

TP5 FUNCIONES

- 1) Solo para entrar en calor, diseñar una función que me muestre un menú, este menú debe ser como se muestra a continuación:

nota: recuerden que este menú es solo texto (un string) que se muestra al usuario.

```
=====
===== Mi genial menu =====
=====
[1] Hacer el histograma en vertical
[2] Hacer el histograma en horizontal
[3] Ordenar de mayor a menor
[4] Ordenar de menor a mayor
[5] Salir
-----:> Ingrese número para seleccionar opción:
```

Si no se ingresa un número entre el 1 y el 5 debe mostrar, un mensaje de error y volver a mostrarme el menú.

Teniendo en cuenta la siguiente lista, **lista=[9, 8, 7, 9, 6, 1, 3, 1,6, 0, 5,5, 8,7, 2]**, si se selecciona la primera opción [1] debe mostrar un histograma en vertical para cada una de las celdas, si se selecciona la segunda opción [2] debe mostrar un histograma en horizontal, si se selecciona la opción [3] se debe ordenar la lista de mayor a menor, si se selecciona la opción [4] se debe ordenar la lista de menor a mayor, y finalmente si se selecciona [5] se debe terminar el programa, hacer una función para cada una de estas opciones.

Ejemplo, supongamos que tenemos una lista, sus histogramas tanto vertical como horizontal es como se muestra en la imagen, y es como se espera observar cuando se seleccione la opción [1] o [2].

lista=[6, 3, 7, 1, 5]

```
*****
***  *
***  *
*  *  *
*  *  *
*  *
*

*****
***
*****
*
*****
```

Datazos: vean que los valores dentro de la lista son números enteros positivos, esto quizá sea una ayuda para poder hacer el histograma en vertical. Una vez que se muestra el histograma, se debe volver a mostrar el menú.

Datazo del datazo: de ahora en adelante eviten a toda costa poner print() o input() dentro de las funciones, a no ser que se les pida como lo es en este ejercicio.

Datazo del datazo del datazo: vean que sucede si usan dos print de la siguiente manera

```
>>print("hola", end=" ")  
>>print("mundo")
```

- 2) Un persona necesita llenar una lista de nombres de forma descendente, es decir de la "A" a la "Z". implementar un algoritmo que permita ingresar un nombre cualquiera y lo ingrese de manera ordenada a la lista, para ello desarrolle una función que se llame "ingresaALista" que va a tener como parámetro la lista de todos los nombres y el nombre que se desea añadir, finalmente muestre en pantalla la lista de nombres.

Datazo: revelan las herramientas que ya tienen hasta ahora, y como se pueden adaptar a este problema. **Un ejemplo:** supongan que inicialmente tengo una lista vacía, y agrego el nombre de Zulma, luego el de Braian, Braian debe estar primero en la lista, ya que comienza con B, luego se ingresa Ramiro, ese se debe ingresar entre Braian y Zulma, y así sucesivamente hasta que el usuario ingrese una clave de salida, que servirá para no ingresar más nombres.

- 3) Unos científicos generaron una lista bastante compleja, que resulto de tomar datos por 5 horas cada media hora, la lista es la siguiente:

```
mediciones = [['dato1', 13], ['dato2', 73], ['dato3', 9], ['dato4', 88], ['dato5',  
12], ['dato6', 79], ['dato7', 13], ['dato8', 96], ['dato9', 16], ['dato10', 83]]
```

Diseñe un programa que a partir de esta lista, el científico pueda seleccionar una medición (**mediciones[1]**) y mostrar la celda del dato numérico(**73**). Hacer que esto se repita hasta que el usuario presione la tecla "q". Se espera que se cree una función con la siguiente cabecera.

def celdaDato(unaMedicion):

Esta función debe retornar el dato propiamente dicho de la medición seleccionada y luego mostrarlo en pantalla.

Datazo: el parámetro **unaMedicion** es de tipo lista, que contiene el nombre del dato en una celda y el dato propiamente dicho en la otra. Tengan en cuenta que el usuario tiene como entradas validas del 0 al 9 para seleccionar un dato, y la letra "q" para terminar el programa.

