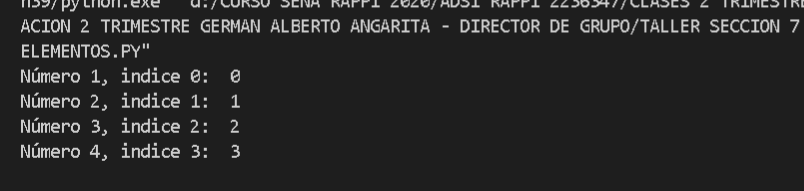
SECCIÓN 7 – 3 PROBLEMAS

1. Cree un programa que implemente un arreglo de cuatro elementos (números del 0 al 3) e imprima cada uno en líneas diferentes. DOS VERSIONES
2. """1. Cree un programa que implemente un arreglo de cuatro elementos
3. (números del 0 al 3) e imprima
4. cada uno en líneas diferentes."""
5. numeros = [0, 1, 2, 3]
6. print("Número 1, indice 0: " , numeros[0])
7. print("Número 2, indice 1: " , numeros[1])
8. print("Número 3, indice 2: " , numeros[2])
9. print("Número 4, indice 3: " , numeros[3])



## implementar un arreglo de 4 elementos (números de 0-3)

#  e imprimirlos cada uno en una línea diferente

j = [8, 4, 9, 1,]

k=len(j)

for i in range(k):

    print ("en la posicion",i,":",j[i])

2. Cree un programa que implemente un arreglo de 10 elementos (los dígitos), ordenados de menor a mayor e imprima cada uno de ellos en el mismo orden en el que se encuentran en el arreglo. Luego, invierta el orden de todos los elementos del arreglo, haciendo que el último pase a ser el primero, el penúltimo pase a ser el segundo, así sucesivamente, e imprima nuevamente cada elemento en el mismo orden en el que ahora se encuentran en el arreglo.

"""2. Cree un programa que implemente un arreglo de 10 elementos

(los dígitos), ordenados de menor a mayor e imprima cada uno de ellos

en el mismo orden en el que se encuentran en el arreglo. Luego,

invierta el orden de todos los elementos del arreglo, haciendo que el

último pase a ser el primero, el penúltimo pase a ser el segundo, así

sucesivamente, e imprima nuevamente cada elemento en el mismo orden en el

 que ahora se encuentran en el arreglo."""

print("ORDENADOS DE MENOR A MAYOR")

numeros = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

print("Número 1, indice 0: " , numeros[0])

print("Número 2, indice 1: " , numeros[1])

print("Número 3, indice 2: " , numeros[2])

print("Número 4, indice 3: " , numeros[3])

print("Número 5, indice 4: " , numeros[4])

print("Número 6, indice 5: " , numeros[5])

print("Número 7, indice 6: " , numeros[6])

print("Número 8, indice 7: " , numeros[7])

print("Número 9, indice 8: " , numeros[8])

print("Número 10, indice 9: " , numeros[9])

print("----#ORDENADOS DE MAYOR A MENOR---")

numeros = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

print("Número 10, indice 9: " , numeros[9])

print("Número 9, indice 8: " , numeros[8])

print("Número 8, indice 7: " , numeros[7])

print("Número 7, indice 6: " , numeros[6])

print("Número 6, indice 5: " , numeros[5])

print("Número 5, indice 4: " , numeros[4])

print("Número 4, indice 3: " , numeros[3])

print("Número 3, indice 2: " , numeros[2])

print("Número 2, indice 1: " , numeros[1])

print("Número 1, indice 0: " , numeros[0])

def mostrar(j):

    k=len(j)

    for i in range(k):

        print ("en la posicion",i,":",j[i])

    print ("--------------------------------------------------\n")

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

Dato = [8, 4, 6, 1, 7, 9, 5, 3, 0, 2]

print ("Datos en el arreglo inicial")

mostrar(Dato)

Dato.reverse()

print ("Datos Invertidos")

mostrar(Dato)

3. Dado el arreglo [1,2,3,4,5,6]

● Iterar por todos los elementos dentro del arreglo utilizando while y mostrarlos en pantalla.

● Iterar por todos los elementos dentro del arreglo utilizando el ciclo “for” y mostrarlos en pantalla.

● Mostrar todos los elementos dentro del arreglo sumándole uno a cada uno.

● Crear una copia del arreglo usando el ciclo “for” pero con todos los elementos incrementados en 1.

