

TEMA 4

ALGUNAS FUNCIONES DEL LENGUAJE C

- 4.1.- Introducción.
- 4.2.- Función de salida exit.
- 4.3.- Función de salida _exit.
- 4.4.- Función de salida abort.
- 4.5.- Función de salida assert.
- 4.6.- Funciones de entrada/salida.
- 4.7.- Función de entrada salida estándar printf, scanf.
- 4.8.- Función de entrada salida putchar, getchar.
- 4.9.- Función de entrada salida puts, gets.
- 4.10.- Funciones de consola.
- 4.11.- Funciones rand, randomize y srand.
- 4.12.- Función clrscr.
- 4.13.- Ejercicios.

4.1.- Introducción.

El objetivo de este tema es describir algunas funciones del lenguaje C. Una función es una porción de código aislado que efectúa una acción determinada según una serie de parámetros separados por comas (puede no tenerlos) y que devuelve un valor dependiente de estas acciones (puede también no devolver valor alguno). Estos parámetros son valores que damos a la función para que trabaje con ellos.

Cuando ejecutamos una función decimos que la llamamos. La sintaxis de una llamada de función en C es:

```
nombre_de_funcion ( lista_de_parametros )
```

Para facilitar el almacenaje y el rápido uso de funciones y variables se utilizan librerías (.LIB) donde se almacenan en un formato ya compilado. Para no tener que incluir todas las funciones de una librería en nuestro programa, las funciones se separan por módulos.

Para utilizar una función de una librería se debe incluir el módulo de esta. Para esto se utiliza la siguiente sintaxis:

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

Cartagena99

4.2.- Función de salida exit.

La función exit se encuentra declarada en la biblioteca estándar (stdlib.h) y da lugar a la terminación automática de un programa. Antes de hacerlo cierra todos los archivos abiertos y los buffers de archivo (En los programas que manejan ficheros, estos han de abrirse, manipularse y cerrarse. Los datos de estos ficheros no se manipulan directamente sino a través de unos almacenamientos temporales llamados buffers) y a continuación llama a la función _exit para cerrar el programa.

La función exit no devuelve nada pero necesita un argumento que es recogido por el programa llamador (normalmente el sistema operativo). Por convención el valor 0 indica terminación normal; valores distintos de 0 indican situaciones anormales.

En la librería stdlib.h existen dos constantes definidas para pasarlas como argumento a la función exit:

exit_success terminación normal tiene valor 0
exit_failure terminación anormal tiene valor 1

de este modo si nos olvidamos si el argumento 0 de exit significa terminación normal o anormal, utilizamos estas constantes que pueden ser más difícil de olvidarse. Su sintaxis es:

exit (exit_success); exit (exit_failure)

4.3.- Función de salida _exit.

La función exit se encuentra declarada en la biblioteca estándar (stdlib.h). La función _exit provoca la terminación inmediata del programa sin realizar el vaciado de los buffers ni cerrar los ficheros. El argumento es el mismo que el de exit.

4.4.- Función de salida abort.

La función abort se encuentra declarada en la biblioteca estándar (stdlib.h). Es similar a la función exit diferenciándose en:

- No acepta ningún argumento. Su sintaxis es abort ();
- No vacía los buffers ni cierra ningún fichero.

The logo for 'Cartagena99' features the text 'Cartagena99' in a stylized, blue, serif font. The '99' is significantly larger and more prominent than the 'Cartagena' part. The text is set against a light blue background with a subtle gradient and a soft shadow effect.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

- - -

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

4.6.- Funciones de entrada/salida.

Se llaman funciones de entrada/salida a aquéllas que transportan datos entre el programa y la entrada salida estándar. La entrada estándar, normalmente, es el teclado y la salida, normalmente, es el monitor.

En las operaciones de entrada/salida, los datos utilizados suelen pasar por buffers. Un buffer es una cantidad de memoria utilizada para meter y sacar datos.

El final de entrada/salida se suele marcar con un carácter especial llamado FinDeFichero EOF (tiene un valor -1). Algunas funciones de C, como scanf, devuelven el valor EOF cuando leen el carácter de fin de fichero.

El C trata varios dispositivos como si fuesen archivos. Haya unos cuantos predefinidos que se abren al iniciar el programa dándole unos descriptores:

Descriptor	Dispositivo por defecto
Stdin	entrada, el teclado
Stdout	salida, la pantalla
Stderr	error, la pantalla
Stdaux	auxiliar, el puerto serie
Stdprn	impresora, LPT1

4.7.- Función de entrada salida estándar printf, scanf.

