cifras impares de %d es %d\n",a,suma);

```

```
printf("\nLas cifras pares de %d es %d\n",a,cont);
```

```
}
```

Ejercicio 43

*/*Pedir un entero mayor que cero, si es mayor que mil contar cuantos treses tiene y si no lo es indicar mediante un mensaje, si es un numero perfecto o no lo es. Un numero es perfecto si es igual a la suma de todos sus divisores exceptuandolo a él, ejemplo 6 28 496*/*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int a,suma,cont,cifra,aux;
```

```
do{
```

```
printf("Introduzca un numero entero:");
```

```
scanf("%d",&a);
```

```
}while(a<0);
```

```
if(a>1000)
```

```
{
```

```
for(aux=a,cont=cifra=0;aux!=0;aux=aux/10)
```

```
{
```

```
cifra=aux%10;
```

```
if(cifra==3)
```

```
cont++;
```

```
}
```

```
printf("\nEl numero %d tiene repetido el 3 %d veces\n",a,cont);
```

```
}
```

```
else
```

```

{
for(cifra=1,suma=0;cont<a;cont++)
{
if(a%cont==0)
suma=suma+cont;
}
if(suma==a)
printf("\nEl numero es perfecto\n");
else
printf("\nEl numero no es perfecto\n");
}
}

```

Ejercicio 44

```

/*Pedir un numero entero mayor que cero e indicar si es capicua */
#include<stdio.h>

void main()
{
int a,aux,cifra,revs;

do{
printf("Introduzca un numero entero:");

scanf("%d",&a);
}while(a<=0);

aux=a; /*Para no perderlo se guarda en un aux*/

revs=0;

cifra=0;

while(aux!=0)

```

```

{
cifra=aux%10; /*Almacenamos los restos de div por 10*/
reves=reves*10+cifra; /*Utilizamos la ponderación base 10*/
aux=aux/10;
}
if(reves==a)
printf("\nEl numero es capicua\n");
else
printf("\nEl numero no es capicua\n");
}

```

ejemplo:

a aux reves cifra(resto de dividir por 10)

2941 2941 0 1

*294 1 4 /*como puede verse se va añadiendo la*

29 14 9 cifra a la variable reves, utilizando la ponderacion de base 10/*

149

Ejercicio 45

*/*Pedir un entero mayor que cero, si es capicua mostrar sus divisores y si no*

lo es, mostrar todos los numeros que hay entre 1 y él, ambos incluidos en orden

descendente/*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int a,aux,cifra,reves,cont;
```

```
do
```

```
{
```

```

printf("Introduzca un numero entero:");

scanf("%d",&a);

}while(a<=0);

aux=a;

reves=0;

cifra=0;

while(aux!=0)

{

cifra=aux%10;

reves=reves*10+cifra;

aux=aux/10;

}

if(reves==a)

{

printf("\nEl numero es capicua y sus divisores son\n");

for(cont=1;cont<=a;cont++)

{

if(a%cont==0)

printf("%d\t",cont);

printf("\n");

}

}

else

{

printf("\nEl numero no es capicua asi que:\n");

for(cont=a;cont>0;cont--)

```

```

{
printf("%d\t",cont);

}

printf("\n");

}

}

```

Bucles anidados

Dos bucles están anidados si un bucle está dentro de otro. Para hacer bucles anidados hay que tener en cuenta dos cuestiones:

1ª) Las variables que controlan los bucles deben ser diferentes.

2ª) Antes de cada bucle, deben inicializarse los controladores y acumuladores que se vayan a utilizar en dicho bucle.

La forma de su estructura es:

While(condicion) /* Las variables se inicializan antes de cada bucle */

```

{
_____;
_____;
while(condicion)
{
_____;
_____;
}
_____;
_____;
}

```

Ejercicio 46

*/*Pedir 6 numeros enteros y mostrar la tabla de multiplicar de cada uno de ellos,*


```

utilizar bucles while*/

#include<stdio.h>

void main()

{

int a,c1,c2tabla;

c1=0;

while(c1<6)

{

printf("Introduzca un numero entero:");

scanf("%d",&a);

printf("\nLa tabla de multiplicar de %d es:\n",a);

c2tabla=1;

while(c2tabla<=10)

{

printf("%d * %d = %d\n",a,c2tabla,a*c2tabla);

c2tabla++;

}

c1++;

}

}

```

Realizar el siguiente ejercicio mediante un bucle externo *while* y un bucle interno *for*

Ejercicio 47

```

/*Pedir 6 numeros enteros mostrar el factorial de los que sean mayores o
iguales que cero y calcular la media de los que no lo sean*/

#include<stdio.h>

void main()

```

```

{
    int a,suma,cont,fact,c,c2;

    cont=0;

    suma=0;

    c2=0;

    while(cont<6)
    {
        printf("Introduzca un numero entero:");

        scanf("%d",&a);

        cont++;

        if(a>=0)
        {
            for(c=fact=1;c<=a;c++)

                fact=fact*c;

            printf("\nEl factorial de %d es %d\n",a,fact);
        }

        else

        {
            suma=suma+a;

            c2++;
        }
    }

    if(c2!=0)

        printf("\nLa media de los que son menores que cero es %.2f\n",(float)suma/c2);

    else

        printf("\nNo se puede hacer la media\n");
}

```

```
}
```

Ejercicio 48

*/*Pedir 6 numeros enteros de cada uno que sea mayor que cero, mostrar sus divisores e indicar la suma de los mismos y de los que no sean mayores que cero, mostrar la tabla de multiplicar (todo con bucles FOR)*/*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int a,c,cont,contdiv,suma;
```

```
for(cont=0;cont<6;cont++)
```

```
{
```

```
printf("Introduzca un numero entero:");
```

```
scanf("%d",&a);
```

```
if(a>0)
```

```
{
```

```
printf("\nLos divisores de %d son:\n",a);
```

```
for(contdiv=1,suma=0;contdiv<=a;contdiv++)
```

```
{
```

```
if(a%contdiv==0)
```

```
{
```

```
printf("%d\t",contdiv);
```

```
suma=suma+contdiv;
```

```
}
```

```
}
```

```
printf("\nLa suma de los divisores es %d\n",suma);
```

```
}
```

```

else

{

printf("\nLa tabla de multiplicar de %d es\n",a);

for(c=0;c<=10;c++)

printf("%d * %d = %d\n",a,c,a*c);

}

}

}

```

Realización del test 3

Ejercicio 49

*/*Pedir 6 numeros enteros de los que sean mayores que cero contar los que sean capicuas, y de los que no sean mayores que cero, mostrar la tabla de multiplicar*/*

```

#include<stdio.h>

void main()

{

int a,aux,cifra,revs,cont,capicuas,cont2;

for(cont=0,capicuas=0;cont<6;cont++)

{

printf("introduzca un numero entero:");

scanf("%d",&a);

if(a>0)

{

aux=a;

revs=0;

cifra=0;

```

```

while(aux!=0)
{
cifra=aux%10;
reves=reves*10+cifra;
aux=aux/10;
}
if(reves==a)
capicuas++;
}
else
{
printf("\nLa tabla de multiplicar de %d es\n",a);
for(cont2=1;cont2<=10;cont2++)
printf("%d * %d = %d\n",a,cont2,a*cont2);
}
}
printf("\nLos numeros capicuas han sido %d\n",capicuas);
}

```

Ejercicio 50

*/*Pedir 6 numeros enteros y calcular la suma de sus factoriales*/*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int a,cont,fact,cont2,suma;
```

```
for(cont=0;cont<6;cont++)
```

```
{
```

```

printf("Introduzca un numero entero:");

scanf("%d",&a);

if(a>=0)

cont2=fact=1;

suma=0;

while(cont2<=a)

{

fact=fact*cont2;

suma=suma+fact;

cont2++;

}

printf("\nLa suma de los factoriales de %d es %d\n",a,suma);

}

}

```

FUNCIONES EN C:

Los programas se pueden dividir en subprogramas, que no son más que trozos de programa que tienen un nombre y que se pueden utilizar cuando sea necesario sin más que llamarlos o invocarlos.

Los programas de C como mínimo tienen siempre una función: *main()*, que es por donde empieza siempre la ejecución. Cuando dentro de *main()* se detecta la llamada a una función, el control pasa entonces a dicha función y se realizan entonces todas sus sentencias. Cuando estas terminan, el control de la ejecución pasa de nuevo al punto de llamada.

Una **función** es una colección de declaraciones y sentencias que sirven para resolver una determinada tarea. Puede ser utilizada cuantas veces sea necesario para lo cual sólo se precisa efectuar una llamada cada vez que sea preciso.

Parámetro: Dato que se le envía a la función cuando se efectúa una llamada a la misma.

Las funciones en C pueden ser de dos clases :

- las que no retornan nada al punto de llamada (realizan cálculos y muestran resultados)
- las que retornan un dato al punto de llamada (realizan cálculos y devuelven un dato al punto de llamada para que sea utilizado en él)

A la hora de hacer funciones en C hay que tener en cuenta tres aspectos:

- Prototipo o declaración de la función
- Llamada a la función
- Definición de la función
-

Prototipo de la función

Los prototipos de las funciones van colocados después de los *#include* y antes de *main()*. En un prototipo se indica:

- el tipo de dato que retorna la función al punto de llamada
- el nombre de la función
- los tipos de dato de los diferentes parámetros que se le envían a la función al llamarla

La forma general de un prototipo es:

tipo nombre_funcion(tipo,tipo,...);

tipo de dato tipos de los parámetros

devuelto por la función

Ejemplos:

int f1 (int,int); El tipo es int, la función denominada f1 que recibe dos enteros como parámetros y retorna un entero al punto de llamada.

float f2 (int); Función denominada f2 que recibe un entero como parámetro y retorna un real al punto de llamada.

int f3(int,char); Función denominada f3 que recibe un entero y un carácter como parámetros y retorna un entero al punto de llamada.

int f4(); También se puede poner **int f4(void);**: Función denominada f4 que no recibe parámetros y retorna un entero al punto de llamada.

void f5(int,float); Función denominada f5 que recibe un entero y un real con parámetros y no retorna nada al punto de llamada.

Llamada a la función

Se hace de forma diferente según que la función retorne un valor o no retorne nada.

Si la función no retorna nada la llamada se efectúa:

nombre_funcion(param1,param2,...);

Si la función retorna un dato al punto de llamada, la llamada a dicha función debe hacerse SIEMPRE desde el interior de una expresión. A la hora de efectuar la llamada a este tipo de funciones hay que determinar qué es lo que se quiere hacer en el punto de llamada con el valor que retorne la función.

Por ejemplo, si tenemos una función cuyo prototipo sea:

int f(int):

desde main() podríamos llamar a la función:

printf(el resultado es %d\n,f(a)); Si lo que queremos es escribir el valor retornado por la función.

x = f(a); Si lo que queremos es guardar en una variable el valor retornado por la función.

x = x + f(a); Si lo que queremos es sumar el valor retornado por la función.

Definición de la función

Consta de una cabecera y un cuerpo. La cabecera es igual que el prototipo pero indicando los nombres de los parámetros recibidos. El cuerpo de una función es un conjunto de declaraciones y sentencias entre llaves. La función termina con una sentencia **return** que retorna el control al punto de llamada.

A la hora de escribir funciones hay que tener en cuenta 2 aspectos fundamentales:

1º.- DEBE haber una CONCORDANCIA TOTAL en cuanto a NUMERO Y TIPO DE LOS PARÁMETROS del prototipo, de la llamada y de la definición.

2ª.- Los parámetros que aparecen en la llamada se denominan parámetros o argumentos actuales y los que aparecen en la cabecera de la función se denominan parámetros o argumentos formales. Cuando se efectúa la llamada a una función, los parámetros formales reciben una copia de los actuales. En realidad al llamar a una función no se le envían datos, sino copias de los mismos. Estas copias (los parámetros formales) pueden recibir cualquier nombre (igual o diferente al de los parámetros actuales).

Las variables que se declaran en una función se denominan variables locales y se destruyen cuando la función termina. Pueden existir variables locales con el mismo nombre en distintas funciones porque en realidad son distintas. Desde una función no se pueden utilizar las variables locales de otra función, o sea las variables locales de una función sólo las puede utilizar dicha función.

Sobre un programa ejemplo analizamos estos apuntes.

