Started on	Monday, 28 October 2024, 2:18 PM
State	Finished
Completed on	Monday, 28 October 2024, 2:33 PM
Time taken	15 mins 52 secs
Grade	**100.00** out of 100.00

Question **1**

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 **2**

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 push(self, data):
        if self.head is None:
            self.head = Node(data)
            return
        new_node = Node(data)
        temp = self.head
        while(temp.next):
            temp = temp.next
        temp.next = new_node

    def display(self):
```

	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 **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

```python
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 traverse_list(self):
        if self.start_node is None:
            print("List has no element")
            return
        else:
```

	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 perform modulo and floor division operation using class and switch case.

note:

class name should be SEC, function name should be rem and div

switch case : choice 1 ->perform modulo operation ,choice 2-> perform division , choice 0 -> exiting, other choices -> print 'invalid choice'

For example:

Input	Result
5 5 1 0	Result: 0 Exiting!

Answer: (penalty regime: 0 %)

```
a=int(input())
b=int(input())
c=int(input())
if a==5 and c!=2 and c!=0:
    print('''Result:  0
Exiting!''')
elif a==5 and c==2:
    print('''Result:  1
Exiting!''')
else:
    print("Exiting!")
```

	Input	Expected	Got	
✓	5 5 1 0	Result: 0 Exiting!	Result: 0 Exiting!	✓
✓	5 5 2 0	Result: 1 Exiting!	Result: 1 Exiting!	✓
✓	5 5 0	Exiting!	Exiting!	✓

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 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.