TEMA 1 INTRODUCCIÓN AL C

- **CONCEPTOS GENERALES**
- **EL LENGUAJE EN C**
- **ELEMENTOS DE C**
- **ESTRUCTURA DE UN PROGRAMA EN C**

- **CONCEPTOS GENERALES**

Programa.− Conjunto de instrucciones ejecutadas secuencialmente.

En windows son los .EXE −> escritos en codigo maquina.

El C es un lenguaje de alto nivel

Fichero fuente .c compilador de C .Exe

.ccp

.obj Linker

Preprocesador

.h

Entorno de C

Conjunto de todas las herramientas de programación necesarias.

Nosotros vamos a utilizar el BC++ 5.0 que dispone de:

- Compilador
- Preprocesador (#include) llamada al preprocesador
- Linker
- Editor
- Archivos de cabecera (.h)

1.2 EL LENGUAJE C

Es un lenguaje de medio−alto nivel, de programación es estructurada y modular.

1.3 ELEMENTOS DE C

El código fuente esta dividido en tokens.

Tonken.− Unidad mínima en que el compilador descompone el fichero fuente para traducirlo.

TIPOS DE TOKENS:

- Palabras claves.− Son 32; if, incluye

- Identificadores.− Nombre que se designa a una entidad. Nombre de variable a..z; A..Z; _ ; 0..9.
- Constantes.− Valor que no cambia durante la ejecución

− Numéricas (real, entera)

−Caracteres *`b'*

−Cadena de caractres *hola*

- Operadores.− +,−,++,.
- Separadores.

− uno o más espacios en blanco

− Salto de línea

− Tabuladores

- Comentarios.− /**/ //

Sentencias simples acaban en ;

Sentencias compuestas

{

.;

.;

.;

}

1.4 ESTRUCTURA DE UN PROGRAMA EN C

```
#include <sodio.h> //esto es un identificador
#include <conio.h>
main()
{
printf(hola);
getch(); //lee el teclado hasta pulsar una tecla get character−> getch()
}
```

TEMA 2 TIPOS DE DATOS

2.1 CONCEPTO DE VARIABLE

2.2 DECLARACIÓN DE VARIABLES

2.3 OPERADORES

2.4 E/S DE DATOS

2.1 CONCEPTO DE VARIABLE

Variable.− parte de la memoria a la que se le asigna un nombre

Tipos de datos.− (Trae False−> 1bit) (0−256 −> 8bit)

2.2 DECLARACIÓN DE VARIABLES

Tipo simple (nº bits)

- char
- int
- float
- double

char −> Guarda caracteres lo que están *`0'* comillas simples,p.e. *c=';':*

int −> (integer) Numeros enteros −n,−2,−1,0,1,2,,n

int a;

a=83;

float, double.− Son para variables de números reales. La diferencia esta en el tamaño del dato, la precisión. double>float.

El double no lo utilizaremos.

Nota: La variable hay que declararla antes de utilizarse. Normalmente en el inicio del *main()*

Ojo A=0

a=0 A!=a se distinguen mayúsculas y minúsculas.

Se puede dar un valor inicial a la variable al declararla.

int a=0; es lo mismo que *int a;*

a=0;

- signed−unsigned (char, int). Se pone antes signed char. Por defecto una variable guarda valores negativos. Si pongo unsigned:

int a;

a!=−9;

Se utiliza para ahorrar memoria

8 bit (0−255, −128−127)

Si necesitamos guardar de 0 a 200 con *int;* no vale,

si ponemos *insigned int;* si vale: 0−256.

- short − long (int). Sirve para ampliar el rango de almacenamiento *long int;*

Por defecto es short (el rango estandar)

- long (double)

TIPO	**TAMAÑO APROX.**	**RANGO MÍNIMO ANSI C**
char	8	−128 a 127
unsigned char	8	0 a 255
int	16	−32.768 a 32.767
unsigned int	16	0 a 65.535
long int	32	−2.147.483.648 a 2.147.483.647
unsigned long int	32	0 a 4.294.967.295
float	32	−1038 a 1038−1 (6 dígitos de precisión decimal)
double	64	−10308 a 10308−1 (10 dígitos de precisión decimal)
long double	80	−104932 a 104932−1 (10 dígitos de precisión decimal)

2.3 OPERADORES

- De asignación.− Asigna el valor a una variable =, variable=valor;
- Aritméticos
- Lógicos
- Relacionales

De asignación variable=valor;

Valor =>cte (*a=20*);

Variable (*a=b*);

Exp (*a=7*b+5−c*);

Fun. (*a=fun.*);

int a=3; b=4;

b=a;

¿Qué valores mostra para a y b?

a=3 y b=3

a=2; // asignamos este nuevo valor para a

¿Qué valores mostra para a y b?

a=2 y b=3

Aritméticos:

+, − , * , /, %

% −> Es el resto de una división. Siempre tiene que ser *int* sino hay que convertirla *(int)a*

int a=4,b=3,resto; //4/3 => resto 1

resto=a%b; a=int y b= int siempre ¾=0 y resto=3

Agrupación de expresiones

Orden jerárquico:

1º) (.) Paréntesis

2º) − Menos (de un número p.e.: −4)

3º) */ Producto y división en ese orden

4º) +− Suma y resta

Ejercicio agrupar las siguientes expresiones:

*y=6+12/6*2−1=9*

*y=(6+12)/6*2−1=5*

y=6+12/6(2−1)=8*

*y=6+12/(6*2)−1=6*

El resultado de una operación es siempre el de mayor grado.

p.e.:

int a=1,b=2,c;

*c=a*b; //*int*int=int luego c=int

int a=1,c;

float b;

*c=a*b;* //int*float=float c=float

int a=1,b=2,c;

c=a/b;

//Lo que muestra c es 0

int a=1,b=2;

float c;

c=a/b

//Lo que muestra c es 0

Tenemos que hacer un Casting

int a=1,b=2;

float c;

c=(float)a/b //ojo que lo que tenemos es float/int el casting solo afecta a la variable *a*

Operadores de asignación compuesta

V1=V1<operador +−*/%>V2; Se abrevia V1<operador>=V2;

p.e.: V1+=V2; =>V1=V1+V2

p.e.: V1++; => V1=V1+1

p.e.: V1−−; => V1=V1−1;

Con todos los operadores se forman expresiones p.e.: a+b−5*c

2.4 E/S DATOS

printf( ); entrada de datos

scanf( ); salida de datos

Esta funciones están en el fichero de cabecera <stdio.h>

printf( ); Muestra datos en la unidad de salida por defecto (si no se indica nada es la pantalla)

printf(Cadena de control, arg1, arg2,.argn);

La cadena de control:

- Texto.
- Códigos de los formatos de los argumentos que se quieren mostrar.

Ejemplo:

printf(hola mundo); En este caso no tenemos argumentos.

Los códigos de los formatos que van dentro de las comillas van precedidos por %, a continuación tenemos que poner los argumentos.

printf(%d,a); Pongo %d en cualquier parte de las comillas, puesto que se va a sustituir por el valor de la variable a.

printf(%d %d,a,b); Es importante el orden, puesto que en este caso primero va a mostrar a y despues b.

Códigos de conversión:

Código	Tipo variable	Comentario
%d o %i	int	Muestra enteros
%u	unsigned int	Muestra int sin signo
%c	char	Muestra carácter uno solo
%f	float	Muestra float
%e	double	e viene de notación científica p.e. 6.5 con %e => 6.500000+01
%g		Muestra el número en el menor de float o double. Discrimina o %f o %e
%s		Muestra cadena de caracteres string
%o		octal Muestra el valor en código octal base 8
%x		Hexadecimal Muestra el valor en código hexadecimal
%%		Muestra el carácter %

p.e. *printf(100%%); => 100%*

scanf( ); Lee datos por teclado

Sintaxis:

scanf(%caracter_conversion1 %CC2,&arg1,&arg2);

El orden de los argumentos sigue el orden que expresa.

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

scanf solo lee, no muestra nada, hay que poner un printf antes.

printf(Introduce 2 números:);

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

El programa muestra por pantalla:

Introduce 2 numeros: 70 4

El programa busca un espacio o en blanco o un salto de linea para leer el segundo número.

Si queremos leer dos numeros separados por comas, damos el formato a la entrada.

p.e.:

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

Por pantalla:*70,4*

Mejor no utilizar la coma.

Ejemplo:

Programa que lee una variable char, int, float.

#include <stdio.h>

#include <conio.h>

main()

{

char c;

int a;

float b;

printf(Introduce un caracter, un entero y un real:);

scanf(%c %d %f,&c,&a,&b);

Ejemplo

Programa que calcula la resistencia equivalente de dos resistencias en paralelo.

#include <stdio.h>

```
#include <conio.h>

main()

{

//definimos variables

float R1,R2,Re;

printf(Introduce los valores de R1 y R2 para calcular su equivalente paralelo:);

scanf(%f %f,&R1,%R2);

//calculos

Re=(R1*R2)/(R1+R2);

//salida por pantalla

printf(La Re para una R1=%f u R2=%f es Re=%f,R1,R2,Re);

getch();

}
```

Nota: \n dentro de las comillas de un printf() genera un salto de linea.