```
#include<stdio.h>
```

```
void tabla(int); /* prototipo o declaración de la función, void no retorna nada */
```

```
void main( )
```

```
{
```

```
int a,b;
```

```
printf(Introduzca un numero:);
```

```
scanf(%d,&a);
```

```
tabla(a); /*llamada a la función tabla, hace lo que sea y vuelve. <<a>> es parámetro actual */
```

```
printf(Introduce otro);
```



```

scanf(%d,&b);

tabla(b); /* otra llamada a la función tabla, hace lo que se pide y vuelve */

tabla(a+b);

tabla(3);

}

/* Viene la definición de la función */

void tabla(int num) /*No lleva <<;>>, es la cabecera de la función. num se declara en la cabecera dentro no y es el parámetro formal (toma el valor del parámetro que aparece en la llamada */

{

int c;

printf(Tabla del %d\n,num);

for (c= 1; c<=10;c+ +)

printf(%d * %d = %d\n,num,c,num*c);

return; /* Solo es obligatorio ponerlo si retorna un dato para trabajar mejor acostumbrarse a ponerlo */

}

```

Como resumen:

main() lo dirige todo y manda a otras funciones que hagan cosas...

Ejercicio 51

*/*Pedir un numero entero mayor o igual que cero y acontinuacion llamar a una*

funcion que calcule y escriba su factorial/*

```
#include<stdio.h>
```

```
void factorial(int);
```

```
void main()
```

```
{
```

```
int a;
```

```
do
```

```
{
```

```

printf("Introduzca un numero entero:");

scanf("%d",&a);

}while(a<0);

factorial(a);

}

void factorial(int num)

{

int cont,fact;

for(cont=fact=1;cont<=num;cont++)

fact=fact*cont;

printf("\nEl factorial de %d es %d\n",num,fact);

return;

}

```

Ejercicio 52

```

/*Pedir un entero mayor que cero y llamar a una funcion que calcule y escriba la
suma de sus cifras*/

#include<stdio.h>

void funcion(int);

void main()

{

int a;

do{

printf("Introduzca un numero entero:");

scanf("%d",&a);

}while(a<=0);

funcion(a);

```

```

}

void funcion(int num)
{
    int cifra,suma,aux;

    for(suma=0,aux=num;aux!=0;)
    {
        cifra=aux%10;

        suma=suma+cifra;

        aux=aux/10;
    }

    printf("\nLa suma de las cifras de %d es %d\n",num,suma);

    return;
}

```

Ejercicio 53

*/*Pedir un numero entero si es mayor que cero, llamar a una función que cuente cuantos divisores tiene que lo indique por pantalla y si no lo es llamar a una funcion que muestre su tabla de multiplicar*/*

```
#include<stdio.h>
```

```
void divisores(int);
```

```
void tablamult(int);
```

```
void main()
```

```
{
```

```
    int a;
```

```
    printf("Introduzca un numero entero:");
```

```
    scanf("%d",&a);
```

```
    if(a>0)
```

```

divisores(a);

else

tablamult(a);

}

void divisores(int num)

{

int cont,contdiv;

for(cont=1,contdiv=0;cont<=num;cont++)

if(num%cont==0)

contdiv++;

printf("\nLos divisores de %d son %d\n",num,contdiv);

}

```

Ejercicio 54

*/*Pedir seis numeros enteros con cada uno de ellos que sea mayor que cero*

llamar a una funcion que indique por pantalla si es o no capicua/*

```
#include<stdio.h>
```

```
void capicua(int);
```

```
void main()
```

```
{
```

```
int a,cont;
```

```
cont=0;
```

```
while(cont<6)
```

```
{
```

```
printf("Introduzca un numero entero:");
```

```
scanf("%d",&a);
```

```
cont++;
```

```

    if(a>0)
        capicua(a);
    }
}

void capicua(int num)
{
    int aux, reves, cifra;

    aux=num;
    reves=0;
    cifra=0;
    while(aux!=0)
    {
        cifra=aux%10;
        reves=reves*10+cifra;
        aux=aux/10;
    }

    if(reves==num)
        printf("\nEl numero es capicua\n");
    else
        printf("\nEl numero no es capicua\n");
    return;
}

```

Ejercicio 55

*/*Pedir un numero entero distinto de cero a continuacion pedir otro mayor o igual que cero y llamar a una funcion que calcule y escriba el resultado de elevar el primero al segundo*/*

```

#include<stdio.h>

void potencia(int,int);

void main()
{
    int a,b;

    do
    {
        printf("Introduzca un numero entero distinto de cero:");

        scanf("%d",&a);

    }while(a==0);

    do
    {
        printf("Introduzca otro numero entero mayor o igual que cero:");

        scanf("%d",&b);

    }while(b<=0);

    potencia(a,b);

}

void potencia(int a,int b)
{
    int cont,producto;

    cont=0;

    producto=1;

    while(cont<b)
    {

        producto=producto*a;

        cont++;
    }
}

```

```

}

printf("%d elevado a %d es %d\n",a,b,producto);

return;

}

```

Ejercicio 56

/ Pedir un numero entero mayor que cero y llamar a una funcion que lo cambie de base 10 a base 5*/*

```

#include<stdio.h>

void cambio_base(int);

void main()

{

int a;

do{

printf("Introduzca un numero entero mayor que cero:");

scanf("%d",&a);

}while(a<=0);

cambio_base(a);

}

void cambio_base(int num)

{

int aux,x,resto,cont;

x=0;

aux=num;

cont=1;

while(aux!=0)

{

```

```

    resto=aux%5;

    x=resto*cont+x;

    cont=cont*10;

    aux=aux/5;

}

printf("\nEl numero %d a base %d es %d\n",num,5,x);

return;

}

```

Cuando la variable es muy grande entonces utilizaremos el comando **long int**. Para escribirlo se utilizará la siguiente expresión: **%ld**.

Volvemos sobre funciones, esta vez las que retornan un dato, lo hacemos mediante un ejercicio ejemplo:

```

#include<stdio.h>

int factorial(int);

void main( )

{

    int a;

    do {

        printf(Número ?:);

        scanf(%d,&a);

    }while(a<0);

    printf(El factorial de %d es %d\n,a,factorial(a));

}

int factorial(int a)

{

    int c,f;

    for(c=f=1;c<=a;c+ +)

        f=f*c;

```



```
return f; /* retorna el valor de f al lugar de la llamada */
```

```
}
```

Ejercicio 57

*/*Pedir un numero entero mayor que cero e indicar cuantas cifras tiene.*

Hacer una funcion que cuente las cifras de un numero

y escribir en "main" el resultado/*

```
#include<stdio.h>
```

```
int cuenta_cifras(int);
```

```
void main()
```

```
{
```

```
int a;
```

```
do{
```

```
printf("Introduzca un numero entero mayor que cero:");
```

```
scanf("%d",&a);
```

```
}while(a<=0);
```

```
printf("\nEl numero %d tiene %d cifras\n",a,cuenta_cifras(a));
```

```
}
```

```
int cuenta_cifras(int a)
```

```
{
```

```
int cont,aux;
```

```
cont=0;
```

```
aux=a;
```

```
while(aux!=0)
```

```
{
```

```
cont++;
```

```
aux=aux/10;
```

```

}

return cont;

}

```

Ejercicio 58

*/*Pedir un numero entero mayor que cero, si la suma de sus cifras es un numero*

impar mostrar su tabla de multiplicar. Hay que hacer una funcion que reciba un numero entero y retorne la suma de sus cifras y otra funcion que reciba un numero entero y muestre su tabla de multiplicar/*

```

#include<stdio.h>

int suma_cifras(int);

void tabla_multiplicar(int);

void main()

{

int a;

do{

printf("Introduzca un numero entero mayor que cero:");

scanf("%d",&a);

}while(a<=0);

if(suma_cifras(a)%2!=0)

printf(tabla_multiplicar(a);

}

int suma_cifras(int a)

