

# Fundamentos de Programación

## Cuaderno de Trabajo 4

Grado en Sistemas de la Información, turno de mañana

Grado en Ingeniería Informática, turnos de mañana y tarde

### Ejercicios Resueltos

1. Escriba un programa en Python que compruebe si una lista de enteros está ordenada ascendentemente.

#### SOLUCION PROPUESTA:

```
def en_orden_ascendente (lista):
    """lista-->boolean
    OBJ: Devuelve True si la lista está ordenada ascendentemente"""
    ordenado=True
    i = 1
    ant = lista [0]
    while i < len(lista) and ordenado:
        if lista [i] < ant:
            ordenado = False
        else:
            ant = lista [i]
            i += 1
    return ordenado

mi_lista = [3,5,1,10]
print('Ordenada: ', en_orden_ascendente(mi_lista))
```

2. Dada una lista de valores enteros, indicar cuántos están por debajo de un valor dado por el usuario.

#### SOLUCION PROPUESTA:

```
def cuantos_por_debajo(lista, umbral):
    """ lista, int --> int
    OBJ: Devuelve cuántos valores en la lista están por debajo del umbral"""
    i=0
    cont=0
    for valor in lista:
        if valor < umbral: cont+=1
    return cont

mi_lista = [3, 5, 1, 10, 41, 23, 19, 33]
umbral = int(input('Introduzca un valor umbral: '))
print('Hay ', cuantos_por_debajo(mi_lista, umbral), ' valores por debajo del umbral')
```

3. Suponer que se dispone de una serie de  $n$  candidatos ( $n$  es introducido por el usuario) para una votación. Escribir un algoritmo que realice:
- Lectura de votos para los candidatos (finalizar introduciendo -)
  - Buscar el candidato ganador.

#### SOLUCION PROPUESTA:

```
def inicializar (lista, n):
    """lista, int-->lista
    OBJ: La lista se inicializa con todos sus valores a 0"""
    for i in range(n):
        lista.append(0)
    return

def votacion(n):
    """int-->lista
    OBJ: Devuelve los resultados de la votación para cada candidato """
    lista_resultados = []
    inicializar(lista_resultados, n)
    voto=int(input('Nº del candidato al que vota (-1 para terminar): '))
    while voto!=-1:
        lista_resultados[voto]+=1
        voto=int(input('Nº del candidato al que vota (-1 para terminar): '))
    return lista_resultados

def ganador(lista_votos):
    """lista-->int
    OBJ: Devuelve la posición del candidato con más votos """
    max = 0
    for i in lista_votos:
        if i > max:
            max = i
            candidato = lista_votos.index(i)
    return candidato

#Prueba
num_candidatos=int(input('Número de candidatos: '))
votos = votacion(num_candidatos)
print('El ganador es el candidato en la posicion: ', ganador(votos))
```

## Ejercicios propuestos

- Escribir una función que sume dos listas de igual longitud y retorne otra lista que contenga la suma de las originales elemento a elemento.
- 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.
- Distribuir los 20 datos enteros en dos listas de tal manera que se vayan generando dos secuencias ordenadas, una creciente y otra decreciente.
- 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).
- Modificar una lista de números reales que representan las calificaciones de los alumnos de una clase, para sustituir los valores numéricos por sus calificaciones alfanuméricas (Suspenso, Aprobado, etc.)
- Implementar una función que compruebe si una palabra es un palíndromo.

7. Implementar una función que pone en mayúsculas todas las primeras letras de las palabras de una frase.
8. Crear una función que compruebe si dos cadenas de caracteres son iguales.
9. Implemente una función que indique si una palabra contiene las cinco vocales: por ejemplo "murciélago". Modifique posteriormente la función para que detecte sólo aquellas palabras que contienen una única vez cada vocal.
10. 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".
11. 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
```