Instrucciones:

- Resuelve los siguientes ejercicios utilizando únicamente Python (contenidos aprendidos hasta el momento).
- Comenta el código donde sea necesario para explicar tu lógica.
- Se valorará el uso correcto y eficiente de estructuras como bucles, condicionales, funciones, comprensión de listas, manejo de archivos, etc.
- Se pide confeccionar los flujogramas.

Ejercicio 1. Conteo condicional

Escribe una función que reciba una lista de números enteros y retorne cuántos números:

- Son positivos
- Son negativos
- Son pares
- Son impares

Usa bucles for, condicionales if-else, y devuelve un diccionario con los conteos.

Ejercicio 2. Manejo de cadenas

Pide al usuario una cadena por consola, y:

- Convierte todo a minúsculas.
- Elimina espacios al principio y al final.
- Devuelve una lista con las palabras que tienen más de 3 letras, ordenadas alfabéticamente.

Ejercicio 3. Función recursiva + valores predefinidos

Define una función recursiva que calcule la suma de los dígitos de un número entero positivo. Usa un parámetro adicional con valor por defecto que acumule la suma. No utilices `str()` para convertir el número.

Ejercicio 4. Lambda, map y filter

Dada la siguiente lista de precios:

precios = [125.5, 99.9, 50.0, 300.0, 75.25]

Usa filter para obtener solo los precios mayores a 100, y luego map con lambda para aplicar un descuento del 15% a esos precios. Devuelve la nueva lista de precios con descuento.

Ejercicio 5. Iteración sobre diccionarios

Crea un diccionario con los nombres y edades de 5 personas. Luego:

- Usa un bucle para imprimir solo las personas mayores de edad.
- Devuelve una nueva lista con strings del tipo "Nombre (Edad años)" para los mayores de edad, usando zip, comprensión de listas y items().

Ejercicio 6. Lectura de archivo

Se proporciona un archivo de texto llamado `datos_usuarios.txt` con líneas en el siguiente formato:

usuario1,25

usuario2,32

usuario3,17

Crea una función que lea el archivo y devuelva un **diccionario** con usuario como clave y edad (int) como valor. Usa manejo de archivos y manejo de strings.

Ejercicio 7. Filtrado y modificación

Con el diccionario del ejercicio anterior:

- Usa filter para obtener solo los usuarios mayores o iguales a 18 años.
- Usa map o comprensión de listas para construir una lista de diccionarios con la estructura:

[{'usuario': 'usuario1', 'edad': 25}, {'usuario': 'usuario2', 'edad': 32}]

Ejercicio 8. Transformación avanzada

A partir de la lista de diccionarios del ejercicio anterior:

- Usa reduce para obtener la suma total de edades.
- Calcula la edad promedio (usa len()).
- Añade una nueva clave a cada diccionario llamada "mayor_que_promedio" (valor booleano) que indique si el usuario tiene más edad que la media.

Ejercicio 9. Generación de informe

Con la estructura modificada del ejercicio anterior:

- Guarda en un archivo llamado informe_usuarios.txt un resumen donde cada línea tenga el siguiente formato:

usuario1 tiene 25 años - Mayor que promedio: False

Ejercicio 10. Función general

Escribe una función que integre todo el proceso desde la lectura del archivo original hasta la escritura del informe final. Usa:

- Parámetros con valores por defecto para los nombres de archivo.
- Parámetros variables para permitir el filtrado por edad mínima (*args o **kwargs).
- Divide el proceso en funciones auxiliares reutilizables.