{

int suma,aux,cifra;

for(suma=0,aux=a;aux!=0; )

{

cifra=aux%10;

suma=suma+cifra;

```

```

aux=aux/10;

}

return suma;

}

void tabla_multiplicar(int a)

{

int cont;

printf("\nLa tabla de multiplicar de %d es:\n");

for(cont=1;cont<=a;cont++)

printf("%d * %d = %d\n,a a*cont);

}

```

Ejercicio 59

*/*Pedir 6 numeros enteros de aquellos que sean mayores que cero contar cuantos tienen mas de dos unos. Es necesario hacer una función que reciba un numero entero y retorne el numero de unos que tiene*/*

```

#include<stdio.h>

int contador(int);

void main()

{

int a,cont,c2;

cont=c2=0;

while(cont<6)

{

printf("Introduzca un numero entero:");

scanf("%d",&a);

cont++;

```

```

if(a>0)

if(contador(a)>2)

c2++;

}

printf("\nHay %d numeros tienen repetido el uno 2 veces\n",c2);

}

int contador(int a)

{

int aux,cuentaunos,cifra;

for(aux=a,cuentaunos=cifra=0;aux!=0;aux=aux/10)

{

cifra=aux%10;

if(cifra==1)

cuentaunos++;

aux=aux/10;

}

return cuentaunos;

}

```

Ejercicio 60

```

/*Pedir 6 numeros enteros y calcular la suma de sus factoriales. Hay que hacer una
función que calcule el factorial de un numero y lo retorne*/

#include<stdio.h>

int factorial(int);

void main()

{

int a,cont,suma;

```

```

cont=0;

suma=0;

while(cont<6)

{

printf("Introduce un numero:");

scanf("%d",&a);

cont++;

if(a<0)

printf("\nNo hay factorial");

else

suma=suma+factorial(a);

}

printf("\nLa suma de sus factoriales es %d\n",suma);

}

int factorial(int a)

{

int cfact,fact;

for(cfact=fact=1;cfact<=a;cfact++)

fact=fact*cfact;

return fact;

}

```

Ejercicio 61

*/*Pedir un numero entero mayor que cero e indicar si es o no es capicua*

Hay que hacer una funcion que reciba un numero y lo retorne al revés. El

mensaje correspondiente debe escribirse en main/*

```
#include<stdio.h>
```

```

int reves(int);

void main()

{

int a;

do{

printf("Introduzca un numero mayor que cero:");

scanf("%d",&a);

}while(a<=0);

if(reves(a)==a)

printf("\nEl numero es capicua\n");

else

printf("\nEl numero no es capicua\n");

}

int reves(int a)

{

int aux,rev,cifra;

aux=a;

rev=0;

cifra=0;

while(aux!=0)

{

cifra=aux%10;

rev=rev*10+cifra;

aux=aux/10;

}

return rev;

```

}

TEST I

1.- ¿Es correcto en C el siguiente comentario?

```
/* este es un comentario/*
```

2.- ¿Es `cont>0` un identificador correcto?

3.- Indica los errores que observes en el siguiente programa.

```
#include<studio.h>
```

```
void main( );
```

```
{
```

```
int a
```

```
a= = 2;
```

```
if(a>0);
```

```
printf(Hola);
```

```
else
```

```
printf(Adios);
```

```
printf(%d\n,2*a)
```

```
printf(%d\n 3*a)
```

```
}
```

4.- Indicar que es lo que escribe este programa.

```
a=5;
```

```
if(a%3= =0)
```

```
printf(%d\n,2*a);
```

```
else
```

```
printf(%d\n, 3*a);
```

```
printf(%d\n, a*a);
```

5.- En el ejercicio anterior¿ y si `<<a>>` se inicializa a 9?

6.- Que aparecerá en pantalla?

```
a=9;

if(a%3==0)

printf("%d\n,2*a);

else

{

printf("%d\n, 3*a);

printf("%d\n, a*a);

}
```

7.- Idem

```
a=5;

b=2;

c=a;

a=b;

b=b+4;

printf("%d %d %d\n, a,b,c);
```

Indica el resultado de las siguientes operaciones

8.- $2 - 8 * 2 + 10 / 2 * 3$

9.- $20 / 5 * 2 - 3 * 1 + 2$

10.- $20 / (5*2) - 3 * 1 + 2$

Resultados TEST I

1.- NO. Al final está al revés el asterisco y la barra.

2.- NO. Un identificador no lleva operadores lógicos.

3.- Hay 8 errores. 1ª línea sobra la <<U>>

2ª línea falta <<;>>

3ª línea falta <<;>>

4ª línea sobra un signo < = > no es asignación.

5ª línea sobra un <<;>>

9ª línea falta <<;>>

10ª línea falta una <<, >> dentro del paréntesis y <<;>> al final.

4.- 15 <<salta una línea>> y escribe 25

5.- 18 <<salta una línea>> y escribe 81

6.- 18

7.- 2 6 5 <<salta una línea>>

8.- 15

9.- 7

10.- 1

TEST II

¿Qué es lo que escribe en la pantalla?

1.- a = 2;

if(a>3)

printf(Hola);

else

printf(Adios);

2.- a = 2;

if(a<3)

printf(Hola);

else

printf(Adios);

3.- a = 2;

if(a<3);

printf(Hola);

```

else

printf(Adios);

4.- a = 2;

if(a>3);

printf(Hola);

5.- a = 2;

if(a<3);

printf(Hola);

6.- a =0;

while (a!= 0)

{

printf(Hola);

a+ +;

}

7.- a = 1;

while (a!=0)

{

printf(Hola);

a+ +;

}

8.- a = 1;

while(a!=0);

{

printf(Hola);

a+ +;

}

```

```

9.- a = 0;

while(a!=0);

{

printf(Hola);

a+ +;

}

```

Resultados TEST II

- 1.- Adios
- 2.- Hola
- 3.- No compila
- 4.- Hola
- 5.- Hola
- 6.- NADA
- 7.- Hola e incrementa <<a>> bucle infinito
- 8.- Se queda atrapado en el bucle y no hace nada
- 9.- Hola e incrementa <<a>> una vez.

Ejercicio 62

/*Pedir un entero mayor que cero e indicar si es o no es capicua.

Hay que hacer una función que reciba un entero y retorne un uno si dicho

entero es capicua y que retorne cero en caso contrario*/

```
#include<stdio.h>
```

```
int capicua(int);
```

```
void main()
```

```
{
```

```
int a;
```

```
do{
```

```

printf("Introduzca un numero entero mayor que cero:");

scanf("%d",&a);

}while(a<=0);

if(capicua(a)==1)

printf("\nEl numero es capicua\n");

else

printf("\nEl numero no es capicua\n");

}

int capicua(int a)

{

int aux,reves,cifra;

aux=a;

reves=0;

cifra=0;

while(aux!=0)

{

cifra=aux%10;

reves=reves*10+cifra;

aux=aux/10;

}

if(reves==a)

return 1;

else

return 0;

}

```

Ejercicio 63

*/*Mostrar por pantalla los 20 primeros numeros capicuas*/*

#include<stdio.h>

int capicua(int);

void main()

{

int a,cont;

cont=0;

a=1;

while(cont<20)

{

if(capicua(a)==1)

{

printf("%d\t",a);

cont++;

}

a++;

}

}

int capicua(int a)

{

int aux,reves,cifra;

aux=a;

reves=0;

cifra=0;

while(aux!=0)

{

```

cifra=aux%10;

reves=reves*10+cifra;

aux=aux/10;

}

if(reves==a)

return 1;

else

return 0;

}

```

En C todo lo que es distinto de 0 es cierto y todo lo que vale 0 es falso

Ejemplo:

if(a!=0) equivale a if(a)Se pregunta si <<a>> es cierto

if(a= 0) equivale a if(!a)..Se pregunta si <<a>> es falso

Caso de errores en asignación o igualación

Ejemplo:

```

a= 5;

if (a = 3) /*Compara si <<a>> y 3 son iguales*/

printf(x);

else

printf(y);

a= 5;

if (a =3) /*Asigna a <<a>> el valor 3 y averigua si <<a>> es cierto,o sea si es distinto de 0*/

printf(x); /* escribe <<x>>, si a=0 es entonces asigna a = 0 y escribiría <<y>>

else

printf(y);

```

En C la forma en que por defecto se envían parámetros a una función es por valor, esto significa que lo que la función recibe es una copia de lo que se envía. Por lo tanto cualquier cambio que dicha copia sufra dentro de

la función no afectará al dato original, que seguirá teniendo el mismo valor que antes de la llamada a la función.

Ejemplo:

```
#include<stdio.h>

void funcion (int);

void main()

{

int a=5;

funcion(a);

printf("%\n",a); /* escribirá 5 */

}

void funcion (int a)

{

a= a-3;

printf("%d\n",a); /* escribe 2*/

return;

}
```

Examen

1.- Implementar un código equivalente al siguiente pero sin utilizar *for* ni *switch*

for (i=0; i<n; i+ +)

Switch (i)

{

case 0: printf("%d, i);

case 1: printf(uno);

case 2: printf(Dos);

break

default: printf(%d,i);

```
}
```

Respuesta:

```
i = 0;
```

```
while(i<n)
```

```
if(i == 0)
```

```
printf("%d uno dos,i);
```

```
else
```

```
if (i == 2)
```

```
printf(Dos);
```

```
else
```

```
printf(%d,i);
```

```
i++;
```

```
}
```

2.- Hacer una función que muestre la posición final de un vehículo (x,y), tomando como parámetros, su posición inicial (,), un vector velocidad (, que indica la dirección y magnitud del movimiento en m/s , y el tiempo transcurrido en segundos.

Fórmula a aplicar:

Posición final = posición inicial + v * tiempo

Respuesta

```
Void funcion( , int t ) /* hay que colocar el tipo delante de cada variable*/
```

```
{
```

```
int x,y;
```

```
x = + * t;
```

```
y = + * t;
```

```
printf(La posición final %d %d es \n, x,y);
```

Ejercicio 64

```
/*Mostrar los 20 primeros numeros cuya suma de cifras sea un numero par*/
```



