Solo se puede buscar un solo numero por comando

Viene separado por comas

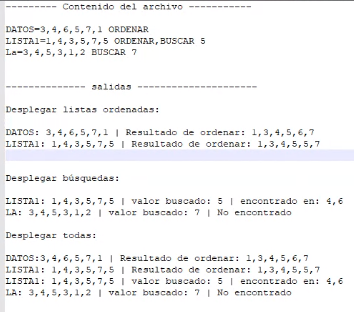
Vienen enteros positivos

**Aplicación de consola**

**Para abrir una ventana se usa una librería (puede ser externa)**

**Par odenar y todo eso: no se pueden usar librerías**

**Cuando selecciono una opción y tal línea no tiene la opción seleccionada esta se omite**

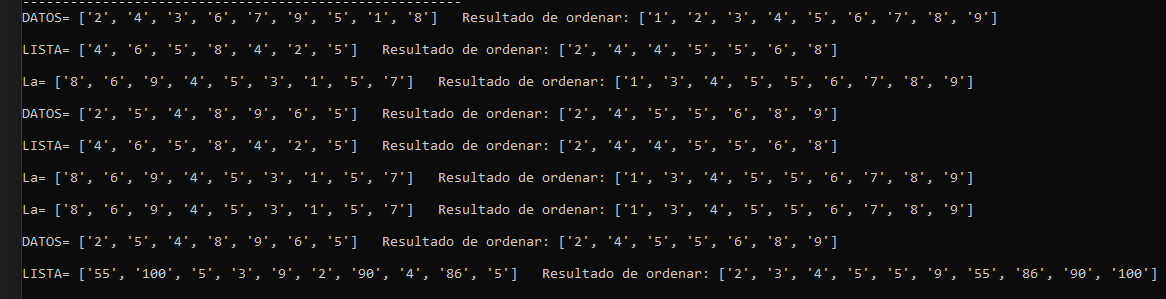


**Pueden cargar n cantidad de archivos (a la hora de cargar uno nuevo se borran los datos del anterior)**

**Las instrucciones de Buscar y todo eso siempre van a estar en mayúsculas**

**Primero viene ORENAR LUEGO BUSCAR (este es el orden a usar)**

**Si presiono ordenar y luego buscar (se busca en el que no se ordeno)**



**Posición donde esta la palabra**

contador2 = 0

    while contador2 < len(nombreOrdenar):

        if nombreOrdenar[contador2][0] == "ORDENAR":

            posiciones.append(int(contador2))

            contador2 = contador2 + 1

        else:

            contador2 = contador2 + 1

    print(posiciones)

import re

from tkinter import \*

from tkinter.filedialog import askopenfilename

# en esta viable ingreso los datos del archivo sin \n todo correcto (datos que voy a manipular)

listaSinSaltos = []

numeros = []  # ---- este es el que utilizo para trabajar) --- me contiene en una lista los numeros con los que voy ordenar o buscar

# ---- este es el que utilizo para trabajar) --- me guarda el nombre de cada lista

nombresListas = []

# ---- este es el que utilizo para trabajar) --- guarda las posiciones que tienen las palabras ordenar

nombreOrdenar = []

numeros\_desordenados =[]  # contiene la lista de lista con numeros desordenados

posiciones =[] # contiene las posiciones donde esta la palabra ordenar en todo el archivo <------- OJO sirve para filtrar con "ORDENAR"

                # es una lista simple

nombres\_listas\_filtro\_ordenar = [] # contiene los nombres de las listas donde esta la palabra ORDENAR         <------ SI resultado

                                    # es una lista simple

numeros\_desordenados\_filtro\_ordenar = [] # contiene los numeros desordenados donde esta la palabra ORDENAR    <------ SI resultado

numeros\_ordenados\_filtro\_ordenar = [] # contiene los numeros ordenados donde esta la palabra ORDENAR    <------ SI resultado

def cargar():

    try:

        root = Tk()

        root.withdraw()

        root.update()

        direccion\_archivo = askopenfilename(

            filetypes=[("Text files", "\*.txt")])

        archivo\_texto = open(direccion\_archivo, "r")

