

# Test básico de algorítmica

*Sopra*

Para realizar esta prueba debes hacer uso de alguno de los siguientes lenguajes de programación: JAVA, C++, C, Python, Matlab... No se evaluarán las pruebas realizadas en pseudocódigo.

Realiza cada uno de los ejercicios en este mismo documento.

Si lo prefieres, también puedes realizar las preguntas en un paquete diferente con un IDE a tu elección. A continuación, sube el proyecto a tu cuenta de github, de manera pública, y compártenos el enlace. En su defecto, comprime el proyecto y háznoslo llegar vía email.

1. Escriba un algoritmo que lea un número entero y determine si es par o impar. Si es par, que escriba todos los pares de manera descendiente desde sí mismo y hasta el cero. Si es impar, que escriba todos los impares de manera descendiente desde sí mismo hasta el uno. Utilice la instrucción *LEER NUMERO* al inicio del programa para cargar un número en la variable *NUMERO*.

[9]:

```
# Instrucción LEER NUMERO: para cargar la variable NUMERO
def LEER_NUMERO():
    """
    Instrucción que sirve para cargar la variable NUMERO
    """

    NUMERO = int(input('Introduce el número que desea leer: '))
    return NUMERO

# Algoritmo que lee un número entero y determine si es par o impar.
def par_impar():
    """
    Algoritmo que lee un número entero y determina si es par o impar escribiendo la lista de
    pares de forma descendiente en caso de ser par, y escribiendo la lista de impares de forma
    descendiente en caso de ser impar.
    """

    numero = LEER_NUMERO()

    lista = list() # Lista que va a contener los números pares o impares de forma descendiente.

    # Si es par. Escribir todos los pares de forma descendiente desde sí mismo hasta 0.
    if numero%2==0:
        lista = [i for i in range(numero+1) if i%2==0]
        lista.reverse()

    # Si es impar. Escribir todos los impares de forma descendiente desde sí mismo hasta 1.
    else:
        lista = [i for i in range(numero+1) if i%2!=0]
        lista.reverse()

    print(lista) # Escribimos por pantalla la lista
```

[10]:

```
par_impar()
```

Introduce el número que desea leer: 6  
[6, 4, 2, 0]

[11]:

```
par_impar()
```

Introduce el número que desea leer: 11

[11, 9, 7, 5, 3, 1]

2. Escriba un algoritmo que visualice una clasificación de 50 personas según edad y sexo. Deberá mostrar los siguientes resultados:
- Cantidad de personas mayores de edad (18 años o más).
  - Cantidad de personas menores de edad.
  - Cantidad de personas masculinas mayores de edad.
  - Cantidad de personas femeninas menores de edad.
  - Porcentaje que representan las personas mayores de edad respecto al total de personas.
  - Porcentaje que representan las mujeres respecto al total de personas.

Utilice la instrucción *LEER PERSONAS* al inicio del programa para cargar los datos de las 50 personas en un variable, *PERSONAS*, que actúa como un vector de 50 posiciones. Cada elemento de *PERSONAS* es de un tipo estructurado que dispone dos campos: *SEXO* y *EDAD*.