```

#include<stdio.h>

int suma_cifras(int);

void main()

{

int a,cont;

cont=0;

a=1;

while(cont<20)

{

if(suma_cifras(a)%2==0)

{

printf("%d\t",a);

cont++;

}

a++;

}

}

int suma_cifras(int a)

{

int suma,aux,cifra;

for(suma=0,aux=a;aux!=0; )

{

cifra=aux%10;

suma=suma+cifra;

aux=aux/10;

}

```

```
return suma;
```

```
}
```

Ejercicio 65

*/*Mostrar los n primeros numeros ("n" se pide por teclado),tales que la*

suma de sus cifras no sea divisible por 3/*

```
#include<stdio.h>
```

```
int suma_cifras(int);
```

```
void main()
```

```
{
```

```
int a,cont,n;
```

```
printf("Introduce un un numero:");
```

```
scanf("%d",&n);
```

```
cont=0;
```

```
a=1;
```

```
while(cont<n)
```

```
{
```

```
if(suma_cifras(a)%3!=0)
```

```
{
```

```
printf("%d\t",a);
```

```
cont++;
```

```
}
```

```
a++;
```

```
}
```

```
}
```

```
int suma_cifras(int a)
```

```
{
```

```

int suma,aux,cifra;

for(suma=0,aux=a;aux!=0; )

{

cifra=aux%10;

suma=suma+cifra;

aux=aux/10;

}

return suma;

}

```

Ejercicio 66

*/*Mostrar los n primeros numeros ("n" se pide por teclado),tales que la suma de sus cifras sea divisible por su última cifra*/*

```

#include<stdio.h>

int suma_cifras(int);

void main()

{

int a,cont,n;

printf("Introduce un un numero:");

scanf("%d",&n);

cont=0;

a=1;

while(cont<n)

{

if(a%10!=0)

{

if(suma_cifras(a)%(a%10)==0)

```

```

{
printf("%d\t",a);

cont++;

}

a++;

}

}

}

int suma_cifras(int a)

{

int suma,aux,cifra;

for(suma=0,aux=a;aux!=0; )

{

cifra=aux%10;

suma=suma+cifra;

aux=aux/10;

}

return suma;

}

```

Ejercicio 67 (funcion para numero primo)

*/*Escribir una funcion en C que, dado un numero devuelva 1 si es primo y
cero en caso contrario.*

*b)Usando la funcion del apartado anterior, escribir una funcion en C que
imprima los n primeros primos (n se pide por teclado), tales que la suma de
sus cifras no sea divisible por 5. Por ejemplo para "n" igual a 10, serían
1,2,3,7,11,13,17,29,31,43 ya que 5,19,23,37,41 siendo primos no cumplen*

la condición pues la suma de sus cifras es divisible por 5*/

```
#include<stdio.h>
```

```
int suma_cifras(int);
```

```
int primo(int);
```

```
void main()
```

```
{
```

```
int a,cont,n;
```

```
printf("Cuantos numeros:");
```

```
scanf("%d",&n);
```

```
cont=0;
```

```
a=1;
```

```
while(cont<n)
```

```
{
```

```
if(primo(a)==1&&suma_cifras(a)%5!=0)
```

```
{
```

```
printf("%d\t",a);
```

```
cont++;
```

```
}
```

```
a++;
```

```
}
```

```
}
```

```
int suma_cifras(int a)
```

```
{
```

```
int suma,aux,cifra;
```

```
for(suma=0,aux=a;aux!=0; )
```

```
{
```

```

cifra=aux%10;

suma=suma+cifra;

aux=aux/10;

}

return suma;

}

int primo(int a) /*cada vez que encuentre que ya
es divisible la funcion acaba*/

{

int c;

for(c=2;c<a;c++)

{

if(a%c==0)

return 0; /*nunca poner un else dentro de for*/

}

return 1;

}

```

Intercambiar valor de 2 variables

Si quiero que a = b (<<a >>coja valor de <>), entonces antes debemos guardar el valor de <<a>>

aux = a; o bien, aux = b; /*así no se pierde el valor*/

a = b; b = a;

b = aux; a = aux

Ejercicio 68

*/*Pedir dos numeros enteros y mostrar por pantalla todos los numeros que*

hay entre ambos en orden ascendente e incluyendolos a los dos/*

#include<stdio.h>

```

void main()

{

int a,b,aux,cont;

printf("Introduzca un numero entero:");

scanf("%d",&a);

printf("Introduzca otro numero entero:");

scanf("%d",&b);

if(a>b)

{

aux=b;

b=a;

a=aux;

}

for(cont=a;cont<=b;cont++) /* contador desde <<a>> hasta <<b>> */

printf("%d\t",cont);

}

```

Ejercicio 69

```

/*Mostrar los 20 primeros numeros de la sucesion de Fibonacci(Sucesion de numeros
naturales donde los dos primeros numeros son el 1 y los siguientes se hallan
sumando los dos anteriores....*/

/*Necesitamos un bucle que escriba 20 numeros, dos variables una que me guarde
el dato anterior y otra que me escriba el numero, luego tendré que escribir
el numero dentro del bucle, luego tengo que hacer un intercambio, que lo que era el
numero ahora sea anterior (mediante un auxiliar y num sea lo que era antes el
numero mas lo que era antes anterior*/

#include<stdio.h>

```

```

void main()

{

int cont,anterior,num,aux;

for(cont=0,anterior=0,num=1;cont<20;cont++)

{

printf("%d\t",num); /*tambien aux=num; num=num+ant;ant=aux*/

aux=anterior;

anterior=num;

num=num+aux;

}

}

```

ARRAYS

Es una colección de datos del mismo tipo. Pueden ser unidimensionales y se denominan vectores y bidimensionales que se denominan matrices.

Todos los datos que se almacenan en un vector se denominan elementos de dicho vector, y cada uno de ellos ocupa una posición dentro del mismo, siendo la 1ª posición 0.

Cuando se trabaja con un vector, no podemos tomar el vector completo sino que hay que ir trabajando uno a uno con sus elementos para poder acceder a un elemento del vector, se indica:

elemento nombrevector[posición del elemento]

Declaración de un vector:

Tipo nombre[numero]

número max de elem. que podrá guardar el vector

Tipo de elementos nombre del vector

Ejemplo:

```
int v[10]; /* Este hasta 10 números enteros*/
```

```
float v[30]; /* Hasta 30 números reales*/
```

No podemos coger todo el vector completo para scanf o printf, para eso debemos recorrer el vector mediante un bucle, ejemplo


```
for( i = 0; i < num. elementos; i++) /* <i >>indica las posiciones, se denomina indice*/
{
    trabajo con el elemento v[i];
}
```

Ejercicio 70

*/*Se quiere cargar un vector con 8 numeros enteros leidos de teclado, mediante un bucle, a continuacion mostrar en una sola linea el contenido del vector, necesitamos declarar el vector y la variable 'i' y un bucle que los muestre. En otra linea mostrar el contenido del vector al revés, a continuacion leer de teclado un numero entero y calcular la media de los elementos del vector menores que dicho numero*/*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int v[8],i,cont,suma,a;
```

```
for(i=0;i<8;i++)
```

```
{
```

```
printf("Introduce un numero entero:");
```

```
scanf("%d",&v[i]);
```

```
printf("\n");
```

```
}
```

```
for(i=0;i<8;i++)
```

```
printf("%d\t",v[i]);
```

```
printf("\n");
```

```
for(i=7;i>=0;i--)
```

```
printf("%d\t",v[i]);
```

```
printf("\n");
```

```

printf("Introduzca un numero entero:");

scanf("%d",&a);

for(i=0,cont=suma=0;i<8;i++)

{

if(v[i]<a)

{

suma=suma+v[i];

cont++;

}

}

if(cont<=0)

printf("No se puede calcular la media\n");

else

printf("La media de los numeros menores de %d es %.2f\n",a,(float)suma/cont);

}

```

En un vector no se puede poner una variable entre corchetes al declararlo, pues no compila el programa. Debemos colocar un numero máximo de elementos y luego una variable que nos indique cuántos elementos va a tener el vector. Ejemplo:

```

int v[20], n, i;

do

{

printf("¿Cuántos elementos tiene el vector?);

scanf("%d",&n);

}while(!(n>0 && n<=20)); /*equivale a (n<=0 || n>20) */

for( i = 0; i <n; i++)

{

acción

```

```
}
```

```
for( )
```

Ejercicio 71

*/*Cargar un vector con n numeros enteros leidos de teclado, n tambien se leera de teclado y tiene que ser mayor que 0 y menor o igual que 20. Mostrar el vector por pantalla, a continuacion calcular y escribir la suma de todos los elementos impares*/*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int v[20],n,i,suma;
```

```
do{
```

```
printf("¿Cuantos elementos tiene el vector?");
```

```
scanf("%d",&n);
```

```
}while(!(n>0 && n<=20));
```

```
for(i=0;i<n;i++)
```

```
{
```

```
printf("Introduce el %d elemento del vector\n",i+1);
```

```
scanf("%d",&v[i]);
```

```
}
```

```
for(i=0;i<n;i++)
```

```
printf("%d\t",v[i]);
```

```
printf("\n");
```

```
for(i=suma=0;i<n;i++)
```

```
if(v[i]%2!=0)
```

```
suma=suma+v[i];
```

```
printf("La suma de los elementos impares es %d\n",suma);
```

```
}
```

Ejercicio 72 (Declaración de vector con variables tipo float)

*/*Pedir un entero `n` mayor que 0 y menor o igual que 30, a continuación*

cargar el vector con `n` numeros reales leídos de teclado, mostrar el vector

*por pantalla visualizando cada elemento con dos dígitos decimales. **Mostrar cual es***

***el elemento mayor del vector.** Para ello inicializamos una variable (max)*

con el primer elemento del vector, 2º recorrer el vector (con bucle for) e ir

comparando cada elemento con max. Cuando encontremos un elemento mayor

que max guardarla en max. 3º Mostrar max por pantalla/*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
float v[30],max;
```

```
int n,i;
```

```
do{
```

```
printf("¿Cuántos elementos tiene el vector?");
```

```
scanf("%d",&n);
```

```
}while(!(n>0 && n<=30));
```

```
for(i=0;i<n;i++)
```

```
{
```

```
printf("Introduce el %d elemento del vector\n",i+1);
```

```
scanf("%f",&v[i]);
```

```
}
```

```
for(i=0;i<n;i++)
```

```
printf("%.2f\t",v[i]);
```

```

printf("\n");

max=v[0];

for(i=0;i<n;i++)

{

if(max<v[i])

max=v[i];

}

printf("El elemento mayor del vector es %.2f\n",max);

}

