Testbásicodealgorítmica

*Sopra*

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

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. Acontinuación, sube el proyecto a tu cuenta de github, de manera pública, y compártenos el enlace.Ensudefecto,comprimeelproyectoyháznoslollegarvíaemail.

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. Sies impar, que escriba todos los impares de manera descendiente desde si sí mismohasta el uno. Utilice la instrucción *LEER NUMERO* al inicio del programa para cargar unnúmeroenlavariable*NUMERO*.

Respuesta:

def numeros\_pares\_impares(num):

    if num % 2 == 0:  # El número es par

        for i in range(num, -1, -2): #-2 para que descienda de dos en dos

            print(i)

    else:  # El número es impar

        for i in range(num, 0, -2):

            print(i)

numeros\_pares\_impares(10)

numeros\_pares\_impares(7)

print("TEST:")

def test\_numeros\_pares\_impares():

    assert numeros\_pares\_impares(10) == [10, 8, 6, 4, 2, 0]

    assert numeros\_pares\_impares(7) == [7, 5, 3, 1]

    assert numeros\_pares\_impares(0) == [0]

    assert numeros\_pares\_impares(1) == [1]

    assert numeros\_pares\_impares(-5) == [-5, -3, -1]

1. Escriba un algoritmo que visualice una clasificación de 50 personas según edad y sexo.Deberámostrarlossiguientesresultados:
   1. Cantidad de personas mayores de edad(18añosomás).
   2. Cantidad de personas menores de edad.
   3. Cantidaddepersonasmasculinasmayoresdeedad.
   4. Cantidaddepersonasfemeninasmenoresdeedad.
   5. Porcentaje que representan las personas mayores de edad respecto al total depersonas.
   6. Porcentajequerepresentanlasmujeresrespectoaltotaldepersonas.

Utilice la instrucción *LEER PERSONAS* al inicio del programa para cargar los datos de las50personasenunvariable,*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*.

#Lista con diccionario con los datos de las personas:

lista\_personas = [

    {"sexo": "M", "edad": 22},

    {"sexo": "F", "edad": 17},

    {"sexo": "M", "edad": 30},

    {"sexo": "F", "edad": 12},

    {"sexo": "F", "edad": 20},

    {"sexo": "M", "edad": 22},

    {"sexo": "F", "edad": 17},

    {"sexo": "M", "edad": 30},

    {"sexo": "F", "edad": 12},

    {"sexo": "F", "edad": 20},

    {"sexo": "M", "edad": 22},

    {"sexo": "F", "edad": 17},

    {"sexo": "M", "edad": 30},

    {"sexo": "F", "edad": 12},

    {"sexo": "F", "edad": 20},

    {"sexo": "M", "edad": 22},

    {"sexo": "F", "edad": 17},

    {"sexo": "M", "edad": 30},

    {"sexo": "F", "edad": 12},

    {"sexo": "F", "edad": 20},

    {"sexo": "M", "edad": 22},

    {"sexo": "F", "edad": 17},

    {"sexo": "M", "edad": 30},

    {"sexo": "F", "edad": 12},

    {"sexo": "F", "edad": 20},

    {"sexo": "M", "edad": 22},

    {"sexo": "F", "edad": 17},

    {"sexo": "M", "edad": 30},

    {"sexo": "F", "edad": 12},

    {"sexo": "F", "edad": 20},

    {"sexo": "M", "edad": 22},

    {"sexo": "F", "edad": 17},

    {"sexo": "M", "edad": 30},

    {"sexo": "F", "edad": 12},

    {"sexo": "F", "edad": 20},

    {"sexo": "M", "edad": 22},

    {"sexo": "F", "edad": 17},

    {"sexo": "M", "edad": 30},

    {"sexo": "F", "edad": 12},

    {"sexo": "F", "edad": 20},

    {"sexo": "M", "edad": 22},

    {"sexo": "F", "edad": 17},

    {"sexo": "M", "edad": 30},

    {"sexo": "F", "edad": 12},

    {"sexo": "F", "edad": 20}]

def clasificar\_personas(lista):

    #valores iniciales

    mayores\_edad = 0

    menores\_edad = 0

    masculinos\_mayores\_edad = 0

    femeninos\_menores\_edad = 0

    mujeres = 0

    #Entramos en la lista y comparamos uno a uno las personas

    for person in lista:

        #filtramos en el diccionario 1. por edad para saber si son mayores de edad o no

        if person["edad"] >= 18:

            mayores\_edad += 1

            if person["sexo"] == "M":

                masculinos\_mayores\_edad += 1

        else:

            menores\_edad += 1

            if person["sexo"] == "F":

                femeninos\_menores\_edad += 1

        if person["sexo"] == "F":

            mujeres += 1

    porcentaje\_mayores\_edad = (mayores\_edad / len(lista)) \* 100

    porcentaje\_mujeres = (mujeres / len(lista)) \* 100

    #Diccionario para indicar todos los valores solicitados

    resultados = {

        "mayores\_edad": mayores\_edad,

        "menores\_edad": menores\_edad,

        "hombres\_mayores\_edad": masculinos\_mayores\_edad,

        "mujeres\_menores\_edad": femeninos\_menores\_edad,

        "porcentaje\_mayores\_edad": porcentaje\_mayores\_edad,

        "porcentaje\_mujeres": porcentaje\_mujeres

    }

    #print(resultados)

    return resultados

print(clasificar\_personas(lista\_personas))

print("TESTS")

def test\_clasificar\_personas():

    personas = [

        {"sexo": "M", "edad": 22},

        {"sexo": "F", "edad": 17},

        {"sexo": "M", "edad": 30},

        {"sexo": "F", "edad": 12},

        {"sexo": "F", "edad": 20},

    ]

    resultados = clasificar\_personas(personas)

    assert resultados["mayores\_edad"] == 3

    assert resultados["menores\_edad"] == 2

    assert resultados["hombres\_mayores\_edad"] == 2

    assert resultados["mujeres\_menores\_edad"] == 1

    assert resultados["porcentaje\_mayores\_edad"] == 60.0

    assert resultados["porcentaje\_mujeres"] == 60.0

    #Añado comentario: no me daba problemas, pero al final me daba un “invalid syntax” en VSC, pero probándolo en la herramienta ‘Thonny’ no daba problemas

1. Desarrolle un algoritmo para el cálculo del salario de un trabajador. El importeliquidado (sueldo) depende de una tarifa o precio por hora establecida y de uncondicionante 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 sueldorecibido por el trabajador en base las horas trabajadas y la tarifa. Utilice lasinstrucciones *LEER HORASTRABAJADAS* y *LEER TARIFA* al inicio del programa paracargarlosvaloresenlasvariables *HORASTRABAJADAS*y*TARIFA*.

horas\_trabajadas = float(input("¿Cuántas horas has trabajado?: "))

tarifa\_hora = float(input("¿Cuál es tu 'tarifa' sueldo/hora?: "))

horas\_minimas = 40

multiplicador\_horas\_extra = 1.5

#si trabaja  horas extra habrá que calcular el importe de ésta

if horas\_trabajadas > horas\_minimas:

   sueldo = ( horas\_minimas \* tarifa\_hora) + ((horas\_trabajadas - horas\_minimas) \* tarifa\_hora \* multiplicador\_horas\_extra)

else:

    sueldo = horas\_trabajadas \* tarifa\_hora

print(f"El sueldo del trabajador es: {sueldo} €")

NOTA: Por tiempo no me he podido comprobar con un try exception si los valores de horas extra y tarifa son float o se introduce valores con error. Por ende si se introduce un string el algoritmo dará error.