



# PRÁCTICAS DE FUNDAMENTOS DE PROGRAMACIÓN

**Koldo Espinosa**  
**Departamento de Ingeniería de Comunicaciones – UPV/EHU**  
**Escuela Técnica Superior de Ingeniería de Bilbao**

## **PRÁCTICAS DE FUNDAMENTOS DE PROGRAMACIÓN**

**© Copyright 1989-2014 Koldo Espinosa**

**Los enunciados de los ejercicios de programación de este documento proceden de una colección de enunciados creada en el año 1989.**

**Grado en Ingeniería Técnica de Telecomunicación**

**2º curso**

**Fundamentos de Programación**

**Año académico 2014/2015**

## **ASIGNACIÓN Y OPERACIONES BÁSICAS**

1. Escribir un programa que permita pasar temperaturas de grados Celsius a Fahrenheit, utilizando la fórmula:

$$F = 1.8 * C + 32$$

donde F es la temperatura en grados Fahrenheit, y C es la temperatura en grados Celsius.

Visualizar el resultado con el siguiente formato:

xxx.xx grados Celsius = yyy.yy grados Fahrenheit

2. Crear un programa que calcule el área de un círculo, un triángulo y un cuadrado, pidiendo previamente los datos necesarios para ello.
3. Escribir un programa que lea de teclado un número entero de tres cifras y escriba en pantalla el número formado por las cifras en orden inverso. (P. ej.: Entrada 123; Salida 321).
4. Escribir un programa que calcule el número de billetes de 5, 10, 20, 50, 100, 200 y 500 euros, y de monedas de 1, 2, 5, 10, 20 y 50 céntimos y 1 y 2 euros necesarios para desglosar una cantidad C de euros, de forma que posea la menor cantidad de monedas y billetes de cada tipo.
5. Leer del teclado tres números que correspondan al día, mes y año. Presentar la fecha en pantalla con el siguiente formato:

dd/mm/aa

## **SENTENCIAS SELECTIVAS**

1. Realizar un programa que imprima en pantalla en orden creciente 3 números enteros introducidos por teclado.
  
2. Un comercio realiza un descuento a sus clientes en función del gasto realizado en la compra, según la siguiente tabla:

GASTO	TIPO DE DESCUENTO	DESCUENTO (%)
$0 < G \leq 6.000$	NULO	0 %
$6.000 < G \leq 10.000$	BAJO	10 %
$10.000 < G \leq 20.000$	MEDIO	20 %
$G > 20.000$	ALTO	30 %

Realizar un programa que tras leer el importe del gasto, proporcione información sobre: la cantidad gastada, el tipo de descuento que le corresponde, el descuento numérico y la cantidad a pagar (con dos decimales de precisión).

## SENTENCIAS REPETITIVAS

1. Calcular el factorial de un número entero positivo leído desde el teclado.
  
2. Realizar un programa que permita simular el "Juego del Número". Dicho juego consistirá en lo siguiente:

El programa tendrá un número "secreto" de dos dígitos, fijado por el programador que el jugador (usuario que ejecuta el programa) deberá acertar. El jugador tendrá como máximo 10 oportunidades para acertar el número, y el programa informará continuamente del número de intentos que le quedan. Si el número es acertado el programa felicitará al usuario y le dará una nota con arreglo al número de intentos que ha necesitado. Si el número no es acertado en la décima oportunidad el programa terminará recomendándole que vuelva a jugar.

3. Realizar un programa para calcular la raíz cuadrada de un número N por el método de Newton que dice:

$$r_{i+1} = \frac{\frac{N}{r_i} + r_i}{2}$$

Continuar el proceso hasta que se verifique:

$$|r_i - r_{i+1}| \leq Epsilon$$

- a) Datos iniciales leídos desde el teclado: el número N, la precisión épsilon y  $r_0$  (aproximación de la raíz cuadrada de N).
- b) Calcular  $r_{i+1}$  y verificar si se cumple la condición de terminación.
- c) Si no se cumple, repetir b).
- d) Si se cumple, escribir la raíz.
  