        # con readlines leo cada linea de mi archivo y la meto en una lista separada por comas

        separado = archivo\_texto.readlines()

        archivo\_texto.close()

        # con este for lo que hago es tomar la variable separado y le quito los \n (es la que uso en todo el programa )

        for i in separado:

            listaSinSaltos.append(i.strip())

        #print(listaSinSaltos)   #<-------------------<------<------<--------<-----<-<-----<<<- quitar

        print("-------------------------------------------------------")

        print("Datos cargados correctamente")

        print("-------------------------------------------------------")

    except:

        print("No selecciono ningun archivo txt, seleccione un para continuar")

def desplegarListaOrdenada():

    # -----------------------------------------# busca la palabra buscar -----------------------------------------------------------

    pattern3 = r"(ORDENAR)"   # busca la palabra buscar

    for i in listaSinSaltos:

        # busco lo que coincide con el pattern2

        listaOrdenada = re.findall(str(pattern3), i)

        if listaOrdenada:

            nombreOrdenar.append(list(listaOrdenada))

        else:

            nombreOrdenar.append(list("f"))  # agrego una f en la posicion donde no este la palabra ordenar

                                             #(para que me sirva para saltar esa posicion y no encontrar el valor de ordenar)

    #print(nombreOrdenar)

    # -----------------------------------------# lista con posiciones donde esta la palabra ordenar ---------------------------------

    contador2 = 0

    while contador2 < len(nombreOrdenar):

        if nombreOrdenar[contador2][0] == "ORDENAR":

            posiciones.append(int(contador2))          # variable posiciones me contiene las posiciones de la palabra ordenar

            contador2 = contador2 + 1

        else:

            contador2 = contador2 + 1

    print(posiciones)

    # -----------------------------------------# busca el nombre de la lista --------------------------------------------------------

    pattern2 = r"(\w+\=)"    # busca el nombre de la lista

    for i in listaSinSaltos:

        # busco lo que coincide con el pattern1

        valorNames = re.findall(str(pattern2), i)

        nombresListas.append(valorNames)  # ingreso a una lista de lista

    #print(nombresListas)

    ###################################################

    # ----------------> #nombres\_listas\_filtro\_ordenar <---- contiene los nombres de las listas donde esta la palabra ORDENAR

    ###################################################

    cont1 = 0

    while cont1 < len(nombreOrdenar):

        if cont1 in posiciones:

            nombres\_listas\_filtro\_ordenar.append(nombresListas[cont1][0])

            cont1 = cont1 + 1

        else:

            cont1 = cont1 + 1

    print(nombres\_listas\_filtro\_ordenar)

    # ['DATOS=', 'LISTA=', 'DATOS=', 'LISTA=', 'DATOS=', 'LISTA=']

   #------------------------------------  lista de lista con numeros desordenados ---------------------------------------------------

    pattern4 = r"(\d+)"     # busca los numeros de la lista

    for i in listaSinSaltos:

        valor = re.findall(pattern4, i)   # busco lo que coincide con el pattern

        numeros\_desordenados.append(valor)  # ingreso ya ordenados los numeros a una lista de lista

    ###################################################

     # ----------> #numeros\_desordenados\_filtro\_ordenar <---- contiene los numeros de las listas filtrado por el nombre BUSCAR

    ###################################################

    cont2 = 0

    while cont2 < len(nombreOrdenar):

        if cont2 in posiciones:

            if "BUSCAR" in listaSinSaltos[cont2]:

                numeros\_desordenados[cont2].pop()

                numeros\_desordenados\_filtro\_ordenar.append(numeros\_desordenados[cont2])

                cont2 = cont2 + 1

            else:

                numeros\_desordenados\_filtro\_ordenar.append(numeros\_desordenados[cont2])

                cont2 = cont2 + 1

        else:

            cont2 = cont2 + 1

    print(numeros\_desordenados\_filtro\_ordenar)