```

Ejercicio 73

*/*Pedir un entero `n' mayor que 0 y menor e igual a 30.*

Cargar 2 vectores con `n' numeros reales leídos de teclado, calcular el vector suma de ambos y mostrar los 3 vectores/*

```
#include<stdio.h>
```

```
void leer (float [],int);
```

```
void ver (float[],int);
```

```
void suma (float[],float[],float[],int);
```

```
void main()
```

```
{
```

```
float v1[30],v2[30],v3[30];
```

```
int n;
```

```
do{
```

```
printf("¿Cuantos elementos tiene el vector?");
```

```
scanf("%d",&n);
```

```
}while(!(n>0 && n<=30));
```

```
printf("Vamos a leer el 1º vector\n");
```

```

    leer(v1,n);

    printf("Vamos a leer el 2º vector\n");

    leer (v2,n);

    printf("Primer vector:\n");

    ver(v1,n);

    printf("Segundo vector:\n");

    ver(v2,n);

    suma(v1,v2,v3,n);

    printf("El vector suma es:\n");

    ver(v3,n);

}

void leer(float v[],int n)

{

    int i;

    for(i=0;i<n;i++)

    {

        printf("Introduce elemento");

        scanf("%f",&v[i]);

    }

    return;

}

void ver(float v[],int n)

{

    int i;

    for(i=0;i<n;i++)

        printf("%.2f\t",v[i]);

```

```

printf("\n");

return;

}

void suma(float v1[],float v2[],float v3[], int n)

{

int i;

for(i=0;i<n;i++)

v3[i]=v1[i]+v2[i];

return;

}

```

Cuando un array se envía como parámetro a una función dicho envío se realiza por referencia. Esto significa que los cambios que se hagan en la función sobre el Array, se conservan una vez que la función termina.

Ejercicio 74

```

/*Pedir un entero `n' mayor que 0 y menor o igual que 15. En una funcion
cargar un vector con `n' enteros leidos de teclado, en otra funcion mostrar el
vector por pantalla, a continuacion pedir un entero `x', y llamar a una funcion
que cuente cuantas veces aparece el numero `x' en el vector y escribir el resultado
en "main"*/

#include<stdio.h>

void leer (int[],int);

void ver (int[],int);

int contar(int[],int,int);

void main()

{

int v[15],n,x;

do{

printf("Introduzca un numero entero:");

```

```

scanf("%d",&n);

}while(!(n>0 && n<=15));

printf("Vamos a leer el vector\n");

leer(v,n);

printf("Vamos a ver el vector\n");

ver(v,n);

printf("Introduzca un numero entero:");

scanf("%d",&x);

printf("\n");

printf("%d aparece %d veces en el vector\n",x,contar(v,n,x));

}

void leer(int v[],int n)

{

int i;

for(i=0;i<n;i++)

{

printf("Introduce un elemento: ");

scanf("%d",&v[i]);

printf("\n");

}

return;

}

void ver(int v[],int n)

{

int i;

for(i=0;i<n;i++)

```



```

printf("%d\t",v[i]);

printf("\n");

return;

}

int contar(int v[], int n,int x) /*la int es porque retorna un dato*/
{
int i,cont;

for(i=cont=0;i<n;i++)

if (x==v[i])

cont++;

return cont;

}

```

Definición de constantes

La sentencia `<< #define nombre constante valor >> /*No lleva <<;>>*/`

Es para definir constantes en vez de pedir por teclado

Ejemplo:

```

#define n 8

void main ( )

{

int v[n] = {--,--,--,--,--} /*Si se ponen menos números en la declaración que los que se indican en la
cte, el programa los rellena con ceros. Si se ponen mas NO COMPILA*/

for(i=0;i<n;i+ +)

_____ ;

_____ ;

for(i=0; i<n; i++)

```

Ejercicio 75

*/*Se tiene un vector con numeros enteros, contar cuantos elementos del vector son*

mayores que la media, el vector puede suponerse el del ejemplo:

int v[8] = {7,3,-5,12,0,14,-4,-10}; pero el programa ha de ser general para que

*funcione con cualquier vector */*

```
#include<stdio.h>
```

```
#define n 8 /*Esta declaracion hace que no tengamos que enviar n*/
```

```
float media(int[]);
```

```
int contar(int[],float);
```

```
void main()
```

```
{
```

```
int v[n]={7,3,-5,12,0,14,-4,-10};
```

```
float m;
```

```
m=media(v);
```

```
printf("La media es %.2f y hay %d elementos > media\n",m,contar(v,m));
```

```
}
```

```
float media(int v[])
```

```
{
```

```
int i,suma;
```

```
suma=0;
```

```
for(i=0;i<n;i++)
```

```
suma=suma+v[i];
```

```
return (float)suma/n;
```

```
}
```

```
int contar(int v[],float m)
```

```
{
```

```
int i,cont;
```

```
cont=0;
```

```

for(i=0;i<n;i++)
{
    if(v[i]>m)
        cont++;
}

return cont;
}

```

Ejercicio 76

*/*Se tiene un vector de numeros enteros, llamar a una funcion que muestre por pantalla los elementos mayor y menor de dicho vector, el vector puede considerarse el del ejemplo pero el programa ha de ser general para cualquier vector. Pedir un entero y llamar a una funcion que modifique el vector multiplicando cada elemento por dicho entero, a continuacion llamar a otra funcion que muestre el vector por pantalla*/*

```

#include<stdio.h>

#define n 6

void funcion(int[]);

void multiplica(int[],int);

void escribir(int[]);

void main()
{
    int a,v[n]={2,5,-8,14,0,3};

    funcion(v);

    printf("Introduzca un numero entero:");

    scanf("%d",&a);

    multiplica(v,a);

    escribir(v);
}

void funcion(int v[])
{

```

```

int i,mayor,menor;

mayor=v[0];

menor=v[0];

for(i=0;i<n;i++)

{

if(v[i]>mayor)

mayor=v[i];

if(v[i]<menor)

menor=v[i];

}

printf("El elem mayor del vector es %d y el elemento menor es %d\n",mayor,menor);

}

void multiplica(int v[], int a)

{

int i;

for(i=0;i<n;i++)

v[i]=v[i]*a;

return;

}

void escribir(int v[])

{

int i;

for(i=0;i<n;i++)

printf("%d\t",v[i]);

printf("\n");

return;

```

```
}
```

Ejercicio 77 (Ejercicio para calcular el producto escalar)

*/*Pedir al usuario que introduzca el numero de elementos de un vector y llamar a una funcion para cargar un vector de reales leído de teclado. A continuacion pedir el numero de elementos que tendra un segundo vector con numeros reales leídos de teclado. Si ambos vectores tienen el mismo numero de elementos llamar a una funcion que calcule su producto escalar y escribir el resultado en "main", en cualquier caso mostrar los dos vectores.El PRODUCTO ESCALAR es un numero, se van multiplicando los vectores elemento a elemento. Si no tienen el mismo numero de elementos, llamar a una función que cuente los elementos negativos del primer vector y escribir el resultado en main() y llamar a otra función que encuentre el mayor y menor elemento del segundo vector*/*

```
#include<stdio.h>
```

```
void cargar(float[], int);
```

```
void ver(float[], int);
```

```
float escalar(float[],float [],int);
```

```
int cuantaneg(float[],int);
```

```
void mayormenor(float[],int);
```

```
void main()
```

```
{
```

```
float v1[30],v2[30];
```

```
int n,m;
```

```
do{
```

```
printf("Introduzca un numero entero mayor que cero,para numero elem:");
```

```
scanf("%d",&n);
```

```
}while(!(n>0));
```

```
printf("\nVamos a cargar el primer vector\n\n");
```

```
cargar(v1,n);
```

```
do{
```

```
printf("Introduzca un numero entero mayor que cero,para num elmentos:");
```

```
scanf("%d",&m);
```

```
}while(!(m>0));
```

```

printf("\nVamos a cargar el segundo vector\n\n");

cargar(v2,m);

printf("\nLos vectores introducidos son:\n\n");

printf("v1= ");

ver(v1,n);

printf("v2= ");

ver(v2,m);

if(n==m)

printf("\nEl producto escalar de los vectores es %.2f\n",escalar(v1,v2,n));

else

{

printf("\nLos numeros negativos del primer vector son %d\n",cuentaneg(v1,n));

mayormenor(v2,m);

}

}

void cargar(float v[],int num)

{

int i;

for(i=0;i<num;i++)

{

printf("Introduzca el elemento %d del vector: ",i);

scanf("%f",&v[i]);

}

return;

}

void ver(float v[],int num)

```

```

{
    int i;

    for(i=0;i<num;i++)

        printf("%.2f\t",v[i]);

    printf("\n");

    return;

}

float escalar(float v1[],float v2[], int n)

{
    int i;

    float suma;

    suma=(float)0;

    for(i=0;i<n;i++)

    {

        suma=suma+v1[i]*v2[i];

    }

    return suma;

}

int cuantaneg(float v1[],int n)

{
    int i,cont1;

    for(i=cont1=0;i<n;i++)

        if(v1[i]<0)

            cont1++;

    return cont1;

}

```

```

void mayormenor(float v2[],int m)
{
    int i;

    float mayor,menor;

    mayor=v2[0];
    menor=v2[0];

    for(i=0;i<m;i++)
    {
        if(v2[i]>mayor)
            mayor=v2[i];
        if(v2[i]<menor)
            menor=v2[i];
    }

    printf("El elem mayor del vector 2 es %.2f y el elem menor es %.2f\n",mayor,menor);
}

```

MATRICES (Arrays bidimensionales)

La posición de un elemento dentro de una matriz, viene determinada por 2 números: su fila y su columna.