Están declaradas en la biblioteca estándar stdio.h. La función printf escribe un mensaje formateado en stdout (salida estándar: pantalla). La función scanf lee datos formateados de stdin (entrada estándar: teclado). La forma general:

```
printf ( "cadena de caracteres ", lista de argumentos);  
scanf ( "cadena de caracteres ", &lista de argumentos );
```

La "cadena de caracteres" está formada por caracteres imprimibles y códigos de formato. Debe haber tantos códigos de formato como argumentos.

% c	Simple carácter	% s	Cadena de caracteres
% d	Entero decimal con signo	% u	Entero decimal sin signo
% i	Entero decimal con signo	% x	Entero hexadecimal sin signo

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

Cartagena99

Las órdenes de formato pueden tener modificadores. Estos modificadores van entre % y la letra identificadora del código.

Si el modificador es un número especifica la anchura mínima en la que se escribe ese argumento (si empieza por 0 los espacios sobrantes se rellenan con 0).

Si el modificador es un número con parte real y parte decimal indica el número de dígitos de la parte real y de la parte decimal a imprimir (si es una cadena de caracteres lo que se va a imprimir indica el número mínimo y máximo de caracteres).

Por defecto la salida se justifica a la derecha (si es negativo el modificador se justifica a la izquierda).

Programa 013 Ejemplos de modificadores

```
#include<stdio.h>

void main ( )
{
    printf(":%f:",123.456);
    printf("\n:%e:",123.456);
    printf("\n:%g:",123.456);
    printf("\n:%-2.5f:",123.456);
    printf("\n:%-5.2f:",123.456);
    printf("\n:%5.5f:",123.456);
    printf("\n:%10s:", "hola");
    printf("\n:%-10s:", "hola");
    printf("\n:%2.3s:", "hola");
    printf("\n:%x:",15);
    printf("\n:%o:",15);
    printf("\n:%05d:",15);
    printf("\n:%x:",2);
}
```

4.8.- Función de entrada salida putchar, getchar.

Están declaradas en la biblioteca estándar stdio.h. Ambas funciones devuelven, en caso de éxito, el carácter procesado (escrito o leído) y en caso de error o fin de fichero, EOF.

La función putchar escribe un carácter en la salida estándar (monitor). Necesita un argumento que es el carácter a escribir.

La función getchar lee un carácter en la entrada estándar (teclado). No recibe ningún argumento.

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, blue, serif font. The '99' is significantly larger and more prominent than the rest of the text. The logo is set against a light blue background with a white arrow pointing to the right, and a yellow and orange gradient bar at the bottom.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

4.9.- Función de entrada salida puts, gets.

Están declaradas en la biblioteca estándar stdio.h.

La función puts escribe una cadena de caracteres y un carácter de nueva línea al final de la cadena de la salida estándar (monitor). Acepta como argumento una cadena (sin formato). Si tiene éxito devuelve el último carácter escrito (siempre es '\n'). En otro caso, devuelve EOF. Esta función es lo mismo que:

```
puts("Esto es un ejemplo.");           printf("Esto es un ejemplo.\n");
```

La función gets lee una cadena de caracteres de la entrada estándar hasta que encuentra el carácter '\n'. Acepta como argumento un puntero al principio de la cadena. Devuelve dicho puntero si tiene éxito o la constante NULL si falla (NULL es una constante definida en stdio.h que tiene el valor 0. Esta constante se suele utilizar para indicar que un puntero no apunta a ningún sitio).

¡Atención! Con las funciones de lectura y escritura hay que tener la precaución de reservar memoria para que entren todos los caracteres que introducimos (si reservamos para 100 no podemos introducir más de 99 caracteres).

4.10.- Funciones de consola.

Están declaradas en la biblioteca estándar conio.h. Las siguientes funciones son parecidas a las anteriores pero trabajan directamente con la pantalla y el teclado (son más rápidas pero no son redireccionables). Además, no introducen un retorno de carro después de cada carácter de nueva línea.

cprintf	printf
cscanf	scanf
cputs	puts
cgets	gets
putch	putchar
getch	getchar
getche	

Las diferencias más notables son:

Estándar	Consola
----------	---------

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, blue, serif font. The '99' is significantly larger and more prominent than the rest of the text. The logo is set against a light blue background with a subtle gradient and a soft shadow effect.

