SPRINT 7

Edoardo Brega

★ 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.

```
# Importo la biblioteca `keyboard` para utilizar el botón 'Esc'.
import keyboard

# Creo una función que acepta dos valores de entrada y los reconoce como 'peso' y 'altura'.
def calc_imc(peso, altura):

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

# Clasifico el IMC calculado en categorías.
   if imc < 18.5: cat = "Bajo Peso"
   elif 18.5 <= imc < 25: cat = "Peso Normal"
   elif 25 <= imc < 30: cat = "Sobrepeso"
   else: cat = "Obesidad"

# Devuelvo el IMC calculado y la categoría clasificada.
   return imc, cat

# Mensaje de inicio del programa.
print("Programa para el cálculo del Índice de Masa Corporal")

# Empiezo el bucle para recibir la entrada correcta del peso.
while True:

# Si el usuario presiona 'Esc', el programa se cierra.
   if keyboard.is_pressed('esc'):
      print("Salida del programa")
      break

# Pido al usuario que ingrese el peso y cambio la coma por el punto si es necesario para poder aceptarlo como número.
   peso = input("Ingresa el peso en 'Kg' o presiona ESC para salir").replace(",", ".")

# Intento convertir el valor ingresado a un número decimal (float).
   try:
      peso = float(peso) l

# Si el valor es un número decimal, verifico que esté dentro del rango de certificación del IMC.
      if 29.9 <= peso <= 250.1:

# Si el peso está en este rango, salgo del bucle con `break` para ir a pedir la altura.
         break

# Si el peso no está en el rango, informo al usuario y vuelvo al principio del bucle `while` para que ingrese nuevamente el peso.
      else:
         print(f"El valor ingresado es {peso} kg\nLa masa corporal se certifica solo para valores de peso entre 30 kg y 250 kg\nIngresa otra vez el peso")
```

```
# Si el valor ingresado no es un número y no es ‘Esc’, informo al usuario y vuelvo al principio del bucle `while` para que ingrese nuevamente el peso.
    except ValueError:
        if not keyboard.is_pressed('esc'):
            print(f"El valor ingresado no es válido: {peso}\nIngresa otra vez el peso")

# Si el usuario no ha presionado ‘Esc’, procedo a pedir la altura con un nuevo bucle.
if not keyboard.is_pressed('esc'):
    while True:

# Si el usuario presiona ‘Esc’, el programa se cierra.
        if keyboard.is_pressed('esc'):
            print("Salida del programa")
            break

# Pido al usuario que ingrese la altura y cambio la coma por el punto si es necesario para poder aceptarlo como número.
        altura = input("Ingresa la altura en 'm' o presiona ESC para salir").replace(",", ".")

# Intento convertir el valor ingresado a un número decimal (float).
        try:
            altura = float(altura)

# Si el valor es un número decimal, verifico que esté dentro del rango de certificación del IMC.
            if 0.9 <= altura <= 2.6:

# Si la altura está en este rango, salgo del bucle con `break` para proceder al cálculo.
                break

# Si la altura no está en el rango, informo al usuario y vuelvo al principio del bucle `while` para que ingrese nuevamente la altura.
            else:
                print(f"El valor ingresado es {altura} m\nLa masa corporal se certifica solo para valores de altura entre 1 m y 2.5 m\nIngresa otra vez la altura")

# Si el valor ingresado no es un número y no es ‘Esc’, informo al usuario y vuelvo al principio del bucle `while` para que ingrese nuevamente la altura.
        except ValueError:
            if not keyboard.is_pressed('esc'):
                print(f"El valor ingresado no es válido: {altura}\nIngresa otra vez la altura")

# Si el usuario ha ingresado correctamente el peso y la altura y no ha presionado ‘Esc’, procedo al cálculo a través de la función.
if not keyboard.is_pressed('esc'):

# Defino dos variables que serán los resultados de la función.
    res_imc, res_cat = calc_imc(peso, altura)

# Muestro en pantalla los resultados.
    print(f"Datos de peso y altura recibidos: {peso} Kg, {altura} m\nIMC = {res_imc:.2f}\nCategoría = {res_cat}")
```