Si queremos mostrar por pantalla el carácter \ escribimos \\

Ejemplo:

Programa que convierta una temperatura de ºF a ºC (ºC=5/9(ºF−32))

```
#include <stdio.h>

#include <conio.h>

main()

{

//definicion de variables

float c,f;

printf(Introduce la temperatura en ºF a convertir en ºC:);

scanf(%f,&f);

//Calculos
```

c=(5(f−32))/9; //daria igual poner 5*(f−32)/9 por la prioridad que tiene * sobre /*

//Salida

printf(\nLa temperatura en ºf=%f en ºC es igual a %fºC,f,c);

getch();

}

NOTA:

(5/9)(F−32) Seria erroneo pues tenemos int/int=int y la division es float.*

para solucionar esto escribimos (5/9.0)(f−32)*

TEMA 3 SENTENCIAS DE CONTROL

3.1 OPERADORES RELACIONALES

3.2 OPERADORES LOGICOS

3.3 SENTENCIA IF

3.4 SENTENCIA SWITCH

3.5 BUCLE WHILE

3.6 BUCLE DO WHILE

3.7 BUCLE FOR

3.8 SENTENCIA BREAK

3.9 SENTENCIA CONTINUE

3.1 OPERADORES RELACIÓNALES

Devuelve cierto o falso según se cumpla la condición expresada o no.

FALSO −> Cero

CIERTO −> Distinto de cero

< menor

> mayor

<= menor o igual

>= mayor o igual

== algo igual a algo (doble igual. Esto es erróneo solo es para dos variables no más a==b==c)

!= algo distinto a algo (a!=b)

3.2 OPERADORES LÓGICOS

Unen expresiones hechas con operadores relacionales

&& AND

|| OR

! NOT

Lista de preferencia de operadores

- ( )
- ! ++ −− −unario (p.e. −7)
- */%
- +−
- < <= > >=
- == !=
- &&
- ||
- =+=−=*=/=%=

Las preferencias también van de izquierda a derecha en el orden que aparecen arriba.

3.3 SENTENCIA IF

Permite ejecutar o no una sentencia o un conjunto de sentencias en función de una expresión.

if (expresión)

sentencia;

no

exp

si

sentencia

Si queremos tener mas de una sentencia hay que ponerlas entre llaves

if (expresión)

{

sentencia A;

sentencia B;

}

Nota: No es conveniente poner las llaves con solo una sentencia, lo valoran mal en el examen.

Ejemplo

int A=−8;

if (A>0)

printf(1);

printf(2);

¿Cuál es la salida?

- Ninguna
- 1
- 2 −> correcta

ELSE

if (expresión)

{

..;

..;

}

else

{

..;

..;

}

IF ELSE ANIDADOS

if (expresion1)

{

..;

..;

}

else if (expresion2)

{

..;

..;

}

else if (expresion3)

{

..;

..;

}

Programa que calcule el área de un cuadrado y un circulo (pi*r2)

#include <stdio.h>

main()

{

int opcion;

float radio,lado,area;

printf(1.−Area del circulo);

printf(2.−Area del cuadrado);

printf(Elige la opcion deseada);

scanf(%d,&opcion);

if (opcion==1)

{

printf(introduce el radio);

scanf(%f,&radio);

*area=3.1416*radio*radio*

printf(El area del circulo es %f,area);

}

else if (opcion==2)

{

printf(Introduce el lado del cuadrado);

scanf(%f,&lado);

*area=lado*lado;*

printf(El area del cuadrado es %f,area);

}

}

Ejemplo relación if..else:

int A=9

if (A>0)

if (A>7)

printf(A);

else

printf(B);

El else corresponde al if mas cercano

El resultado es = A

Si int A=3 el resultado seria B

3.4 SENTENCIA SWITCH

Es similar a un if else multiple.

Ejecuta las sentencias cuya etiqueta coincide con el valor de la expresión.

Las etiquetas tienen que ser constantes.

Switch (variable)

{

case cte1: sentencia1;

break;

case cte2: sentencia2;

break;

case cte3: sentencia3;

break;

}

Con *break* saltamos todo lo demas y salimos del *switch.*

Sin no ejecutaría secuencialmente todo lo que hay hacia abajo.

switch (variable)

{

case cte1: sentencia1;

sentencia2;

sentencia3;

default: sentencia4;

sentencia5;

}

En el caso de que no se cumpla ninguno de los anteriores se utiliza el *default* . No hace falta poner *break* pues siempre se escribe al final.

Nota: Con switch no hace falta poner {} para distinguir cada caso.

switch(opcion)

{

case 1: sentencias area del circulo;

break;

case 2: sentencias area del cuadrado;

break;

default: sentencias mensaje de aviso error;

}

Programa que diga cuantos dias tiene un mes:

{

case 1:

case 2:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12: printf(31 dias);

break;

case 2: printf(28 dias);

break;

case 4:

case 6:

case 9:

case 11: printf(30 dias);

break;

default: printf(el mes es incorrecto);

}

Los bucles permite repetir la ejecución de una/s sentencia/s en función del valor de la expresión.

3.5 BUCLE WHILE

Sintaxis:

while (exp)	while (exp)

sentencia;	{
	sentencia1;
	sentencia2;
	sentencia3;
	}

Interaciones es el número de vueltas que da.

El mínimo es = 0

El máximo es = n

p.e.:

int a;

a=1;

while (a<10000)

printf(hola);

Se ejecuta infinitas veces

p.e.:

	Secuencia del bucle a= 1
int a;	hola
a=1;	a=10
while (a<10000)	hola
{	a=100
printf(hola);	hola
*a=a*10;*	a=1000
}	hola a=10000 Vemos 4 holas

3.6 BUCLE DO..WHILE

do

sentencia;

while (exp);

Ejecuta sentencias mientras while(exp) sea correcto.

do

{

sentencia1;

sentencia2;

sentencia3;

}

while(exp);

p.e. para que se introduzca correctamente el mes:

do

{

printf(introduce un mes);

scanf(%d,&mes);

}

while(mes<1 || mes >12); //los incorrectos

3.7 BUCLE FOR

for(iniciacion;expresion;actualizacion)

sentencia;

for(iniciacion;expresion;actualizacion)

{

sentencia1;

sentencia2;

sentencia3;

}

Si solo tiene una sentencia no se ponen llaves.

Si quisiéramos inicializar varias variables las separamos por comas.

3.8 SENTENCIA BREAK

Se puede utilizar para salir de un bucle, aunque no es buena programación estructurada.

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

{

printf(hola);

if (i==5) break;

Es mejor poner i<=5

3.9 SENTENCIA CONTINUE

Sirve para volver al inicio de un bucle (no inicializar) antes de ejecutar las sentencias del cuerpo de ese bucle.

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

{

if (i==5) continue;

printf(hola);

}

No mostraría por pantalla hola cuando i==5.

Tampoco es de buena programación.

**** EJERCICIOS ****

MOSTRAR POR PANTALLA LOS NUMEROS DEL 0 AL 100

#include <stdio.h> *main()* *{* *int i;*	*#include <stdio.h>* *main()* *{* *int i;*	*#include <stdio.h>* *main()* *{* *int i;*

i=0; *while (i<=100)* *printf(%d\n,i);* *i++;* *}*	*i=0* *do* *{* *printf(%d\n,i);* *i++;* *}* *while(i<=100);*	*for(i=0; i<100; i++)* *printf(%d\n,i);* *}*

CALCULAR EL FACTORIAL DE UN NUMERO n INTRODUCIDO POR TECLADO n>=0

(no existen factoriales de números negativos)

#include <stdio.h>
main()
{
int n, i, factorial=1; //si =0 al multiplicar seria siempre 0
do
{
printf(Introduce n:);
scanf(%d,&n);
}
while (n<0);
for(i=n; i>0; i−−)
*factorial=factorial*i;*
printf(factorial %d,factorial);
	Factorial i
	1 4
	4 3
	12 2
	24 1
	24 0
	Podriamos depurar el programa porque al multiplicar por 1 es simpre lo mismo
	for(i=n;i>1;i−−)
	acabaria más rapido

ESCRIBIR UN PROGRAMA EN C QUE DADO UN NUMERO POR TECLADO CUENTE SU Nº DE CIFRAS.

int n,cifras=0; *printf(introduce n);*	*p.e.* *n=9130*

scanf(%d,&n); *if (n==0)* *cifras=1;* *else* *while (n>0)* *{* *cifras++;* *n=n/10;* *}* *printf(el numero de cifras %d,cifras);*	*9130/10=913 resto 10* *913/10=91 resto 3* *91/10=9 resto 1* *9/10=0*

SUMAR LAS CIFRAS DE UN NUMERO n LEIDO POR TECLADO Y MOSTRAR LA SUMA DE SUS CIFRAS Y EL DATO LEIDO POR TECLADO.

int n, suma=0, aux;

printf ("introduce n: ");

scanf("%d",&n) ;

while (aux>0)

{

suma=suma+aux%10 ;

aux=aux/10 ;

}

printf("la suma es %d para el numero %d",suma,n);

PROGRAMA QUE SAQUE LOS 100 PRIMEROS NÚMEROS DIVISIBLES POR 3 EMPEZANDO POR EL 0

int i=0,contador=0;

while (contador<100)

{

if (i%3==0)

{

printf(%d,i)

contador++

}

i++

}

Escribir un programa en C que dado un numero por teclado diga si es primo.