Diferencia entre cgets y gets.

Vamos a introducir una frase , por ejemplo: “hola, que tal estas”, para ello reservamos 80 espacios mediante char frase[80].

Con cgets reserva los dos primeros elementos de memoria para: en frase[0] escribe el número de caracteres reservados, en este caso 80, en frase[1] escribe el número de caracteres realmente introducidos, en esta caso 19. A continuación escribe la frase: en el lugar de memoria frase[2] escribe “h”, en el lugar de memoria frase[3] escribe ”o”

Con gets no reserva nada y toma “retorno de carro” en el último elemento.

4.11.- Funciones rand, randomize y srand.

Están declaradas en la librería stdlib.h. La función rand devuelve un valor pseudoaleatorio, genera siempre los mismos números, entre 0 y RAND_MAX (suele ser el valor del mayor número entero (int) con signo.

Programa 014 Ejemplo de la función rand

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

void main ( )
{
    int i;
    for(i=0;i<10;i++)
        printf("\t%d",rand());
}
```

Para que realmente los números sean aleatorios, C y en C++, tenemos que usar la función randomize(), además, si queremos que los números estén en un intervalo por ejemplo, del 1 al 99, hemos de operar a rand () con % 100.

Programa 015 Ejemplo de la función randomize

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

void main ( )
{
    int i;
    randomize();
```

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, blue, serif font. The '99' is significantly larger and more prominent than the rest of the text. The logo is set against a background of a light blue and white abstract shape that resembles a stylized 'C' or a wave.

Programa 015A Ejemplo de la función srand

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

void main ()
{
    int i;
    srand(time(NULL));
    for(i=0;i<10;i++)
        printf("\t%d",rand()%100);
}
```

Programa 16 Generación de un vector y ordenamos por el método de la Burbuja

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

void main ()
{
    int vector [100];
    int i;
    srand(time(NULL));
    for(i=0;i<=99;i++)
        vector[i]=rand()%1000;
}
```

Como no sale por pantalla nada no podemos ver lo que pasa

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

void main ()
{
    int vector [100];
    int i;
    srand(time(NULL));
    for(i=0;i<=99;i++)
        vector[i]=rand()%100;
    printf("El vector desordenado es: \n");
    for(i=0;i<100;i++)
        printf("%d\t",vector[i]);
}
```

The logo for 'Cartagena99' features the text 'Cartagena99' in a stylized, blue, serif font. The '99' is significantly larger and more prominent than the rest of the text. The logo is set against a background of a light blue and white starburst or explosion shape.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

```

#include<stdio.h>
#include<stdlib.h>
#include<time.h>

void main ()
{
    int vector [100];
    int i,j,aux;
    srand(time(NULL));
    for(i=0;i<=99;i++)
        vector[i]=rand()%100;
    printf("El vector desordenado es: \n\n");
    for(i=0;i<100;i++)
        printf("%d\t",vector[i]);
    for(i=98;i>=0;i--)
        for(j=0;j<=i;j++)
            if(vector[j]>vector[j+1])
                {
                    aux=vector[j];
                    vector[j]=vector[j+1];
                    vector[j+1]=aux;
                }
    printf("\n\nEl vector ordenado es: \n\n");
    for(i=0;i<100;i++)
        printf("%d\t",vector[i]);
}

```

4.12.- Función clrscr.

En algunos compiladores al ejecutarse el programa no limpia la pantalla del programa anterior. A veces nos interesa limpiar la pantalla, borrar lo que hay en ella y así solo ver lo que nos interesa. En C++ esto se consigue con la función **clrscr** () que se encuentra en la librería **conio.h**.

En Visual C no existe dicha función. Para limpiar la pantalla llamamos al sistema mediante **system** y pedimos que lo limpie mediante el argumento **“cls”**. Para ejecutar la función **system** debemos incluir la librería de cabecera **stdlib.h**. Osea:

- 1º Incluir stdlib.h
- 2º Dentro de main ejecutamos system(“cls”);



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, dark blue font. The '99' is significantly larger and more prominent than the rest of the text. The logo is set against a light blue background with a white arrow pointing to the right, and a white arrow pointing to the left, creating a sense of motion or direction.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**