4. Escribir un programa que lea del teclado una secuencia de caracteres y que diga cuántos blancos, cuántos dígitos y cuantas letras se han leído. El final de la entrada será un carácter fijado previamente.
  
5. Escribir un programa que calcule, para un número entero (**numero**) leído por teclado:

$$\sum_{i=1}^{numero} i^i$$

6. Escribir un programa que imprima la tabla de caracteres ASCII a partir del carácter cuyo código ASCII es 33. La salida por pantalla deberá ser de la siguiente manera:

En cada línea imprimirá 8 caracteres, y a la derecha su correspondiente número del código ASCII (en total 28 líneas).

Columna	7	10	17	20	27	30
línea 1	<b>33</b>	!	<b>34</b>	"	<b>35</b>	#
línea 2	<b>41</b>	)	<b>42</b>	*	<b>43</b>	+

7. Escribir un programa que lea una sucesión de caracteres y blancos terminada en punto, e imprima la sucesión obtenida eliminando los blancos y sustituyendo cada letra por la que le sigue a una distancia d (número comprendido entre 0 y 26), considerando circular el alfabeto y por tanto el carácter 'A' como siguiente al 'Z'.

## FUNCIONES

1. Se deja caer una pelota desde una altura H. La capacidad de rebote de la pelota se puede describir por medio de un número R, menor que 1.0, denominado factor de rebote. En cada rebote la pelota alcanza una altura igual a R veces su altura anterior.

El programa pedirá la altura inicial H y el factor de rebote de la pelota R. Luego, pedirá una altura de rebote (mayor que 0), y calculará el número de orden del primer rebote en el cual la altura alcanzada es igual o inferior a la leída, visualizando también la altura alcanzada en ese rebote.

**NOTA:** Se deberá utilizar una función que calcule la altura alcanzada en un rebote, conocidos los valores: altura inicial, factor de rebote y número de orden del rebote.

2. Realizar un programa que calcule la fracción irreducible equivalente a una fracción introducida por teclado. El programa leerá desde el teclado el numerador y denominador de la fracción (números enteros positivos), y deberá escribir el resultado en forma de fracción, tal como se muestra en el ejemplo siguiente:

$$\frac{330}{126} = \frac{55}{21}$$

**NOTA:** Se deberá utilizar un subprograma que calcule el MÁximo Común Divisor de dos números, NUM1 y NUM2, utilizando el siguiente algoritmo iterativo:

- 1.- Si los números X e Y son iguales ir al punto 4.
- 2.- Al mayor de ellos restarle el menor.
- 3.- Volver al punto 1.
- 4.- El MÁximo Común Divisor es el valor X = Y.

3. El desarrollo en serie de Taylor de la función coseno es

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

donde el ángulo x viene expresado en radianes. Escribir un programa que calcule el valor aproximado de cos(x), utilizando para ello los N primeros términos de la serie; el número de términos N dependerá del valor del último, es decir, se añadirán términos a la serie hasta que el valor absoluto de un término sea menor que 0.0005.

Para escribir el programa se utilizará un subprograma que permita evaluar potencias y otro que calcule el factorial de un número.

4. Un centro numérico es un número que separa una lista de números enteros (comenzando desde el 1) en dos grupos de números cuyas sumas son iguales.

El primer centro numérico es el 6, que separa la lista (1, 2, 3, 4, 5) y (7, 8) cuyas sumas son ambas iguales a 15. El segundo centro numérico es el 35, y separa una lista desde 1 a 49 en dos (1 a 34) y (36 a 49). La suma de los números de cada grupo es 595.

Escribir un programa que halle los centros numéricos entre 1 y N, donde N es un número, menor de 7000, que se introducirá por teclado.

Utilizar para ello:

- Una función que compruebe y devuelva si un número es centro numérico o no.
- Una función que devuelva la suma de los números de cada grupo.

**NOTA:** Centros numéricos menores de 7000: 6, 35, 204, 1189, 6930.

## ARRAYS UNIDIMENSIONALES

1. Escribir un programa que lea por teclado en un array lineal una sucesión de caracteres e imprima el carácter alfabético que aparezca con más frecuencia en la sucesión.

