

## Nivell 1

## - Exercici 1

Calculadora de l'índex de massa corporal

Escriu una funció que calculi l'IMC ingressat per l'usuari/ària, és a dir, qui ho executi haurà d'ingressar aquestes dades. Pots obtenir més informació del seu càlcul en:

La funció ha de classificar el resultat en les seves respectives categories

```
#Solicitamos los datos
peso = float(input("Ingrese su peso en kg:"))

altura = float(input("Ingrese su altura en metros:"))

#Calculamos el IMC
imc = peso / (altura ** 2)

# Determinamos la clasificación del IMC
if imc < 18.5:
    clasificacion = "Peso inferior al normal"
elif 18.5 <= imc <= 24.9:
    clasificacion = "Normal"
elif 25.0 <= imc <= 29.9:
    clasificacion = "Peso superior al normal"
else:
    clasificacion = "Obesidad"

#Mostramos los resultados

print("%.2f" % imc) #Redondeamos a 2 decimales

print(clasificacion)
```

✓ 10.7s

24.44

Normal

## - Exercici 2

Convertidor de temperatures.

Existeixen diverses unitats de temperatura utilitzades en diferents contextos i regions. Les més comunes són Celsius (°C), Fahrenheit (°F) i Kelvin (K). També existeixen altres unitats com Rankine (°Ra) i Réaumur (°Re). Selecciona almenys 2 i construeix el convertidor.

```
def fahrenheit_celsius():  
    fahrenheit = int(input('Ingrese la temperatura en grados Fahrenheit: '))  
    celsius = (fahrenheit - 32) * 5.0/9.0  
    return '{} grados Fahrenheit son {}'.format(fahrenheit, celsius)  
  
def celsius_fahrenheit():  
    celsius = int(input('Ingrese la temperatura en grados Celsius: '))  
    fahrenheit = 9.0/5.0 * celsius + 32  
    return '{} grados Celsius son {}'.format(fahrenheit, celsius)  
  
while True:  
    print('1.- Fahrenheit a Celsius')  
    print('2.- Celsius a Fahrenheit')  
  
    try:  
        opcion = int(input('Seleccione una opción: '))  
        if opcion == 1:  
            print(fahrenheit_celsius())  
        elif opcion == 2:  
            print(celsius_fahrenheit())  
        else:  
            raise ValueError  
    except ValueError:  
        print('Ingrese solo números.(1/2)')
```

```
1.- Fahrenheit a Celsius  
2.- Celsius a Fahrenheit  
124 grados Fahrenheit son 51.111111111111114 grados Celsius
```

## - Exercici 3

Comptador de paraules d'un text.

Escriu una funció que donat un text, mostri les vegades que apareix cada paraula.

```
def contar_palabras(texto):
    # Creamos un diccionario para almacenar las palabras y sus frecuencias
    frecuencia_palabras = {}

    # Dividimos el texto en palabras usando el espacio como separador
    palabras = texto.split()

    # Iteramos sobre cada palabra en el texto
    for palabra in palabras:
        # Eliminamos los signos de puntuación alrededor de la palabra
        palabra = palabra.strip('.,!?:;"()[]{}')

        # Convertimos la palabra a minúsculas para evitar diferencias de capitalización
        palabra = palabra.lower()

        # Si la palabra ya está en el diccionario, incrementamos su frecuencia
        if palabra in frecuencia_palabras:
            frecuencia_palabras[palabra] += 1
        # Si la palabra no está en el diccionario, la inicializamos con una frecuencia de 1
        else:
            frecuencia_palabras[palabra] = 1

    # Mostramos las palabras y sus frecuencias
    for palabra, frecuencia in frecuencia_palabras.items():
        print(f"{palabra}: {frecuencia}")

# Ejemplo
texto = """En el código de arriba, hemos añadido .1 entre % y f en el operador . Esto significa que queremos redondear el número hasta un dígito decimal.
Nota que obtendrás un resultado similar al primer ejemplo si omites el punto (.) que viene antes del dígito que le entregamos en medio % y f."""
contar_palabras(texto)

✓ 0.0s Python
```

```
en: 3
el: 4
código: 1
de: 1
arriba: 1
hemos: 1
añadido: 1
1: 1
```

#### - Exercici 4

Diccionari invers.

Resulta que el client té una enquesta molt antiga que s'emmagatzema en un diccionari i els resultats els necessita al revés, és a dir, intercanviats les claus i els valors. Els valors i claus en el diccionari original són únics; si aquest no és el cas, la funció hauria d'imprimir un missatge d'avertiment.

```
def diccionari_invers(diccionari):
    # Comprovem si els valors i les claus són únics
    valors_unicos = len(set(diccionari.values())) == len(diccionari.values())
    claus_unicas = len(set(diccionari.keys())) == len(diccionari.keys())

    if not (valors_unicos and claus_unicas):
        print("Advertència: Els valors i les claus al diccionari original no són únics.")
        return None

    # Intercanviem claus i valors
    diccionari_invers = {valor: clau for clau, valor in diccionari.items()}

    return diccionari_invers

# Exemple d'ús
diccionari_original = {'a': 1, 'b': 2, 'c': 3}
diccionari_invers_resultant = diccionari_invers(diccionari_original)

if diccionari_invers_resultant is not None:
    print("Diccionari original:", diccionari_original)
    print("Diccionari invers:", diccionari_invers_resultant)

✓ 0.0s
```

```
Diccionari original: {'a': 1, 'b': 2, 'c': 3}
Diccionari invers: {1: 'a', 2: 'b', 3: 'c'}
```

