**ASSIGNMENT N0 02**

***SUBMITTED BY: - Muhammad Umar***

***R0LL N0: - 20011556-037***

***SECTI0N: - BS-IT (B)***

*SUBJECT: -* DATA STRUCTURE AND ALGORITHMS

***SEMESTER: - 7TH***

class Node:

def \_init\_(self, data):

self.data = data

self.next = None

class LinkedList:

def \_init\_(self):

self.head = None

def display(self):

current = self.head

while current:

print(current.data, end=" -> ")

current = current.next

print("None")

def insert\_at\_position(self, data, position):

new\_node = Node(data)

if position == 1:

new\_node.next = self.head

self.head = new\_node

return

current = self.head

for \_ in range(position - 2):

if current is None:

print("Invalid position")

return

current = current.next

new\_node.next = current.next

current.next = new\_node

def delete\_from\_beginning(self):

if self.head:

self.head = self.head.next

else:

print("List is empty")

def delete\_from\_end(self):

if not self.head or not self.head.next:

self.head = None

else:

current = self.head

while current.next.next:

current = current.next

current.next = None

def delete\_from\_position(self, position):

if position == 1:

self.head = self.head.next

return

current = self.head

for \_ in range(position - 2):

if current is None or current.next is None:

print("Invalid position")

return

current = current.next

current.next = current.next.next

def search(self, data):

current = self.head

position = 1

while current:

if current.data == data:

return position

current = current.next

position += 1

return -1

def update\_at\_position(self, data, position):

current = self.head

for \_ in range(position - 1):

if current is None:

print("Invalid position")

return

current = current.next

current.data = data

# Example usage:

linked\_list = LinkedList()

linked\_list.insert\_at\_position(1, 1)

linked\_list.insert\_at\_position(2, 2)

linked\_list.insert\_at\_position(3, 3)

linked\_list.display()

linked\_list.update\_at\_position(4, 2)

linked\_list.display()

linked\_list.delete\_from\_beginning()

linked\_list.display()

linked\_list.delete\_from\_end()

linked\_list.display()

linked\_list.delete\_from\_position(2)

linked\_list.display()

position = linked\_list.search(4)

if position != -1:

print(f"Data found at position {position}")

else:

print("Data not found")