➔ 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 conversors, de tal manera que en introduir una temperatura retorni, com a mínim, dues conversions.

```
# Importo la biblioteca `keyboard` para utilizar el botón 'Esc'.
import keyboard

# Creo las funciones para convertir las temperaturas.
def celsius_2_fahrenheit(celsius): return (celsius * 9/5) + 32
def celsius_2_kelvin(celsius): return celsius + 273.15
def fahrenheit_2_celsius(fahrenheit): return (fahrenheit - 32) * 5/9
def fahrenheit_2_kelvin(fahrenheit): return (fahrenheit - 32) * 5 / 9 + 273.15
def kelvin_2_celsius(kelvin): return kelvin - 273.15
def kelvin_2_fahrenheit(kelvin): return (kelvin - 273.15) * 9 / 5 + 32

# Mensaje de inicio del programa.
print("Programa para la conversión de temperaturas entre Celsius (c), Fahrenheit (f) e Kelvin (k)")

# Empiezo un bucle para pedir la temperatura.
while True:

# Si el usuario presiona 'Esc', el programa se cierra.
    if keyboard.is_pressed('esc'):
        print("Salida del programa")
        break

# Pido al usuario que ingrese la temperatura, recordando ingresar también la unidad de medida; quito el espacio, la coma o el símbolo º si es necesario.
    temp_in = input("Ingresa la temperatura que quieres convertir con su unidad de medida\n(ejemplos: 25C, 77F, 300K): ").strip().replace(',', '.').replace("°", "")

# Este control me ayuda a evitar mensajes duplicados o errores en caso de que el usuario presione 'Esc' o no ingrese ningún valor.
    if not temp_in:
        continue

# Empiezo el reconocimiento del valor ingresado.
    try:

# Si se reconoce el valor como Celsius, procedo a las conversiones.
        if temp_in[-1].upper() == 'C':

# Quito el último carácter, que debería ser la unidad de medida, e intento la conversión a número decimal.
            celsius = float(temp_in[:-1])

# Verifico que la temperatura esté dentro de los valores permitidos por la unidad de medida.
            if -273.15 <= celsius <= 1000:

# Muestro en pantalla las conversiones.
                print(f"{celsius}°C = {celsius_2_fahrenheit(celsius):.2f}°F = {celsius_2_kelvin(celsius):.2f}K")
                break
            else:

# Si el valor está fuera del rango, informo al usuario para que lo ingrese correctamente.
                print("Las temperaturas en Celsius tienen que ser entre -273.15°C y 1000°C")
```

```
# Si se reconoce el valor como Fahrenheit, procedo a las conversiones.
    elif temp_in[-1].upper() == 'F':
        fahrenheit = float(temp_in[:-1])

# Verifico que la temperatura esté dentro de los valores permitidos por la unidad de medida.
        if -459.67 <= fahrenheit <= 1832:

# Muestro en pantalla las conversiones.
            print(f"{fahrenheit}°F = {fahrenheit_2_celsius(fahrenheit):.2f}°C = {fahrenheit_2_kelvin(fahrenheit):.2f}K")
            break
        else:

# Si el valor está fuera del rango, informo al usuario para que lo ingrese correctamente.
            print("Las temperaturas en Fahrenheit tienen que ser entre -459.67°F y 1832°F")

# Si se reconoce el valor como Kelvin, procedo a las conversiones.
    elif temp_in[-1].upper() == 'K':
        kelvin = float(temp_in[:-1])

# Verifico que la temperatura esté dentro de los valores permitidos por la unidad de medida.
        if 0 <= kelvin <= 1273.15:

# Muestro en pantalla las conversiones.
            print(f"{kelvin}K = {kelvin_2_celsius(kelvin):.2f}°C = {kelvin_2_fahrenheit(kelvin):.2f}°F")
            break
        else:

# Si el valor está fuera del rango, informo al usuario para que lo ingrese nuevamente.
            print("Las temperaturas en Kelvin tienen que ser entre 0K y 1273.15K")

# Si el último carácter no es uno de los tres requeridos, informo al usuario para que vuelva a ingresar la temperatura.
    else:
        print("Unidad de medida no reconocida. Recuerda añadir C, F o K después del valor de la temperatura")

# En caso de que el valor no sea un número después de quitar el último carácter, informo al usuario para que vuelva a ingresar la temperatura.
  except ValueError:
    print("Ingresa un valor numérico válido para la temperatura")
```