## Nivell 2

### Exercici 1

#### Diccionari invers amb duplicats

Continuant amb l'exercici 4 del nivell 1: al client es va oblidar de comentar un detall i resulta que els valors en el diccionari original poden duplicar-se i més, per la qual cosa les claus intercanviades poden tenir duplicats. En aquest cas, en l'exercici anterior imprimies un missatge d'avertiment, ara, els valors del diccionari resultant hauran d'emmagatzemar-se com una llista. Tingues en compte que si és un valor únic no ha de ser una llista.

```
def diccionari_invers(diccionari):
    # Intercanviar claus i valors, gestionant duplicats
    diccionari_invers = {}
    for clau, valor in diccionari.items():
        if valor in diccionari_invers:
            if isinstance(diccionari_invers[valor], list):
                diccionari_invers[valor].append(clau)
            else:
                diccionari_invers[valor] = [diccionari_invers[valor], clau]
        else:
            diccionari_invers[valor] = clau

    return diccionari_invers

# Exemple d'ús
diccionari_original = {'a': 1, 'b': 2, 'c': 1, 'd': 3, 'e': 2}
diccionari_invers_resultant = diccionari_invers(diccionari_original)

print("Diccionari original:", diccionari_original)
print("Diccionari invers:", diccionari_invers_resultant)
```

✓ 0.0s

```
Diccionari original: {'a': 1, 'b': 2, 'c': 1, 'd': 3, 'e': 2}
Diccionari invers: {1: ['a', 'c'], 2: ['b', 'e'], 3: 'd'}
```

## Exercici 2

### Conversió de tipus de dades

El client rep una llista de dades i necessita generar dues llistes, la primera on estaran tots els elements que es van poder convertir en flotants i l'altra on estan els elements que no es van poder convertir. Exemple de la llista que rep el client: ['1.3', 'one', '1e10', 'seven', '3-1/2', ('2',1,1.4,'not-a-number'), [1,2,'3','3.4']]

```
def dividir_llistes(dades):
    elements_convertits = []
    elements_no_convertits = []

    for element in dades:
        try:
            # Intenta convertir l'element a float
            valor = float(element)
            # Si es pot convertir, afegim a la llista d'elements convertits
            elements_convertits.append(valor)
        except ValueError:
            # Si no es pot convertir, afegim a la llista d'elements no convertits
            elements_no_convertits.append(element)

    return elements_convertits, elements_no_convertits

# Llista proporcionada pel client
dades = ['1.3', 'one', '1e10', 'seven', '3-1/2', ('2', '1', '1.4', 'not-a-number', '1', '2', '3', '3.4')]

# Dividir la llista en dos
elements_convertits, elements_no_convertits = dividir_llistes(dades)

print("Elements convertits a flotants:", elements_convertits)
print("Elements no convertits a flotants:", elements_no_convertits)
```

✓ 0.0s

Elements convertits a flotants: [1.3, 10000000000.0, 2.0, 1.0, 1.4, 1.0, 2.0, 3.0, 3.4]  
Elements no convertits a flotants: ['one', 'seven', '3-1/2', 'not-a-number']

## Nivell 3

### Exercici 1

Comptador i endreçador de paraules d'un text.

El client va quedar content amb el comptador de paraules, però ara vol llegir arxius TXT i que calculi la freqüència de cada paraula ordenades dins de les entrades habituals del diccionari segons la lletra amb la qual comencen, és a dir, les claus han d'anar de la A a la Z i dins de la A hem d'anar de la A la Z. Per exemple, per a l'arxiu "tu\_me\_quieres\_blanca.txt" la sortida esperada seria:

```

from collections import defaultdict
import string

def comptar_paraules(text):
    paraules = text.split()
    comptador = defaultdict(int)
    for paraula in paraules:
        paraula = paraula.strip(string.punctuation).lower()
        comptador[paraula] += 1
    return comptador

def ordenar_i_mostrar(comptador):
    paraules_ordenades = sorted(comptador.items(), key=lambda x: (x[0][:2], x[0]))
    for paraula, freq in paraules_ordenades:
        print(f"{paraula}: {freq}")

def main():
    text = """ Tú me quieres alba,
me quieres de espumas,
me quieres de nácar.
Que sea azucena
Sobre todas, casta.
De perfume tenue.
Corola cerrada .

Ni un rayo de luna
filtrado me haya.
Ni una margarita
se diga mi hermana.
Tú me quieres nívea,
tú me quieres blanca,
tú me quieres alba.

Tú que hubiste todas
las copas a mano,
de frutos y mieles
los labios morados.
Tú que en el banquete

```

```

que por las alcobas
se quedó enredada,
entonces, buen hombre,
preténdeme blanca,
preténdeme nivea,
preténdeme casta.
"""

comptador = comptar_paraules(text)
ordenar_i_mostrar(comptador)

if __name__ == "__main__":
    main()

```

✓ 0.0s

```

: 1
a: 3
agua: 1
al: 2
alba: 4
alcobas: 1
alimenta: 1
alma: 1
amarga: 1
azucena: 1
baco: 1
banquete: 1
bebe: 1
blanca: 3
boca: 1
bosques: 1
buen: 1
cabañas: 1
carnes: 2
casta: 3
cerrada: 1
con: 4
conservas: 1
copas: 1
corola: 1
...
vete: 1
vive: 1
y: 5
ime: 1

```