PUNTO 1

#*Desarrolle un algoritmo que permita leer los primeros n números enteros, los almacene en un vector y los imprima en pantalla de forma ascendente y descendente.*

numeros = int(input("Ingrese los n numeros para el vector "))

array1=[]

*for* i in range(0,numeros):

    array1.append(int(input('Ingrese un numero entero: ')))

ascendente = sorted(array1)

descendente = sorted(array1, reverse=True)

print(f'Los numeros del vector ordenados en forma ascendente son: {ascendente}')

print(f'Los numeros del vector ordenados en forma ascendente son: {descendente}')

PUNTO 2

#*Realizar un algoritmo que permita leer un vector de tamaño n, imprimirlo en pantalla*

#*de forma original y luego calcular el promedio de los elementos múltiplos de 3*

numeros = int(input("Ingrese los n numeros para el vector "))

array=[]

impares = []

*for* i in range(0,numeros):

    array.append(int(input('Ingrese un numero entero: ')))

*for* i in array:

*if* i% 3==0:

        impares.append(i)

print(impares)

PUNTO 3

#*3) Realizar un algoritmo que permita calcular la media y la mediana de los elementos de*

#*un vector de tamaño n. Se debe mostrar en pantalla los elementos del vector.*

*import* statistics as stats

numeros = int(input("Ingrese los n numeros para el vector "))

array=[]

*for* i in range(0,numeros):

    array.append(int(input('Ingrese un numero entero: ')))

print("La media del vector dado es: " + str(stats.mean(array)))

print("La mediana del vector dado es:" + str(stats.median(array)))

PUNTO 4

#*4) Realice un algoritmo que permita ordenar un vector de 8 números enteros de menor*

#*a mayor utilizando el método de la burbuja. Este vector debe contener números*

#*repetidos.*

#*Ejemplo: Vnums = {5,3,8,4,3,7,5,9}. Al final se debe mostrar en pantalla el vector original,*

#*el vector ordenado y la cantidad de elementos repetidos en el vector.*

def *bubbleSort*(array):

    length = len (array) - 1

*for* i in range (0,length):

*for* j in range(0, length):

*if* array[j]> array [j+1]:

                aux = array [j]

                array[j] =array [j+1]

                array[j+1] = aux

*return* array

scores = [70, 90, 0, 80, 60, 85]

print("Antes de ordenar: ")

print(scores)

print("Despues de ordenar: ")

print(bubbleSort (scores))

PUNTO 5

#*Realice un algoritmo que permita leer los elementos de una matriz de tamaño M x N*

#*y escriba en pantalla los elementos de esta elevados a la potencia 2.*

*import* numpy as np

a = np.array([ [1,1],

               [0,1] ])

result = np.linalg.matrix\_power(a, 2)

print(result)

PUNTO 6

#*Realice un algoritmo que permita buscar un elemento en una matriz de tamaño N x*

#*N y escriba en pantalla la posición de ese elemento y cuántas veces se encuentra*

#*repetido.*

*from* random *import* \*

fil=3

col=3

a=[[randint(1,100) *for* i in range (fil)]*for* j in range(col)]

#*for f in range(fil):*

#*for c in range(col):*

#*print (a[f][c], end= ' ')*

#*print()*

*for* f in a:

    print(f)

c = int (input('Digite columna a obtener'))

b=[]

*for* f in range (len(a)):

    b.append(a[f][c])

print(b)

PUNTO 7

#*7) Realice un algoritmo que permita calcular el promedio de los elementos de una matriz*

#*de tamaño N x N y escriba en pantalla todos los elementos de esta.*

matriz = [

    [7.8, 2.5, 10],

    [20, 15.2, 12],

    [89, 9, 6.77],

]

elementos = 0

sumatoria = 0

*for* fila in matriz:

*for* elemento in fila:

        sumatoria += elemento

        elementos += 1

promedio = sumatoria / elementos

print(

    f"La suma es {sumatoria} y el promedio es {promedio}, para la matriz que tiene {elementos} elementos"

)

PUNTO 8

#*8)Realice un algoritmo que permita imprimir los elementos de la diagonal principal o*

#*secundaria de una matriz cuadrada.*

fila = 3

columna = 3

matriz =[]

*for* i in range(fila):

    matriz.append([0]\*columna)

print (matriz)

#*#les da valor a la matriz 1a9*

contador=1

*for* i in range(fila):

*for* j in range(columna):

        matriz[i][j]=contador

        contador+=1

print(matriz)

#*# ordena la matriz*

*for* k in matriz:

    print (k)

#*#Diagonal principal*

a=[]

*for* i in range(fila):

    a.append(matriz[i][i])

print(a)

#*#Diagonal secundaria*

a=[]

*for* i in range(fila):

    a.append(matriz[i][(columna-1)-i])

print(a)