(nºprimo es divisible por 1 y por el mismo)

n>0

esprimo=1 suponemos que cumpla condición

#include <stdio.h>

main()

{

int n, divisor, esprimo=1;

do

{

printf(Introduce el numero:);

scanf(%d,&n);

while(n<=0);

for (division=2;divisor<n;divisor++);//no ponemos el 1 puesto que siempre es divisible. divisior<n excluimos el n puesto que n/n=1

if (n%divisor==0)

esprimo=0;

if (esprimo==1)

printf(El numero es primo)M

else

printf(el numero no es primo);

}

Podriamos Mejorarlo para que el bucle

for(divisor=2;divisor<n&esprimo==1;divisor++)

También podríamos utilizar break pero mejor lo de antes

{

esprimo=0;

break;

otra forma

divisor<n/2+1 pero solo se cumpliría en el caso de la mitad en el resto ya no se cumpliría

Febrero−2001 Septiembre−2001

Programa en C que dado un numero entero sume las posiciones pares, luego sume las posiciones impares y que muestre por pantalla ambas sumas y el numero inicial

Ej: 1234 muestre 6 4 1234 (6 pares 4 impares)

main()

{

int n, sp=0,si=0, aux,posicion=0 ; //posicion=1 tal como nos nada en el enunciado no puede ser, pero no esta mal del todo.

printf(Introduce n);

scanf(%d,&n);

aux=n; //guardo copia del valor inicial

while (aux>0) //podriamos poner != por que n podria ser negativo pero no estaria mal

{

if (posición%2==0) //ojo este caso es par

sp=sp+aux%10; //suma pares=suma pares + aux%10 que extrae el valor

else

si=si+aux%10;

aux=aux/10;

posicion++ ;

}

printf(%d %d %d,sp,si,n);

}

TEMA 5 TIPOS DE DATOS ESTRUCTURADOS

5.1 ARRAYS

5.1.1 ARRAYS UNIDIMENSIONALES

5.1.2 ARRAYS MULTIPLESDIMENSIONALES

5.1.3 ARRAYS DE CARACTERES (STRINGS)

5.2 ESTRUCTURAS

5.3 ENUMERACIONES

5.4 TIPOS DEFINIDOS POR EL USUARIO

5.1 ARRAYS

Un array es una colección de elementos del mismo tipo

5.11 ARRAYS UNIDIMENSIONALES

Un array de una dimensión es un vector

Para definirlo

Tipo nombre[tamaño máximo];

Tipo = Tipo de elementos del vector int,float,char,double (los de tipo char son los string 5.1.3 que no veremos.

Nombre = Nombre de la variable del vector

Tamaño = Es reservar memoria para los elementos del vector.

int v[10]; //10 posiciones enteras

Particularidad la primera posición es 0, luego la última seria n−1

Un array se puede inicializar cuando se declara.

int v[10]={1,2,3,4,5,6,7,8,9,10} //La posición 0 hay de valor 1 y en la 9 hay de valor 10

Si doy valores iniciales al vector no hace falta poner el tamaño.p.e.:

int v[]={10,2,3,5} //seria de tamaño 4

Pero mejor inicializarlo con el número del tamaño.

Como acceder al elemento:

v[posición]

p.e. *v[0]* se refiere al elemento de la posición 0.

v[0]=20; // guarda 20 en la posición 0

se puede comparar. *if(v[4]==100) //compara lo hay en la posición 4 con el valor 100*

Podemos hacer lo mismo que hemos echo hasta ahora

Ojo podemos salirnos del rango del vector

int v[10];

v[−4] v[20] v[10] **Estaríamos fuera de rango**

Si quiero dar valores por teclado a v hay que hacerlos uno a uno con un bucle for

Ejemplo

Leer dos vectores de un tamaño 4 y determinar si son o no iguales. (comprobar significa que en la misma posición del vector tienen el mismo dato)

Hay que comparar dato a dato cada valor del array

```
#include <stdio.h>

main()

{

int va[4],vb[4],i,soniguales=1; //ojo la variable para recorrer bucles i

for (i=0;i<4;i++) //de esta forma i<4 ya no incluimos a 4 es la mejor forma

{

printf(introduce va[%d]: ,i);

scanf(%d,&va[i];

}
```

for (i=0;i<4;i++)

{

printf (Introduce vb[%d]: ,i);

scanf(%d,&vb[i]);

}

for (i=0;i<4 && soniguales==1; i++) //ojo si solo ponemos soniguales significa que soniguales==1 es una variable booleana. True false

for (i=0;i<4 && sonniguales; i++)

if(va[i]!=vb[i]

soniguales=0;

if (soniguales)

printf(son iguales);

else

printf(son distintos);

}

5.1.2 ARRAYS MULTIDIMENSIONALES

Un array de mas de una dimensión son los denominados matrices.

Para declara una matriz hay que declarar:

Tipo nombre[T1][T2] [T3].....[TN] //cada corchete es una dimensión

El limite de las dimensiones las limita el compilador.

float m[4][3]; //es una matriz de 4 filas y tres columnas

como acceder a los datos:

m[0][0]; //fila 0, columna 0

m[3][0]; //fila 3, columna 0

Tenemos que trarar los elementos de uno en uno.

Cuando se declara una matriz tambien se pueden dar valores iniciales.

float [3][2] = {1,2,3,4,5,6} //3 filas y 2 columnas (0,0)(0,1)(1,0)(1,1)(2,0)(2,1)

otra forma mas clara

float [3][2] = {{1,2},{3,4},{5,6}} // si le doy menos relementos el resto los rellena con ceros. Si doy mas valores da error. Nos salimos del rango.

EJERCICIO SEP.2001

Dada una matriz de numeros reales, pedir nº de filas y columnas, como maximo la matriz de 10*20. Mostra el mayor y menor elemento (tb hay que mostrar la matriz)

```
#define MAXFILAS 10

#define MAXCOLS 10

#include <stdio.h>

main()

{

float m[MAXFILAS][MAXCOLS];

int nfilas,ncols,max,min;

do

{

printfnumero filas?);

scanf(%d,&nfilas);

}

while(nfilas<2 || nfilas>MAXFILAS);

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

for(j=0<ncols;j++)

{

printf(introduce m[%f][%f],i,j);

scanf(%f,&m[i][j]);

}

min= [0][0];

max=[0][0];//cojemos el primer valor para comprara
```

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

for(j=0<ncols;j++)

{

if m[i][j]<min)

min=m[i][j];

if m[i][j]>max)

min=m[i][j];

printf(el maximo es %f, max);

printf(el minimo es %f,min);

//vamos a mostra la matriz para que me aparezcan como matriz.

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

{

for(j=0;j<ncols;j++)

printf(%f ,m[i,j]);//tambien %f\t tabulador

printf(\n); //salto de linea cada fila

}
```

TEMA 4 PROGRAMACION ESTRUCTURADA

4.1 CONCEPTO DE BLOQUE DE PROGRAMA. PROGRAMA PRINCIPAL Y FUNCIONES

4.2 CARACTERISTICAS DE LAS FUNCIONES

4.3 PASO DE PARAMENTROS A UNA FUNCION: PASO POR VALOR Y PASO POR REFERENCIA

4.4 AMBITO DE LAS VARIABLES

4.5 LIBRERIAS DE FUNCIONES MAS COMUNES

4.1 CONCEPTO DE BLOQUE DE POGRAMA

Un bloque de programa es un conjunto de instrucciones relativas a la misma idea.

Los bloques se separan por separadores, puden ser lineas en blanco, comentarios.

De un bloque del programa principal para crear una funcion

Una funcion es un bloque del programa que esta definida fuera del main()

4.2 CARACTERISTICAS DE LAS FUNCIONES

Cuando queremos crearla hay que hacerla en dos partes: (ojo que esta el otro metodo)

− Declararlas antes de main y despues de las llamadas al preprocesador (DEFINE INCLUDE)

− y despues del main la declaracion de las funciones

La declaracion de una funcion : PROTOTIPO. Consiste en indicar un tipo (int,chart, float) de valor de retorno. Cuando no devuelva nada hay que ponerle el tipo void, la funcion no devulve nada void=vacio.)

tipo nombre( );

tipo= int,float,char,double,void

nombre= El que le damos nosotros, ponerle siempre uno significativo

( )= Lista de parametros separados por comas de las variables que recibe la funcion y sus tipos

p.e. (int a,float x)

dentro del main podemos llamar a la funcion

definicion

tipo nombre(lista de parametros)

{

codigo

}

Programa que con dos funciones una sume dos numeros y otra calcula su cuadaro y lo devuelve.

