



Grado en Ingeniería Informática

Introducción a la Programación- Curso 15-16

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

Para lograr los objetivos descritos anteriormente a continuación se plantean algunos ejercicios que el alumno debe resolver por sí mismo haciendo uso del compilador de C.

- 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 menú también será visualizado por una función sin argumentos, que devolverá como resultado la opción elegida.

La ejecución será de la siguiente forma:

- 1.-Sumar
- 2.-Restar
- 3.-Multiplicar
- 4.-Dividir
- 5.-Salir

Seleccione la operación deseada: 3

Introducir dato1 : 2.5

Introducir dato2 : 10

Resultado 25

4.- Escribe una función que devuelva el número de cifras de un entero que se introduzca por pantalla.

5.- Realiza un programa que lea dos números 10 veces y realice en la primera lectura el producto de los números sin utilizar el operador * y en la segunda lectura la división sin usar el operador / y así sucesivamente. Finalmente debe imprimir los resultados. Implementarlo utilizando funciones (Producto_sumas, división_restas).
(Realiza la comprobación de si el número es par o impar a través de un macro).

6.- 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

7.- Escribe una función que realice el intercambio de dos variables enteras. La cabecera de la función será void swap (int *x, int *y) y la llamada swap(&x, &y)

8.- 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);
}
```

9.- Realiza un programa que reciba un número y llame a una función que calcule el factorial de ese número, posteriormente con el resultado del paso anterior, debe llamar a la función fibonacci que calcule la sucesión de números hasta ese. (realízalo de dos formas, una usando funciones que devuelvan los valores y otra simulando el paso de parámetros por referencia).

10.- Escribe una función llamada múltiplo que determine para un par de enteros, si el segundo de ellos es múltiplo del primero. La función debe tomar dos argumentos enteros y regresar 1 (verdadero) si el segundo es múltiplo del primero, y 0 (falso) de no ser así. Utilice ésta función en un programa que introduzca una serie de pares enteros.

11.- Escriba un programa que contenga una función que nos permita calcular el valor más pequeño de tres números reales

12.- Realiza un programa para calcular el coeficiente del binomio con una función factorial.

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

13.- Realiza un programa que, utilizando funciones calcule el valor del número e como suma de la serie:

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

14.- Realiza un programa utilizando funciones que dado un número entero nos dé su descomposición en factores primos. Todo número entero tiene una única descomposición dada por una multiplicación de potencias de números primos. El programa deberá encontrar estos números primos y además su potencia asociada

15.- Escribe un programa utilizando funciones que calcule los números perfectos menores que 1000 (Un número es perfecto si la suma de sus divisores, excepto él mismo es igual al propio número). Implementa la función suma_divisores utilizando la siguiente cabecera *int suma_divisores(int numero)*

16.- Escribe un programa utilizando funciones que determine 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). Implementa la función suma_divisores usando la siguiente cabecera *void suma_divisores (int numero, int *suma)*

17.- Escribe funciones que lleven a cabo lo siguiente:

- a) Calcule la parte entera del cociente cuando el entero a se divide por el entero b
- b) Calcule el residuo entero cuando el entero a es dividido por el entero b
- c) Utilice las funciones desarrolladas en a y b para describir una función que introduzca un entero entre 1 y 32.767 y lo visualice como una serie de dígitos, estando separados cada par de dígitos por dos espacios. Por ejemplo el entero 4562 deberá ser visualizado como 4 5 6 2

18.- Escribe un programa que lea varios números y que utilice una función que le permita redondear el número al entero más cercano. Para cada número procesado debe mostrar tanto el número original como el número redondeado.

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

20.- 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)*

21.- Escribe un programa, que mediante el uso de funciones, visualice el calendario correspondiente a un mes determinado dado el mes y el año. Por ejemplo el calendario correspondiente al mes de septiembre de 2004 sería:

L	M	X	J	V	S	D
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

El usuario debe indicar únicamente el mes y el año. La fórmula para calcular el día de la semana correspondiente a una fecha dada es:

a) Meses de Enero y Febrero:

$$n \leftarrow a + 31 * (m - 1) + d * (a - 1) \text{ div } 4 - 3 * ((a + 99) \text{ div } 100) \text{ div } 4;$$

b) Restantes meses:

$$(n \leftarrow a + 31 * (m - 1) + d - (4 * m + 23) \text{ div } 10 + a \text{ div } 4 - (3 * (a \text{ div } 100 + 1)) \text{ div } 4) - 1$$

donde a = año, m = mes, día = día

n mod 7 indica el día de la semana (0= domingo, 1 = lunes, 2 = martes...)

div se refiere a la división entera

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;
}
```