Datazo del datazo: este ejercicio es para que puedan crear una función para tomar un dato dentro de una lista de listas, viene bien ya que nos va a servir más adelante para hacer cosas más interesantes, recuerden que dentro de la función celda() no debe tener print() ni input()

- 4) Cuando se comienza a programar se cometen ciertos errores, y debemos estar atentos a ellos como futuros programadores, vean el siguiente código

```
def maximo(a,b):
    if x>y:
        return x
    else:
        return y

def minimo(a,b):
    if x<y:
        return x
    else:
        return y

#programa principal
x=int(input("Un número: "))
y=int(input("Otro número: "))
print(maximo(x-3, minimo(x+2, y-5)))
```

Este programa debería mostrarme en pantalla el número 2, si se le ingresan como valores $x=5$, $y=1$ pero en su lugar muestra en pantalla el número 5. ¿Qué hay que corregir?

Datazo: es bueno ver y entender códigos que no son propios, esta es una muy buena práctica para resaltar esas habilidades.

- 5) En un curso de la primaria hay 10 alumnos, un día realizaron un examen, el profesor para registrar las calificaciones de cada alumno, lo hace de la siguiente manera, **alumno=["nombreDelAlumno", nota]**, donde "alumno" es una lista donde, la primera celda está el nombre del alumno y en la segunda celda esta la nota, cada alumno está dentro otra lista llamada "clase", que es de la siguiente manera **clase=[alumno1, alumno2, alumno3....]**, así hasta completar 10 alumnos, generar un algoritmo donde puedan definir una lista "clase" que tenga los nombres y notas de 10 alumnos, la nota van a ir de 0 al 10, además deben hacer lo siguiente:

- Hacer una función que ordene la lista clase por la nota, de mayor a menor.
- Hacer una función que cambie la forma de calificar de la siguiente manera: aquellos que tienen una nota de 7 o más debe aparecer como "Aprobado" y "Desaprobado" en caso contrario

Datazo: aquí hay algo interesante, no hay una única forma de resolver este ejercicio, investiguen **como ordenar** una lista que contiene listas en cada una de sus celdas, para ello les recomiendo usar la función `sorted()`, vean el siguiente link: <https://www.geeksforgeeks.org/sorted-function-python/>

Datazo del datazo: sorted “tiene tres parámetros” el primer parámetro es una lista, tupla o una cadena, lo llamaremos iterable, ya que dentro de la función sorted va a recorrer la lista que se pase, celda por celda, luego el segundo parámetro se llama key, sirve como referencia de comparación para su ordenamiento, es decir, sorted va a recorrer la lista internamente y va a mirar cada celda con key para ir ordenando la lista con respecto a ese “algo”, por **ejemplo:** si tienen una lista general con una lista en cada una de ellas, como la que se muestra a continuación:

lista = [[1, 3, 1] , [4], [1,9]]

¿que tenemos aquí?, tenemos una lista de listas, lista[0] tiene la lista [1, 3, 1], la lista[1] tiene una lista con un solo valor que es [4], la lista[2] tiene una lista [1, 9], ahora, supongan que quiero ordenar las lista general por la longitud de cada una de las listas internas, para ello uso sorted()

listaOrdenada=sorted(lista, key= len)

vean dos cosas, a sorted le paso la lista que quiero ordenar, que en mi caso se llama “lista”, luego le digo ordéname la lista usando una función que se llame “len”, vean que solo necesita el nombre de la función, ya que automáticamente sorted va a recorrer cada una de las celdas de la lista general y va a aplicarle a cada celda la funcion “len” y en base a eso va a ordenar la lista. Y como sorted retorna una lista ordenada, creo una variable que guarde el resultado.

Finalmente sorted tiene un tercer parámetro que se llama “reverse” que es de tipo booleano, que si es False, ordena de menor a mayor y si es True ordena de mayor a menor.

listaOrdenada=sorted(lista, key= len, reverse=True)

- 6) Generen una lista con 10 números aleatorios entre el 1 y el 99. Luego se debe ordenar la lista de menor a mayor, una vez ordenada, generar de manera automática los valores faltantes entre cada una de las celdas contiguas.

Ejemplo: supongan que tenemos una lista más chica que la del enunciado, **lista=[1,6,3]**, primero se debe ordenar, es decir vean como armar una función que ordene la lista, tal que quede así **lista=[1,3,6]**, luego se debe generar los valores faltantes entre las celdas, tal que el resultado obtenido sea **lista=[1,2,3,4,5,6]**, finalmente se debe mostrar la lista.