● Calcular el promedio de todos los elementos del arreglo

"""3. Dado el arreglo [1,2,3,4,5,6]

● Iterar por todos los elementos dentro del arreglo utilizando while y

mostrarlos en pantalla.

● Iterar por todos los elementos dentro del arreglo utilizando el ciclo “for”

y mostrarlos en pantalla.

● Mostrar todos los elementos dentro del arreglo sumándole uno a cada uno.

● Crear una copia del arreglo usando el ciclo “for” pero con todos los

elementos incrementados en 1.

● Calcular el promedio de todos los elementos del arreglo"""

print("----CICLO while-----")

i=0

arreglo\_while = [1, 2, 3, 4, 5, 6]

while i < len((arreglo\_while)):

    print("arreglo con ciclo while: ",arreglo\_while[i])

    i=i+1

print("----CICLO for-----")

arreglo\_for = [1, 2, 3, 4, 5, 6]

for i in range(len(arreglo\_for)):

    print("arreglo con ciclo for: ", arreglo\_for[i])

print("----AUMENTAR ELEMENTOS DE UN ARREGLO CON for-----")

arreglo\_aumento = [1, 2, 3, 4, 5, 6]

for i in range(len(arreglo\_aumento)):

    arreglo\_aumento[i] = arreglo\_aumento[i] + 1

print("arreglo despues de aumento: ", arreglo\_aumento)

print("----COPIA DEL ARREGLO CON for-----")

arreglo = [1, 2, 3, 4, 5, 6]

arreglo\_copia = arreglo.copy()

print("contenido antes de modificar arreglo", arreglo\_copia)

for i in range(len(arreglo)):

    arreglo[i] = arreglo[i] + 1

print("arreglo despues de aumento: ", arreglo)

print("----promedio de todos los elementos del arreglo-----")

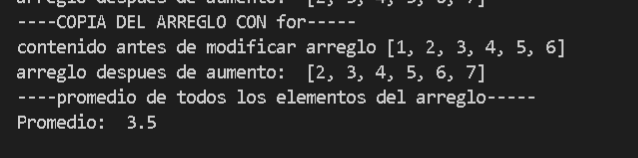
arreglo = [1, 2, 3, 4, 5, 6]

promedio=0

for i in range(len(arreglo)):

    promedio = promedio+arreglo[i]

print("Promedio: ", promedio/(len(arreglo)))



#LISTAS: conjunto de elementos preferiblemente del mismo

#tipo. Los elementos se encuentran ordenados por índices.

#es una estructura de datos.

#sintaxis

#se define el identificador, seguido del operador de

#asignación =, y entre corchetes [] se ponen los elementos

#separados por ,

#arreglo que contiene numeros

numeros = [13, 14, 15, 200.5, -100]

#arreglo(lista) vacia

lista\_vacia = []

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

#lectura de elementos

#se usa el identificador seguido de corchetes que contienen el

#índice del elemento que se quiere leer

print("elemento de indice 3: ", numeros[3])

print("----asignacion-sobreescritura----")

#para asignar o sobreescribir elementos se usa

#el operador de asignacion =

#aca sobreescribimos el elemento de indice 4

#y le asignamos un nuevo valor 10000

numeros[4] = 10000

print("verificamos si el elemento cambio: ", numeros[4])

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

print("Arreglo antes del borrado: ", numeros)

#para eliminar usamos el metodo pop evaluandolo en el indice

#del elemnto que queremos eliminar

#aca eliminamos el elemento 10000 cuyo indice es 4

numeros.pop(4)

print("Arreglo despues del borrado: ", numeros)

#aca eliminamos el elemento 14 cuyo indice es 1

numeros.pop(1)

print("Arreglo despues del segundo borrado: ", numeros)

print("-------agregacion de elementos--------")

#para agregar se usa el metodo append evaluado en el nuevo

#elemento que queremos agregar

#aca agragamos el elemento 25

print("Arreglo antes de agregar un elemento: ", numeros)

numeros.append(25)

print("Arreglo despues de agregar un elemento: ", numeros)

#que pasa cuando tratamos de acceder a un indice que no existe?

#aca tratamos de acceder al indice 20 de la lista numeros

#numeros[20]

print("---copia de arreglos-----")

#copia de arreglos. Para realizar una copia de arreglos independiente

#debemos usar el metodo copy.

