Started on	Tuesday, 15 April 2025, 3:10 PM
State	Finished
Completed on	Tuesday, 15 April 2025, 4:02 PM
Time taken	51 mins 45 secs
Grade	**100.00** out of 100.00

Question **1**

Incorrect

Mark 20.00 out of 20.00

Write a Python Program to Count the common characters in the Two Inputted Strings "manager" and "trainer"

For example:

Input	Result
---	a n a e r are the common characters

Answer: (penalty regime: 0 %)

```
'''if "a" in n1 and n2:
    print("a")

if "n" in n1 and n2:
    print("n")

if "a" in n1 and n2:
    print("a")

if "e" in n1 and n2:
    print("e")

if "r" in n1 and n2:
    print("r")
'''
for i in n1.split():
    if i in n2.split():
        print(i)
```

	Input	Expected	
✖	---	a n a e r are the common characters	✖

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

Incorrect

Marks for this submission: 0.00/20.00.

Question **2**

Correct

Mark 20.00 out of 20.00

Define the 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 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_last(self):
      temp=self.head
```

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

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 elements at the beginning of the doubly linked list.

Answer: (penalty regime: 0 %)

Reset answer

```
class Node:
    def __init__(self, data):
        self.item = data
        self.nref = None
        self.pref = None

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

    def insert_in_emptylist(self, data):
        if self.start_node is None:
            new_node = Node(data)
            self.start_node = new_node
        else:
            print("list is not empty")

    def insert_at_start(self, data):
        n=Node(data)
        temp=self.start_node
        temp.pref = n
        n.nref=temp
```

	Expected	Got	
✔	10 20 30 40	10 20 30 40	✔

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 display the elements in doubly linked list.

Answer: (penalty regime: 0 %)

Reset answer

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

class doublyLinkedList:
    def __init__(self):
        self.start_node = None

    def InsertToEmptyList(self, data):
        if self.start_node is None:
            new_node = Node(data)
            self.start_node = new_node
        else:
            print("The list is empty")

    def InsertToEnd(self, data):
        if self.start_node is None:
            new_node = Node(data)
            self.start_node = new_node
            return
```

	Expected	Got	
✔	Element is: 10 Element is: 20 Element is: 30 Element is: 40 Element is: 50 Element is: 60	Element is: 10 Element is: 20 Element is: 30 Element is: 40 Element is: 50 Element is: 60	✔

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 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.