

Universidad Americana

Facultad de Ingeniería y Arquitectura



Algoritmos y Estructuras de Datos, Grupo 4

Manejo de Funciones, Clases y Paquetes

Realizado por:

Joaquín Alberto Pérez Zúñiga

Docente:

MSc. César Marín López

1. Ejercicio 1:

```
1  """
2  Implementar un método que recibe una lista de enteros L y un número
3  entero n de forma que modifique la lista mediante el borrado de
4  todos los elementos de la lista que tengan este valor.
5  """
6
7  from linked_list import LinkedList
8
9
10 def main() -> None:
11     """
12     Ejecución del programa.
13     """
14
15     lista = LinkedList()
16     print()
17
18     while True:
19         elemento = input(
20             "Ingrese el elemento que desea agregar a la lista (dejar vacío para terminar): "
21         )
22
23         if elemento == "":
24             break
25         if not elemento.isnumeric():
26             input("\n¡Error! Elemento ingresado no es un número, intente nuevamente...")
27             continue
28
29         lista.add_at_end(int(elemento))
30
31     while True:
32         n = input("Ingrese el valor a eliminar de la lista: ")
33         if not n.isnumeric():
34             input("\n¡Error! String ingresado no es un número, intente nuevamente...")
35             continue
36         break
37
38     # delete() esta implementado para eliminar
39     # todos los nodos con el valor especificado
40     lista.delete(int(n))
41     print("\n-----\n")
42     print(f"{lista}\n")
43
44
45 if __name__ == "__main__":
46     main()
47
```

2. Ejercicio 2:

```
1  """
2  Construir un método cantVocales que determine la
3  cantidad de vocales almacenadas en una lista de caracteres.
4  """
5
6  from linked_list import LinkedList
7
8
9  def encontrar_vocales(lista: LinkedList) -> int:
10     """
11     Encuentra la cantidad de vocales en una lista enlazada de caracteres.
12     """
13
14     cantidad = 0
15     current = lista.head
16
17     while current is not None:
18         if current.data in ("a", "e", "i", "o", "u"):
19             cantidad += 1
20         current = current.next
21
22     return cantidad
23
24
25 def main() -> None:
26     """
27     Ejecución del programa.
28     """
29
30     lista = LinkedList()
31     print()
32
33     while True:
34         elemento = input(
35             "Ingrese el caracter que desea agregar a la lista (dejar vacío para terminar): "
36         )
37
38         if elemento == "":
39             break
40
41         lista.add_at_end(elemento)
42
43     print("\n-----\n")
44     print(f"Cantidad de vocales en lista: {encontrar_vocales(lista)}\n")
45
46
47 if __name__ == "__main__":
48     main()
49
```

3. Ejercicio 3:

```
1  """
2  Construir un método imprime_inverso() que imprima los elementos de una lista
3  enlazada de enteros en orden inverso a partir de una posición p.
4  """
5
6  from linked_list import LinkedList
7
8
9  def imprimir_inverso(lista: LinkedList, p: int) -> None:
10     """
11     Imprime los elementos de una lista enlazada de
12     enteros en orden inverso a partir de una posición p.
13     """
14
15     i = 0
16     inversa = []
17     current = lista.head
18
19     while current.next is not None: # type: ignore
20         inversa.append(current.data) # type: ignore
21         current = current.next # type: ignore
22         i += 1
23
24     if p == i - 1:
25         break
26
27     print(f"{inversa[::-1]}\n")
28
29
30 def main() -> None:
31     """
32     Ejecución del programa.
33     """
34
35     lista = LinkedList()
36     print()
37
38     while True:
39         elemento = input(
40             "Ingrese el elemento que desea agregar a la lista (dejar vacío para terminar): "
41         )
42
43         if elemento == "":
44             break
45         if not elemento.isnumeric():
46             input("\n¡Error! Elemento ingresado no es un número, intente nuevamente...")
47             continue
48
49         lista.add_at_end(elemento)
50
51     while True:
52         p = input("Ingrese el índice a partir para imprimir los elementos en orden inverso: ")
53         if not p.isnumeric():
54             input("\n¡Error! String ingresado no es un número, intente nuevamente...")
55             continue
56         break
57
58     print("\n-----\n")
59     imprimir_inverso(lista, int(p))
60
61
62 if __name__ == "__main__":
63     main()
64
```