

## SISTEMAS DE GESTIÓN EMPRESARIAL

Python\_3

### Ejercicios propuestos

- 1.** Escribe un código que implemente el siguiente comportamiento: "Si la compra es superior a 100EUR se aplica un descuento del 5% si se paga al contado, pero si el pago es con tarjeta sólo se aplica el 2%". Asegúrate de que el importe de la compra es un número válido antes de proceder a los cálculos (pista: usa try para comprobar que es posible convertir la entrada a un float).
- 2.** Una universidad acaba de modificar su sistema de representación de la calificación de los alumnos, que como es sabido son valores entre 0 y 10. A partir de ahora, se calificarán como "A" las notas mayores o iguales a 8,5, "B" las mayores o iguales a 6,5, "C" las calificaciones mayores o iguales a 5, "D" las calificaciones mayores o iguales a 3,5, y "F" todas las demás. Programa una función que reciba una calificación numérica y retorne la letra con la nueva calificación. Asegúrate de que la calificación introducida es válida (idea: programa una función lo suficientemente genérica que se pueda luego reutilizar en programas que necesiten una validación similar).
- 3.** Escribe un algoritmo que tras leer tres enteros los escriba en pantalla en orden creciente. Como siempre, valida las entradas.
- 4.** Codifica un subprograma que reciba un número entero, y si es entre 1 y 12 escriba un mensaje por pantalla indicando el mes a que dicho número corresponde. En caso contrario deberá mostrar un mensaje de error.
- 5.** Escribe un programa que, después de preguntar cuántos números se van a introducir, pida esos números (enteros o reales) y devuelva su media aritmética, el mayor y el menor. El programa debe controlar que la cantidad de números es mayor de 2 y en caso contrario ha de mostrar un mensaje de error.
- 6.** Escribir una función que sume dos listas de igual longitud y retorne otra lista que contenga la suma de las originales elemento a elemento.
- 7.** Modifique el ejercicio anterior permitiendo que las listas sean desiguales. Los elementos sobrantes de la lista más larga se añadirán al final de la lista resultante.
- 8.** Distribuir los 20 datos enteros en dos listas de tal manera que se vayan generando dos secuencias ordenadas, una creciente y otra decreciente.
- 9.** Crear una lista de enteros, inicializarlos según valores aleatorios en el rango 1.20 y computar la media de los valores, el valor más alto y el más bajo (todo ello utilizando listas).
- 10.** Implementar una función que compruebe si una palabra es un palíndromo.
- 11.** Implementar una función que pone en mayúsculas todas las primeras letras de las palabras de una frase.
- 12.** Crear una función que compruebe si dos cadenas de caracteres son iguales.
- 13.** Implemente una función que indique si una palabra contiene las cinco vocales: por ejemplo "murciélagos".

**14.** Escriba un programa que “codifique” una frase modificando todas las vocales según el siguiente código: a por 4, e por 3, i por 1, o por 0 y u por el símbolo #. Por ejemplo, la frase: “Un perro del hortelano”, deberá devolverse: “#n p3rr0 d3l h0rt3l4n0”.

**15.** Realizar un programa que lea palabras hasta que se introduzca “fin”, mostrando una estadística de las longitudes de las palabras, es decir, el número total de palabras de longitud 1 que se hayan introducido, el total de longitud 2, etc. La máxima longitud de las palabras deberá ser de 25 caracteres. Una posible salida de este programa sería:

```
Palabras longitud 1: 0  
Palabras longitud 2: 10  
...  
Palabras longitud 25: 1
```

**16.** Implementa las estructuras de datos que permitan almacenar información anónima sobre personas con objeto de hacer un estudio estadístico. Así, se deberá almacenar el número de secuencia, sexo y edad de cada persona. Programe además un par de funciones para

- leer por teclado datos relativos a una persona
- para mostrar dichos datos por pantalla.

**17.** Haciendo uso de la estructura Fecha de los ejercicios resueltos, implementa una estructura Cronología que permita llevar la cuenta de 3 fechas importantes en la vida de toda persona: día de su nacimiento, fecha en que celebró su matrimonio y fecha de deceso.

**18.** Reutilizando lo programado en el **ejercicio 17**, implementa funciones para leer las fechas importantes de una persona y comprobar si son o no correctas (el orden ha de ser nacimiento, matrimonio, deceso, obligatoriamente). Si no lo fueran debería indicarse y solicitarse de nuevo.

**19.** Implemente una aplicación que solicite 10 valores y los almacene en un array. A continuación, recorra el array restándole a cada valor el valor que se encuentre en la siguiente posición (ej.: [1 3 5], resultado [(1-3) (3-5) (5-1)] = -2 -2 4). Modularice su solución.

**20.** Implemente una aplicación para ayudar en la gestión de cobros de una gasolinera. Mediante 3 arrays deberá calcular el gasto de un total de 10 clientes. El primer array será utilizado para almacenar el gasto de cada cliente en gasolina, el segundo array será utilizado para almacenar el gasto en productos de la tienda de la gasolinera, el tercer array almacenará la suma de los dos arrays anteriores. La aplicación:

- Solicitará por teclado los gastos de gasolina y de la tienda para cada uno de los 10 clientes.
- Mostrará por pantalla la suma de los gastos para cada cliente.

**21.** Crear un diccionario “agenda telefónica” donde la clave sea el nombre de una persona y el valor sea el teléfono. Programar una función para llenarlo y otra para mostrarlo. Para llenarlo, es necesario ir pidiendo contactos hasta el usuario diga que no quiere insertar más. Obviamente, no es válido introducir nombres repetidos.

**22.** Programar una función que reciba un diccionario y una lista y que como salida genere dos listas:

- una lista con todos los valores de aquellos elementos de la lista que están en el diccionario,
- otra lista con los valores de los elementos que NO están en el diccionario.