Started on	Thursday, 24 April 2025, 1:19 PM
State	Finished
Completed on	Thursday, 24 April 2025, 1:51 PM
Time taken	31 mins 49 secs
Grade	**80.00** out of 100.00

Question **1**

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 **2**

Not answered

Mark 0.00 out of 20.00

Write a Python program to find the first appearance of the substring 'not' and 'poor' from a given string, if 'not' follows the 'poor', replace the whole 'not'...'poor' substring with 'good'. Return the resulting string.

For example:

Input	Result
`The lyrics is not that poor!`	`The lyrics is good!`

Answer: (penalty regime: 0 %)

```
1
```

Question **3**

Correct

Mark 20.00 out of 20.00

Write a python program to insert an element (String) after the specified element 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 AI DS ML After inserting elements at the beginning CS AI DS ML Inserting elements after the specified item CS AI DS R_PGM ML	After inserting elements at the end AI DS ML After inserting elements at the beginning CS AI DS ML Inserting elements after the specified item CS AI DS R_PGM ML	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **4**

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 append(self, new_data):
        new_node = Node(new_data)
        if self.head is None:
            self.head = new_node
            return
        last = self.head
        while last.next:
            last = last.next
        last.next = new_node
        new_node.prev = last
        return

```

	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 **5**

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.