#copia no independiente

edades = [10, 15, 90, 20]

edades\_copia = edades

print("contenido antes de modificar edades", edades\_copia)

edades.pop(2)

print("contenido despues de modificar edades", edades\_copia)

#copia independiente usamos copy()

edades\_2 = [100, 12, 19, 5]

edades\_copia\_2 = edades\_2.copy()

print("contenido antes de modificar edades\_2", edades\_copia\_2)

edades\_2.pop(2)

print("contenido despues de modificar edades", edades\_copia\_2)

print("-------operador len--------")

#operador len: nos sirve para saber al ongitud de p.e.

#arreglos, cadenas...

#operador len aplicado a una cadena, devuelve la longitud

#de la cadena, es decir, el numero de caracteres de la cadena

#sintaxis: se una len evaluado en el elemento del cual queremos

#saber su longitud

cadena = "abc  zx"

print("longitud de la cadena: ", len(cadena))

#len aplicado a LISTAS

#devuelve el numero de elementos de la lista

lista\_len = [True, "ADSI RAPPI", 10.6, 100]

print("longitud de la LISTA: ", len(lista\_len))

print("-----1 for aplicado a un arreglo------")

#imprimir cuatro nombres de personas que esten previamente

#guardadas

#SIN ARREGLOS

#aca tenemos una colecccion de nombres de

nombre1 = "Maria"

nombre2 = "Pedro"

nombre3 = "Raul"

nombre4 = "Luisa"

print(nombre1, nombre2, nombre3, nombre4)

#las colecciones las trabajamos con estructuras de datos

#CON ARREGLOS

arreglo\_nombres = ["Maria", "Pedro", "Raul", "Luisa"]

#iteracion sobre el arreglo

for nombre in arreglo\_nombres:

    print("nombres: ", nombre)

#iteracion con range, usando len

#nos permite trabajar con el indice

#si eliminamos un elemento el for no tira

#errores, igual si agregamos elementos

#EL LIMITE SUPERIOS EN LOS CICLOS CUANDO RECORREMOS

#ARREGLOS LO DEBEMOS TRABAJAR SUJETO A LA LONGITUD

#DEL ARREGLOS

arreglo\_nombres.pop(2)

for i in range(0, len(arreglo\_nombres)):

    print("elemento: ", arreglo\_nombres[i])

print("----AUMENTAR ELEMENTOS DE UN ARREGLO CON for-----")

#aumentar los elementos del siguiente arreglo en 1

#use for

arreglo\_aumento = [5, 6, 7, 8]

for i in range(len(arreglo\_aumento)):

    arreglo\_aumento[i] = arreglo\_aumento[i] + 1

print("arreglo despues de aumento: ", arreglo\_aumento)

SECCIÓN 8 – 2 PROBLEMAS

1. Dado el siguiente arreglo [ [1, 2, 3], [4, 5, 6], [7, 8, 9] ]:

● Iterar por todos los elementos del arreglo utilizando while y mostrarlos en pantalla.

● Iterar por todos los elementos del arreglo utilizando el ciclo “for” y mostrarlos en pantalla.

● Crear otro arreglo que sea idéntico al arreglo dado, pero con todos los elementos incrementados en 1.

Para esto debe usar ciclo for y a través de iteraciones ir formando el nuevo arreglo con sus elementos tal cual se piden.

● Usando el ciclo “for” Calcular el promedio de todos los elementos del arreglo. DOS VERSIONES

""" Dado el siguiente arreglo [ [1, 2, 3], [4, 5, 6], [7, 8, 9] ]:

 ● Iterar por todos los elementos del arreglo utilizando while y mostrarlos en pantalla.

 ● Iterar por todos los elementos del arreglo utilizando el ciclo “for” y mostrarlos

 en pantalla.

 ● Crear otro arreglo que sea idéntico al arreglo dado, pero con todos los elementos

 incrementados en 1.

 Para esto debe usar ciclo for y a través de iteraciones ir formando el nuevo

 arreglo con sus elementos tal  cual se piden.