➔ exercici 3

- **Comptador de paraules d'un text:**

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

```
# Importo la biblioteca `keyboard` para utilizar el botón ‘Esc’.
import keyboard

# Creo la función para contar las palabras.
def comptador_paraules(texto):

# Convierto el texto en minúsculas.
    texto = texto.lower()

# Convierto el texto en una lista de palabras usando el espacio como separador.
    palabras = texto.split()

# Inicializo un diccionario para almacenar y contar las palabras.
    contador = {}

# Empiezo un ciclo `for` para contar las palabras.
    for word in palabras:

# Verifico si la palabra ya está en el diccionario y aumento en uno el contador de la palabra.
        if word in contador:
            contador[word] += 1

# Si la palabra no está en el diccionario, la añado e inicializo el valor de la clave a 1.
        else:
            contador[word] = 1

# Devuelvo el diccionario.
    return contador

# Mensaje de inicio del programa.
print("Programa para contar las veces que aparece cada palabra o cada número en un texto")

# Empiezo un bucle para que el usuario ingrese el texto.
while True:

# Si el usuario presiona ‘Esc’, el programa se cierra.
    if keyboard.is_pressed('esc'):
        print("Salida del programa")
        break

# Pido al usuario que ingrese el texto.
    texto_in = input("Escribe el texto: \n")

# Llamo a la función para el recuento de las palabras ingresadas.
    resultado = comptador_paraules(texto_in)

# Si el usuario no ha presionado 'esc', muestro en pantalla el resultado.
    if not keyboard.is_pressed('esc'):
        print("\nResultado:")

# Utilizo sorted para ordenar alfabéticamente el diccionario.
        for palabra in sorted(resultado.keys()):

# Muestra cada palabra con su recuento y finaliza el programa.
            print(f"'{palabra}' = {resultado[palabra]} ")
        break
```

➔ 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'advertiment.

```
# Creo una función para invertir un diccionario.
def diccionario_inverso(diccionario_original):

# Inicializo un diccionario que almacenará las llaves y los valores invertidos.
    diccionario_inv = {}

# Inicializo un diccionario que almacenará los duplicados.
    duplicates = {}

# Inicio un ciclo for para recorrer los elementos del diccionario original.
    for key, value in diccionario_original.items():

# Verifico si el valor que quiero como nueva llave ya existe en el diccionario invertido.
        if value in diccionario_inv:

# Si el elemento no está como llave en el diccionario de los duplicados, lo añado.
            if value not in duplicates:
                duplicates[value] = [diccionario_inv[value]]

# Si el elemento ya está como llave en el diccionario de los duplicados, añado la llave actual a la lista de claves asociadas a ese elemento duplicado.
            duplicates[value].append(key)

# Si el valor no es duplicado, lo agrego al diccionario invertido con la clave como valor.
        else:
            diccionario_inv[value] = key

# La función devuelve el diccionario invertido y el diccionario de duplicados.
    return diccionario_inv, duplicates

# Diccionario de prueba 1 con claves únicas.
diccionario_original1 = {
    'a': 1,
    'b': 2,
    'c': 3,
}

# Diccionario de prueba 2 con claves duplicadas.
diccionario_original2 = {
    'x': 'apple',
    'y': 'banana',
    'z': 'banana',
}

# Muestro en pantalla el primer diccionario original.
print(f"***\nDiccionario original 1:\n {diccionario_original1}")

# Llamo a la función.
dic_inv1, duplicados1 = diccionario_inverso(diccionario_original1)

# Muestro el diccionario invertido.
print(f"\nDiccionario invertito 1: \n{dic_inv1}")
```

```
# Si el diccionario de duplicados existe, muestro también los duplicados.
if duplicados1:
    print(f"\nError: No se ha invertido todo el diccionario\nSe han encontrado estos duplicados: \n{duplicados1}\n")

# Si el diccionario de duplicados no existe, muestro el mensaje en pantalla para informar al usuario.
else:
    print("\nEl diccionario ha sido invertido correctamente\n")

# Muestro en pantalla el segundo diccionario original.
print(f"\n***\nDiccionario original 2:\n {diccionario_original2}")

# Llamo a la función.
dic_inv2, duplicados2 = diccionario_inverso(diccionario_original2)

# Muestro el diccionario invertido.
print(f"\nDiccionario invertito 2: \n{dic_inv2}")

# Si el diccionario de duplicados existe, muestro también los duplicados.
if duplicados2:
    print(f"\nError: No se ha invertido todo el diccionario\nSe han encontrado estos duplicados: \n{duplicados2}\n")

# Si el diccionario de duplicados no existe, muestro el mensaje en pantalla para informar al usuario.
else:
    print("\nEl diccionario ha sido invertido correctamente\n")
```

