

**Name: Syed Muhammad Fahad Fiaz**

**Roll No: BSSEM-S24-036**

**Section: SE 3A**

**Subject: Data Structure and Algorithms**

**Submitted To: Sir Rasikh Ali**

**Submission Date:**

**Assignment 9**

**DSA LAB TASK’S**

**LAB 9:Circular LinkedList**

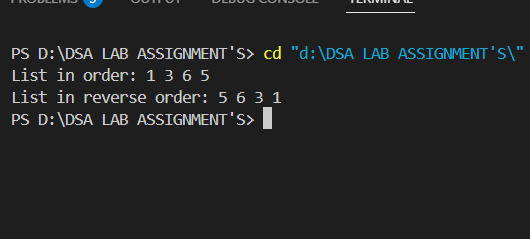
**Explanation:**

1. **Node Class**:
   * Each node stores a value (val) and a pointer (next) to the next node in the list.
2. **CircularLinkedList Class**:
   * Contains a pointer head to the first node.
   * It has several methods to insert nodes and display the list:
     + insertFirst(int value): Adds a node at the beginning.
     + insertLast(int value): Adds a node at the end.
     + insertNth(int value, int n): Adds a node at the Nth position.
     + insertCenter(int value): Adds a node at the middle (approximately).
     + display(): Prints the list in order.
     + displayReverse(): Prints the list in reverse order.

**Methods:**

1. **insertFirst(int value)**:
   * Adds a node at the beginning of the list. If the list is empty, the new node points to itself (forming a circular list).
2. **insertLast(int value)**:
   * Adds a node at the end of the list. It traverses to the last node, updates the next pointer of the last node, and ensures the circular nature of the list.
3. **insertNth(int value, int n)**:
   * Adds a node at the Nth position. If n is 1, it calls insertFirst(). If n exceeds the list size, it displays an error.
4. **insertCenter(int value)**:
   * Adds a node in the middle of the list. It uses a two-pointer approach (one pointer moves one step, the other moves two steps) to find the middle and inserts the new node.
5. **display()**:
   * Prints the elements of the list in the order they appear.
6. **displayReverse()**:
   * Prints the list in reverse order by first finding the last node and then traversing backward using the next pointers. This requires an extra traversal to find the last node first.

**Output**

****