

SEGUNDO PARCIAL
Faryd walteros (fwalteros@unal.edu.co)

1. Desarrollar un programa que determine si en una lista hay elementos repetidos.

```
##FIRST POINT
##verificar si la lista cuenta con elemtos repetidos
def x(lista):
    for i in range(len(lista)):
        for j in range(i + 1, len(lista)):
            if lista[i] == lista[j]:
                return True
    return False

lista = [1,2,3,4,5,1]

if x(lista):
    print("La lista tiene elementos repetidos")
else:
    print("Esta lista no tiene elementos repetidos")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
& C:/Users/Estudia
La lista tiene elementos repetidos
PS C:\Users\Estudiante\ia_practica> []
```

2. Desarrollar un programa que determine si en una lista se encuentra una cadena de caracteres con dos o más vocales. si la cadena existe debe imprimirla y si no existe debe imprimir "no existe".

```
Desarrollar un programa que determine si en una lista se encuentra una cadena de caracteres con dos o mas vocales.
def x(cadena):
    vocales = "aeiouAEIOU"
    contador = 0

    for letra in cadena :
        if letra in vocales :
            contador +=1
        if contador >=2 :
            return True
    return False

def y(lista):
    for elemento in lista :
        if type(elemento) == str and x(elemento):
            print(elemento)
            return True
    return False

lista= ["gato" ,"perro" , "carro" , "raton"]
# y(lista):
print (("esta cadena tiene dos o mas vocales"))
else:
    print("NO EXISTE")
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

gato
esta cadena tiene dos o mas vocales
PS C:\Users\Estudiante\ia_practica> & C:/Users/Estudiante/App
gato
esta cadena tiene dos o mas vocales
PS C:\Users\Estudiante\ia_practica> |
```

3. desarrollar un programa que dadas dos listas determine que elementos tiene la primera lista que la segunda no tenga

```
##THIRD POINT
##desarrollar un programa que dadas dos listas determine que elementos tiene la primera lista que la segunda no
def x(lista1,lista2):
    resultado =[]
    for elemento in lista1:
        if elemento not in lista2:
            resultado.append(elemento)
    return resultado

lista1 =[1,2,3,4,5,6]
lista2 = [1,2,4,5,8,9]

diferencia = x(lista1,lista2)
if diferencia:
    print("Estos elementos no estan en la segunda",diferencia)
else:
    print("Todos los elementos estan en las dos listas")
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

esta cadena tiene dos o mas vocales
PS C:\Users\Estudiante\ia_practica> & C:/Users/
Estos elementos no estan en la segunda [3, 6]
PS C:\Users\Estudiante\ia_practica> & C:/Users/
Estos elementos no estan en la segunda [3, 6]
PS C:\Users\Estudiante\ia_practica> |
```

4. Desarrollar un algoritmo que calcule el promedio de un arreglo de reales.

```
#FOURTH POINT
#Desarrollar un algoritmo que calcule el promedio de un arreglo de reales.
def x(arreglo):
    if len(arreglo) == 0:
        return 0
    suma=0
    for numero in arreglo:
        suma+= numero
    return suma /len(arreglo)

numeros =[2.5 , 4.2 , 6.5 , 8.1]
promedio= x(numeros)
print("El promedio es : ",promedio)
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Estos elementos no estan en la segunda [3, 6]
PS C:\Users\Estudiante\ia_practica> & C:/Users/
Estos elementos no estan en la segunda [3, 6]
PS C:\Users\Estudiante\ia_practica> & C:/Users/
El promedio es : 5.324999999999999
PS C:\Users\Estudiante\ia_practica>
```

5. desarrollar un algoritmo que determine la mediana de un arreglo de enteros. La mediana es el numero que queda en la mitad del arreglo después de ser ordenado

```
##FIFTH POINT
##desarrollar un algoritmo que determine la mediana de un arreglo de enteros. La mediana es el numero que queda en la mitad del arreglo después de ser ordenado
def x(lista):
    n = len(lista)
    for i in range(n):
        for j in range(0, n-i-1):
            if lista[j] > lista[j+1]: ##aca ordenamos las listas por si estan desordenadas
                lista[j], lista[j+1] = lista[j+1], lista[j]
    return lista

def y(arreglo):
    a = x(arreglo)
    n = len(a)
    medio = n//2

    if n % 2 == 1: ##por si llega a ser impar
        return a[medio]
    else:
        return (a[medio-1] + a[medio]) / 2

numeros = [7, 2, 5, 10, 9]
mediana = y(numeros)
print("La mediana es ", mediana)
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Estos elementos no estan en la segunda [3, 6]
● PS C:\Users\Estudiante\ia_practica> & C:/Users/Estudiante/
El promedio es : 5.324999999999999
● PS C:\Users\Estudiante\ia_practica> & C:/Users/Estudiante/
La mediana es 5
○ PS C:\Users\Estudiante\ia_practica>
```