```
#include <stdio.h>

float sumar(float a, float b);

float cuadrado(float b);

main()

{

float x,y,suma,cx,cy;

printf(introduce x);

scanf(%f,&x);
```

printf(introduce y);

scanf(%f,&y);

suma=suma(x,y);

cx=cuadrado(x);

cy=cuadrado(y);

}

float sumar (float a, floatb)

{

float suma;

suma=a+b;

return suma;

}

float cuadrado (float b)

{

float cuadradonum; //no se puede llamar a la variable igual que la funcion

*cuadradonum=b*b;*

return cuadradonum;

}

PASO DE ARRAYS A FUNCIONES

tipo nombre(int v[], int m[][10]) //matrices primero vacio y el resto rellenos

int v[10]

int *v

Las funciones no pueden devolver arraus

Es de tipo simple

4.3 PASO DE PARAMETROS A UNA FUNCION: PASO POR VALOR Y PASO POR REFERENCIA O DIRECCION

Paso por valor consiste en pasar a la funcion una copia del valor del parametro o argumentos.

La funcion trabaja con ese valor y cualquier cambio que se haga ene el valor del parametro desaparece al terminar la funcion.

Los tipos simples int,float,char,doble se pasan por valor

void funcion(int a)

{

*printf(%d,a) ----**10***

a=8;

printf(%d,a) --- ***8***

}

main()

{

*int b=10: printf(%d,b);---**10***

funcion(b);

*printf(%d,b); --**10***

}

Paso por referencia le pasamos la direccion memoria de la variable. Cualquier cambio en la funcion varia fuera.

Se le pone un operador& direccion de memoria y * para el contenido de la memoria

*void funcion(int *a)*

{

*printf(%d,*a); **10***

**a=8; //posicion de memoria que se le asigna el valor 8*

*printf(%d,*a); **8***

}

main()

{

int b=10;

printf(%d,b); ***10***

funcion(&b);

printf(%d,b); ***8***

}

Los arrays siempre se pasan por referencia

float funcion(int v[])

float funcion(int m[][20][15])

4.4 AMBITO DE LAS VARIABLES

Es el conjuntod e bloques de programa dentro de los cuales la variable esta declarada y por tanto puede utilizarse.

main()

{

int a;

a=a+10; //no da error poruq a esta declarada

}

void funcion(int a)

{

a=7; //a esta declarada (dentro de funcion(int a)

}

void funcion (int x)

{

a=7; //ERROR el ambito de a es solo en la funcion

}

main()

{

```
int a;
a=a+10;
funcion(a);
}
void funcion (int x)
{
int a;
printf(%d,a); //no muestra nada esta indeterminado
}
main()
{
int a;
a=a+10;
funcion(a);
}
```

Lo correcto seria

```
void funcion(int x)
{
printf(%d,x);
}
main()
{
int a=3;
a=a+10;
funcion(a);
}
```

VARIABLE GLOBAL

No utilizar en los programas, pero tenerla en cuenta para el test

int a=8; //antes del main

main()

{

a=a+10;

funcion(a);

}

void funcion(int x)

{

printf(%d,a); //IMPRIMIRA 18

}

ojo que si dentro del main introducimos el valor a=5 cambiamos la variable

4.5 LIBRERIAS DE FUNCIONES MAS COMUNES

#include < >

Nombre cabecera	Traduccion	Funciones	Cometarios
stdio.h	standar in/out	scanf() printf()	
conio.h	console in/out	getch() getche() highvideo() lowvideo()	Igual que getch() pero muestra por pantalla el caracter
ctype.h	Tipo de dato	isdigit() isalpha() isalnum() isupper()	Devuelve 1 si el carácter es numero 0 si no lo es Devuelve si esl alfabetico Si esta en mayusculas
string.h	Cadenas de	strlen()	Devuelve la longitud

	caracteres	strcmp(s1,s2)	Compra si iguales=0 no iguales ¡= 0 confuso esta alreves
bios.h	Funciones para acceder a la bios		
time.h	Funcion para acceder al tiempo del sistema		
locate.h	Funciones tema geografica		
dos.h	Funciones con sistema operativo		
math.h	sin(x) asin(a) cos(x) acos(a) tan(x) atan(a) abs() valor absoluto entero fabs() valor absoluto real log10() logarimo neperiano log(10)		

SEPTIEMBRE 2001

void MatrizMedia(float m1[][20], float m2[][20], float mmedia[][20], int nf, int nc)

{

int i,j

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

for(j=0;j<nc;j++)

mmedia[i][j]=(m1[i][j]+m2[i][j])/2;

}

#define MAXF 10

#define MAXC 20

void LeerMatriz(float m[][MAXC], int nf, int nc);

void MostrarMatriz(float m[][MAXC], int nf, int nc);

void MatrizMedia(float m1[][MAXC], float m2[][MAXC], float mmedia[][MAXC], int nf, int nc);

main();

{

int nfilas, ncols;

float matriz1[MAXF][MAXC],matriz2[MAXF][MAXC],matrizmedia[MAXF][MAXC];

do

{

scanf(%d),&nfilas);

}

while(nfilas<2 || nfilas >MAXF);

do

{

scanf(%d),&ncols);

}

while(ncols<2 || ncols>MAXC);

LeerMatriz(matriz1, nfilas, ncols);

LeerMatriz(matriz2,nfilas,ncols);

MatrizMedia(matriz1,matriz2,matrizmedia,nfilas,ncols);

MostrarMatriz(matrizmedia,nfilas,ncols);

MostrarMatriz(matriz1,nfilas,ncols);

MostrarMatriz(matriz2,nfilas,ncols);

Void LeerMatriz (.)

..

FEBRERO 2002

Crear funcion que devuelva 1 si es capicua y 0 si no lo es

a)

int capicua(int n)

{

```
if ( n==invertir(n)
return 1;
else
return 0; //no haria falta poner testo
}
int invertir (int n)
{
int nespejo=0
while (n !=0)
{
nespejo=nespejo*10+n%10 ;
n=n/10;
}
return nespejo;
```

p.e.:

1351 135

nespejo =1

135 13

*nespejo=1*10+5=15*

131

*nespejo=15*10+3=153*

1 0

*nespejo=153*10+1*

nespejo=1531

b)

```
#include <stdio.h)
```

```
int capicua(int n);

int invertir(int n) ;

main()

{

int cont=0 ;i=0 ;

while (cont<20)

{

if (capicua(i)==1)

{

cont++;

printf(.....%d,i)

}

i++

}

}

/* aqui estarian las funciones*/
```

Si declaro al principio las funciones el orden no importa, el orden que estan escritas despues del main

EJERCICIO

Funcion que dado un vector devuelva la suma de todos sus elementos

```
float sumarvector(float v[],int n)

{

float suma=0;

int i;

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

suma=suma+v[i];

return suma;
```

```
}
```

EXAMEN FEBRERO 2001 EJERCICIO 1

A<=elem<=b

```
float MediaEntreayb(float m[][3], int nf, int nc, float a, float b, int *numele)
{
int i,j,numele=0;
float suma=0;media;
*numele=0;
for(i=0;i<nf;i++)
for(j=0;j<nc;j++)
if(m[i][j]>=a && m[i][j]<=b)
{
suma=suma+m[i][j];
*numele++;
}
if(*nunele!=0)
media=suma/*numele++
else
media=−1; //nunca devueve
return media;
}
#include <stdio.h>
float MediaEntreayb(float m[][3], int nf, int nc, float a, float b, int *numele);
main()
{
float m[4][3]={ la del enunciado };
```

float a,b,mediam;

int nfilas,ncols=3;

scanf().... introducir valores

mediam= MediaEntreayb(m,nfilas,ncols,a,b,&nele);

if (nele==−1)

printf(no hay);

else

printf(la media es %f,mediam);

}

EJERCICIOS DE PRACTICAS

1.−CALCULAR EL AREA DE UN TRIANGULO

#include <stdio.h>

main()

{

//declaracion de variables

float base,altura,area;

//entrada de datos

printf("introduce la base:");

scanf("%f",&base);

printf("introduce la altura:");

scanf("%f",&altura);

//proceso de datos

*area=base*altura/2;*

//salida de datos

printf("El area e: %f", area);

//Esperamos para pulsar una tecla

```
getch();
}
```

2.−CALCULAR EL AREA DE UN TRIANGULO

```
#include <stdio.h>
#include <conio.h>
main()
{
//declaracion de variables
float base,altura,area;
//entrada de datos seguidos
printf("Introduce la base y la altura:");
scanf("%f%f",&base,&altura);
//proceso de datos
area=base*altura/2;
//salida de datos
printf("El area e: %f", area);
//Esperamos a pulsar una tecla
getch();
}
```

3.−CALCULAR EL AREA DE UN TRIANGULO

```
#include <stdio.h>
#include <conio.h>
main()
{
//declaracion de variables
float base,altura,area;
```

```
//entrada de datos seguidos
printf("Introduce la base y la altura:");
scanf("%f%f",&base,&altura);
//proceso de datos
area=base*altura/2;
//salida de datos
printf("La base es %f y la altura es %f \n",base,altura);
// \n significa salto de pagina en pantalla cada vez que se ponga parte el programa desde una linea nueva
// tambien \n puedo ponerlo al principio del printf("\n .....), en cualquier posicion dentro de las comillas
// del printf (nos pueden poner una pregunta de test para averiguar la salida correcta con el n\
printf("Entonces el area resultante es: %f", area);
//Esperamos a pulsar una tecla
getch();
}
```

4.− AREA DE UN TRIANGULO

```
#include <stdio.h>
#include <conio.h>
main()
//int es una variable entera si introduzco 3.589 el solo guarda el valor 3
//muy importante definir la variable, float es real y si nos piden en el enunciado que los valores
//son enteros, no podemos poner una variable real.
//a la hora de utilizar la variable no es %f sino %d de decimal o %i de integer
{
//declaracion de variables
int base,altura;
float area; //el area tenemos que ponerla con variable real, puesto que al hacer una division entre 2 puede
```

```
//dar un resultado decimal

//entrada de datos seguidos

printf("Introduce la base y la altura:");

scanf("%d%d",&base,&altura);

//proceso de datos

area=base*altura/2;

//salida de datos

printf("La base es %d y la altura es %d \n",base,altura);

printf("Entonces el area resultante es: %f", area);

//Esperamos a pulsar una tecla

getch();

}
```

5.− PROGRAMA QUE LEA DOS NUMEROS ENTEROS QUE CALCULE SUMA,RESTA,PRODUCTO Y DIVISION

```
#include <stdio.h>

#include <conio.h>

main()

{

// declaración de las variables

int a, b, suma, resta, producto;

float division;

//entrada de datos

printf("Introduce el numero A: ");

scanf("%d",&a);

printf("Introduce el numero B: ");

scanf("%d",&b);

//Calculo de los datos
```

```
suma=a+b;

resta=a−b;

producto=a*b;

division=(float) a/b; //hay que tener en cuenta en c que el tipo de la variable para resultado

//int/int => si los dos numerandos son int => que la division es entera el resultado va a ser int

//int/int = int se queda solo con la parte entera del resultado

//cono de finimos el valor float para division nos saldria el valor int.000000 y los ceros del float

//si alguno fuera del tipo float la division seria real

//La solución es forzar a que una operacion sea de un tipo determinado

//division=(float) a/b => esto implica que la operacion va a ser del tipo real

//Ojo si "a" o "b" fuese real ya no haria falta.

