

Práctica de Funciones

1. Ejecuta el siguiente programa y responde:

```
main()
{
    printf("%lf",      sqrt(floor(fabs(-
16.8))));
    getch();
    return 0;
}
```

- ✓ Qué realiza cada función predefinida?
- ✓ Cuántos parámetros necesita cada una de esas funciones?

2. El siguiente programa indica si un número leído desde el teclado es par:

```
main ()
{int numero;
scanf ("%d", &numero);
if (numero%2 == 0)
```

```
printf ("Es un número par");
else
printf ("Es un número impar");}
```

- ✓ Convierte el programa en una función que reciba como parámetro el número y devuelva como resultado un valor de *tipo lógico que indique si el número es o no par*.
- ✓ Modifica la función anterior para que sirva para comprobar si el número es múltiplo de un número variable *n*, que será otro parámetro de entrada de la función.

3. El siguiente programa lee dos números enteros del teclado y calcula e imprime el mayor de ellos:

```
main()
{int a, b, maximo;

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

```
if (a> b) maximo = a;
else maximo = b;
printf ("\nEl máximo de %d y %d es
%d", a, b, maximo);}
```

- ✓ Convierte el programa en una función que reciba ambos números como parámetros de entrada y devuelva el mayor de ellos como resultado.

4. El siguiente programa calcula el factorial de un número:

```
main ()
{
int n, i;
long factorial;
printf ("\nIntroduzca un número
entero positivo: ");
}
```

```
scanf ("%d", &n);
factorial = 1;
for (i=1; i<=n; i++)
factorial = factorial * i;
printf ("\nEl factorial de %d es:
%d",n, factorial);
```

- ✓ Modifícalo transformándolo en una función que reciba el número como parámetro de entrada y devuelva el resultado como salida. Utiliza el código que se proporciona a continuación.

```
main ()
{int n;
printf ("\nIntroduzca un
número entero positivo: ");
```

```
scanf ("%d", &n);
printf ("\nEl factorial de
%d es: %ld",n,
factorial(n));}
```

5. El siguiente programa calcula la potencia de un número entero *x*.

```
main ()
{
    int n, i;
    double x, potencia;

    printf ("\nIntroduzca la
    base y el exponente: ");
```

```
scanf ("%lf%d", &x, &n);
potencia = 1;
for (i=0; i<n; i++)
    potencia = potencia * x;
printf ("\nLa potencia es
    %lf", potencia);}
```

- ✓ Modifícalo transformándolo en una función que reciba los valores de x y n como parámetro de entrada y devuelva el resultado como salida.
- 6. El siguiente programa incompleto imprime todas las tablas de multiplicar del 1 al 10.

```
main ()
{
    int i;
    for (i=1; i<=10; i++)
        {printf ("\nTabla de multiplicar del %d:\n", i);
        tabla_multiplicar (i);}}
```

- ✓ Completa el programa escribiendo la función `tabla_multiplicar` que imprime la tabla de multiplicar del número que recibe como parámetro.

- 7. Sea el siguiente programa incompleto:

```
main ()
{
    int x1, x2, suma;

    printf ("\nIntroduce dos valores: ");
    scanf ("%d%d", &x1, &x2);
    suma = suma_intervalo (x1, x2);
    printf ("\nLa suma de los valores del intervalo es: %d", suma);
}
```

- ✓ Completa el programa escribiendo la función `suma_intervalo` que calcula la suma de todos los números enteros comprendidos entre los dos parámetros de entrada, ambos inclusive.

- 8. Escribe una función que imprima la pirámide

```

      1
     121
    12321
   1234321
  123454321
```

La declaración de la función será la siguiente: `void piramide (int niveles);` siendo *niveles* el número de filas de la pirámide (5 en el ejemplo).

- 9. Escribe la función booleana `PRIMO`, emite "verdadero" si un número dado es primo.
- 10. Escribe la función booleana `MULTIPLO`, que recibe dos valores enteros y emite "verdadero" si el primero es múltiplo del segundo.
- 11. Escribe la función `FIBO`, que emite el n-ésimo término de la sucesión de Fibonacci: Sucesión de Fibonacci: 1,1,2,3,5,8,13

12. Dado el siguiente código

```
1  main()
2  {
3  int a=1, b=2;
4  mmm (___a, ___b);
5  printf ("%d,%d", a, b);
6}
```

