

## Actividad número 3

Niedferon Vargas

### Problemas

I

1. Desarrollar un programa que determine si en una lista no existen elementos repetidos.
2. Desarrollar un programa que determine si un elemento de una lista es una cadena palíndrome. Si la cadena existe debe imprimirla y si no existe debe imprimir 'No existe'.
3. 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'.
4. Desarrollar un programa que determine si una lista es palíndrome. Una lista es palíndrome si el elemento en la posición  $i$  es el mismo de la posición  $n - 1 - i$  con  $n$  la longitud de la lista.

1.

```
1  # Punto número uno
2  def elementos_repetidos(lista):
3      vistos = []
4      for elemento in lista:
5          if elemento in vistos:
6              return True
7          vistos.append(elemento)
8      return False
9
10 lista = [4, 5, 6, 7, 8]
11 if elementos_repetidos(lista):
12     print("Hay elementos repetidos en la lista.")
13 else:
14     print("No hay elementos repetidos en la lista.")
15
16 lista_con_repetidos = [4, 5, 6, 7, 8, 7]
17 if elementos_repetidos(lista_con_repetidos):
18     print("Hay elementos repetidos en la lista.")
19 else:
20     print("No hay elementos repetidos en la lista.")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\niedf\OneDrive\Documents\Programación> & C:/Users/nie
Actividad3.py
No hay elementos repetidos en la lista.
Hay elementos repetidos en la lista.
```

2.

```
1  ## Problema número 2
2
3  def palindromo(cadena):
4      return cadena == cadena[::-1]
5
6  def encontrar_palindromos(lista):
7      palindromos = [palabra for palabra in lista if palindromo(palabra)]
8      if palindromos:
9          print("Palíndromos encontrados:", ", ".join(palindromos))
10     else:
11         print("No existe")
12
13     lista_palabras = ["aerea", "aguacate", "radar", "lechuga", "rallar"]
14     encontrar_palindromos(lista_palabras)
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\niedf\OneDrive\Documents\Programación> & C:/Users/niedf/AppData/Local/M  
nts/Programación/Actividad3  
Palíndromos encontrados: aerea, radar, rallar

Actividad3 > ...

```
1  ## Problema número 2
2
3  def palindromo(cadena):
4      return cadena == cadena[::-1]
5
6  def encontrar_palindromos(lista):
7      palindromos = [palabra for palabra in lista if palindromo(palabra)]
8      if palindromos:
9          print("Palíndromos encontrados:", ", ".join(palindromos))
10     else:
11         print("No existe")
12
13     lista_palabras = ["aguacate", "lechuga", "hamburguesas"]
14     encontrar_palindromos(lista_palabras)
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\niedf\OneDrive\Documents\Programación> & C:/Users/niedf/AppData/Local/M  
nts/Programación/Actividad3  
No existe  
PS C:\Users\niedf\OneDrive\Documents\Programación>

### 3.

```

1  ## Problema número 3
2  def tiene_mas_de_dos_vocales(cadena):
3      vocales = "aeiouAEIOU"
4      contador_vocales = 0
5
6      for caracter in cadena:
7          if caracter in vocales:
8              contador_vocales += 1
9          if contador_vocales >= 2:
10             return True
11
12     return False
13
14     lista3 = ["murcielago", "sol", "tomate", "cielo", "mar", "paz"]
15
16     encontrado = False
17
18     for elemento in lista3:
19         if tiene_mas_de_dos_vocales(elemento):
20             print(f"Cadena con dos o más vocales encontrada: {elemento}")
21             encontrado = True
22             break
23
24     if not encontrado:
25         print("No existe")

```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```

PS C:\Users\niedf\OneDrive\Documents\Programación> & C:/Users/niedf/AppData/Local/Microsoft/Programación/Actividad3
Cadena con dos o más vocales encontrada: murcielago
PS C:\Users\niedf\OneDrive\Documents\Programación>

```

```

14 lista3 = ["sol", "mar", "paz"]
15
16 encontrado = False
17
18 for elemento in lista3:
19     if tiene_mas_de_dos_vocales(elemento):
20         print(f"Cadena con dos o más vocales encontrada: {elemento}")
21         encontrado = True
22         break
23
24 if not encontrado:
25     print("No existe")

```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```

PS C:\Users\niedf\OneDrive\Documents\Programación> & C:/Users/niedf/AppData/Local/
nts/Programación/Actividad3
No existe
PS C:\Users\niedf\OneDrive\Documents\Programación>

```

4.

```
Actividad3 > es_palindromo
1  ## Problema 4
2
3  def es_palindromo(lista4):
4      n = len(lista4)
5      lista_invertida = []
6      for i in range(n - 1, -1, -1):
7          lista_invertida.append(lista4[i])
8      return lista4 == lista_invertida
9
10 lista5 = [1, 2, 3, 2, 1]
11 otralista = [1, 2, 3, 2]
12
13 if es_palindromo(lista5):
14     print("La lista es un palíndromo.")
15 else:
16     print("La lista no es un palíndromo.")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\niedf\OneDrive\Documents\Programación> & C:/Programación/Actividad3  
La lista es un palíndromo.  
PS C:\Users\niedf\OneDrive\Documents\Programación>

```
Actividad3 > ...
1  ## Problema 4
2
3  def es_palindromo(lista4):
4      n = len(lista4)
5      lista_invertida = []
6      for i in range(n - 1, -1, -1):
7          lista_invertida.append(lista4[i])
8      return lista4 == lista_invertida
9
10 lista5 = [1, 2, 3, 2, 1]
11 otralista = [1, 2, 3, 2]
12
13 if es_palindromo(otralista):
14     print("La lista es un palíndromo.")
15 else:
16     print("La lista no es un palíndromo.")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\niedf\OneDrive\Documents\Programación> & C:/Programación/Actividad3  
La lista no es un palíndromo.  
PS C:\Users\niedf\OneDrive\Documents\Programación>