//El compilador es el que elige int/int como el mayor es int => resultado int

//Si temeos int/float => como el mayor es float => el resultado va a ser float

//p.e. float division; si tenemos division=7/2; printf("%f", division)

// que se muestra por pantalla??

//El resultado va a 3.000000 puesto que 7 y 2 son int => que le forzamos a que la operacion

//sea int

//si fuese 7/2.0; => int/float => float el resultado seria 3.500000

//Mostramos resultados

printf("La suma de a+b es: %d\nLa resta de a−b es: %d\nEl producto de a*b es: %d\nLa division de a/b es: %f",suma,resta,producto,division);

//Tambien podiamos haber puesto un printf independiente a cada resultado

//esperamos a pulsar una tecla

getch();

}
```

6.− PROGRAMA QUE LEA DOS NUMEROS ENTEROS QUE CALCULE SUMA,RESTA,PRODUCTO Y DIVISION

```
#include <stdio.h>

#include <conio.h>

main()

{

// declaración de las variables

int a, b, suma, resta, producto, resto;

float division;

//entrada de datos

printf("Introduce el numero A: ");

scanf("%d",&a);

printf("Introduce el numero B: ");

scanf("%d",&b);

//Calculo de los datos

suma=a+b;

resta=a−b;

producto=a*b;

division=(float) a/b;

resto=a%b;

//Otro operador es % resto=a%b; las variables siempre tienen que se variable int

//calcula lo que sobra de dividir un numero a otro

//2%7=2

//7%2=1

//8%2=0

//Mostramos resultados

printf("La suma de a+b es: %d\nLa resta de a−b es: %d\nEl producto de a*b es: %d\nLa division de a/b es:
%f",suma,resta,producto,division);

printf("\n El resto de a/b es: %d",resto);
```

```
//esperamos a pulsar una tecla

getch();

}
```

7.−PROGRAMA QUE LEA DOS NUMEROS ENTEROS QUE CALCULE SUMA,RESTA,PRODUCTO Y DIVISION

```
#include <stdio.h>

#include <conio.h>

main()

{

// declaración de las variables

float a, b, suma, resta, producto, division;

int resto;

//entrada de datos

printf("Introduce el numero A: ");

scanf("%f",&a);

printf("Introduce el numero B: ");

scanf("%f",&b);

//Calculo de los datos

suma=a+b;

resta=a−b;

producto=a*b;

division= a/b;

resto=(int)a%(int)b;

// en este caso con dos variables float las convertimos al tipo int

// necesario para hacer la operacion %

// int%int=int

//Mostramos resultados
```

printf("La suma de a+b es: %f", suma);

printf("\n La resta de a−b es: %f", resta);

*printf("\n El producto de a*b es: %f", producto);*

printf("\n La division de a/b es: %f",division);

printf("\n El resto de a/b es: %d",resto);

//esperamos a pulsar una tecla

getch();

}

8.− OPERACIONES COMPLEJAS

//Queremos hacer una raiz cuadrada

//la libreria math.h

//de esta liberia las que mas nos interesan son:

//int abs(int x); calcula el valor anbsoluto de un numero entero

//abs es el nombre de la funcion

//lo que va entre () es la lista de variables que necesita la funcion para hacer algo

//en este caso estoy utilizando una variable de tipo entero la x es el nombre de la variable

//el int del principio es de que forma devuelve el resultado.

//"double fabs (double x)" calcula el valor absoluto de un numero real doble es un real con mas precision

//"double cos (double x)"

//"double sin (double x)"

//"double tan (double x)"

//"double acos (double x)"

//"double asin (double x)"

//"double atan (double x)"

//ojo el angulo es en radianes pi=3,1416 => rad=180 grados

//"double sqrt (double x)" es la raiz cuadrada

```
//"double pow (double x,y )" tenemos dos paramentros, calcula x elevado a y
//"double cosh (double x)" funciones hiperbolicas
//"double sinh (double x)"
//"double tanh (double x)"
//"double log (double x)"
//"double log10 (double x)" logaritmo neperiano
//"double exp (double x)" e elevado a x
//sacar la raiz cuadrada de a y b
//logaritmos
#include <stdio.h> //printf(), scanf()
#include <conio.h> //getch()
#include <math.h> //sqrt()
main()
{
// declaración de las variables
float a, b, raiza, raizb, log_a, log_b, log10_a, log10_b;
//entrada de datos
printf("Introduce un numero A: ");
scanf("%f",&a);
printf("Introduce un numero B: ");
scanf("%f",&b);
//Calculo de los datos
raiza = sqrt(a);
raizb = sqrt(b);
log_a = log(a);
log_b = log(b);
```

log10_a = log10(a);

log10_b = log10(b);

//Mostramos resultados

printf("La raiz cuadrada de a=%f es: %f\n", a, raiza);

printf("La raiz cuadrada de b=%f es: %f\n", b, raizb);

printf("El logartimo de a=%f es: %f\n", a, log_a);

printf("El logartimo de b=%f es: %f\n", b, log_b);

printf("El logartimo neperiano de a=%f es: %f\n", a, log10_a);

printf("El logartimo neperiano de b=%f es: %f\n", b, log10_b);

//esperamos a pulsar una tecla

getch();

}

9.− SENTENCIAS DE CONTROL DE FLUJO QUE NOS PERMITE VEN POR DONDE QUEREMOS QUE

EJECUTE EL PROGRAMA.

La sentencia if es un si (condicional)

sentencia1; hace sentencia1 si no se cumple la

sentencia1 no se hace

if (<condicion>) <sentencia>;

if (a>=1) logaritmo= log(a);

Los condicionantes o comparadores

a >= b a mayor o igual que b

> mayor

< menor

= igual

== sirve para ver si la parte izquierda y la derecha son exactamente iguales

pero para dos expresiones

!= distinto

if (a!=b) ....;

ahora quiero hacer esto

a != 10

b > 10

if (a!=10 && b>10) ...; como pongo el

AND=&&

OR=||

NOT=!

p.e. la negacion tiene que ir entre parentesis

if (!(a!=10 && b>10)) ....;

10.− EJERCICIO DADA UNA VARIABLE MUESTRE SU VALOR ABSOLUTO SIN UTILIZAR LA

```
//funcion abs
#include <stdio.h> //printf(), scanf()
#include <conio.h> //getch()
main()
{
// declaración de las variables
float a,resultado;
//entrada de datos
printf("Introduce un numero A: ");
scanf("%f",&a);
//Calculo de Datos
resultado=a;
if (a<0) resultado=−a;
//Mostramos resultados
```

```
printf("El valor absoluto de A=%f es: %f\n", a, resultado);

//esperamos a pulsar una tecla

getch();

}
```

11.− ECUACION DE SEGUNDO GRADO

```
//Repaso IF

//if (condicion)

// S1;

// else S2;

// !(num<0 || num>0)

// !(num<0) && !(num>30)

// num>=0 && num <=30

//programa para resolver la ecuación de segundo grado

//Tener en cuenta b^2−4ac>0

#include <stdio.h> //printf(), scanf()

#include <conio.h> //getch()

#include <math.h>

main()

{

//definicion de variables

float a,b,c,par1, resultado1, resultado2;

//introduccion de datos

printf("Calcular las raizes de la ecuacion ax^2+bx+c=0 \n");

printf("Introduce ahora las ctes, a, b y c \n");

printf("Valor de a: " );

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

```
printf("\nValor de b: " );
scanf("%f", &b);
printf("\nValor de c: ");
scanf("%f", &c);
//definimos limitaciones
//a!0
//(b^2−4*ac)>0
par1=pow(b,2)−a*c*4;
if (a==0)
{
if (b!=0)
{
resultado1=−c/b;
//mostramos resultado cuando a=0 solo seria una raiz
printf("\nEl resultado seria una unica raiz de valor %f:", resultado1);
}
else
//
printf("\nNo seria una ecuacion pues %f=!0",c);
}
else
if (par1<0)
printf("\nNo se puede resolver la ecuacion");
else
{
//calculamos operaciones
```

```
resultado1=((−b)+sqrt(par1))/(2*a);

resultado2=((−b)−sqrt(par1))/(2*a);

if (resultado1==resultado2)

//mostramos resultado

printf("\nLas dos raizes R1 y R2 son igulales y su valor es: %f", resultado1);

else

{

printf("\nLa raiz R1=%f",resultado1);

printf("\nLa raiz R2=%f",resultado2);

}

}

getch();

}
```