Tanto las filas como las columnas comienzan a numerarse en cero.

Declaración de una matriz

*tipo nombre [n°] [n°]; /*Siempre va primero el número de filas y luego el de columnas*/*

Un elemento de la matriz se especifica así:

Elemento nombre[fila del elemento] [columna del elemento]

Ejemplo: int m [4] [3];

Fila 0 9 0 3

Fila 11 7 -2

Fila 28 5 4

Fila 33 7 1

col 0 col 1 col 2

ejemplo:

El número 9 será : m [0] [0]

El número 5 será : m [2] [1]

etc.

Formas de recorrer una matriz:

Hay 2 formas por filas o por columnas:

Recorrido de una matriz por filas

Es el mas usado, su esquema es:

for(i=0; i < número de filas; i+ +)

{

// Empieza una fila

for(k=0; k< numero de columnas; k+ +)

{

trabajo con el elemento m [i] [k];

}

// Acaba la fila.

Comprobación, ejemplo:

`int m [3] [2];`

`m 1 9 /* Los bucles van por filas mirando por columnas*/`

`2 0`

• 6

Ejercicio 78

*/*Cargar una matriz de 3 filas y 3 columnas con numeros enteros leidos de teclado, a continuacion mostrar dicha matriz por pantalla*/*

`#include<stdio.h>`

```

void main()

{

int i,k;

int matriz [3][3];

for(i=0;i<3;i++)

for(k=0;k<3;k++)

{

printf("\nIntroduzca elemento de fila %d y columna %d : ",i+1,k+1);

scanf("%d",&matriz[i][k]);

}

printf("\nLa matriz introducida fue: \n"); /*con los bucles vemos la matriz*/

for(i=0;i<3;i++)

{ se vuelva a la posición inicial en la siguiente fila*/

for(k=0;k<3;k++)

printf(" %d\t",matriz[i][k]);

printf("\n");

}

}

printf("\n");

}

```

Ejercicio 79

*/*Cargar una matriz de 3x3 con numeros enteros pedidos de teclado. A continuacion mostrar la matriz por pantalla. Pedir un numero entero y contar cuantas veces aparece dicho numero en la matriz, ademas de indicar en que filas y columnas aparece*/*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```

int m[3][3],i,j,a,cont;

for(i=0;i<3;i++)

{

printf("Introduce los elementos de la fila %d\n",i+1);

for(j=0;j<3;j++)

{

printf("Introduce un numero:");

scanf("%d",&m[i][j]);

}

}

printf("La matriz es:\n"); 0

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

printf("%d\t",m[i][j]);

printf("\n");

}

printf("\n");

printf("\nAhora introduceme un numero cualquiera:");

scanf("%d",&a);

cont=0;

for(i=0;i<3;i++)

for(j=0;j<3;j++)

if(a==m[i][j])

{

cont++;

```

```

printf("El numero %d aparece en la posicion [%d][%d]\n",a,i+1,j+1);

}

printf("El numero %d aparece %d veces\n",a,cont);

}

```

Ejercicio 80

*/*Pedir al usuario que introduzca el numero de filas y columnas de un matriz, ambos numeros deberan ser mayores que 0 y menores o iguales que 10. A continuación cargar una matriz de numeros reales leídos de teclado y despues mostrar dicha matriz, visualizando cada elemento con 2 digitos decimales. Calcular la media de los elementos de la matriz negativos y contar los que no sean negativos*/*

```

#include<stdio.h>

void main()

{

int i,j,f,c,noneg,neg;

float m[10][10],suma;

do{

printf("Introduce el numero de filas:");

scanf("%d",&f);

}while(!(f>0 && f<=10));

do{

printf("\nIntroduce el numero de columnas:");

scanf("%d",&c);

}while(!(c>0 && c<=10));

for(i=0;i<f;i++)

{

printf("Elementos de la fila %d:\n",i+1);

for(j=0;j<c;j++)

{

printf("\nIntroduce un elemento:");

```

```

scanf("%f",&m[i][j]);

}

printf("\n");

}

for(i=0;i<f;i++)

{

for(j=0;j<c;j++)

printf("%.2f\t",m[i][j]);

printf("\n");

}

suma=(float)0;

neg=noneg=0;

for(i=0;i<f;i++)

{

for(j=0;j<c;j++)

{

if(m[i][j]<0)

{

suma=suma+m[i][j];

neg++;

}

else

noneg++;

}

}

if(neg!=0)

```

```
printf("La media vale %.2f\n",(float)suma/neg);

else

printf("No hay media posible\n");

printf("Hay %d numeros no negativos,noneg);

}
```

Ejercicio 81

/ Pedir el numero de filas y columnas de una matriz (mayores que cero y menores o iguales que 10), llamar a una funcion para cargar una matriz de numeros enteros leidos por teclado, llamar a otra funcion que muestre la matriz por pantalla, a continuacion llamar a otra funcion que encuentre el elemento menor de la matriz y escribirlo en main, llamar a una funcion que muestre por pantalla la suma*

*de cada fila, llamar a una funcion que muestre, cuantos elementos impares hay en cada columna */*

```
#include<stdio.h>

void leer(int[][10],int,int);

void ver(int[][10],int,int);

int menor(int[][10],int,int);

void sumafila(int[][10],int,int);

void impares(int[][10],int,int);

void main()

{

int m[10][10],f,c;

do {

printf("Introduce el numero de filas:");

scanf("%d",&f);

}while(!(f>0 && f<=10));

do{

printf("Introduce el numero de columnas:");

scanf("%d",&c);

}while(!(c>0 && c<=10));
```

```

printf("\n");

leer(m,f,c);

ver(m,f,c);

printf("\n\nEl numero menor de la matriz es:%d\n",menor(m,f,c));

sumafila(m,f,c);

impares(m,f,c);

}

void leer (int m[][10], int f,int c)

{

int i,j;

for(i=0;i<f;i++)

{

printf("Vamos a introducir los elementos de la fila:%d\n",i+1);

for(j=0;j<c;j++)

{

printf("Introduce un elemento de la matriz:");

scanf("%d",&m[i][j]);

}

printf("\n");

}

return;

}

void ver(int m[][10],int f,int c)

{

int i,j;

for(i=0;i<f;i++)

```

```

{
for(j=0;j<c;j++)
printf("%d\t",m[i][j]);
printf("\n");
}
return;
}

int menor(int m[][10],int f,int c)
{
int i,j, menor;
menor=m[0][0];
for(i=0;i<f;i++)
for(j=0;j<c;j++)
if(m[i][j]<menor)
menor=m[i][j];
return menor;
}

void sumafila( int m[][10],int f,int c)
{
int suma,i,j;
for (i=0;i<f;i++)
{
for(suma=0,j=0;j<c;j++)
suma=suma+m[i][j];
printf("La suma de la fila %d es %d\n",i+1,suma);
}
}

```



```

return;

}

void impares(int m[][10],int f,int c) //para recorrer una matriz por columnas solo se cambia el orden
{ //de los bucles

int i,j,ci;

printf("\n");

for(j=0;j<c;j++)

{

for(ci=i=0;i<f;i++)

if(m[i][j]%2!=0)

ci++;

printf("En la columna %d hay %d impares\n",j+1,ci);

}

}

```

Ejercicio 82

*/*Se tiene una matriz de numeros enteros, llamar a una funcion que escriba la media de cada fila, y llamar a otra funcion que muestre el elemento mayor de cada columna. La matriz puede suponerse una concreta pero el programa ha de ser general*/*

```

#include<stdio.h>

#define f 4

#define c 4

void media(int[f][c]);

void mayor(int[f][c]);

void main()

{

int m[f][c]={1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}};

media(m);

```

```

    mayor(m);

}

void media(int m[i][j])

{

    int suma,i,j;

    for(i=0;i<f;i++)

    {

        suma=0;

        for(j=0;j<c;j++)

            suma=suma+m[i][j];

        printf("La media de la fila %d es %.2f\n",i+1,(float)suma/c);

    }

    return;

}

void mayor(int m[f][c])

{

    int i,j,mayor;

    for(j=0;j<c;j++)

    {

        mayor=m[0][j];

        for(i=0; i<f;i++)

            if(m[i][j]>mayor)

                mayor=m[i][j];

        printf("El mayor de la columna %d es %d\n",j+1,mayor);

    }

    return;

```

```
}
```

Ejercicio 83

*/*Escribir una funcion en C que reciba una matriz de numeros reales de, como maximo,10x12, y que devuelva un vector en el que cada elemento contenga el producto de cada uno de los elementos de la columna correspondiente de la matriz*/*

```
void matriz(float m[f][c],float v[c]) //con el return no se puede devolver un vector
```

```
{
```

```
int i,j;
```

```
float prod;
```

```
for(j=0;j<c;j++)
```

```
{
```

```
for(i=0,prod=1;i<f;i++)
```

```
prod=prod*m[i][j];
```

```
v[j]=prod;
```

```
}
```

```
}
```

Ejercicio 84

*/*Escribir una funcion que calcule el factorial de un numero. Utilizando esta funcion escribir otra que calcule el numero combinatorio m sobre n ($m!/n!(m-n)!$)*/*

```
int combinatorio(int m,int n)
```

```
{
```

```
int comb;
```

```
comb=(factorial(m))/(factorial(n)*factorial(m-n));
```

```
return comb;
```

```
}
```

```
int factorial(int a)
```

```
{
```

```
int fact,cont;
```

```

fact=1;

for(cont=1;cont<=a;cont++)

fact=fact*cont;

return fact;

}

```

Ejercicio 85

*/*Escribir un programa en C que posea una funcion que dada una matriz de, como*

maximo 10x10, calcule y devuelva en cada elemento de un vector la media de la

columna correspondiente. El programa principal debera pedir la matriz por teclado e imprimir el vector resultado por pantalla/*

```

#include<stdio.h>

void leer(int[][10],int,int);

void mostrar(int[][10],int,int);

void media(int[][10],float[],int,int);

void main()

{

int f,c,m[10][10],k;

float v[10];

do{

printf("Introduce el numero de filas:");

scanf("%d",&f);