 ● Usando el ciclo “for” Calcular el promedio de todos los elementos del arreglo"""

#matriz - arreglo... 3 filas x 3 columnas

A = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

#matriz - arreglo... ciclo while

x=0

while x < len(A):

    print(A[x])

    x=x+1

#matriz - arreglo... ciclo for

for i in range (len(A)):

    for j in range (len(A)):

        print(A[i][j])

print(A)

#matriz - arreglo... ciclo incrementado en 1

for i in range (len(A)):

    for j in range (len(A)):

        A[i][j]=A[i][j]+1

print(A) #INCREMENTADO EN 1

#matriz - arreglo... ciclo for y promedio

suma=0

for i in range (len(A)):

    for j in range (len(A)):

        suma=suma+A[i][j]

promedio=suma/(len(A)\*len(A[0]))

print(A)

print("El promedio es:" ,promedio)

def mostrar(j):

    i=0

    while (i<3):

        print ("en la posicion",j,",",i,":",Dato[j][i])

        i=i+1

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

i=0

con=0

Dato = [ [1, 2, 3], [4, 5, 6], [7, 8, 9] ]

copia\_dato = [ [0, 0, 0], [0, 0, 0], [0, 0, 0] ]

suma=0

print ("Datos en el arreglo usando 'while'")

mostrar(con)

con=con+1

mostrar(con)

con=con+1

mostrar(con)

print ("--------------------------------------------\n")

print ("Datos en el arreglo usando 'for'")

for i in range(len(Dato)):

    for j in range(len(Dato[i])):

        print(Dato[i][j], end=' ')

    print()

print ("--------------------------------------------\n")

print ("Datos en el arreglo usando 'for'+1 en cada valor")

for i in range(len(Dato)):

    for j in range(len(Dato[i])):

        copia\_dato[i][j]=Dato[i][j]+1

        print(copia\_dato[i][j], end=' ')

    print()

for i in range(len(Dato)):

    for j in range(len(Dato[i])):

        suma=Dato[i][j]+suma

promedio= suma/9

print ("El proedio de los datos es :",promedio)

2. Programe una función que permita encontrar el número menor contenido en una matriz recibida como parámetro.

'''

Función que permita encontrar el número menor contenido en una matriz recibida como parámetro.

'''

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

def vet\_matriz(b):

    a=b # se igualan para que tenga el mismas dimensiones

    val=len(b) # se conose la cantidad de vectores dentro de la matriz

    for i in range(val):

        b[i].sort() # se organiza de menor a mayor los datos de cada vector

        a[i]=b[i][0] # se agrega el primer dato de cada vector al vector a[]

    a.sort()    # se ordena de menor a mayor el vector a[]

    print("El nimimo valor en la matriz es:",a[0])

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

matriz = []

v\_col=[]

filas = int(input("Ingrese el numero de filas: \n"))

columnas = int(input("Ingrese el numero de columna: \n"))

for i in range(filas):

    for j in range(columnas):

        dat = int(input('Ingrese dato (%d,%d):'%(i,j)))

        v\_col.append(dat) # se toma cada dato y se ingresa en un vector

    matriz.append(v\_col) # se ingresa el vector completo en la matriz

vet\_matriz(matriz) # se invoca la funcion y se le pasa como parametro un vector de n x m elementos

SECCIÓN 9

1. Cree un programa que use una función de un parámetro, a la cual se le pase como argumento un arreglo, de tal manera que la función retorne la longitud(número de elementos) del arreglo.

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

vector = [1,2,3,4,5,6,7,8,9,0]

def  fun(c):

    a=len(c)

    return (a)

fun(vector)

longitud=fun(vector)

print ("El vector es:", vector)

print("La longitud del vector es",longitud)

2. Dado los siguientes arreglos arreglo1 = [ [“A”, “B”, “C”], [“D”, “E”, “F”], [“G”, “H”, “I”] ] y arreglo2 = [ [“J”, “K”, “L”], [“M”, “N”, “O”], [“P”, “Q”, “R”] ] , use ciclos para intercambiar los elementos de los arreglos.

'''

Dado los siguientes arreglos arreglo1 = [ [“A”, “B”, “C”], [“D”, “E”, “F”], [“G”, “H”, “I”] ] y

arreglo2 = [ [“J”, “K”, “L”],[“M”, “N”, “O”], [“P”, “Q”, “R”] ] ,

use ciclos para intercambiar los elementos de los arreglos.