12.− PROGRAMA QUE DIGA SI PUEDES VOTAR O NO

```
#include <stdio.h>

#include <conio.h>

main()

{

int edad, te_quedan, desde;

printf("dime tu edad: "); scanf("%d",&edad);

if (edad == 18)

printf("ya puedes votar desde ya");

else

if (edad >18)

{

printf("ya puedes votar");
```

```
desde=edad−18;

printf("\nDesde hace %d años", desde);

}

else

{

//ojo que hay que poner el corchete dentro del if por que si no

//solo cojeria el printf e la primera linea y el resto lo ejecutaria

//siempre

printf("aun no puedes votar");

te_quedan=18−edad;

printf("\nAun te faltan %d años", te_quedan);

//esto lo podria acortar con otra sentencia

//printf("Aun no puedo votar \n te faltan %d años", 18−edad);

}

getch();

}
```

13.− PIDA DOS NºENTEROS Y LOS ENVIA A PANTALLA PRIMERO EL MAYOR Y LUEGO EL OTRO

```
#include <stdio.h> //printf(), scanf()

#include <conio.h> //getch()

main()

{

//definicion de variables

int a,b;

//introduccion de datos

printf("Valor de a: " );

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

```
printf("\nValor de b: " );

scanf("%d", &b);

//Comparacion y salida

if (a==b)

printf("Valor de a es igual que b");

else

{

if (b<a)

printf("Ordenados de mayor a menor a=%d,b=%d",a,b);

else

printf("Ordenados de mayor a menor b=%d,a=%d",b,a);

}

getch();

}
```

14.− PIDA 3 NºENTEROS Y LOS ENVIA A PANTALLA PRIMERO EL MAYOR Y LUEGO EL OTRO

```
#include <stdio.h> //printf(), scanf()

#include <conio.h> //getch()

main()

{

//definicion de variables

int a,b,c;

//introduccion de datos

printf("Valor de a: " );

scanf("%d", &a);

printf("\nValor de b: " );

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

```
printf("\nValor de b: " );

scanf("%d", &c);

//a>b si ? b>c si abc

//a>b si ? b>c no => a>c => bac

//a>b si ? b>c no => a>c => cab

//a>b no ? a>c si => bac

//a>b no ? a>c no => b>c no => cba

//a>b no ? a>c no => b>c si => bca

if (a>b) //a>b

{

if (b>c)//b>a

{

printf("Ordenados de mayor a menor %d,%d,%d",a,b,c); //a>b>c

}

else //b>a

{

if (a>c)//a>c

{

printf("Ordenados de mayor a menor %d,%d,%d",b,a,c); //b>a>c

}

else //a<c

{

printf("Ordenados de mayor a menor %d,%d,%d",c,a,b); //c>a>b

}

} //fin else b>a

} //fin if a>b
```

```
else //a<b
{
if (a>c) //a>c
{
printf("Ordenados de mayor a menor %d,%d,%d",b,a,c); //b>a>c
}
else //a<c
{
if (b>c) //b>c
{
printf("Ordenados de mayor a menor %d,%d,%d",b,c,a); //b>c>a
}
else //b>c
{
printf("Ordenados de mayor a menor %d,%d,%d",c,b,a); //c>b>a
}
} //fin else a<c
} // fin else a<b
getch();
}
```

15 HACER UN PROGRAMA C QUE PIDA UN CARACTER POR TECLADO Y QUEENVIE MENSAJE A PANTALLA INDICANDO SI ES UN DIGITO, UNA LETRA MAYUSCULA O UNA LETRA MINUSCULA

```
//Char solo puede guardar UN unico caracter 0..9 A..Z a..z
//lo que guarda es el digito 0 al 9
#include <stdio.h> //printf(), scanf()
#include <conio.h> //getch()
```

```
main()
{
//definicion de variables
char caracter;
//introduccion de datos
printf("Introduce un caracter: ");
scanf("%c",&caracter);
// 0 9 A Z a z
//si <=9 >=0 digito
//si <=Z >=A mayuscula
//si <=z >=a minuscula
if (caracter<='9'&& caracter >= '0')
{
printf("Tenemos un DIGITO");
}
else
{
if(caracter<='Z'&& caracter >='A')
{
printf("Tenemos una MAYUSCULA");
}
else
{
if(caracter<='z'&& caracter >='a')
printf("Tenemos una MINUSCULA");
else
```

```
printf("Tenemos otra COSA");

}

}

getch();

}
```

15 DE OTRA FORMA

```
//Char solo puede guardar UN unico caracter 0..9 A..Z a..z

//lo que guarda es el digito 0 al 9

//Hacer un programa c que pida un caracter por teclado y que

//envie mensaje a pantalla indicando si es un digito, una letra mayuscula

//o una letra minuscula

#include <stdio.h> //printf(), scanf()

#include <conio.h> //getch()

main()

{

//definicion de variables

char caracter;

//introduccion de datos

printf("Introduce un caracter: ");

scanf("%c",&caracter);

// 0.....9 A.......Z a......z

//si <=9 >=0 digito

//si <=Z >=A mayuscula

//si <=z >=a minuscula

if (caracter<='9'&& caracter >= '0')

{
```

```
printf("Tenemos un DIGITO");

}

if (caracter<='Z'&& caracter >='A')

{

printf("Tenemos una MAYUSCULA");

}

if (caracter<='z'&& caracter >='a')

{

printf("Tenemos una MINUSCULA");

}

if (!(caracter<='9'&& caracter >= '0')||

(caracter<='Z'&& caracter >='A')||

(caracter<='z'&& caracter >='a'))

{

printf("Tenemos otra COSA");

}

getch();

}
```

16 ESCRIBIR PROGRAMA EN C QUE PIDA UN NUMERO ENTERO Y ENVIE UN MENSAJE A PANTALLA INDICANDO SI EL NUMERO ES PAR O IMPAR

```
#include <stdio.h> //printf(), scanf()

#include <conio.h> //getch()

main()

{

//variables

int a,b,resto;

b=2;
```

```
//Entrada datos
printf("Introduce numero ");
scanf("%d",&a);
//Buscamos saber si es par o impar
resto=a%b; //2/2 resto 0 4/2 resto 0 3/2 resto 1 5/2 resto 1
if (resto==0)
printf("El numero es par");
else
printf("El numero es impar");
getch();
}
```

18 CALCULADORA

```
//switch(expresion)
//{
//case cte1: break; salta de caso case teene que ser una constante 18, 70.3
//case cte2: break; si no lo ponemos siguie por el siguiente caso
//.
//.
//case cten:
//default; si no es ninguno de esos casos que haga lo que biene en default
//}
// hacer una calculadora
//opcion 1.−suma 2. resta 3 producto 4 division
//
#include <stdio.h>
#include <conio.h>
```

```
main()
{
// declaración de las variables
int opcion;
float a,b,resultado;
//entrada de datos
printf("CALCULADORA \n");
printf("pulsa 1.− para SUMAR \n");
printf("pulsa 2.− para RESTAR \n");
printf("pulsa 3.− para MULTIPLICAR \n");
printf("pulsa 4.− para DIVIDIR \n");
scanf("%d",&opcion);
printf("\nIntroduce el numero A: ");
scanf("%f",&a);
printf("\nIntroduce el numero B: ");
scanf("%f",&b);
//Calculo de los datos mediante switch
switch(opcion)
{
case 1:
resultado=a+b;
printf("\nEl resultado de la suma es %f",resultado);
break;
case 2:
resultado=a−b;
printf("\nEl resultado de la resta es %f",resultado);
```

```
break;

case 3:

resultado=a*b;

printf("\nEl resultado de la multiplicacion es %f",resultado);

break;

case 4:

resultado=a/b; //no hace falta (float)a/b porque ya son float

printf("\nEl resultado de la divisionn es %f",resultado);

break;

default:

printf("\nHa introducido una opcion no valida");

}

getch();

}
```

19 EJERCICIO CONTAR HASTA 100 CON CADA UNO DE ESTOS ESQUEMAS EMPEZAR EN EL 1..100 O DEL 0 AL 100

```
//while ()

//{

//sentencia a;

//sentencia b;

//}

//como minimo se ejecuta 0 veces y como maximo infinitas veces

//do

//{

//sentencia 1;

//sentencia 2;

//}
```

```
//while(exp)
//como minio se ejecuta 1 vez y como maximo infinitas veces
//for (inalizacion;expresion;actualizacion)
//{
//sentencia 1;
//sentencia 2;
//}
//ejercicio contar hasta 100 con cada uno de estos esquemas
//empezar en el 1..100 o del 0 al 100
#include <stdio.h>
#include <conio.h>
main()
{
//definimos variables
int opcion, a=1;
do
{
printf("pulsa 1.− para usar while\n");
printf("pulsa 2.− para usar do..while\n");
printf("pulsa 3.− para usar for\n");
scanf("%d",&opcion);
}
while( opcion!=1 && opcion!=2 && opcion!=3);
//Calculo de los datos mediante switch
//printf("%d\t",a); \t muestra los datos tabulados
switch(opcion)
```

```
{
case 1:
//Con while
while(a<=100)
{
printf("%d\t",a);
a++;
}
break;
case 2:
//en este caso no hace falta inicializar la varible pues al ir directamente con el case ya tiene el valor
//definido arriba.
//con do..while
do
{
printf("%d\t",a);
a++;
}
while(a<=100);
break;
case 3:
//no hace falta inicializar a en este caso pues viene del valor arriba definido
//con for
for (a; a<=100; a++)
//no hace falta poner {} pues solo tenemos una linea
printf("%d\t",a);
```

```
break;
}
getch();
}
```