}while(!(f>=0 && f<10));

do{

printf("\nIntroduce el numero de columnas:");

scanf("%d",&c);

}while(!(c>=0 && c<10));

leer(m,f,c);

```

```

mostrar(m,f,c);

media(m,v,f,c);

printf("El vector es:\n");

for(k=0;k<f;k++)

printf("%.2f\t",v[k]);

}

void leer(int m[][10],int f,int c)

{

int i,j;

for(i=0;i<f;i++)

{

printf("Elementos de la fila %d:\n",i+1);

for(j=0;j<c;j++)

{

printf("Introduce un elemento:\n");

scanf("%d",&m[i][j]);

}

printf("\n");

}

return;

}

void mostrar(int m[][10],int f,int c)

{

int i,j;

for(i=0;i<f;i++)

{

```

```

for(j=0;j<c;j++)

printf("%d\t",m[i][j]);

printf("\n");

}

return;

}

void media(int m[][10],float v[],int f,int c)

{

int i,j,suma;

for(j=0;j<c;j++)

{

for(i=0,suma=0;i<f;i++)

suma=suma+m[i][j];

v[j]=(float)suma/f;

}

return;

}

```

Ejercicio 86

*/*Escribir una funcion que reciba 2 matrices de numeros reales de igual numero de filas y de columnas y que devuelva una tercera en la que cada elemento [i][j] contenga la media de los elementos i, j de las otras 2 matrices. Obligatoriamente la funcion ha de recibir como parametros las 3 matrices, el numero de filas y el numero de columnas*/*

```

void funcion(float m[][10],float n[][10],float w[][10],int f,int c)

{

int i,j;

for(i=0;i<f;i++)

for(j=0;j<c;j++)

w[i][j]=(m[i][j]+n[i][j]) / 2;

```

```
}
```

Ejercicio 87

*/*Escribir una funcion que reciba 2 matrices de numeros reales de igual numero de filas y de columnas y que devuelva una tercera en la que cada elemento [i][j] contenga el elemento menor de las otras 2 matrices. Obligatoriamente la funcion ha de recibir como parametros las 3 matrices, el numero de filas y el numero de columnas*/*

```
void funcion(float m[][10],float n[][10],float w[][10],int f,int c)
```

```
{
```

```
int i,j,menor;
```

```
for(i=0;i<f;i++)
```

```
for(j=0;j<c;j++)
```

```
{
```

```
if(m[i][j]<n[i][j])
```

```
w[i][j]=m[i][j];
```

```
else
```

```
w[i][j]=n[i][j];
```

```
return;
```

```
}
```

Ejercicio 88

*/*Hacer un programa que lea de teclado una lista de 5 puntos en 3 dimensiones*

(coordenadas x, y, z) y nos muestre a continuacion en orden inverso a como fueron introducidos/*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int i,j,m[5][3];
```

```
for(i=0;i<5;i++)
```

```
{
```

```
printf("Elementos del punto %d:\n",i+1);
```

```

for(j=0;j<3;j++)
{
printf("Introduce un valor:\n");

scanf("%d",&m[i][j]);

}

printf("\n");

}

for(i=4;i>=0;i--)
{

for(j=2;j>=0;j--)

printf("%d\t",m[i][j]);

printf("\n");

}

printf("\n");

}

```

Ejercicio 89

/ a) Escribir una funcion que, a partir de un numero entero recibido como parametro, calcule y devuelva la suma de la mayor de sus cifras con la menor de sus cifras.*

Si el numero es 2108, la suma es $0 + 8 = 8$.

b) Aprovechando la funcion del apartado anterior, escribir un programa en C que pida un digito 'd' por teclado (entre 1 y 9) y calcule y muestre los n primeros numeros (n se pide por teclado) que verifican que la suma de la mayor de sus cifras con la menor de sus cifras es divisible exactamente por el digito 'd' leído/*

```
#include<stdio.h>
```

```
int suma_mayor_menor(int);
```

```
void main()
```



```

{
    int d,n,cont,a;

    do{

        printf("Introduce un digito:");

        scanf("%d",&d);

    }while(!(d>0 && d<10));

    do{

        printf("\nAhora introduce un numero cualquiera:");

        scanf("%d",&n);

    }while(n<=0);

    cont=0;

    a=1;

    while(cont<n)

    {

        if(suma_mayor_menor(a)%d==0)

        {

            printf("%d\t",a);

            cont++;

        }

        a++;

    }

}

int suma_mayor_menor(int a)

{

    int mayor,menor,aux,cifra;

    mayor=menor=a%10;

```

```
for(aux=a;aux!=0;aux=aux/10)
```

```
{
```

```
cifra=aux%10;
```

```
if(cifra<menor)
```

```
menor=cifra;
```

```
if(cifra>mayor)
```

```
mayor=cifra;
```

```
}
```

```
return mayor+menor;
```

```
}
```

Ejercicio 90

/ a) Escribir una funcion que dada una matriz de reales de, como maximo 10 x 10, calcule y devuelva un vector con la media de cada columna de la matriz.*

b) Aprovechando la funcion del apartado anterior escribir un programa que para una matriz dada por teclado calcule y muestre por pantalla cual es la columna con menor media, y el valor de esta media/*

```
#include<stdio.h>
```

```
void leer(float[][10],int,int);
```

```
void ver(float[][10],int,int);
```

```
void media(float[][10],float v[10],int, int);
```

```
void main()
```

```
{
```

```
int a,b,i,j;
```

```
float m[10][10],v[10],aux;
```

```
do{
```

```
printf("Introduzca un numero de filas:");
```

```
scanf("%d",&a);
```

```
}while(!(a>0 && a<=10));
```

```

do{

printf("Introduzca un numero de columnas:");

scanf("%d",&b);

}while(!(b>0 && b<=10));

leer(m,a,b);

ver(m,a,b);

media(m,v,a,b);

printf("\nEl vector de medias es v= ");

for(i=0;i<a;i++)

printf("%.2f\t",v[i]);

printf("\n");

aux=v[0];

for(i=j=0;i<a;i++)

if(v[i]<aux)

{

aux=v[i];

j=i;

}

printf("\nLa menor media esta en la columna %d y la media es %.2f\n",j+1,aux);

}

void leer(float m[][10],int a,int b)

{

int i,k;

for(i=0;i<a;i++)

for(k=0;k<b;k++)

{

```

```

printf("Introduzca el elemento (%d , %d): ",i,k);

scanf("%f",&m[i][k]);

}

return;

}

void ver(float m[][10],int a, int b)

{

int i,k;

for(i=0;i<a;i++)

{

for(k=0;k<b;k++)

printf(" %.2f\t",m[i][k]);

printf("\n");

}

printf("\n");

return;

}

void media(float m[][10],float v[],int a,int b)

{

int i,k;

float suma;

for(k=0;k<b;k++)

{

suma=(float)0;

for(i=0;i<a;i++)

suma=suma+m[i][k];

```

```
v[k]=suma/a;
```

```
}
```

```
return;
```

```
}
```

VARIABLES TIPO CARACTER

En una variable tipo char guardamos un caracter (letras, (,+, [,*, etc).

char x;

Lectura

Para poder leer un caracter hay 2 maneras:

1ª.- scanf("%c",&x);

2ª.- x = getchar();

Escritura

También existen 2 formas:

1ª.- printf("%c,x);

2ª.- putchar (c);

El valor de una variable de tipo <<char>> debe ir entre apóstrofes (comillas simples).

Ejemplo:

if(x == 'a') // Si la variable x contiene la letra a

if(x == a) // Si las variables x y a son iguales

También se pueden utilizar con la sentencia switch:

```
switch (x)
```

```
{
```

```
case 'a': _____;
```

```
case 'b': _____;
```

Ejercicio 91

*/*Pedir al usuario que pida un caracter, a continuacion pedir un numero 'n' mayor que 0 y menor o igual que 20. Mostrar por pantalla 'n' lineas conteniendo cada una de ellas 'n' veces el caracter leído*/*

```

#include<stdio.h>

void main()

{

char c;

int n,i,j;

printf("Introduce un caracter:");

scanf("%c",&c);

do{

printf("\nIntroduce un numero:");

scanf("%d",&n);

}while(!(n>0 && n <=20));

for(i=0;i<n;i++)

{

for(j=0;j<n;j++)

printf("%c\t",c);

printf("\n\n\n");

}

}

```

Ejercicio 92

*/*Pedir un numero entero y mostrar su tabla de multiplicar, a continuación preguntar al usuario si quiere seguir introduciendo numeros, y si la respuesta es afirmativa volver a empezar*/*

```

#include<stdio.h>

void main()

{

int a,cont;

char c;

do{

```

```

printf("Introduce un numero entero:");

scanf("%d",&a);

for(cont=1;cont<=10;cont++)

printf("\n%d x %d = %d",a,cont,a*cont);

printf("\n ¿Quieres continuar (s/n)?");

getchar(); //Ponerlo antes de leer un caracter cuando previamente se haya pedido otro dato

scanf("%c",&c);

}while(c=='s' || c=='S');

}

```

Cadena de caracteres

Un cadena de caracteres es en realidad un array unidimensional de caracteres, cada caracter es una posición de un vector.

ejemplo: char cad[80]; cada carácter toma una posición.

Lectura

1ª- gets(cad);

2ª- scanf("%s,cad);

La instrucción que sirve para leer una palabra o una frase es: *gets(cad);* entre paréntesis la cadena que quiero leer.

También, aunque no es muy usual, se puede utilizar:

scanf("%s,c); // No lleva <<&>> y solo sirve para leer una palabra

Cuando el ordenador recibe la orden << gets>> se encarga de leer todos los caracteres que el usuario teclea, hasta detectar el salto de línea y los guarda en el vector, ocupando cada carácter una posición. Además después del último carácter guarda un caracter especial <<\0>> que es el caracter de fin de cadena.

Ejemplo:

Si tenemos una cadena de caracteres <<c>> de máximo 80 posiciones c[80]; y la leemos mediante la instrucción <<gets>>, tendríamos un vector (array) de 80 posiciones:

En la pantalla del Pc escribimos la palabra HOLA y <<return>>, el ordenador coge los caracteres y los guarda.