'''

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

from copy import copy, deepcopy

Auxi=[]

Arreglo1 = [["A","B", "C"], ["D", "E", "F"], ["G", "H", "I"]]

Arreglo2 = [["J", "K", "L"],["M", "N", "O"], ["P", "Q", "R"]]

print ("Datos Arreglo 1 Y 2")

print (Arreglo1)

print (Arreglo2)

Auxi=deepcopy(Arreglo1)

#Auxi=Arreglo1

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

tama=len(Arreglo1)

for i in range(tama):

    for j in range(tama):

        Arreglo1[i][j]=Arreglo2[i][j]

        Arreglo2[i][j]=Auxi[i][j]

print ("Id de los arreglos")

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

print ("La id de AR1 =", id(Arreglo1), "->", Arreglo1)

print ("La id de AR2 =", id(Arreglo2), "->", Arreglo2)

print ("La id de Aux =", id(Auxi), "->", Auxi)

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

print ("Datos Arreglos intercambiados")

print(Arreglo1)

print(Arreglo2)

3. Cree un programa que implemente una función que reciba como argumento un arreglo y retorne la suma de los cuadrados de sus elementos, luego, imprima el resultado.

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

Vector= []

suma=0

def calculo(uno):

    suma=0

    num=len(uno)

    for i in range(num):

        suma=suma+(uno[i]\*uno[i])

    return suma

n\_datos = int(input("Ingrese el numero de datos: \n"))

for i in range(n\_datos):

    dat = int(input('Ingrese dato (%d):'%(i)))

    Vector.append(dat)

print (Vector)

sumas=calculo(Vector)

print("la suma de los cuadrados del los",n\_datos,"elementos es:",sumas)

SECCIÓN 10

Diseñe y desarrolle los siguientes programas de acuerdo a los requerimientos dados.

1. Cree un programa que pida al usuario el nombre de un producto existente en una tienda, luego, que le muestre el precio del producto. El usuario debe poder elegir de entre por lo menos cinco productos. Use diccionarios.

'''

programa que pida al usuario el nombre de un producto existente en una tienda,

luego, que le muestre

el precio del producto. El usuario debe poder

elegir de entre por lo menos cinco productos. Use diccionarios.

'''

Productos = {'Detergente':2500,'Golosinas':800,'Aceite':3800,'Pilas':2750,'Condones':500,}

print("Productos en la tienda son:\n")

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

for i in Productos:

    print (i)

print ("----------------------------------------\n")

dato = str(input("Ingrese el producto como aparece en el listado: \n"))

print ("el valor de",dato,"es de:",Productos[dato])

2. Cree un programa que almacene los documentos y nombres de diez usuarios, donde a cada documento corresponda su respectivo nombre. Use diccionarios. Imprima todos los nombres de los usuarios usando el ciclo for.

programa que almacene los documentos y nombres de diez usuarios, donde a cada documento

corresponda su respectivo nombre. Use diccionarios. Imprima todos los nombres

de los usuarios usando el ciclo

for.

'''

Nombres = {} #SE CREA UN DICCIONARIO VACIO

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

for i in range(10): # PARA CADA I=0 HASTA 10  NO INCLUYE EL DIEZ

    dato = int(input("Ingrese el Documento: \n"))

    dato1= str(input("Ingrese el Nombre: \n"))

    Nombres[dato]=dato1 # CLAVE - DATO /// DATO1 CONTENIDO DE LA CLAVE

    print (Nombres) ## se va mostrando el crecimiento y los datos del Diccionario

#print ("----------------------------------------\n")

print("los nombres ingresados son:")

for i in Nombres:

    print (Nombres[i]) # IMPRIME EL CONTENIDO DE LAS LLAVE - NOMBRES

print ("----------------------------------------\n")

3. Cree un programa que solicite al usuario el nombre de uno de los cinco continentes, luego, muestre cinco países del continente seleccionado por el usuario. Use diccionarios. Use el ciclo for.

'''

Cree un programa que solicite al usuario el nombre de

uno de los cinco continentes, luego muestre cinco paises del continente seleccionado

por el usuario. Use diccionarios. Use ciclo for

'''

Continentes = {'America':['Canada','Estados Unidos','Mexico','Colombia','Brasil'],'Asia':['China','Corea del Norte','Corea del sur','India','Japón'],'Africa':['Congo','Sudafrica','Camerun','Egipto','Tunez'], 'Europa':['Alemania','España','Francia','Inglaterra','Escocia'], 'Oceania':['Australia','Islas Mauricio','Nueva Zelanda','Islas Solomon','Islas Fiji']}

print("los Continentes ingresados son:")

for i in Continentes:

    print (Continentes[i])

print ("----------------------------------------\n")

conti = str(input("Ingrese el Continente: \n"))

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

print ("Paises del continente",conti)

for i in range(5):

    print (Continentes[conti][i])

print ("----------------------------------------\n")

4. Cree un programa que cumpla con los siguientes requerimientos: En una clínica, se requiere un programa donde el usuario pueda consultar el día de su cita mediante su documento. La cita debe tener día y fecha. Si el usuario consulta, el programa debe mostrarle sus nombres, seguidos del día y hora de su cita. Una vez hecha la consulta, el programa le debe mostrar al usuario un mensaje preguntándole si desea cambiar el día ó fecha de su cita, de ser así el programa debe realizar tal cambio y mostrarle al usuario que el cambio solicitado ha sido exitoso. Use diccionarios.