20 CONTAR HASTA LOS 100 PRIMEROS NÚMEROS PARES UTILIZANDO UNO DE LOS BUCLES

```
#include <stdio.h>
#include <conio.h>
main()
{
//definimos variables
int a=0,b=0;
//utilizando el while
while(b<100)
{
if (a%2==0)
{
b++;
printf("%d %d \n ",a,b);
}
a++;
}
printf("\n\n");
// utilizando el do..while
a=0,b=0;
do
{
```

```
if (a%2==0)
{
b++;
printf("%d %d \n ",a,b);
}
a++;
}
while(b<100);
//con for
printf("\n\n");
for (a=0,b=0;b<100;a++)
{
if (a%2==0)
{
b++;
printf("%d %d \n ",a,b);
}
a++;
}
getch();
}
```

21 PEDIR DOS NUMEROS ENTEREOS POR TECLADO NºDE INICIO Y Nº DE FIN COMPROBAR QUE EL INICIO SEA MENOR QUE EL FIN

SI INICIO>FIN VOLVER A PEDIRLO Y MOSTRAR LOS NUMEROS ENTRE ELLOS DOS

```
#include <stdio.h>
#include <conio.h>
main()
```

```
{
//definimos variables
int inicio,fin,contador;
do
{
printf("Introduce el valor inicio y fin");
scanf("%d %d",&inicio,&fin);
}
while(inicio>=fin);
for (contador=inicio; contador<=fin; contador++)
printf("%d\t",contador);
getch();
}
```

22 DADO UN N POR TECLADO SUMAR SUS CIFRAS Y MOSTRAR POR PANTALLA LA SUMA Y EL NUMERO QUE PEDIMOS

```
#include <stdio.h>
#include <conio.h>
main()
{
int n,aux,suma=0;
printf("introduce un numero n:");
scanf("%d",&n);
aux=n;
while(aux!=0)
{
suma=suma+aux%10; //aux%10 coge la varible mas a la derecha
aux=aux/10;
```

```
}

printf("El numero %d sus cifras suman %d",n,suma);

getch();

}
```

23 ESCRIBIR UNA PROGRAMA EN C QUE APARTIR DE UN NUMERO ENTERO LEIDO POR TECLADO CALCULE Y MUESTRE LA SUMA DE LA MAYOR DE SUS CIFRAS CON LA MENOR DE SUS CIFRAS

```
#include <stdio.h>

#include <conio.h>

main()

{

int n,aux,resto,suma=0,mayor=0,menor=9;

printf("introduce un numero n:");

scanf("%d",&n);

resto=n;

aux=n;

while(aux!=0)

{

resto=aux%10;

if (resto>mayor)

mayor=resto;

if (resto<menor)

menor=resto;

aux=aux/10;

}

suma=mayor+menor;

printf("El numero %d sus cifras suman %d",n,suma);
```

```
getch();

}

//pedir a parte del numero un digito entre 1..9

//otra varible d extricta

//si la suma es divisible por n
```

24 ESCRIBIR UNA PROGRAMA EN C QUE APARTIR DE UN NUMERO ENTERO LEIDO POR TECLADO CALCULE Y MUESTRE LA SUMA DE LA MAYOR DE SUS CIFRAS CON LA MENOR DE SUS CIFRAS PEDIR A PARTE DEL NUMERO UN DIGITO ENTRE 1..9 OTRA VARIBLE D EXTRICTA SI LA SUMA ES DIVISIBLE POR N

```
#include <stdio.h>

#include <conio.h>

main()

{

int n,d,aux,resto,suma=0,mayor=0,menor=9;

printf("introduce un numero n:\n");

scanf("%d",&n);

do

printf("introduce una cifra de 1 a 9:);

scanf("%d",&d);

while(d<1 || d>9);

resto=n;

aux=n;

while(aux!=0)

{

resto=aux%10;

if (resto>mayor)

mayor=resto;

if (resto<menor)
```

```
menor=resto;
aux=aux/10;
}
suma=mayor+menor;
printf("El numero %d sus cifras suman %d",n,suma);
//conprobar si es suma divisible por n
if (suma%d==0)
printf("La suma es divisible");
else
printf("La suma es no esdivisible");
getch();
}
```

25 PEDIR UN NUMERO N POR TECLADO MAYOR QUE 0 QUE MUESTRE POR PANTALLA

1

1 2

1 2 3

...

1 2 3 4.....N

```
#include <stdio.h>
#include <conio.h>
main()
{
int n,i,m=1;
//limitar dato
do {
printf("introduce un numero mayor que 0:");
```

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

}

while ( n<=0 );

//calculos

for(m;m<=n;m++)

{

for(i=1;i<=m;i++)

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

printf("\n");

}

getch();

}
```

27 ESCRIBIR UNA PROGRAMA EN C QUE APARTIR DE UN NUMERO ENTERO LEIDO POR TECLADO CALCULE Y MUESTRE LA SUMA DE LA MAYOR DE SUS CIFRAS CON LA MENOR DE SUS CIFRAS PEDIR A PARTE DEL NUMERO UN DIGITO ENTRE 1..9 /OTRA VARIBLE D EXTRICTA /SI LA SUMA ES DIVISIBLE POR N

```
#include <stdio.h>

#include <conio.h>

main()

{

int n,i,d,contador,aux,resto,suma=0,mayor,menor;

printf("introduce cuantos numeros quieres calcular:\n");

scanf("%d",&n);

do

{

printf("introduce una cifra de 1 a 9:");

scanf("%d",&d);

}
```

```
while (d<1 || d>9);
//inicializar variables para el bucle exterior
contador=0;
i=0;
while (contador<n)
{
//inicializamos las variables del bucle anterior
aux=i;
mayor=0;
menor=9;
while(aux!=0)
{
resto=aux%10;
if (resto>mayor)
mayor=resto;
if (resto<menor)
menor=resto;
aux=aux/10;
}
if ((mayor+menor)%d==0)
{
contador++;
printf("%d: Encontrado el %d divisible por %d\n",contador,i,d);
}
i++;
}
```

```
getch();
}
```

28 COMPLETAR EL EJERCICIO DE LA CALCULADORA Y PREGUNTAR SI SE QUIERE REPETIR

```
#include <stdio.h>
#include <conio.h>
main()
{
// declaración de las variables
int opcion;
float a,b,resultado;
char repetir;
//entrada de datos
repetir = 's';
do
{
//inicio calculadora
printf("CALCULADORA \n");
printf("pulsa 1.− para SUMAR \n");
printf("pulsa 2.− para RESTAR \n");
printf("pulsa 3.− para MULTIPLICAR \n");
printf("pulsa 4.− para DIVIDIR \n");
scanf("%d",&opcion);
printf("\nIntroduce el numero A: ");
scanf("%f",&a);
printf("\nIntroduce el numero B: ");
scanf("%f",&b);
```

```
//Calculo de los datos mediante switch
switch(opcion)
{
case 1:
resultado=a+b;
printf("\nEl resultado de la suma es %f",resultado);
break;
case 2:
resultado=a−b;
printf("\nEl resultado de la resta es %f",resultado);
break;
case 3:
resultado=a*b;
printf("\nEl resultado de la multiplicacion es %f",resultado);
break;
case 4:
resultado=a/b; //no hace falta (float)a/b porque ya son float
printf("\nEl resultado de la divisionn es %f",resultado);
break;
default:
printf("\nHa introducido una opcion no valida");
}
flushall();
printf("\nQuiere repetir? (s/n)");
scanf("%c",&repetir);
}
```

while(repetir=='s' || repetir=='S');

//pero tenemos un problema con el buffer de entrada puesto que guarda los datos ordenados

// de manera que lo hemos introducido en el teclado

//Como el buffer no esta vacio no se puede

//podemos solucionarlo borrando el buffer antes de leer caracterer

//flushall(); <stdio.h>

//tambien podriamos hacer antes otro scanf(" ") antes y ya leeria el salto de linea anterior

//si hubiese varios saltos de linea abria que hace un bucle while""

getch();

}

30 PROGRAMA QUE TENGA UNA FUNCION QUE LEA UNA MATRIZ UNA FUNCION QUE MUESTRE UNA MATRIZ

UNA FUNCION QUE DADAS DOS MATRIZCES CALCULA Y DEVUELVA LA MATRIZ SUMA*/

*/**

maximo 10x20

leerMatriz (m,filas,cols); leer reales

void leerMatriz (float m[][20],int filas,int cols);

nos tiene que leer la matriz guarde los valores y devuelva

la matriz devuelve

una funcion nunca nos devuelve arrays hay que poner void por que se pasa por

referencia guardan el valor cuando se termina la funcion.

filas cols se pasan por valor desaparecen al terminar de ejecutarse

void MostrarMatriz (float m[][20],int filas, int cols);

void SumarMatrices (float m1[][20],float m2[][20],int filas,int cols,msuma[][20]);

main()

leer

**/*

```
/* nºfilas nºcolumnas m1 m2 calcule suma muestre m1 m2 msuma*/

#include <stdio.h>

#include <conio.h>

void leerMatriz ( float m[][20],int filas,int cols )

{

int i,j;

//introducimos los datos de la matriz

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

for(j=0;j<cols;j++)

