
Escuela Politécnica Superior de Elche

# FUNDAMENTOS DE INFORMÁTICA
## 1º Ingeniería Industrial

---

**PRÁCTICA 3:**
**Funciones**

---

**CURSO 2007-2008**

---

**División de Ingeniería de Sistemas y Automática**

---

# Ejemplos resueltos

**EJEMPLO 1:**

Escribir una función llamada *ConvierteMayuscula* que reciba como parámetro una letra minúscula y devuelva la mayúscula correspondiente. Si el parámetro recibido no es una letra minúscula, la función debe devolver el carácter '0'.

A continuación, escribir una función *main* que vaya pidiendo al usuario letras hasta que introduzca un cero, pase estas letras a mayúscula (utilizando la función creada) y las muestre por pantalla.

```
#include <stdio.h>

char ConvierteMayuscula(char letra);

void main(void)
{
   char x, y;

   // Bucle infinito. Finalizará mediante un break cuando el usuario introduzca el caracter '0'.
   while(1)
   {
      printf("Introduzca letra (0 para acabar): ");
      fflush(stdin);  // Necesario cada vez que se lea un char
      scanf("%c", &x);
      if(x == '0')
         break;

      // Se llama a la función ConvierteMayuscula pasando como parametro el carácter
      // introducido por el usuario y se almacena el resultado en la variable y.
      y = ConvierteMayuscula(x);

      // Si la función devuelve el caracter '0', es que no se ha introducido una letra minúscula.
      if(y == '0')
         printf("Letra incorrecta\n\n");
      else
         printf("%c convertida a mayuscula es %c\n\n", x, y);
   }

   system("PAUSE");
}

char ConvierteMayuscula(char letraMin)
{
    // Si el caracter recibido no es una letra minúscula, la función finaliza devolviendo un cero.
    // Si sí que lo es, se le resta 32 a su código ASCII para calcular la máyusc. correspondiente.
    if((letraMin < 'a') || (letraMin > 'z'))
       return('0');
    else
       return(letraMin-32);
}
```

**EJEMPLO 2:**

Escribir función llamada *MediaDe3* que lea tres números enteros y calcule la media de los tres y la devuelva como resultado en formato *float*.

Escribir a continuación una función *main* desde la que se pidan tres números al usuario, se calcule su media llamando a la función anterior y por último se muestre la media con tres decimales de precisión.

```
#include <stdio.h>

float MediaDe3(int a, int b, int c);

void main(void)
{
  int x, y, z;
  float media;

  // Se piden al usuario los tres valores
  printf("Introduzca valor 1: ");
  scanf("%d", &x);
  printf("Introduzca valor 2: ");
  scanf("%d", &y);
  printf("Introduzca valor 3: ");
  scanf("%d", &z);

  // Se calcula la media llamando a la función
  media = MediaDe3(x, y, z);

  // Se muestra la media con tres decimales
  printf("La media es %.3f\n", media);

  system("PAUSE");
}

// La función MediaDe3 recibe como parámetros tres números
// enteros y devuelve su valor medio en formato float
float MediaDe3(int a, int b, int c)
{
    float result;

    // Para que el resultado sea un valor float, es necesario
    // forzar alguna de las variables en la expresión matemática
    result = (float)(a + b + c) / 3;

    return(result);
}
```

**EJEMPLO 3:**

Escribir una función llamada *LeerNumeroPositivo* que se encargue de pedir un número entero al usuario. En caso de que el número introducido sea cero o negativo debe volver a pedirlo, las veces que sean necesarias, hasta que el usuario introduzca un número positivo. Al final, la función devuelve el número positivo introducido.

Escribir a continuación una función llamada *MuestraDivisores* que reciba como parámetro un número entero y que calcule y muestre por pantalla los divisores de dicho número.