```
7  void mmm (___ x, ___ y)
8  {9      ___x=___y+1;
10     ___y=___y+___x;
11 }
```

✓ Si el resultado impreso es: 1 y 5 (1,5) y teniendo en cuenta que algunos de los vacíos pueden no contener nada, indica cuál de las siguientes afirmaciones es cierta:

- ☐ En la línea 10 hay que colocar un total de tres *
- ☐ En la línea 7 las dos variables x e y son de tipo pint
- ☐ En la línea 9 hay que colocar un * delante de la variable x
- ☐ En la línea 4 hay que colocar un & delante de la variable b
- ☐ Hay más de una respuesta correcta

13. Desarrolla una función que emita los múltiplos de n menores que m. Tanto n como m serán dos parámetros de entrada de la función.

14. Construye una función que imprima la suma de los cuadrados de los n primeros números naturales, siendo n un valor recibido como parámetro.

15. Escribe dos funciones: Cada una de ellas debe calcular la superficie de un rectángulo en función de la base y la altura (ambos datos tipo int). La primera función llamada areal() recibe los argumentos por valor y la segunda, area2(), por referencia. Los resultado se deben imprimir en la función main()

16. Sea el siguiente código

```
void mmm (int ___);
main()
{ int a,b;
  mmm (___a);
  mmm (___b);
```

```
printf ("%d", a+b);}

void mmm (int ___ x)
{ scanf ("%d", ___x);}
```

Si se introducen desde el teclado los valores 1 y 2, el resultado en pantalla es un 3. Teniendo en cuenta que algunas de las líneas en blanco pueden no contener nada, indique cuántos símbolos & y * faltan en el código.

- ☐ Faltan dos * y dos &
- ☐ Faltan dos * y tres &
- ☐ No falta nada
- ☐ Faltan tres * y dos &
- ☐ Faltan tres & y ningún *

17. Realiza una función de nombre Siguiete tal que, recibiendo un número primo mayor que uno, devuelva el número primo inmediatamente siguiente y superior a dicho número primo. Por ejemplo, si se invoca Siguiete(7), la función devolverá el número 11.

18. Una función de nombre logaritmo recibe como parámetros un número real de doble precisión y un entero que representa la base, y calcula el logaritmo de dicho número en dicha base. Escriba los prototipos en lenguaje C de dicha función en los dos casos siguientes:

- ✓ La función devuelve el valor del logaritmo obtenido.
- ✓ El logaritmo es un parámetro de salida de la función.
- ✓ Realiza una llamada para cada prototipo, declarando las variables adecuadas.

19. Escribe una función:

```
void maxmin (int x1, int x2, int max, int min);
```

- ✓ que reciba como parámetros de entrada dos números enteros x1 y x2 y devuelva a través de los parámetros de salida max y min el máximo y el mínimo, respectivamente, de ambos números.
- ✓ Modifica la función anterior para que calcule también el valor medio de ambos números. El nuevo prototipo de la función será ahora el siguiente: void maxmin (int x1, int x2, int max, int min, int media);

20. Realiza una función que reciba como parámetros dos números enteros y los devuelva ordenados conteniendo el primer parámetro el mayor de los dos números y el segundo el menor de ellos. El prototipo de la función es el siguiente:

```
void ordena_dos_numeros (int mayor, int menor);
```

21. **La función:**

```
int valida_dato (int dato, int min, int max, int defecto)
{
    int resul=defecto;
    if (dato >= min && dato <= max)
        resul = dato;
    return (resul);
}
```

- ✓ Comprueba si el parámetro de entrada dato se encuentra comprendido entre los dos límites min y max. En caso afirmativo, devuelve el propio valor dato como resultado, mientras que en caso negativo devuelve el valor del parámetro defecto. Modifique esta función para que adopte el siguiente prototipo: void valida_dato (int dato, int min, int max, int defecto);

Es decir, que dato pase a ser un parámetro de entrada salida.

22. **La función** Logico fecha_valida (int dia, int mes, int anyo);

- ✓ Recibe como entrada una fecha (día, mes y año) y devuelve Cierto si la fecha es válida y Falso si no lo es. Modifique el prototipo de esta función para que el valor devuelto pase a ser un parámetro de salida.

23. Realiza una función que reciba como parámetros tres números enteros y los devuelva ordenados. El prototipo de la función es el siguiente: void ordena_tres_numeros (int mayor, int medio, int menor);

24. Escribe una función Logico fecha_valida (int dia, int mes, int anyo);

- ✓ Que reciba como entrada una fecha (día, mes y año) y devuelva Cierto si la fecha es válida y Falso si no lo es. Para que una fecha sea válida el valor del mes debe estar comprendido entre 1 y 12 y el valor del día debe estar de acuerdo con el mes (en el caso del mes de febrero, debe tenerse

en cuenta además si el año es bisiesto o no).

25. Escribe una función que reciba como parámetro de entrada el valor de un año y devuelva como resultado un valor de tipo lógico que indique si es o no un año bisiesto. Recuerda la regla: *"Un año es bisiesto si es divisible por 400, o bien si es divisible por 4 pero no por 100"*
26. Escribe una función que reciba como parámetros de entrada dos fechas dadas por un día, mes y año y devuelva como resultado el número de días transcurridos entre ambas. Tené en cuenta los posibles años bisiestos que puedan existir entre ellas (puedes utilizar la función realizada en el ejercicio anterior).
27. En C puedes pasar argumentos a funcione por valor y por referencia. Investiga ¿Cuándo interesa cada una de las formas?
28. Escribe el siguiente programa, compílalo y ejecútalo. ¿Por qué no funciona correctamente? Corrígelo.