[23]:

```
import random
# Instrucción LEER PERSONAS: cargar Los datos de 50 personas en una variable PERSONAS.
# PERSONAS es un vector de 50 posiciones. Cada posición del vector PERSONAS es de un tipo
# estructurado que dispone de dos campos: SEXO y EDAD

def LEER_PERSONAS():
    """
    Instrucción que carga los datos de 50 personas en una variable PERSONAS.
    La variable PERSONAS tiene 50 posiciones y cada posición tiene dos campos SEXO y EDAD.
    """
    PERSONAS = list()
    num_personas = 50
    for persona in range(num_personas):
        # Como introducir uno a uno Los datos de las personas es largo, se va a introducir de
        # forma aleatoria empleando random.choice.
        SEXO = random.choice(['Femenino', 'Masculino'])
        EDAD = random.choice(range(100)) # Se va a suponer que las personas no van a tener más
        # de 99 años.
        PERSONAS.append({'SEXO':SEXO, 'EDAD':EDAD})
    return PERSONAS
```

```
# Algoritmo que visualiza una clasificación de 50 personas según edad y sexo.
def clasificacion():
    """
    Algoritmo que realiza la clasificación de 50 personas según su edad y sexo.
    En concreto obtendrá:
    a. Cantidad de personas mayores de edad (18 años o más).
    b. Cantidad de personas menores de edad.
    c. Cantidad de personas masculinas mayores de edad.
    d. Cantidad de personas femeninas menores de edad.
    e. Porcentaje que representan las personas mayores de edad respecto al total de personas.
    f. Porcentaje que representan las mujeres respecto al total de personas.
    """
    personas = LEER_PERSONAS()

    # Número de personas mayores de edad (+18).
    n_mas18 = len([p for p in personas if p['EDAD']>=18])
    print('Número de personas mayores de edad (+18):')
    print(n_mas18)

    # Número de personas menores de edad (-18).
    n_menores = len([p for p in personas if p['EDAD']<18])
    print('Número de personas menores de edad (-18):')
    print(n_menores)

    # Número de personas masculinas mayores de edad.
    n_masc_mas18 = len([p for p in personas if p['EDAD']>=18 and p['SEXO']=='Masculino'])
    print('Número de personas masculinas mayores de edad:')
    print(n_masc_mas18)

    # Número de personas femeninas menores de edad.
    n_fem_menores = len([p for p in personas if p['EDAD']<18 and p['SEXO']=='Femenino'])
    print('Número de personas femeninas menores de edad:')
    print(n_fem_menores)

    # Porcentaje que representan las personas mayores de edad respecto al total de personas.
    porc_mas18 = n_mas18/len(personas)*100
    print('Porcentaje que representan las personas mayores de edad respecto al total de personas:')
    print(str(porc_mas18)+'%')

    # Porcentaje que representan las mujeres respecto al total de personas.
    n_mujeres = len([p for p in personas if p['SEXO']=='Femenino'])
    porc_mujeres = n_mujeres/len(personas)*100
    print('Porcentaje que representan las mujeres respecto al total de personas:')
    print(str(porc_mujeres)+'%')
```

[31]:

```
clasificacion()
```

```
Número de personas mayores de edad (+18):
41
Número de personas menores de edad (-18):
9
Número de personas masculinas mayores de edad:
18
Número de personas femeninas menores de edad:
2
Porcentaje que representan las personas mayores de edad respecto al total de personas:
82.0%
Porcentaje que representan las mujeres respecto al total de personas:
50.0%
```

3. Desarrolle un algoritmo para el cálculo del salario de un trabajador. El importe liquidado (sueldo) depende de una tarifa o precio por hora establecida y de un condicionante sobre las horas trabajadas: si la cantidad de horas trabajadas es mayor a 40 horas, la tarifa se incrementa en un 50% para las horas extras. Calcular el sueldo recibido por el trabajador en base las horas trabajadas y la tarifa. Utilice las instrucciones *LEER HORASTRABAJADAS* y *LEER TARIFA* al inicio del programa para cargar los valores en las variables *HORASTRABAJADAS* y *TARIFA*.

[47]:

```
# Instrucción LEER_HORASTRABAJADAS: para cargar el valor de la variable HORASTRABAJADAS.
def LEER_HORASTRABAJADAS():
    """
    Instrucción que sirve para cargar la variable HORASTRABAJADAS.
    """
    HORASTRABAJADAS = int(input('Introduce el número de horas trabajadas: '))
    return HORASTRABAJADAS

# Instrucción LEER_TARIFA: para cargar el valor de la tarifa.
def LEER_TARIFA():
    """
    Instrucción que sirve para cargar la variable TARIFA
    """
    TARIFA = int(input('Introduce la tarifa establecida: '))
    return TARIFA

# Algoritmo que calcula el salario de un trabajador.
def calcularSalario():
    """
    Función que calcula el salario de un trabajador en función de la tarifa estipulada
    y las horas trabajadas. Además, se tiene en cuenta que si se han realizado horas extra
    en esas horas se incrementa la tarifa en un 50%.
    """
    tarifa = LEER_TARIFA()
    horas_trabajadas = LEER_HORASTRABAJADAS()

    # El sueldo se calcula a partir de la tarifa y de las horas trabajadas.
    if horas_trabajadas <= 40:
        sueldo = horas_trabajadas * tarifa
    # Si el trabajador ha trabajado más de 40 horas, las horas extras se pagan como un 50% más que
    # tarifa establecida.
    else:
        h_extra = horas_trabajadas - 40
        nueva_tarifa = tarifa * 1.5
        sueldo = 40 * tarifa + h_extra * nueva_tarifa

    print('Salario semanal: ' + str(sueldo) + '€')
```

[48]:

```
calcularSalario()
```

```
Introduce la tarifa establecida: 10
Introduce el número de horas trabajadas: 40
Salario semanal: 400€
```

[49]:

```
calcularSalario()
```

```
Introduce la tarifa establecida: 10
Introduce el número de horas trabajadas: 42
Salario semanal: 430.0€
```