Por último, escribir una función *main* que pregunte al usuario cuantos números quiere pedir. A continuación, debe ir pidiendo números (haciendo uso de la primera función) y mostrando por pantalla sus divisores (haciendo uso de la segunda función) hasta que la cantidad de números introducida alcance el número deseado por el usuario.

```
#include <stdio.h>

int LeerNumeroPositivo(void);
void MostrarDivisores(int a);

void main(void)
{
  int n, x, i;

  // Se pide al usuario que introduzca la cantidad de numeros
  printf("Cuantos numeros? ");
  scanf("%d", &n);

  // Se crea un bucle que vaya pidiendo los números y mostrando los divisores.
  for(i=0; i<n; i++)
  {
     x = LeerNumeroPositivo();
     MostrarDivisores(x);
  }

  system("PAUSE");
}

// La función LeerNumeroPositivo no recibe ningún parámetro.Se encarga de pedir un número
// al usuario, comprobar si es positivo y devolverlo como resultado. Si el número es
// negativo, lo vuelve a pedir al usuario
int LeerNumeroPositivo(void)
{
    int num;

    printf("Introduzca valor: ");
    scanf("%d", &num);
    while(num <= 0)
    {
       printf("Valor incorrecto\n");
       printf("Introduzca valor: ");
       scanf("%d", &num);
    }
    return(num);
}
```

```
// La función MostrarDivisores recibe un número entero y se encarga de mostrar por pantalla los
// divisores del mismo. No devuelve ningún resultado.
void MostrarDivisores(int a)
{
    int i;
    printf("Los divisores de %d son: ", a);
    for(i=1; i<=a; i++)
    {
        if(a%i == 0)
        {
            printf("%d, ", i);
        }
    }

    // La siguiente línea sirve para borrar la coma que aparece en pantalla tras el último divisor
    // (se hace retroceder 2 veces el cursor para borrar el ultimo espacio y la coma y se escribe
    // un espacio en blanco encima de la coma), e inserta un salto de linea.
    printf("\b\b \n");
    return;
}
```

**EJEMPLO 4:**

Escribir un programa que pida un número entero y muestre por pantalla todos los números primos menores o iguales a dicho número. Para ello, se debe programar una función llamada *EsPrimo*, que reciba como parámetro un número entero y que devuelva 0 si dicho número no es primo y 1 si sí que lo es.

```
#include <stdio.h>

short EsPrimo(int x);

void main(void)
{
  int n, i;
  short primo;

  // Se pide al usuario que introduzca un número.
  printf("Introduzca numero: ");
  scanf("%d", &n);

  printf("Los numeros primos menores o iguales a %d son:\n", n);

  // El siguiente bucle saca por pantalla los números primos menores o iguales a n.
  // Para comprobar si estos números son primos, se usa la función EsPrimo.
  for(i=1; i<=n; i++)
  {
     primo = EsPrimo(i);
     if(primo == 1)
        printf("%d, ", i);
  }
  printf("\b\b \n");

  system("PAUSE");
}

// La función EsPrimo comprueba si el número que se le pasa como parámetro es primo.
// Si sí que lo es, devuelve resultado 1, y si no lo es, devuelve 0.
short EsPrimo(int x)
{
   int i;

   for(i=2; i<x; i++)
   {
      if(x % i == 0)
      {
         // Si x es divisible por algún número comprendido entre 2 y x-1, entonces x no es primo.
         return(0);
      }
   }
   // Si la función ha llegado a este punto es porque no ha existido ningún
   // divisor, por tanto el número es primo.
   return(1);
}
```

## Ejercicios Propuestos.

**EJERCICIO 1:** Escribir una función en C llamada *CalculaSueldo*, que calcule la nómina de un trabajador. La función recibe dos parámetros enteros llamados *ord* (número de horas ordinarias trabajadas) y *extra* (número de horas extra trabajadas). Con esta información, la función debe devolver un entero igual a la nómina del trabajador teniendo en cuenta las siguientes consideraciones:

1.- Las 30 primeras horas ordinarias se pagan a 10 €/hora
2.- Las siguientes 10 horas ordinarias se pagan a 15 €/hora
3.- Las restantes horas ordinarias se pagan a 20 €/hora
4.- Las horas extra se pagan a 30 €/hora
5.- Sobre el sueldo resultante se aplican los siguientes impuestos:
    5.1.- Hasta los primeros 500 €. un 10%
    5.2.- Al resto del sueldo un 15%

**EJERCICIO 2:** Escribir una función llamada *NumDigitos*, que tome como parámetro un número entero llamado *n* y devuelva como resultado otro entero tipo short igual al número de dígitos de *n*.

