

Práctica 3 - Funciones en C

TIEMPO ASIGNADO: 6 Horas de laboratorio

OBJETIVOS:

- Que el alumno conozca los conceptos de ámbito y persistencia de variables en C y sepa aplicarlos a la práctica.
- Que el alumno conozca y sea capaz de definir funciones en C usando el paso de parámetros por valor
- Que el alumno conozca y sea capaz de utilizar variables de tipo puntero
- Que el alumno conozca y sea capaz de implementar funciones en C usando el paso de parámetros por referencia
- Que el alumno sea capaz de definir y usar macros sencillas
- Que el alumno comprenda el concepto de puntero a función y su uso.
- Que el alumno sea capaz de implementar usando el IDE Code::Blocks 12.11 pequeños programas en lenguaje C que hagan uso de distintas funciones

EJERCICIOS PROPUESTOS

1.- Realiza una función que reciba como entrada las coordenadas cartesianas de un punto del plano(x,y) y devuelva el cuadrante en el que se encuentra el punto (1, 2, 3, 4), debe devolver 0 si se encuentra en un eje.

2.- Realiza tres funciones que tengan como argumento un valor radio, y calculen el perímetro de la circunferencia, área del círculo y volumen de la esfera. Se debe implementar una función main que haga uso de estas funciones dependiendo de una opción del usuario.

3.- Realiza un programa que lea dos números enteros y en función de una opción del usuario realice la suma, resta producto o división entera de los mismos, utilizando funciones. El programa deberá ejecutarse mientras el usuario desee.

4.- Realiza un programa que lea un entero por pantalla, defina un puntero que apunte a esa variable entera y escriba:

- a) Dirección y contenido del entero
- b) Dirección y contenido del puntero
- c) Valor referenciado por el puntero

Analiza los resultados obtenidos

5.- Escribe una función que reciba dos variables enteras inicializadas e intercambie sus valores.

6.- Realiza un programa que calcule la intensidad que circula por una resistencia dados R y V. $I=V/R$ suponiendo que la función main del programa sea:

```
main()
{
float res, vol, intensidad;

obtener_valores(&res,&vol);
intensidad= calcular(res, vol);
imprimir_respuesta(intensidad);
return 0;
}
```

7.- Escribe un programa en C que escriba la suma de los n términos de la serie armónica, utilizando para ello, una función que reciba como parámetro el valor de n y devuelva la suma de la serie:

$$1 + 1/2 + 1/3 + 1/4 + 1/5 \dots 1/n$$

8.- Escribe un programa en C que calcule la suma de la serie: $x - x^3 + x^5 - \dots$ utilizando para ello, dos funciones: una función que reciba el valor de x, y el número de términos y devuelva la suma de la serie y otra función que devuelva la potencia dadas la base y el exponente.

9.- Escribe un programa en C que, , usando funciones, calcule el valor del número e como suma de la serie:

$$\sum \frac{1}{i!} = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{n!}$$

10.- Realiza un programa para calcular el coeficiente del binomio utilizando dos funciones. Una de las funciones deberá recibir los valores de n y k como parámetros y devolverá el valor del binomio. Para ello, deberá hacer uso de la función factorial que devolverá el valor del factorial a través de un puntero (sin usar la sentencia return)

$$\binom{n}{k} = \left(\frac{n!}{(k! * (n-k)!)} \right)$$

11.- Escribe un programa en C que acepte el valor de x y el número de términos y realice la suma de la serie: $1 - (x^2/2!) + (x^4/4!) - \dots$ utilizando para ello tres funciones. (Una de las funciones deberá devolver el resultado como parámetro de salida, en vez de a través de la sentencia *return*).

12.- Escribe dos versiones de una función que calcule la suma de los divisores de un número utilizando las siguientes cabeceras:

a) `void add_divisors (int num, int *add)`

b) `int add_divisors(int num)`

13.- Escribe un programa que utilice la función del apartado a) del problema 12 para calcular los números perfectos menores que 1000 (Un número es perfecto si la suma de sus divisores, excepto él mismo es igual al propio número).

14.- Escribe un programa que utilice la función del apartado b) del problema 12 para determinar si dos números enteros positivos son amigos (Dos números son amigos si la suma de los divisores del primero excepto el mismo es igual al segundo y viceversa).

15.- Escribe un programa que, utilizando funciones, convierta un número decimal en binario.

16.- Escribe un programa que, a través del uso de funciones, lea un número y lo visualice separando sus dígitos por dos espacios. Por ejemplo el entero 4562 deberá ser visualizado como 4 5 6 2.

17.- Escribe un programa para determinar si un número es capicúa. Ejemplo 123321

18.- Escribe un programa que acepte un número decimal y, usando funciones, redondee el número a un solo decimal. El programa deberá permitir al usuario repetir el proceso mientras desee. Para cada número procesado debe mostrar tanto el número original como el número redondeado.

19.- Escribe un programa en C que reciba el número de filas y escriba el triángulo siguiente, usando funciones.

Ejemplo: número de filas: 5

```
1
01
101
0101
10101
```

20.- Escribe un programa que ordene tres números por el método del intercambio usando una **macro** para realizar el intercambio de dos variables.

21.- Realiza un programa que reciba dos números, visualice un menú con la operación a realizar y en función de dicha opción del usuario realice la suma, resta, producto o división de dichos números a través de una función *opera* que recibe los dos enteros y un **puntero a función**: *int (*operación)(int, int)*

22.- Dado el siguiente programa, realiza un seguimiento indicando los valores que toman cada una de las variables después de ejecutarse cada una de las sentencias y qué es lo que muestra por pantalla. (realiza el ejercicio con papel y lápiz y posteriormente comprueba los resultados)

```
include <stdio.h>
#include <stdlib.h>
int x=2;

void a()
{
    printf("%d\n",x);
    x=x*2;
    printf("%d\n",x);
}

void b()
{
    static int x=4;
    x*=3;
    printf("%d\n",x);
}

void c(int x)
{
    x=x*2;
    printf("%d\n",x);
    b();
}

void d(int *x)
{
    *x=*x+1;
    printf("%d\n",*x);
}

int main()
{
    int x=5;
    {
        int x=7;
        printf("%d\n",x);
    }
    printf("%d\n",x);
    a();
    printf("%d\n",x);
    b();
    printf("%d\n",x);
    c(x);
    printf("%d\n",x);
    b();
    printf("%d\n",x);
    d(&x);
    printf("%d\n",x);
    system("PAUSE");
    return 0;
}
```