El programa deberá contemplar el caso en el cual dos o más caracteres aparezcan con la máxima frecuencia, e imprima todos.

2. Escribir un programa que lea N datos dentro de cada uno de los arrays X, Y de tamaño 10, y compare cada uno de los elementos de X con el elemento correspondiente de Y.

En el elemento correspondiente de otro array Z deberá almacenar:

1	si	$x > y$
0	si	$x = y$
-1	si	$x < y$

Después imprimirá una tabla que contenga el contenido de los arrays X,Y,Z con un contador del número de elementos de X que sean menores que los de Y, y otro para los que sean mayores.

3. Un banco local ha instituido un nuevo sistema de créditos y le gustaría tener unos números para que las cuentas fueran autocomprobadas, de modo que existiera alguna protección frente al fraude y los errores de introducción de datos. Han pensado en el siguiente sistema: los números de las cuentas tendrán 9 dígitos y el noveno (el de más a la derecha) puede calcularse sumando, restando o multiplicando todos los dígitos que le preceden, y cogiendo el dígito más a la derecha del resultado.

El trabajo consiste en escribir un programa que compruebe la validez de los números de las cuentas enviadas al banco como parte de sus transacciones diarias.

Entrada:

1. Un dígito correspondiente a una operación.
2. Un número de cuenta de nueve dígitos.

En el código de operación 1 es suma, 2 es resta y 3 es multiplicación. También es preciso controlar la corrección del código de operación.

Salida:

1. Imprimir el eco de la entrada.
2. Bajo cada línea del eco, poner si el número de la cuenta es válido o no o si el código de operación es incorrecto.

Ejemplo:

Entrada 1 123456786

1 representa la suma por lo que  $1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36$ .  
 6 es el dígito más a la derecha de 36, por lo tanto la cuenta es válida.

4. Realizar un programa que vaya leyendo desde el teclado números enteros hasta que se diga que no se desea introducir más números o hasta que se hayan introducido 50.

El programa deberá calcular:

- a) La media aritmética:

$$\bar{x} = \frac{\sum x}{N}$$

- b) La moda: El valor que se presenta con mayor frecuencia.

- c) La desviación típica:

$$\sigma = \sqrt{\frac{\sum (\bar{x} - x)^2}{N}}$$

## ARRAYS BIDIMENSIONALES

1. Se llama "**punto de silla**" de una matriz de dos dimensiones, al elemento de la matriz que sea el mínimo de la fila a la que pertenece y el máximo de la columna a la que pertenece (o a la inversa). Escribir un programa que lea los elementos (enteros y todos distintos entre sí) de una matriz de dos dimensiones y determine la posición del punto de silla, si existe.
  
2. Realizar un programa que tome un array que contenga la representación digitalizada de un dibujo del cielo por la noche y localice en él las estrellas.

Cada elemento del array representa la cantidad de luz que hay en esa porción de la imagen al tomar el dibujo. El rango de intensidad va de 0 a 20, por eso debe ser rechazada la introducción de intensidades fuera de dicho rango.

Una estrella está en el área cubierta por el elemento  $i, j$  del array si se da el siguiente caso:

$(\text{Suma de la intensidad de dicho elemento más las intensidades de los } 8 \text{ elementos circundantes}) / 9 > 6.0$

Se deben ignorar las posibles estrellas que estén en las aristas del array, es decir, no se calcula el valor de la expresión anterior para los puntos de la imagen que se encuentran en los bordes, ya que dichos puntos no tienen 8 elementos circundantes.

La salida será un plano de las estrellas. La presencia de una estrella en una posición se indicará por un blanco seguido de un asterisco y otro blanco. La no presencia de una estrella en una posición se indicará por un blanco seguido de un punto y otro blanco.