```
#include <stdio.h>

float max_doble(float, float);

void main(void)
{
    float x, y;
    printf("\nDeme dos números separados por un blanco: ");
    scanf("%f%f", &x, &y);
    printf("El doble-máximo de %f y %f es %f\n", x, y, max_doble(x, y));
}

float max_doble(float a, float b) {
    float max;
    if (a>b)
        max=a;
    else
        max=b;
    max=max*2;
}
```

29. Escribe una función que reciba como parámetro de entrada un número entero y devuelva como resultado el número de cifras del número. Para ello, considera lo siguiente: dado un número cualquiera, como por ejemplo el 173, se tiene que $\rightarrow 173 / 10 = 17$ (todas las cifras salvo la última). Habrá que repetir pues este proceso hasta que el número resultante de la división tenga una sola cifra.
30. Escribe una función que reciba como parámetros de entrada un valor entero y compruebe si se encuentra comprendido entre dos valores constantes MIN y MAX. La función devolverá un valor lógico cierto en caso afirmativo, y falso en caso contrario.
31. Modifica la función anterior para que los dos límites del intervalo sean parámetros de la función. Escribe varias posibles llamadas a la función, e indica el valor devuelto en cada caso.
32. El siguiente programa da tres "warning", de los cuales uno de ellos asegura que la función cuadrado está "redeclarada". ¿Cuál es el problema? Corrígelo.

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

```

void main(void)
{
    float x;
    printf("\nDeme un número: ");
    scanf("%f",&x);
    printf("El cuadrado de %f es %f\n",x,cuadrado(x));
}
float cuadrado(float y)
{
    float z;
    z=y*y;
    return(z);
}

```

33. Escribe una función que reciba como parámetros de entrada dos instantes de tiempo expresados en horas, minutos y segundos y devuelva como resultado un 1 ó un 2 según el primer instante de tiempo sea anterior al segundo o viceversa.

34. Escribe una función que reciba como parámetros de entrada tres números enteros que representan las longitudes de tres segmentos rectilíneos, y devuelva como resultado un valor de tipo lógico que indique si dichos segmentos pueden formar o no un triángulo (la condición necesaria pero no suficiente es que ninguno de los segmentos tenga una longitud superior a la suma de los otros dos).

35. Escribe un programa teniendo en cuenta las siguientes funciones:

- *lee_opcion* lee la opción deseada y comprueba su validez.
- *menú* muestra el menú en la pantalla.
- *cuadrado*, *circulo*, *rectángulo*, *trapezio*, *triángulo* calculan la superficie correspondiente.

```

=====
==CÁLCULO DE SUPERFICIES (versión 1.0)==
=====
1. Cuadrado      (lado*lado)
2. Círculo       (pi*radio*radio)
3. Rectángulo    (base*altura)
4. Trapecio      (base1+base2)*altura/2)
5. Triángulo     (base*altura)/2)
0. Salir del programa
=====

```

36. Realiza un programa que lea un número de hasta 8 cifras y emita por pantalla la cifra resultante de aplicar el siguiente proceso:

- a) Sumar el valor absoluto de todas las cifras del número.
- b) Si el valor resultante tiene más de una cifra, volver a sumar todas sus cifras sucesivamente hasta obtener un valor de una única cifra.

Ejemplos: Valor introducido por el usuario: 68543210 → 6+8+5+4+3+2+1+0 = 29;
 29 → 2+9 = 11; 11 → 1+1 = 2: Valor a mostrar: 2

37. Construir una función que permita procesar un número desconocido de datos de productos vendidos, e ingresar por cada uno:

1. identificación del producto (número entero > 0 y menor a 1000), y por cada producto ingresado:
2. precio de costo, ej: 5.30
3. precio de venta, ej: 9.50
4. cantidad vendida, ej: 25

El ciclo finaliza con identificación del producto 0 (cero).

o Calcular y emitir:

- a) por cada producto, cuánto dinero le deja de ganancia la venta del mismo. Ej: $9.5 - 5.3 * 25 = 105$;
- b) La función main debe emitir el total de productos procesados (*no la cantidad vendida*) y las ganancias que se obtuvieron por esas ventas. Ej: se procesaron 43 productos, las ganancias obtenidas fueron \$2398.

Funciones recursivas

1. Escriba funciones recursivas para:

- ✓ Mostrar los números del 1 al N en orden creciente.
- ✓ Mostrar los números del 1 al N en orden decreciente.

2. Considere la siguiente función recursiva:

