

Parcial #2

Universidad Nacional de Colombia
Juan Pablo Gómez Sequeda

Punto 1

Desarrollar un programa que identifique si en una lista se encuentran dos apariciones del mismo elemento

Código completo:

```
lista = [2,3,2,4,5,1,6,0,8,2]
lista2 = [2,3,4,5,6,7,8,9,10]
def repeticion(x):
    for i in x:
        if x.count(i) >= 2:
            print("Se encuentra un elemento repetido que es: ",i)
        else:
            print("no se encuentran dos apariciones del mismo elemento")

print(repeticion(lista))
print(repeticion(lista2))
```

Explicación

Se toma como ejemplo dos listas en las cuales en una de ellas se encuentra repetido un parámetro.

```
lista = [2,3,2,4,5,1,6,0,8,2]
lista2 = [2,3,4,5,6,7,8,9,10]
```

luego se crea una función donde con el elemento count se determina si un elemento se encuentra repetido 2 o más veces para así generar un print que afirme que se encuentra un elemento repetido y posteriormente lo muestre, si no se encuentra ningún elemento repetido, se imprime que no se encuentran dos apariciones del mismo elemento.

```
def repeticion(x):
    for i in x:
        if x.count(i) >= 2:
            print("Se encuentra un elemento repetido que es: ",i)
        else:
            print("no se encuentran dos apariciones del mismo elemento")

print(repeticion(lista))
print(repeticion(lista2))
```

luego se evalúa la función con las dos listas de ejemplo y en la terminal da como resultado:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
& C:/Users/Estudi

#2/punto_1.py
Se encuentra un elemento repetido que es: 2
Se encuentra un elemento repetido que es: 2
Se encuentra un elemento repetido que es: 2
no se encuentran dos apariciones del mismo elemento
None
no se encuentran dos apariciones del mismo elemento
None
PS C:\Users\Estudiante\Documents\parcial#2> |
```

Punto 2

Desarrollar un programa que identifique si en una lista se identifican cadenas de dos o mas vocales e imprimirlas

Codigo completo:

```
lista = ["hola", "buenas", "tardes", "adios"]

def identificar_cade(x):
    for i in x:
        voc = 0
        for j in i:
            if j in "aeiou":
                voc += 1
                if voc >= 2:
                    print(i)
                    break
            else:
                voc = 0

print(identificar_cade(lista))
```

Explicación

```
def identificar_cade(x):
    for i in x:
        voc = 0
```

Se crea una lista de valores floats
luego se crea una función donde se evalúa la cantidad de vocales empezando por 0 Esto debido a que vamos a identificar si hay dos o más vocales

```
        for j in i:
            if j in "aeiou":
                voc += 1
                if voc >= 2:
                    print(i)
                    break
            else:
                voc = 0
```

Posteriormente se evalúa cada elemento j dentro del elemento i para así identificar si se encuentran dos o más vocales seguidas en el elemento i. pedimos que si se encuentran se sumen e impriman y si no se deje el contador en cero 0

Terminal

```
#2/punto_2.py
buenas
adios
None
PS C:\Users\Estudiante\Documents\parcial#2>
```

Punto 3

Desarrollar un programa que dadas dos listas identifique que elementos faltan en una que la otra posea

```
lista1 = [2,3,4,5,6,7]
lista2 = [5,6,7,0,1,10]

def repeticion():
    for i in lista1:
        if i not in lista2:
            print(i)

print (repeticion())
```

Explicación:

Se crean dos listas con elementos diferentes y algunos en común
se crea la función repeticion para identificar quienes están en los dos elementos y se crea un if not para identificar si se encuentra en la lista #2

Terminal

```
#2/punto_3.py
2
3
4
None
```

Punto 4

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

Codigo completo

```

arreglo = [1.2,2.4,5.6,4.6]

def promedio():
    suma = 0
    for i in arreglo:
        suma += i
    promedio = suma / len(arreglo)
    return promedio

print(promedio())

```

Explicación

se crea una lista de números reales

luego se crea una función para determinar el promedio haciendo una suma de cada elemento sobre la longitud con la función len retomando el promedio

Terminal

```

PS C:\Users\Estudiante\Docum
#2/punto_4.py
3.4499999999999997

```

Punto 5

determinar la mediana de un arreglo de enteros

```

arreglo = [12,11,3,5,6,43,7,1,32]

def mediana(x):
    x.sort()
    if len(x) % 2 == 0:
        calculo = (x[len(arreglo)//2] + x[len(arreglo)//2 - 1]) / 2
    else:
        calculo = x[len(arreglo)//2]
    return calculo

print(mediana(arreglo))

```

Explicacion

se genera el arreglo

luego la funcion que se ordena con el elemento sort de menor a mayor

se evalua la longitud con el elemento len y si esta es par

se calcula la mediana con la formula

si no es par se calcula con la otra formula

Terminal

```
#2/punto_5.py  
7
```