PROGRAMACION II

TRABAJO PRACTICO N° 4

Codificar los siguientes ejercicios en **Python**. Poner en cada programa de Python una primera línea de comentario con su Apellido y Nombre, y nombrar los archivos con el siguiente formato: “TP4_Ejercicio_xx.py”. Por ejemplo: ***“TP4_Ejercicio_01.py”***. Subir los archivos comprimidos y agrupados en uno solo.

1- Crear un diccionario vacío. El diccionario debe llamarse "stock". Luego de crear el diccionario complételo con el siguiente stock:

tornillos = 100
tuercas = 150
arandelas = 300

Los nombres tornillos, tuercas y arandelas son las claves (keys) del diccionario mientras que las cantidades son los valores (values). Una vez armado el diccionario imprimirlo en pantalla con print.

2- Basado en el ejercicio anterior ej1, utilizaremos el diccionario como una base de datos. Comenzaremos con un diccionario de stock de nuestros productos en cero:
stock = {'tornillos': 0, 'tuercas': 0, 'arandelas': 0}
Crear un bucle utilizando while que se ejecute de forma infinita *while True.....*
Dentro de ese bucle consultar al usuario por consola que producto desea agregar al stock. Si el usuario ingresa "FIN" como producto se debe finalizar el bucle. Si el usuario ingresa un producto no definido en el stock se debe enviar un mensaje de error (si desea investigar esto se resuelve muy bien utilizando el operador "in" con diccionarios).
Luego de haber ingresado el producto se debe ingresar por consola cuanto stock de ese producto se desea agregar al stock. Si teníamos 20 tornillos y el usuario desea agregar 10 tornillos más, en nuestro diccionario deben quedar 30 tornillos (debe acumular). Cuando el usuario ingrese "FIN" y se termine el bucle, debe imprimir en pantalla con print el diccionario con el stock final.

3- Realice un programa que abra el archivo **'stock.csv**' (se adjunta) en modo lectura y cuente el stock total de tornillos a lo largo de todo el archivo, sumando el stock en cada fila del archivo. Para eso debe leer los datos del archivo con "csv.DictReader", y luego recorrer los datos dentro de un bucle y solo acceder a la columna "tornillos" para cumplir con el enunciado del ejercicio.

4- Realice un programa que abra el archivo CSV "**propiedades.csv**" (se adjunta) en modo lectura. Recorrer dicho archivo y contar la cantidad de departamentos de 2

ambientes y la cantidad de departamentos de 3 ambientes disponibles. Al finalizar el proceso, imprima en pantalla los resultados. Tener cuidado que hay departamentos que no tienen definidos la cantidad de ambientes, verifique que el texto no esté vacío antes de convertirlo a entero con "int(..)". Si desea investigar puede evitar que el programa explote utilizando "try except".

Matploblib

5- Realizar un gráfico "plot" con: *years como "x"* y *poblacion como "y"*; con los siguientes datos:

Years: 1900, 1970, 1990, 2000, 2020
Población: 1650, 3692, 5263, 6070, 7800

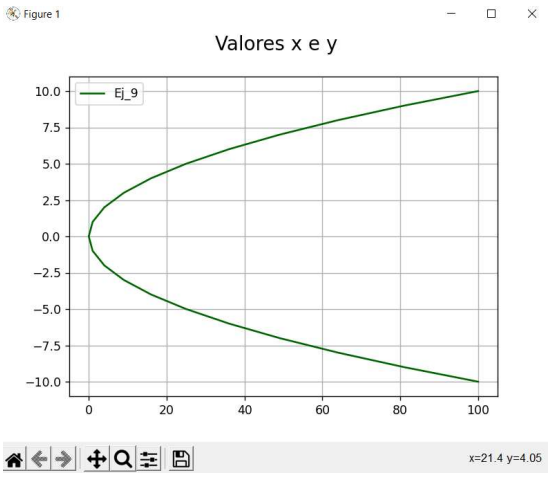
6- Realizar un gráfico “multi line plot”. A partir de los siguientes datos:

Mes: 3, 4, 5, 6
gasto_carne: 1650, 2600, 3100, 4000
gasto_verdura: 2500, 2200, 1800, 600
Calcular el gasto total como la suma de las listas gasto_carne y gasto_verdura.
Realizar un gráfico "plot" con: mes como "x"; gasto_carne como "y1”; gasto_verdura como "y2"; gasto_total como "y3".

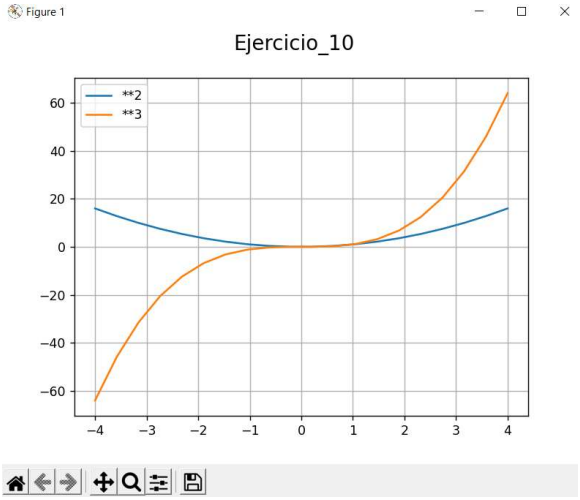
7- Realizar dos gráficos conjuntos “plot” y “scatter” (dentro de un mismo figure). Utilizar los datos del archivo población.csv (adjunto). *Year como “x” y población como “y”*. Título del gráfico “Población Histórica Mundial”. Plot con líneas verdes; Scatter con puntos rojos.

8- Realizar un gráfico de barras. Utilizar los datos del archivo población.csv (adjunto). *Year como “x” y población como “y”*. Título del gráfico “Población Histórica Mundial”. Visualizar solo los datos de los años 2000, 2005, 2010, 2015, 2020. Es decir, partiendo del csv mencionado, recorrerlo creando la estructura de datos necesaria para luego graficar la población de los años mencionados.

9- Realizar el código necesario para visualizar este gráfico (**2):



10- Realizar el código necesario para visualizar este gráfico (**2 y **3):



NumPy

11- Dado el siguiente array numpy: [1, 5, -2, 10, 2], realizar las siguientes operaciones:

a. Suma
b. Promedio
c. Ordenarlo

Mostrar todos los resultados.

12- Crear un array a partir de una lista: *l1 = list(range(10))*.
Luego, usando where:

a) Crear un nuevo array que solo tengo los números mayores a 3 del array numy y los demás reemplazar por cero.
b) Crear un nuevo array que solo tengo los números pares del array numy y los demás reemplazar por cero.

Mostrar todos los resultados.

13- Crear un array numpy: [1, 2, 4, 7]. Luego:

a. Crear la máscara para solo quedarnos con los números mayores a 1.
b. Crear la máscara para solo quedarnos con los números pares.

Mostrar todos los resultados.

14- <u>Ejercicio Final Integrador:</u>

Inventar un ejercicio de enunciado <u>propio</u> y desarrollo <u>individual</u>, que contenga todo lo visto en la materia, <u>al menos una vez</u> cada herramienta vista. Es decir, la resolución del enunciado debe tener:

- Cálculos secuenciales
- Ingreso de datos por parte del usuario, y mostrarle resultados
- Funciones para el tratamiento de cadenas de texto

- Estructuras de selección
- Bucles
- Diccionario de Datos
- Lectura de un archivo csv
- Gráficos con Matploblib
- Array NumPy