Name:M Ijaz

Submitted to

Sir Rasikh

Roll no . 073

Subject . Lab Data structure

Lab .9

Implement functions to insert node at first, last, Nth location, and centre of a circular linked list. And display in order and display in reverse order.

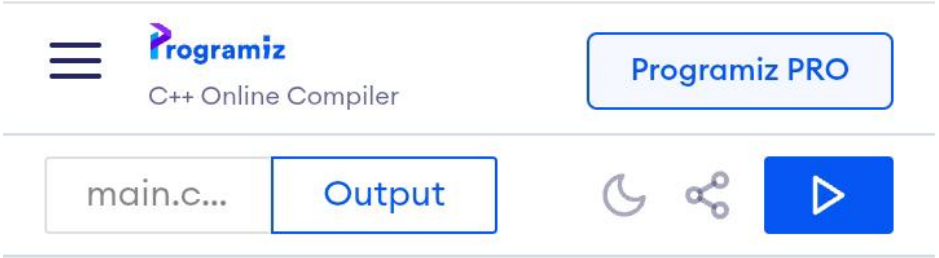
```
#include <iostream>
using namespace std;

class Node {
public:
    int data;
    Node* next;
    Node(int val) : data(val), next(nullptr) {}
};

class CircularLinkedList {
public:
    Node* head;
    CircularLinkedList() : head(nullptr) {}

    void insertFirst(int val) {
        Node* newNode = new Node(val);
```

```
    if (!head) {
        head = newNode;
        newNode->next = head;
    } else {
        Node* temp = head;
        while (temp->next != head) temp = temp->next;
        temp->next = newNode;
        newNode->next = head;
        head = newNode;
    }
}

void insertLast(int val) {
    Node* newNode = new Node(val);
    if (!head) {
        head = newNode;
        newNode->next = head;
    } else {
        Node* temp = head;
        while (temp->next != head) temp = temp->next;
        temp->next = newNode;
        newNode->next = head;
    }
}

void insertNth(int val, int n) {
    Node* newNode = new Node(val);
```

```
    if (n == 1) {
        insertFirst(val);
        return;
    }
    Node* temp = head;
    for (int i = 1; temp->next != head && i < n - 1; i++) {
        temp = temp->next;
    }
    if (temp->next == head) return;
    newNode->next = temp->next;
    temp->next = newNode;
}

void insertCentre(int val) {
    if (!head) return;
    Node* slow = head;
    Node* fast = head;
    while (fast->next != head && fast->next->next != head) {
        slow = slow->next;
        fast = fast->next->next;
    }
    Node* newNode = new Node(val);
    newNode->next = slow->next;
    slow->next = newNode;
}

void display() {
```

```
        if (!head) return;
        Node* temp = head;
        do {
            cout << temp->data << " ";
            temp = temp->next;
        } while (temp != head);
        cout << endl;
    }

    void displayReverse() {
        if (!head) return;
        Node* temp = head;
        do {
            temp = temp->next;
        } while (temp->next != head);
        while (temp != head) {
            cout << temp->data << " ";
            Node* prev = head;
            while (prev->next != temp) prev = prev->next;
            temp = prev;
        }
        cout << temp->data << endl;
    }
};

int main() {
    CircularLinkedList cll;
```

```
cll.insertFirst(10);
cll.insertLast(20);
cll.insertLast(30);
cll.insertNth(25, 3);
cll.insertCentre(15);
cout << "Circular Linked List: ";
cll.display();
cout << "Reverse Circular Linked List: ";
cll.displayReverse();
return 0;
```




```
Circular Linked List: 10 20 15 25 30
Reverse Circular Linked List: 30 25 15 20
    10


=== Code Execution Successful ===
```

}