Started on	Wednesday, 7 May 2025, 3:44 PM
State	Finished
Completed on	Thursday, 8 May 2025, 11:49 AM
Time taken	20 hours 5 mins
Overdue	18 hours 5 mins
Grade	**80.00** out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

Define a function to delete the last 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 delete_last:
    def __init__(self):
        self.head = None

    def removeLastNode(self):
        #{{TYPE THE CODE}}
        if self.head.next==None:
            self.head=None
        else:
            temp=self.head
            while temp.next.next!=None:
                temp=temp.next
            temp1=temp.next
            temp.next=None
            temp1=None

    def push(self, data):
```

	Input	Expected	Got	
✓	5 10 20 30 40 50	Enter the number of elements to push: 10 20 30 40	Enter the number of elements to push: 10 20 30 40	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **2**

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

```python
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 **3**

Not answered

Mark 0.00 out of 20.00

Write a Program in Python to Generate Fibonacci series for the number '7'

For example:

Input	Result
---	0 1 1 2 3 5 8

Answer: (penalty regime: 0 %)

Question **4**

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

```python
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 **5**

Correct

Mark 20.00 out of 20.00

Type a python function to insert words at the beginning and display the sentence 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, end=" ")
```

	Input	Expected	Got	
✓	4 step on step carefully	Enter the number of words to display. Enter the data to push Enter the data to push Enter the data to push Enter the data to push Traversal in forward direction carefully step on step Traversal in reverse direction step on step carefully	Enter the number of words to display. Enter the data to push Enter the data to push Enter the data to push Enter the data to push Traversal in forward direction carefully step on step Traversal in reverse direction step on step carefully	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.