★ 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'advertiment, 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.

```
# Creo una función para invertir un diccionario.
def diccionario_inverso(diccionario_original):

# Inicializo un diccionario que almacenará las llaves y los valores invertidos.
    diccionario_inv = {}

# Inicializo un diccionario que almacenará los duplicados.
    duplicates = {}

# Inicio un ciclo for para recorrer los elementos del diccionario original.
    for key, value in diccionario_original.items():

# Verifico si el valor que quiero como nueva llave ya existe en el diccionario invertido.
        if value in diccionario_inv:

# Si el elemento ya existe pero aún no es una lista, lo convierto en una lista para almacenar todos los duplicados de la misma llave.
            if not isinstance(diccionario_inv[value], list):
                diccionario_inv[value] = [diccionario_inv[value]]

# Añado la llave a la lista de duplicados.
            diccionario_inv[value].append(key)

# Añado esta lista de duplicados al diccionario de duplicados.
            duplicates[value] = diccionario_inv[value]

# Si el valor no es un duplicado, lo añado al diccionario invertido.
        else:
            diccionario_inv[value] = key

# La función devuelve el diccionario invertido y el diccionario de duplicados.
    return diccionario_inv, duplicates

# Diccionario de prueba 1 con claves únicas.
diccionario_original1 = {'a': 1, 'b': 2, 'c': 3}

# Diccionario de prueba 2 con claves duplicadas.
diccionario_original2 = {'x': 'apple', 'y': 'banana', 'z': 'banana'}

# Muestro en pantalla el primer diccionario original.
print(f"\n**********\nDiccionario original 1:\n{diccionario_original1}")

# Llamo a la función.
dic_inv1, duplicados1 = diccionario_inverso(diccionario_original1)
```

```
# Si el diccionario de duplicados existe, muestro el diccionario invertido y el diccionario de duplicados.
if duplicados1:
    print(f"\Diccionario con duplicados transformado en lista:\n{dic_inv1}\n\nValores duplicados encontrados:
{duplicados1}\n")

# Si el diccionario de duplicados no existe, muestro el diccionario invertido e informo al usuario.
else:
    print(f"\nDiccionario invertido, no habian duplicados:\n{dic_inv1}\n")

# Muestro en pantalla el segundo diccionario original.
print(f"\n**********************************************************************\nDiccionario original 2:\n{diccionario_original2}")

# Llamo a la función.
dic_inv2, duplicados2 = diccionario_inverso(diccionario_original2)

# Si el diccionario de duplicados existe, muestro el diccionario invertido y el diccionario de duplicados.
if duplicados2:
    print(f"\nDiccionario con duplicados transformado en lista:\n{dic_inv2}\n\nValores duplicados encontrados:
{duplicados2}\n")

# Si el diccionario de duplicados no existe, muestro el diccionario invertido e informo al usuario.
else:
    print(f"\nDiccionario invertido, no habian duplicados:\n{dic_inv2}\n")
```