### Especificaciones:

El programa principal leerá las dimensiones del array de entrada. Habrá que utilizar como mínimo las siguientes funciones:

- Función de lectura de un array bidimensional.
- Función que guarde en un array bidimensional el plano de estrellas y lo imprima en pantalla de la siguiente manera:

```
* . . . .
. . * . . .
. . . * . .
. . . . * .
. . . . * .
. . . . . .
```

## **CADENAS DE CARACTERES**

1. Escribir un programa que lea una línea de texto que contenga espacios adicionales, al principio, entre palabras y al final de la frase. El programa deberá eliminar todos los espacios adicionales y contar las palabras que tiene la frase. Para ello se utilizarán los siguientes subprogramas:

<b>iniciales</b>	Recibe una cadena de caracteres y devuelve la misma cadena sin los espacios del principio de la frase.
<b>finales</b>	Idem con los espacios del final de la frase.
<b>intermedios</b>	Idem con los espacios intermedios adicionales, dejando un único espacio entre palabras.
<b>cuenta_palabras</b>	Contará las palabras que tiene la frase.

2. Escribir un programa que lea una frase de texto y nos diga si se trata de un palíndromo o no. Se deberá utilizar para ello un subprograma que elimine del string todos aquellos caracteres que no sean letras y otro que pase todos los caracteres a mayúsculas. Finalmente, mediante un subprograma comprobar si la frase es palíndromo.

**NOTA:** Palíndromo es una frase cuyo conjunto de letras es capicúa.

3. Realizar un programa que permita calcular el N.I.F., conocido el D.N.I. de una persona.

El programa deberá leer, sobre un string, el número del D.N.I. del interesado. Seguidamente deberá averiguar si es un valor válido (todos los valores numéricos y representando un valor entre 100000 y 99999999). Si la entrada es válida se deberá calcular el N.I.F. y sobre el string inicial se deberán insertar los puntos de los millares y millones en las posiciones correspondientes, así como la letra del N.I.F., al final del string separada por un espacio en blanco.

### **Cálculo de la letra del N.I.F.:**

Se obtiene el resto de la división del número del D.N.I. entre 23, y en función del resultado se asigna un carácter según la siguiente tabla:

0 = 'T'	7 = 'F'	14 = 'Z'	21 = 'K'
1 = 'R'	8 = 'P'	15 = 'S'	22 = 'E'
2 = 'W'	9 = 'D'	16 = 'Q'	
3 = 'A'	10 = 'X'	17 = 'V'	
4 = 'G'	11 = 'B'	18 = 'H'	
5 = 'M'	12 = 'N'	19 = 'L'	
6 = 'Y'	13 = 'J'	20 = 'C'	

## ESTRUCTURAS

1. Se quiere llevar el control del alquiler de los stands de la Feria de Muestras. Para ello se utilizará un array unidimensional en el que se almacenará la siguiente información relativa a los stands alquilados por las empresas:

**nombre** de la empresa  
**numero** de pabellón  
**dias** de alquiler  
**metros** cuadrados del stand  
**precio** total del alquiler

La información de este array estará almacenada por orden alfabético del nombre de las empresas.

Se utilizará otro array unidimensional en el que se almacenará la siguiente información relativa a cada uno de los CINCO pabellones de la Feria:

**tarifa** ( $\text{€}/\text{m}^2$  día)  
**metros\_alquilados** de cada pabellón

Inicialmente, el programa preguntará por las tarifas de cada uno de los pabellones.

Posteriormente, el programa continuará su ejecución bajo el control del siguiente menú de opciones:

- 1.- Añadir un stand
- 2.- Eliminar un stand
- 3.- Modificar la tarifa de un pabellón
- 4.- Visualizar la información de los stands y de los pabellones
- 5.- Salir del programa

### Opción 1

Pedirá la información de un nuevo stand (nombre, número de pabellón, días, metros). A partir de estos datos, el programa calculará el precio total del alquiler, visualizándolo y añadiéndolo al array de stands.

En ese momento habrá que actualizar los metros cuadrados alquilados del pabellón correspondiente.

### Opción 2

Pedirá el nombre de la empresa, y eliminará el stand.

En ese momento habrá que actualizar los metros cuadrados alquilados del pabellón correspondiente.

