**Assignment 2**

**Implement a Doubly Linked List**

class Node:

def \_\_init\_\_(self, data):

self.data = data

self.next = None

self.prev = None

class DoublyLinkedList:

def \_\_init\_\_(self):

self.head = None

def append(self, data):

new\_node = Node(data)

if not self.head:

self.head = new\_node

else:

current = self.head

while current.next:

current = current.next

current.next = new\_node

new\_node.prev = current

def prepend(self, data):

new\_node = Node(data)

if not self.head:

self.head = new\_node

else:

new\_node.next = self.head

self.head.prev = new\_node

self.head = new\_node

def delete(self, data):

current = self.head

while current:

if current.data == data:

if current.prev:

current.prev.next = current.next

if current.next:

current.next.prev = current.prev

if current == self.head:

self.head = current.next

return

current = current.next

def display(self):

current = self.head

while current:

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

current = current.next

print("None")

# Example usage:

dll = DoublyLinkedList()

dll.append(1)

dll.append(2)

dll.append(3)

dll.prepend(0)

dll.display()

dll.delete(2)

dll.display()