El que teclea HOLA no ve que se guarda en la posición 4 : <<\0>>

Escritura

1ª- puts(cad);

2ª- printf(%s,cad);

Recorrido de una cadena

for(i=0; cad[i] != '\0'; i+ +) // hasta que cad[i] sea distinto de << \0 >>

{

trabajar con c[i] que es un carácter

}

Ejercicio 93

*/*Leer una cadena de caracteres y decir el porcentaje de blancos que contiene sobre el total de caracteres.*

*'La universidad de oviedo' tiene 24 caracteres de los cuales 3 son blancos, y el % es (3/24)*100=12.5 %/*

#include<stdio.h>

void main()

{

char cad[50];

int i,blanco,cont;

printf("Introduce una frase cualquiera:");

gets(cad);

blanco=cont=0;

for(i=0;cad[i]!='\0';i++)

{

cont++;

if(cad[i]==' ')

blanco++;

}

printf("Nº de blancos:%d Porcentaje:%.2f",blanco,(float)blanco/cont*100);


```
}
```

Otras funciones de caracteres

STRLEN– Función que recibe una cadena y retorna su longitud

```
char cad[30];
```

```
gets (cad);
```

```
printf(La cadena %s tiene %d caracteres,cad,strlen(cad));
```

Recorrido:

```
for(i=0;cad[i]!='\0';i++) for(i=0;i<strlen(cad);i++)
```

STRCPY– Función que recibe 2 cadenas de caracteres y hace que la primera sea una copia de la segunda. Esta función es útil ya que no se pueden realizar asignaciones entre cadenas:

```
char cad1[80],cad2[80]; strcpy(cad2,cad1)
```

```
cad2=cad1;
```

STRCAT– Función que recibe 2 cadenas de caracteres y realiza la concatenación de ambas. En la 1ª cadena guarda la concatenación de las 2 cadenas:

```
strcat(cad1,cad2)
```

Ejercicio 94

*/*Escribir un programa que pida un numero entero por teclado y que imprima por pantalla la suma de las cifras que ocupan las posiciones pares, despues la suma de las cifras que ocupan las posiciones impares y a continuacion mostrar el numero leido, las posiciones comienzan en 1,correspondiendo este valor a la posicion mas a la derecha. No utilizar arrays*/*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int a,cont,cifra,aux,sumpar,sumimpar;
```

```
printf("Introduce un numero:");
```

```
scanf("%d",&a);
```

```

sumpar=suminpar=0;

cont=1;

aux=a;

while(aux!=0)

{

cifra=aux%10;

if(cont%2!=0)

sumimpar=sumimpar+cifra;

else

sumpar=sumpar+cifra;

cont++;

aux=aux/10;

}

printf("%d,%d,%d,sumimpar,sumpar,a);

}

```

Ejercicio 95

*/*Escribir un programa que calcule los 'n' primeros numeros (n se pide por teclado)*

tales que la suma de su primera cifra con su ultima cifra sea impar/*

```
#include<stdio.h>
```

```
int sumacifras(int);
```

```
void main()
```

```
{
```

```
int n,cont,a;
```

```
printf("Introduce un numero:");
```

```
scanf("%d",&n);
```

```
a=1;
```

```

cont=0;

while(cont<n)

{

if(sumacifras(a)%2!=0)

{

printf("%d\t",a);

cont++;

}

a++;

}

}

int sumacifras(int num)

{

int aux,cifra;

for(aux=num;aux!=0;aux=aux/10)

cifra=aux%10;

return cifra+num%10;

}

```

Test

- ¿Qué valor tomará la variable `m' suponiendo que es de tipo entero?

$m = 10.2 * (4/5);$

- 8
- 8.16
- 0
- Ninguna de las anteriores
- ¿ Y si `m' fuese de tipo flotante?

Valdría 0 también ya que $4/5$ es una división de numeros enteros, por tanto su cociente también será un numero entero. $m = 10.2 * 0 = 0$

- ¿Qué sentencia es correcta?

- int a==3;
- char c=a;
- float b=3;
- Más de una de las anteriores es correcta

Ejercicio 96

*/*Cargar una matriz de numeros enteros leidos de teclado, previamente se pedira el numero de filas y de columnas que, como máximo sera 10. Llamar a una funcion que devuelva la matriz traspuesta y mostrarla.*

Una matriz esta traspuesta cuando le cambiamos las filas por columnas/*

```
#include<stdio.h>

void cargar(int[10][10],int,int);

void traspuesta(int[10][10],int,int,int[10][10]);

void mostrar(int[10][10],int,int);

void main()
{
    int f,c,m[10][10],t[10][10];

    do{

        printf(Introduce el numero de filas:);

        scanf(%d,&f);

    }while(!(f>0 && f<=10));

    do{

        printf(Introduce el numero de columnas:);

        scanf(%d,&c);

    }while(!(c>0 && c<=10));

    cargar(m,f,c);

    traspuesta(m,f,c,t);

    mostrar(t,f,c);

}

void cargar(int m[10][10],int f,int c)
```

```

{
    int i,j;

    for(i=0;i<f;i++)

        printf(Elementos de la fila %d:\n,i+1);

    for(j=0;j<c;j++)

    {

        printf(Introduce un elemento:);

        scanf(%d,&m[i][j]);

    }

}

void traspuesta(int m[10][10],int f,int c,int t[10][10])

{

    int i,j;

    for(i=0;i<f;i++)

    for(j=0;j<c;j++)

        t[i][j]=m[j][i];

}

void mostrar(int m[10][10],int f,int c)

{

    int i,j;

    for(i=0;i<f;i++)

    for(j=0;j<c;j++)

        printf(%d,m[i][j]);

}

```

Libreria math.h

Algunas funciones de esta libreria son:

- **abs(i)**: función que recibe un entero y devuelve su valor absoluto
- **fabs(f)**: función que recibe un real y devuelve su valor absoluto
- **sqrt(d)**: función que recibe un entero o real y devuelve su raíz cuadrada
- **pow(d1,d2)**: función que recibe 2 números reales o enteros y devuelve el resultado de elevar el primero al segundo
- **sin(d)**: función que recibe un número real o entero correspondiente al valor de un ángulo en radianes y devuelve el valor de su seno
- **cos(d)**: función que recibe un número real o entero correspondiente al valor de un ángulo en radianes y devuelve el valor de su coseno

Cuando queramos utilizar alguna de estas funciones deberemos escribir en el encabezamiento junto con la librería stdio.h la de math.h.

```
#include<stdio.h>
```

```
#include<math.h>
```

```
void main()
```

Ejercicio 97

*/*Escribir una función en C que reciba una matriz de números reales de, como máximo 10x12, e imprima cual o cuales son los elementos [i][j] que se aproximan más a la media de todos los elementos de la matriz*/*

```
void cercamedia(float m[10][12],int f,int c)
```

```
{
```

```
float x,difmenor;
```

```
int i,j;
```

```
x=media(m,f,c);
```

```
difmenor=fabs(m[0][0]-x);
```

```
for(i=0;i<f;i++)
```

```
for(j=0;j<c;j++)
```

```
if(fabs(m[i][j]-x)<difmenor)
```

```
difmenor=fabs(m[i][j]-x);
```

```
for(i=0;i<f;i++)
```

```
for(j=0;j<c;j++)
```

```
if(fabs(m[i][j]-x)==difmenor)
```

```
printf(Elemento mas cercano a media es m[%d][%d]=%f,i,j,m[i][j]);
```

```

}

float media(float m[10][12],int f,int c)

{

float suma;

int i,j;

suma=(float)0;

for(i=0;i<f;i++)

for(j=0;j<c;j++)

suma=suma+ m[i][j];

printf(La media es %f,suma/f);

}

```

Ejercicio 98

*/*Hacer un programa que pida los coeficientes de una ecuación de segundo grado y a continuación muestre sus raíces*/*

```

#include<stdio.h>

#include<math.h>

void main()

{

float a,b,c;

printf(Introduce los coeficientes de la ecuacion:);

scanf(%f %f %f,&a,&b,&c);

if(a==0)

{

if(b==0)

printf(No hay ecuacion);

else

printf(Raiz= %f,-c/b);

```

```

}

else

{

dis=pow(b,2)-4*a*c;

if(dis<0)

printf(Las raices son imaginarias);

else

{

printf(Raiz 1= %f,(-b+sqrt(dis))/(2*a));

printf(Raiz 2= %f,(-b-sqrt(dis))/(2*a));

}

}

}

```

Ejercicio 99

*/*Hacer una función que reciba 2 matrices y devuelva la matriz producto de ambas.*

Para multiplicar 2 matrices tenemos que asegurarnos de que la primera tiene el mismo numero de columnas que filas tienes la segunda (c1=f2)

*m1(f1,c1)*m2(f2,c2)=m3(f1,c2) */*

```
void multiplica(int m1[10][10],int m2[10][10],int m3[10][10],int f,int c1,int c2)
```

```

{

int i,j,k,suma;

for(i=0;i<f;i++)

for(suma=k=0;k<c2;k++)

{

for(j=0;j<c1;j++)

suma=suma+( m1[i][j]*m2[j][k] );

m3[i][k]=suma;

```



```
}
```

```
}
```

Ejercicio 100

*/*Pedir un numero positivo de 3 digitos y reorganizar sus digitos para que quede ordenado de menor a mayor*/*

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
int a,u,d,c,aux;
```

```
do{
```

```
printf("Introduce un numero:");
```

```
scanf("%d",&a);
```

```
}while(!(a>=100 && a<=999));
```

```
u=a%10;
```

```
d=(a%100)/10;
```

```
c=a/100;
```

```
if(c>d)
```

```
{
```

```
aux=c;
```

```
c=d;
```

```
d=aux;
```

```
}
```

```
if(d>u)
```

```
{
```

```
aux=d;
```

```
d=u;
```

```
u=aux;
```

```
}  
  
if(c>d)  
{  
  
    aux=c;  
  
    c=d;  
  
    d=aux;  
  
}  
  
a=c*100+d*10+u;  
  
printf("El numero ordenado es %d",a);  
  
}
```