'''

programa que cumpla con los siguientes requerimientos: En una clínica, se requiere un programa

donde el usuario pueda consultar el día de su cita mediante su documento.

La cita debe tener día y fecha. Si el usuario consulta, el programa debe mostrarle

sus nombres, seguidos del día y hora de su cita.

Una vez hecha laconsulta, el programa le debe mostrar al usuario un mensaje preguntándole

si desea cambiar el día ó fecha de su cita, de ser así el programa debe realizar tal cambio

y mostrarle al usuario que el cambio solicitado ha sido exitoso. Use diccionarios.

'''

import sys

Pacientes = {1:['lola','01/30/2021','10:35'],

             2:['lole','02/24/2021','10:55'],

             3:['loli','03/24/2021','11:15'],

             4:['lolo','04/25/2021','11:35'],

             5:['lolu','05/26/2021','11:55'],

             }

Docu = int(input("Ingrese el Documento:"))

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

print ("Nombre del Paciente:",Pacientes[Docu][0])

print ("Fecha de la cita:",Pacientes[Docu][1])

print ("Hora:",Pacientes[Docu][2])

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

pre=str(input("¿Decea cambiar la fecha y hora de la cita S/N?: "))

if ((pre=='S') or (pre=='s')):

    Pacientes[Docu][1]=str(input("¿ingrese la nueva fecha dd/mm/aaaa: "))

    Pacientes[Docu][2]=str(input("¿ingrese la nueva Hora de cira hh:mm: "))

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

    print ("Nombre del Paciente:",Pacientes[Docu][0])

    print ("Fecha de la cita:",Pacientes[Docu][1])

    print ("Hora:",Pacientes[Docu][2])

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

    sys.exit()

else:

    sys.exit()

**TALLER EVALUATIVO LISTAS - DICCIONARIOS - PRÁCTICA**

**PYTHON 3**

1. Dado la siguiente lista:

lista = [1,2,3,4,5,6,7,8,9,10];

a) Usando ciclo for recorra la lista imprimiendo cada uno de sus elementos.

b) Usando ciclo while recorra la lista imprimiendo cada uno de sus elementos.

c) Usando ciclo for recorra la lista imprimiendo cada uno de sus elementos aumentados en 3.

d) Usando ciclo while recorra la lista imprimiendo cada uno de sus elementos aumentados en 3.

e) Usando ciclo for recorra la lista imprimiendo únicamente los números impares.

f) Usando ciclo while recorra la lista imprimiendo únicamente los números pares.

'''

1. Dado la siguiente lista = [1,2,3,4,5,6,7,8,9,10];

a) Usando ciclo for recorra la lista imprimiendo cada uno de sus elementos.

b) Usando ciclo while recorra la lista imprimiendo cada uno de sus elementos.

c) Usando ciclo for recorra la lista imprimiendo cada uno de sus elementos aumentados en 3.

d) Usando ciclo while recorra la lista imprimiendo cada uno de sus elementos aumentados en 3.

e) Usando ciclo for recorra la lista imprimiendo únicamente los números impares.

f) Usando ciclo while recorra la lista imprimiendo únicamente los números pares.

 '''