    #-------------------------------------------------------------ordeno la lista de numeros--------------------------------------------

    pattern = r"(\d+)"     # busca los numeros de la lista

    for i in listaSinSaltos:

        # busco lo que coincide con el pattern

        valor = re.findall(pattern, i)

    #########################   sirve para ordenar ya la lista #################################

        for recorrido in range(1,len(valor)):

            for posicion in range(len(valor)- recorrido):

                if int(valor[posicion]) > int(valor[posicion + 1]):

                    temp = valor[posicion]

                    valor[posicion]=valor[posicion + 1]

                    valor[posicion + 1 ] = temp

        numeros.append(valor)  # ingreso ya ordenados los numeros a una lista de lista

    print("")

    print("")

    print(numeros)   # (eliminar ) solo era para probar

    """cont3 = 0

    while cont3 < len(nombreOrdenar):

        if cont3 in posiciones:

            if "BUSCAR" in listaSinSaltos[cont3]:

                numeros[cont3].pop()

                numeros\_ordenados\_filtro\_ordenar.append(numeros[cont3])

                cont3 = cont3 + 1

            else:

                numeros\_ordenados\_filtro\_ordenar.append(numeros[cont3])

                cont3 = cont3 + 1

        else:

            cont3 = cont3 + 1

    print("")

    print("")

    print(numeros\_ordenados\_filtro\_ordenar)"""

    #------------------------------------  imprimo respuesta de desplegar listas ordenadas --------------------------------------------

    '''   esto si sirve   alsljasdjiajsildjijeijdioqjoiwjdoijqiojiodqjwiodjqojwdoijqowidjoqiwjdoijqwiojdojqwoidjioqwjd

    contador = 0

    while contador < len(nombresListas):

        print(str(nombresListas[contador][0]) +str(" ") + str(numeros\_desordenados[contador]) + "   Resultado de ordenar: " + str(numeros[contador]))

        print("")

        contador = contador + 1

    '''

    '''for i in nombreOrdenar:

                for j in i:

                    if j == "ORDENAR":

                        numeros\_desordenados.append(valor)

                    else:

                        continue'''

**0 DATOS=2,4,3,6,7,9,5,1,8 ORDENAR**

**1 LISTA=4,6,5,8,4,2 ORDENAR,BUSCAR 5**

**2 La=8,6,9,4,5,3,1,5 BUSCAR 7**

**3 DATOS=2,5,4,8,9,6,5 ORDENAR**

**4 LISTA=4,6,5,8,4,2 ORDENAR,BUSCAR 5**

**5 La=8,6,9,4,5,3,1,5 BUSCAR 7**

**6 La=8,6,9,4,5,3,1,5 BUSCAR 7**

**7 DATOS=2,5,4,8,9,6,5 ORDENAR**

**8 LISTA=55,100,5,3,9,2,90,4,86 ORDENAR,BUSCAR 5**

    #-------------------------------------------------------------ordeno la lista de numeros--------------------------------------------

    pattern = r"(\d+)"     # busca los numeros de la lista

    for i in listaSinSaltos:

        # busco lo que coincide con el pattern

        valor = re.findall(pattern, i)

    #########################   sirve para ordenar ya la lista #################################

        for recorrido in range(1,len(valor)):

            for posicion in range(len(valor)- recorrido):

                if int(valor[posicion]) > int(valor[posicion + 1]):

                    temp = valor[posicion]

                    valor[posicion]=valor[posicion + 1]

                    valor[posicion + 1 ] = temp

        numeros.append(valor)  # ingreso ya ordenados los numeros a una lista de lista

    print("")

    print("")

    #print(numeros)   # (eliminar ) solo era para probar

**Nueva lista fitrada números desordenados 2**