```
int misterio(int a, int b){  
    if (b==0) misterio = 0;  
    else if (b%2==0) misterio = misterio(a+a, b%2);  
    else misterio = misterio(a+a, b%2) + a;  
}
```

- ✓ Realice una traza para las siguientes llamadas:
 - o `misterio(2,25);`
 - o `misterio(3,11);`
 - o Si se reemplaza en la línea 3 por `misterio = 1` y las operaciones `+` por `*` en las líneas 4 y 5, ¿Qué función matemática queda definida?

3. Escribe funciones recursivas que retornen:

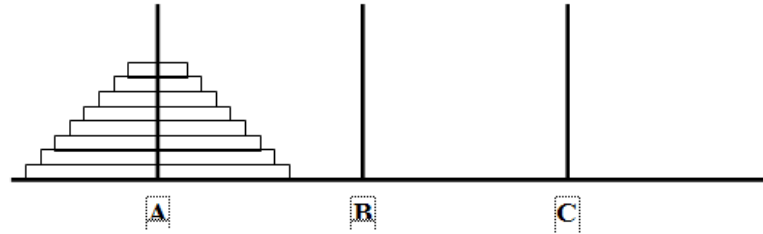
- a) El producto de dos números enteros recibidos en función de la suma.
- b) La división entera de dos números enteros recibidos por parámetro en función de la resta.
- c) El máximo común divisor entre dos números enteros.
- e) Un valor booleano indicando si un número dado por parámetro es capicúa.
- f) El n-ésimo término de la sucesión de Fibonacci.

4. Escribe una función recursiva para imprimir una media pirámide de dígitos como se muestra en la siguiente figura.

```
    21  
   321  
  4321  
 54321  
654321  
7654321  
87654321  
987654321
```

5. Escribe un programa que lea un número entero correspondiente a la cantidad de discos a usar y emita los movimientos correspondientes al juego de las Torres de Hanoi. Ejemplo: Se dan tres barras verticales y n discos de diferentes tamaños. Los discos pueden apilarse en las barras formando "torres". Los discos aparecen inicialmente en la primera barra en orden decreciente y la tarea es mover los n discos de la primer barra a la tercera de manera que queden ordenados de la misma forma inicial. Las reglas a cumplir son:

- ✓ En cada paso se mueve exactamente un disco desde una barra a otra.
- ✓ En ningún momento pueden situarse un disco encima de otro más pequeño.



Vamos a intentar resolverlo probando con 1 ó 2 discos:

1 disco

Mover el disco 1 de A a C

2 discos

Mover el disco 1 de A a B

Mover el disco 2 de A a C

Mover el disco 1 de B a C

Veamos qué pasa con 3 y 4 discos:

3 discos

Mover el disco 1 de A a C

Mover el disco 2 de A a B

Mover el disco 1 de C a B

Mover el disco 3 de A a C

Mover el disco 1 de B a A

Mover el disco 2 de B a C

Mover el disco 1 de A a C

4 discos

Podemos, partir de los movimientos para 3 discos, resolver así: pasamos

todos los discos de A a B menos el disco base usando la barra C como auxiliar. Luego movemos la base a C y volvemos a repetir los movimientos pasando los discos de la barra B a C usando A como auxiliar

Mover el disco 1 de A a B

Mover el disco 2 de A a C

Mover el disco 1 de B a C

Mover el disco 3 de A a B

Mover el disco 1 de C a A

Mover el disco 2 de C a B

Mover el disco 1 de A a B

Mover el disco a de A a C (Movemos el disco base)

Mover el disco 1 de B a C

Mover el disco 2 de B a A

Mover el disco 1 de C a A

Mover el disco 3 de B a C

Mover el disco 1 de A a B

Mover el disco 2 de A a C

Mover el disco 1 de B a C

6. Escribe una función recursiva que lea un par de enteros (analiza las restricciones) y emita el resultado de elevar el primer número a la potencia indicada por el segundo.
7. Escribe el planteo recursivo e implemente en C una función recursiva que calcule la suma de los dígitos que ocupan posiciones impares para un número natural. Se considera que la posición 1 es la posición del dígito menos significativo (lugar de la unidad), la posición 2 es la posición de la decena, etc. Por ejemplo, si se considera el natural 587, el 7 está en la posición 1, el 8 en la posición 2 y el 5 en la posición 3. En el ejemplo, la función debería retornar 12 (7+5).