Lista = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

copia\_Lista=[0,0,0,0,0,0,0,0,0,0]

aumento=3

print ("--------------------------------------------\n")

print ("Datos en el arreglo usando 'for'")

print(Lista)

for i in range(len(Lista)):

    print("Dato",i,"de la lista: ",Lista[i])

print ("--------------------------------------------\n")

print ("Datos en el arreglo usando 'While'")

i=0

while (i<len(Lista)):

    print ("En la posicion",i,"El dato es:",Lista[i])

    i=i+1

print ("--------------------------------------------\n")

i=0

print ("Datos en el arreglo usando 'for' + 3")

for i in range(len(Lista)):

    copia\_Lista[i]=Lista[i]+3

    print("Dato",i,"de la lista aumentada en 3: ",copia\_Lista[i])

print ("--------------------------------------------\n")

i=0

print ("Datos en el arreglo usando 'While' + 3")

while (i<len(Lista)):

    print ("En la posicion",i,"El dato aumentado en 3 es:",copia\_Lista[i])

    i=i+1

print ("--------------------------------------------\n")

print ("Datos impares en el arreglo usando 'for'")

print (Lista)

i=0

for i in range(0,len(Lista),2):

    print("Dato",i,"de la lista: ",Lista[i])

print ("--------------------------------------------\n")

i=0

print ("Datos pares en el arreglo usando 'While'")

print (Lista)

while (i<len(Lista)):

    if ((Lista[i]%2)==0):

        print ("En la posicion",i,"El dato par es:",Lista[i])

    i=i+1

2. Dado el siguiente diccionario:

persona = {“edad”: 20, “peso”: 75, “nombres”: “Pedro”, “apellidos”: "Pérez

Pérez"};

Consulte por los métodos keys() y values() en la página:

https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion3/tipo\_diccionarios.html

y usando ciclo for y el método keys(), imprima todas las claves del diccionario dado, también, usando ciclo for y el método values() imprima todos los valores del diccionario. Luego, sobre el mismo diccionario realice las siguientes operaciones:

● Cambie el valor de la clave peso por 77

● Cambie el valor de la clave edad por 21

● Elimine la clave apellidos usando método pop

● Imprim el valor de la clave peso usando el método get

'''

persona = {“edad”: 20, “peso”: 75, “nombres”: “Pedro”, “apellidos”: "Pérez

Pérez"};

Usando ciclo for y el método keys(), imprima todas las claves del diccionario dado, también, usando ciclo

for y el método values() imprima todos los valores del diccionario. Luego, sobre el mismo diccionario

realice las siguientes operaciones:

● Cambie el valor de la clave peso por 77

● Cambie el valor de la clave edad por 21

● Elimine la clave apellidos usando método pop

● Imprim el valor de la clave peso usando el método get

'''

Persona = {"edad":20, "peso":75, "nombres":"Pedro", "apellidos":"Pérez Pérez"}

print("Claves del diccionario:\n")

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

for key in Persona.keys():

    print ("Llave",key)

print ("----------------------------------------\n")

for value in Persona.values():

    print("Valor: ", value)

Persona["peso"]=77

Persona["edad"]=21

Persona.pop("apellidos")

print("llave peso: ", Persona.get("peso"))

print("Cambios:",Persona)

3. Dado el siguiente arreglo [ “Colombia”, “es”, “un”, “país”, “extraordinario” ], imprima en consola el mensaje “Colombia es un país extraordinario” usando el método join el cual puede consultar en el siguiente link: <https://blog.carreralinux.com.ar/2017/07/uso-split-y-join-python/>.

# Dado el siguiente arreglo [ “Colombia”, “es”, “un”, “país”, “extraordinario” ], imprima en consola el

# mensaje “Colombia es un país extraordinario” usando el método join

# join se utiliza para convertir una lista en una cadena, separada por un caracter deseado

mensaje = [ "Colombia", "es", "un", "país","extraordinario" ]

mensaje\_join = "@".join(mensaje)

print(mensaje\_join)

#Colombia@es@un@país@extraordinario

4. Dada la siguiente cadena “Programar es algo genial, todos deberían intentarlo, atrévete!” genere el siguiente arreglo [“Programar es algo genial”, “todos deberían intentarlo”, “atrévete!”] usando el método split el cual puede consultar en el siguiente link:

<https://blog.carreralinux.com.ar/2017/07/uso-split-y-join-python/>..

'''Dada la siguiente cadena “Programar es algo genial, todos deberían intentarlo, atrévete!” genere el

siguiente arreglo [“Programar es algo genial”, “todos deberían intentarlo”, “atrévete!”] usando el método

split el cual puede consultar en el siguiente link:

https://blog.carreralinux.com.ar/2017/07/uso-split-y-join-python/..'''

# SPLIT convierte una cadena a una lista separadas por un caracter deseado

mensaje = "Programar es algo genial, todos deberían intentarlo, atrévete!"

mensaje\_split = mensaje.split(",")

print(mensaje\_split)

#['Programar es algo genial', ' todos deberían intentarlo', ' atrévete!']