### Opción 3

Preguntará por el número de pabellón y su nueva tarifa. Tras ello, se actualizará dicha tarifa así como el precio total de alquiler de todos los stands que se encuentran en ese pabellón.

#### Opción 4

Visualizará la información de todos los stands de la Feria y de cada uno de los CINCO pabellones.

#### Opción 5

Terminará la ejecución del programa.

**NOTA:** Puede darse el caso de que el array de stands no contenga ningún elemento (por ejemplo, al principio). En dicho caso, si elegimos las opciones 2 ó 4 nos avisará de que no hay ningún stand alquilado.

## **FICHEROS BINARIOS DE DATOS**

1. Dado un fichero conteniendo el censo de la población bilbaína, se pide escribir un programa que cree un fichero que contenga la información referente a las personas con más de 60 años, o que los cumplan dentro del año actual (dicha información no deberá borrarse del fichero original).

El programa deberá además calcular el porcentaje que suponen estas personas respecto del total del censo.

El tipo de elementos que contiene el fichero del censo es, por cada ciudadano, un registro con los siguientes datos: nombre, dos apellidos, año de nacimiento y dirección.

2. Debido a un fallo en el procesamiento de la información sobre los trabajadores de una factoría, se descubre que la información sobre dichos trabajadores ha quedado desdoblada en dos ficheros diferentes.

La información que se tenía sobre los trabajadores consistía en:

```
nombre
apellidos
domicilio
fecha_ingreso_empresa
numero_seg_social
numero_cta_corriente
```

Los datos sobre el nombre, apellidos, domicilio y la fecha de ingreso en la empresa se encuentra en un fichero de nombre PERSONAL, y el resto en otro de nombre CUENTAS.

Afortunadamente el orden de la información no ha sido alterado de forma que la información del primer registro del fichero PERSONAL se completa con la información que se encuentra en el primer registro del fichero CUENTAS, y así sucesivamente.

Se pide hacer un programa que recomponga la información para poder disponer de toda ella en un único fichero.

3. Realizar un programa que lleve el control de las salidas de vuelos de un aeropuerto con la siguiente información almacenada en un fichero:

<b>numero_vuelo :</b>	entero
<b>destino :</b>	cadena de caracteres
<b>horario_salida :</b>	cadena de caracteres (Para la hora introduciremos un cadena de 5 caracteres con el siguiente formato hh:mm)
<b>numero_plazas_libres :</b>	entero

El fichero estará ordenado en todo momento según el campo **numero\_vuelo**, el cual será calculado según el horario de salida de los vuelos de ese día.

El programa se ejecutará en torno al siguiente menú:

- 1.- Información de vuelos
- 2.- Añadir un nuevo vuelo
- 3.- Cancelar un vuelo
- 4.- Modificar horario de salida de un vuelo
- 5.- Actualizar el número de plazas libres de un vuelo
- 0.- Salir del programa

### Opción 1

Visualizará de la siguiente manera la información de todos los vuelos almacenados:

Nº VUELO	DESTINO	HORA DE SALIDA	Nº PLAZAS LIBRES
1	Madrid	9 : 30	15
2	Londres	11 : 15	25

### Opción 2

Nos preguntará el destino, el horario de salida y el número de plazas libres del vuelo a añadir.

Si tenemos 5 vuelos y queremos añadir un vuelo cuyo número de vuelo sea el 3, entonces el vuelo 3 que ya existía pasará a ser el 4, el 4 pasará a ser el 5, y el 5 pasará a ser el 6.

### Opción 3

Nos preguntará el número de vuelo a cancelar. Si no existe un vuelo con el número de vuelo indicado, se visualizará un mensaje que lo indique.

Si tenemos 5 vuelos y queremos cancelar el vuelo 3, entonces el vuelo 4 pasará a ser el 3, y el vuelo 5 pasará a ser el 4.

### Opción 4 y Opción 5

Nos preguntará el número de vuelo y modificará el campo correspondiente del vuelo especificado.

**NOTA:** El fichero debe estar actualizado en todo momento, es decir, tras la ejecución de cada opción del menú el fichero debe quedar actualizado.