➔ exercici 2

- **La gerència està interessada a analitzar més a fons les vendes en relació amb el mes. Per tant, et demanen que facis els ajustos necessaris per a mostrar la informació d'aquesta manera.**

```
# Creo una función para separar los números de los elementos que no son números.
def separa_numeros(lista):

# Inicializo una lista para los elementos convertibles.
    lista_de_convertibles = []

# Inicializo una lista para los elementos no convertibles.
    lista_de_no_convertibles = []

# Defino una función interna para manejar la recursividad.
    def procesa_elemento(elemento):

# Verifico si el elemento es una tupla o una lista.
        if isinstance(elemento, (tuple, list)):

# Si el elemento es una tupla o una lista, inicio un ciclo for para verificar los elementos que contiene.
            for sub_item in elemento:
                procesa_elemento(sub_item)

# Intento convertir el elemento en un número decimal float y añadirlo a la lista de convertibles.
                else:
                    try:
                        lista_de_convertibles.append(float(elemento))

# Si no se puede convertir, lo añado a la lista de no convertibles.
                    except (ValueError, TypeError):
                        lista_de_no_convertibles.append(elemento)

# Inicio un ciclo for para verificar los elementos de la lista recibida.
                for item in lista:
                    procesa_elemento(item)

# La función devuelve las dos listas.
                return lista_de_convertibles, lista_de_no_convertibles

# Lista de prueba.
            lista_prueba = ['1.3', 'one', '1e10', 'seven', '3-1/2', ('2', 1, 1.4, 'not-a-number'), [1, 2, '3', '3.4']]

# Llamo a la función.
            lista_convertibles, lista_no_convertibles = separa_numeros(lista_prueba)

# Muestro en pantalla las dos listas.
            print(f"Elementos convertibles: \n{lista_convertibles}\nElementos no convertibles: \n{lista_no_convertibles}\n")
```

★ 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:

```
# Creo una función para contar la frecuencia de las palabras desde un archivo.
def cuenta_palabras(file):

# Inicializo un diccionario para contar la frecuencia de cada palabra.
    frecuencia = {}

# Intento abrir el archivo en modo lectura con codificación UTF-8.
    try:
        with open(file, 'r', encoding='utf-8') as file:

# Leo el archivo, lo convierto en minúsculas y elimino la puntuación encontrada.
            palabras = file.read().lower().translate(str.maketrans('', '', ',.!?;:()¡'))

# Convierto el texto en una lista con los elementos divididos por espacios.
            palabras = palabras.split()

# Inicio un ciclo for para verificar cada palabra de la lista de palabras.
            for word in palabras:

# Si la palabra ya está en el diccionario, incremento su contador.
                if word in frecuencia:
                    frecuencia[word] += 1

# Si la palabra no está ya en el diccionario, la añado como llave e incremento su contador.
                else:
                    frecuencia[word] = 1

# Inicializo un diccionario para agrupar y ordenar las palabras alfabéticamente.
        diccionario_ordenado = {}

# Inicio un ciclo for para verificar las palabras ordenadas alfabéticamente con sorted.
        for word in sorted(frecuencia.keys()):

# Inicializo una variable con la primera letra de la palabra.
            primeraletra = word[0]

# Verifico si la primera letra de la palabra no está en el diccionario ordenado y la agrego.
            if primeraletra not in diccionario_ordenado:
                diccionario_ordenado[primeraletra] = {}

# Agrego la palabra y su frecuencia al diccionario ordenado por la letra correspondiente.
            diccionario_ordenado[primeraletra][word] = frecuencia[word]
```

```
# Muestro el resultado con dos ciclos for: primero para la primera letra y luego para las palabras correspondientes.
        for letra, palabras in diccionario_ordenado.items():
            print(f"\nPalabras con '{letra}':")
            for palabra, count in palabras.items():
                print(f"{palabra}: {count}")

# Si el archivo no se encuentra, muestro un mensaje de error.
    except FileNotFoundError:
        print("File no encontrado")

# Elijo el archivo para leer.
file = 'tu_me_quieres_blanca.txt'

# Llamo a la función.
cuenta_palabras(file)
```