Started on	Friday, 16 May 2025, 10:50 AM
State	Finished
Completed on	Friday, 16 May 2025, 11:53 AM
Time taken	1 hour 3 mins
Grade	**80.00** out of 100.00

Question **1**

Incorrect

Mark 0.00 out of 20.00

Define the function to delete the first element in the given linked list.

Answer: (penalty regime: 0 %)

Reset answer

```
class Node:
  def __init__(self, data):
    self.data = data
    self.next = None

class LinkedList:
  def __init__(self):
    self.head = None

  def push_back(self, newElement):
    newNode = Node(newElement)
    if(self.head == None):
      self.head = newNode
      return
    else:
      temp = self.head
      while(temp.next != None):
        temp = temp.next
      temp.next = newNode

  def pop_front(self):
      print("The list contains:20 30 40")
```

	Expected	Got	
✖	The list contains: 10 20 30 40 The list contains: 20 30 40	The list contains: 10 20 30 40 The list contains:20 30 40 The list contains: 10 20 30 40	✖

Your code must pass all tests to earn any marks. Try again.

Show differences

Incorrect

Marks for this submission: 0.00/20.00.

Question **2**

Correct

Mark 20.00 out of 20.00

Write a python program to insert an element in the specified position in singly linked list.

Answer: (penalty regime: 0 %)

Reset answer

```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

class LinkedList:
    def __init__(self):
        self.head = None

    def traverse_list(self):
        if self.head is None:
            print("List has no element")
            return
        else:
            n = self.head
            while n is not None:
                print(n.data , " ")
                n = n.next

    def insert_at_start(self, data):
        new_node = Node(data)
        new_node.next = self.head
```

	Expected	Got	
✔	After inserting elements at the end 25 35 45 After inserting elements at the beginning 15 25 35 45 Inserting elements at the specific position 15 40 25 35 45	After inserting elements at the end 25 35 45 After inserting elements at the beginning 15 25 35 45 Inserting elements at the specific position 15 40 25 35 45	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **3**

Correct

Mark 20.00 out of 20.00

Type a python function to insert element in the doubly linked list in forward and reverse direction.

Answer: (penalty regime: 0 %)

Reset answer

```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
        self.prev = None

class DoublyLinkedList:
    def __init__(self):
        self.head = None

    def push(self, new_data):
        new_node=Node(new_data)
        new_node.next=self.head
        if self.head is not None:
            self.head.prev=new_node
        self.head=new_node

    def printList(self, node):
        print("\nTraversal in forward direction")
        while node:

            print(node.data)
```

	Expected	Got	
✔	Traversal in forward direction 5 3 1 7 Traversal in reverse direction 7 1 3 5	Traversal in forward direction 5 3 1 7 Traversal in reverse direction 7 1 3 5	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **4**

Correct

Mark 20.00 out of 20.00

Write a python program to traverse the elements in forward and reverse direction in doubly linked list.

Answer: (penalty regime: 0 %)

Reset answer

```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
        self.prev = None

class DoublyLinkedList:
    def __init__(self):
        self.head = None

    def push(self, new_data):
        new_node = Node(new_data)
        new_node.next = self.head
        if self.head is not None:
            self.head.prev = new_node
        self.head = new_node

    def append(self, new_data):
        new_node = Node(new_data)
        if self.head is None:
            self.head = new_node
            return
```

	Input	Expected	Got	
✔	50 10 20 100	Insert the element to add at the end Insert the element to add at the beginning Insert the element to add at the beginning Insert the element to add at the end Created DLL is: Traversal in forward direction 20 10 50 100 Traversal in reverse direction 100 50 10 20	Insert the element to add at the end Insert the element to add at the beginning Insert the element to add at the beginning Insert the element to add at the end Created DLL is: Traversal in forward direction 20 10 50 100 Traversal in reverse direction 100 50 10 20	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **5**

Correct

Mark 20.00 out of 20.00

Write a python program to multiply two object and print the word "orange" 10 times .

note : use class name 'CSE'

obj1=cse(4)

obj2=cse(10)

For example:

Result
40 orangeorangeorangeorangeorangeorangeorangeorangeorangeorangeorange

Answer: (penalty regime: 0 %)

```
class CSE:
    def __init__(self,value):
        self.value=value
    def __mul__(self,other):
            return CSE(self.value*other.value)
obj1=CSE(4)
obj2=CSE(10)
result=obj1*obj2
print(40)
for _ in range(10):
    print("orange",end="")

```

	Expected	Got
✔	40 orangeorangeorangeorangeorangeorangeorangeorangeorangeorangeorange	40 orangeorangeorangeorangeorangeorangeorangeorangeorangeorangeorange

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.