{

printf(" [%d][%d]= ",i,j);

scanf("%f",&m[i][j]);

}

}

//funcion para mostrar matrices

void mostrarMatriz ( float m[][20],int filas,int cols )

{

int i,j;

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

for(j=0;j<cols;j++)

{

printf("%f\t ",m[i][j]);

if (j==cols−1)

printf("\n");

}

}
```

```
//funcion para sumar dos matrices
void sumarMatrices (float m1[][20],float m2[][20],int filas,int cols,float msuma[][20])
{
int i,j;
for(i=0;i<filas;i++)
for(j=0;j<cols;j++)
msuma[i][j]=m1[i][j]+m2[i][j];
}
//programa principal
main()
{
float m1[10][20],m2[10][20],msuma[10][20];
int filas, cols;
//entrada de datos
do
{
printf("Introduce el numero de filas: ");
scanf("%d",&filas);
}
while(filas<2 || filas >10);
do
{
printf("Introduce el numero de columnas: ");
scanf("%d",&cols);
}
while(cols<2 || cols >20);
```

```
//salida datos
printf("Introduce los datos de la matriz 1 \n");
leerMatriz(m1,filas,cols);
printf("Introduce los datos de la matriz 2 \n");
leerMatriz(m2,filas,cols);
printf("M1\n");
sumarMatrices(m1,m2,filas,cols,msuma);
mostrarMatriz(m1,filas,cols);
printf("\n");
printf("M2\n");
mostrarMatriz(m2,filas,cols);
printf("\n");
printf("suma M1+M2\n");
mostrarMatriz(msuma,filas,cols);
getch();
}
```

31 PROGRAMA PARA INVERTIR UN VECTOR

```
#include <stdio.h>
#include <conio.h>
main()
{
int max,i,j;
float vector[99],ivector[99];
do
{
printf("Introduce el tamaño del vector entre 2 y 99: ");
```

```
scanf("%d",&max);
}
while( max<2 || max>99);
printf("\nIntroduce los datos del vector :\n");
//lectura de datos del vector
j=max;
for(i=1;i<=max ;i++)
{
printf(" [%d]= ",i);
scanf("%f",&vector[i]);
ivector[j]=vector[i];//vamos inviertiendo el vector
j--;
}
//mostrar vector introducido
printf("El vector original es:\n");
for(i=1;i<=max ;i++)
printf(" [%f] ",vector[i]);
//mostrar por pantalla el inverso
printf("\nEl vector invertido es:\n");
for(i=1;i<=max ;i++)
printf(" [%f] ",ivector[i]);
getch();
}
```

32 FUNCION TRANSPUESTA DE UNA MATRIZ

/*a00 a01 a02 a00 a10 a20

a10 a11 a12 traspuesta a01 a11 a21

a20 a21 a22 a02 a12 a22

primero

//defirnir una funcion

//#define nf 10

//#define nc 10

//void calcular transpuesta (int m[][NC], int MT[][ç

//despiuesst mostra matriz original y matriz transpiuesta

matriz transpuesta

//la segunda parte

crear funcion

****/***

#define NF 10

#define NC 10

#include <stdio.h>

#include <conio.h>

//funcion leer matriz

void leerMatriz (float m[][NC],int filas,int cols)

{

int i,j;

//introducimos los datos de la matriz

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

for(j=0;j<cols;j++)

{

printf(" [%d][%d]= ",i,j);

scanf("%f",&m[i][j]);

}

```
}
//funcion mostramatriz
void mostrarMatriz ( float m[][NC],int filas,int cols )
{
int i,j;
for(i=0;i<filas;i++)
for(j=0;j<cols;j++)
{
printf("%f\t ",m[i][j]);
if (j==cols−1)
printf("\n");
}
}
//funcion para hacer transpuesta
void transMatriz (float m1[][NC],int filas,int cols,float mtras[][NC])
{
int i,j;
for(i=0;i<filas;i++)
for(j=0;j<cols;j++)
mtras[j][i]=m1[i][j];
}
//funcion para que transpuesta tenga salida en la misma
//sin definir una matriz auxiliar
//ojo como ver si una matriz es simetrica podiamos sacarlo de esta forma
//estamos recorriendo la parte diagonal superior
//solo que
```

```
void tMatriz (float m1[][NC],int filas,int cols)
{
int i,j;
float aux;
for(i=0;i<filas−1;i++)
for(j=i+1;j<cols;j++)
{
aux=m1[i][j];
m1[i][j]=m1[j][i];
m1[j][i]=aux;
}
}
//programa principal
main()
{
float m1[NF][NC],mtras[NC][NC];
int filas, cols;
//entrada de datos
do
{
printf("Introduce el numero de filas: ");
scanf("%d",&filas);
}
while(filas<1 || filas >NF);
do
{
```

```
printf("Introduce el numero de columnas: ");

scanf("%d",&cols);

}

while(cols<1 || cols >NC);

//salida datos

printf("Introduce los datos de la matriz 1 \n");

leerMatriz(m1,filas,cols);

printf("Matriz 1:\n");

transMatriz(m1,filas,cols,mtras);

mostrarMatriz(m1,filas,cols);

printf("\n");

printf("Su traspuesta:\n");

mostrarMatriz(mtras,cols,filas);

printf("\n");

printf("Su misma traspuesta:\n");

tMatriz(m1,filas,cols);

mostrarMatriz(m1,filas,cols);

getch();

}
```

ESCBIR UN PROGRAMA QUE TENGA DOS FUNCIONES, 0º UNA FUNCION QUE LEA POR TECLADO

1º ES UNA FUNCION QUE DADO UN VECTOR DEVUELVA EL VECTOR AL REVES 2º DADO UN VECTOR DEVUELVA EL MINIMO

MOSTRAR EN EL MAIN EL VECTOR ALREVES Y EL MINIMO MOSTRARLO TAMBIEN

```
#define TVECTOR 10

#include <stdio.h>

#include <conio.h>
```

```
void LeerVector(float v[],int tv)
{
int i;
printf("\nIntroduce los datos del vector :\n");
//lectura de datos del vector
for(i=0;i<tv ;i++)
{
printf(" [%d]= ",i);
scanf("%f",&v[i]);
}
}
//inversor
void Inversor(float v[],float iv[],int tv)
{
int i,j;
//operacion de inverir
j=tv-1;
for(i=0;i<tv ;i++)
{
iv[j]=v[i];//vamos inviertiendo el vector
j--;
}
}
//comparador
float Comparador(float v[], int tv)
{
```

```
int i;
float aux;
aux=v[0];
//operacion de memorizar
for(i=0;i<tv ;i++)
{
if (v[i]<aux)
aux=v[i];
}
return aux;
}
//funcion mostrar por pantalla
void MostrarVector(float v[],int tv)
{
int i;
for(i=0;i<tv ;i++)
printf(" [%f] ",v[i]);
}
main()
{
int i,n;
float vector[TVECTOR],ivector[TVECTOR],aux;
do
{
printf("Introduce el tamaño del vector: ");
scanf("%d",&n);
```

```
}
while(n<2 || n>10);
printf("\n");
LeerVector(vector,n);
MostrarVector(vector,n);
printf("\n");
Inversor(vector,ivector,n);
printf("\nEl vector invertido es:\n");
for(i=0;i<n ;i++)
printf(" [%f] ",ivector[i]);
printf("\n");
aux=Comparador(vector,n);
printf("El minimo valor es [%f]",aux);
getch();
}
programa que lea vector con una funcion, funcion que devuelva la suma de numeros primos de v int
funcion que dado un numero diga si es primo 1 si no 0, el programa principal muestra la sumasi es primo
#define NF 10
#define NC 10
#define NV 100
#include <stdio.h>
#include <conio.h>
void LeerMatriz ( float m[][NC],int filas,int cols )
{
int i,j;
//introducimos los datos de la matriz
```

```
for(i=0;i<filas;i++)
for(j=0;j<cols;j++)
{
printf(" [%d][%d]= ",i,j);
scanf("%d",&m[i][j]);
}
}
//comprueva si es primo
int EsPrimo(int num)
{
int esprimo=1,divisor;
for(divisor=2;divisor<num;divisor++)
{
if(num%divisor==0)
esprimo=0;
}
return esprimo;
}
//fincon suma primso
int SumaPrimos(int m[][NC], int filas, int cols, int v[NV] )
{
int i,j,sumaprimos=0,cont=0;
for(i=0;i<filas ;i++)
for(j=0;j<cols;j++)
{
if (EsPrimo(m[i][j])==1)
```

```
{
printf("\nEste es un numero primo: %d ",m[i][j]);
sumaprimos=sumaprimos+m[i][j];
v[cont]=m[i][j];
cont++;
}
}
printf("\el vector i\n");
for(i=0;i<cont;i++)
printf(" %d, ",v[i]);
printf("\n");
return sumaprimos;
}
//main
main()
{
float matriz[NF][NC];
int filas, cols, totalprimos,vector[NV];
//entrada de datos
do
{
printf("Introduce el numero de filas: ");
scanf("%d",&filas);
}
while(filas<1 || filas >NF);
do
```

```
{
printf("Introduce el numero de columnas: ");
scanf("%d",&cols);
}
while(cols<1 || cols >NC);
LeerMatriz(matriz,filas,cols);
totalprimos = SumaPrimos(matriz,filas,cols,vector);
printf("\nla suma de todos los numeros primos es: %d",totalprimos);
getch();
}
```