5. Dado el siguiente arreglo [1, 2, 3, 4, 500, 420, -100] elimine el elemento de índice 4 y a continuación, elimine el último elemento de la lista usando el método pop(si al método pop no se le pasa argumento, eliminará el último elemento de la lista) el cual puede consultar aquí:

https://www.geeksforgeeks.org/python-list-pop/ . Agregue el elemento 520 usando el método append.

Finalmente, haga una copia “independiente del arreglo” usando el método copy el cual puede consultar

acá: <https://www.programiz.com/python-programming/methods/list/copy>

'''. Dado el siguiente arreglo [1, 2, 3, 4, 500, 420, -100] elimine el elemento de índice 4 y a continuación,

elimine el último elemento de la lista usando el método pop(si al método pop no se le pasa argumento,

eliminará el último elemento de la lista) el cual puede consultar aquí:

https://www.geeksforgeeks.org/python-list-pop/ . Agregue el elemento 520 usando el método append.

Finalmente, haga una copia “independiente del arreglo” usando el método copy el cual puede consultar

acá: https://www.programiz.com/python-programming/methods/list/copy'''

from copy import copy, deepcopy

print("lista original")

lista = [1, 2, 3, 4, 500, 420, -100]

print (lista)

print("borra el elemento 5 con el valor 500 y el indice 4, muestra como quedaria la lista")

print(lista.pop(4), lista)

print("------------------------------------\n")

print("borra el ultimo elemento  y muestra como quedaria la lista")

print(lista.pop(), lista)

print("agregamos a la lista el elemento con el valor 520")

lista.append(520)

print(lista)

print("------------------------------------\n")

print("se genera la copia en diferente posicion en memoria - lista es la original")

lista\_copia = deepcopy(lista) # se genera una copia profunda de lista

print("a la copia se le añade un elemento - lista\_copia es la copia con el nuevo elemento en diferente posicion en memoria")

lista\_copia.append("ARQSOFT")

print(lista\_copia)

print ("La id de lista =", id(lista), "->", lista)

print ("La id de lista\_copia =", id(lista\_copia), "->", lista\_copia)

6. En una empresa se requiere un sistema de información que registre sus nuevos usuarios. Cada vez que un usuario se registra debe introducir los siguientes datos: documento, nombres, apellidos, edad, peso, estatura. Cree un sistema de información que reciba los datos de registro de cada cliente por teclado, luego de realizado el registro el sistema debe mostrar al usuario el mensaje “Registro exitoso”. Si el usuario desea consultar su registro debe poder hacerlo a través de su documento.

# En una empresa se requiere un sistema de información que registre sus nuevos usuarios.

# Cada vez que un usuario se registra debe introducir los siguientes datos:

# documento, nombres, apellidos, edad, peso, estatura.

# Cree un sistema de información que reciba los datos de registro de cada cliente por

# teclado, luego de realizado el registro el sistema debe mostrar al usuario el mensaje

# “Registro exitoso”. Si el usuario desea consultar su registro

# debe poder hacerlo a través de su documento

import sys

def agregar():

    usuarios = {}

    nuevo\_usuario="s"

    while nuevo\_usuario=="s":

        documento=int(input("Ingrese el documento: "))

        nombre=input("Ingrese el nombre: ")

        apellido=input("Ingrese el apellido: ")

        edad=input("Ingrese su edad: ")

        peso=input("Ingrese su peso: ")

        usuarios[documento]=(nombre,apellido,edad,peso)

        nuevo\_usuario=nuevo\_usuario=input("Desea agregar otro usuario[s/n]? ")

    print("REGISTRO EXITOSO")

    return usuarios

def imprimir(usuarios):

    print("Usuarios Registrados en el Sistema")

    for documento in usuarios:

       print("Documento:",documento,"- " "Nombres:",usuarios[documento][0], "- Apellidos:" ,usuarios[documento][1],"- Edad:",usuarios[documento][2],"- Peso:", usuarios[documento][3])

def consulta(usuarios):

    documento=int(input("Ingrese el documento del usuario a consultar: "))

    if documento in usuarios:

         print("Los datos del usuario con el documento:",documento,"- Nombres:",usuarios[documento][0], "- Apellidos:" ,usuarios[documento][1],"- Edad:",usuarios[documento][2],"- Peso:", usuarios[documento][3])

usuarios=agregar()

imprimir(usuarios)

consulta(usuarios)