**EJERCICIO 3:** Se puede calcular el seno de x en puntos cercanos a 0 de forma aproximada, sumando los n primeros términos de la serie infinita siguiente:

$$senx = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \ldots$$

Se pide escribir una función llamada *CalculaSeno* que tome como parámetros el valor de *x* (double) y de *n* (entero) y devuelva como resultado la aproximación del seno de *x*, siendo *n* el exponente del numerador del último término que se debe sumar. El resultado se debe devolver en formato double.

Nota. Además de la función *CalculaSeno*, se debe escribir una función llamada *CalculaPotencia*, que reciba un double *x* y un float *n* (base y exponente) y devuelva el resultado $x^y$, en formato double y otra función llamada *CalculaFactorial*, que reciba un entero (*n*) y devuelva n! en formato long. La función *CalculaSeno* debe hacer uso de estas dos funciones para calcular las potencias y factoriales que le sean necesarios.

**EJERCICIO 4:** Escribir una función llamada *ConvierteBase10* que reciba como parámetro un número *n* (long int) expresado en base 2 y devuelva su correspondiente en base 10. El resultado debe ser un entero. Si el número introducido es incorrecto (contiene algún dígito diferente de 0 o 1), la función debe devolver un valor -1 (ese valor indicará que se ha producido un error).

**EJERCICIO 5:** Escribir una función llamada *CalculaArea* que permita calcular el área bajo una curva de la forma $a \cdot x^2 + b \cdot x + c$ utilizando la regla del trapecio. El programa debe tomar como parámetros las cantidades a, b y c (tipo float), el número de puntos que se toma para calcular el área, n (tipo int), y los dos puntos entre los cuales se calcula el área x1, x2 (tipo float). El resultado debe estar en formato double.

**EJERCICIO 6**: Escribir un programa que imprima el siguiente menú:

```
1.- Calcular sueldo
2.- Calcular numero de digitos
3.- Calcular seno
4.- Convertir a base 10
5.- Calcular area bajo una curva
6.- Salir
Elegir opcion:
```

de tal forma que el usuario introduzca una opción (1 a 6) y se ejecute la función correspondiente de los ejercicios anteriores (1, 2, 3, 4, 5). La opción 6 se utilizará para salir del programa. Se debe utilizar una sentencia *switch*.

- Si el usuario selecciona la opción 1, en primer lugar debe pedirse al usuario que introduzca número de horas ordinarias y extraordinarias, leerlas y llamar a la función *CalculaSueldo* con estos datos. Por último, se debe sacar por pantalla el resultado de la función.

- Si el usuario selecciona la opción 2, en primer lugar debe pedirse al usuario que introduzca un número entero, debe leerse y a continuación debe llamarse a la función *NumDigitos*. Por último, se debe sacar por pantalla el resultado devuelto por la función.

- Si el usuario selecciona la opción 3, primero deben pedirse al usuario los parámetros de la misma (cantidad de la que quiere calcularse el seno y exponente máximo de la aproximación). Después de la llamada a la función se imprimirá la aproximación de sen(x), que es el valor devuelto por la función.

- Si el usuario selecciona la opción 4, debe pedirse primero un número al usuario y a continuación debe llamarse a la función *ConvierteBase10* pasando como parámetro ese número. Por ultimo debe mostrarse el resultado de la función. Si el usuario ha introducido un número incorrecto, debe mostrarse un mensaje de error.

- Si el usuario selecciona la opción 5, deben pedirse primero todos los datos necesarios (coeficientes de la función, punto inicial y final del intervalo y número de puntos). A continuación, debe llamarse a la función *CalculaArea*. Por ultimo debe mostrarse el resultado de esta función.

Una vez realizados los cálculos correspondientes y mostrados los resultados, debe volver a aparecer el menú inicial y se debe repetir esta secuencia hasta que se introduzca la opción 6.