SUPERIOR UNIVERSITY

NAME :ZOHAIB KHAN

ROLL_ NO : 022

SUBJECT: DSA LAB

LAB NO 9

Task: 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.

```cpp
#include <iostream>
using namespace std;
struct Node {
    int data;
    Node* next;
};
class CircularLinkedList {
private:
    Node* head;
public:
    CircularLinkedList() : head(nullptr) {}
    void insertAtFirst(int value);
    void insertAtLast(int value);
    void insertAtNth(int value, int n);
    void insertAtCentre(int value);
    void displayOrder();
    void displayReverse();
};
void CircularLinkedList::insertAtFirst(int value) {
    Node* newNode = new Node{value, nullptr};
    if (!head) {
        newNode->next = newNode;
        head = newNode;
    } else {
        Node* temp = head;
        while (temp->next != head) temp = temp->next;
        temp->next = newNode;
        newNode->next = head;
        head = newNode;
    }
}
void CircularLinkedList::insertAtLast(int value) {
    Node* newNode = new Node{value, nullptr};
    if (!head) {
        newNode->next = newNode;
        head = newNode;
    } else {
        Node* temp = head;
```

```
    }
    Node* temp = head;
    for (int i = 0; i < n - 1 && temp->next != head; i++) temp = temp->next;
    if (temp->next == head && n > 0) {
        cout << "Position out of range!" << endl;
        return;
    }
    Node* newNode = new Node(value, temp->next);
    temp->next = newNode;
}

void CircularLinkedList::insertAtCentre(int value) {
    if (!head) {
        insertAtFirst(value);
        return;
    }
    Node *slow = head, *fast = head;
    while (fast->next != head && fast->next->next != head) {
        slow = slow->next;
        fast = fast->next->next;
    }
    Node* newNode = new Node(value, slow->next);
    slow->next = newNode;
}

void CircularLinkedList::displayOrder() {
    if (!head) return;
    Node* temp = head;
    do {
        cout << temp->data << " -> ";
        temp = temp->next;
    } while (temp != head);
    cout << "HEAD" << endl;
}
void CircularLinkedList::displayReverse() {
    if (!head) return;
    Node* temp = head;
    do {
        temp = temp->next;
    } while (temp->next != head);
    while (temp != head) {
        cout << temp->data << " -> ";
        temp = temp->next;
    }
    cout << temp->data << " -> HEAD" << endl;
}
int main() {
    CircularLinkedList list;
    list.insertAtFirst(10); list.displayOrder();
    list.insertAtLast(20);  list.displayOrder();
    list.insertAtNth(15, 1); list.displayOrder();
    list.insertAtCentre(25); list.displayOrder();
    cout << "Reverse order: "; list.displayReverse();
    return 0;
}
```

```
10 -> HEAD
10 -> 20 -> HEAD
10 -> 15 -> 20 -> HEAD
10 -> 15 -> 25 -> 20 -> HEAD
Reverse order: 20 -> 10 -> HEAD

--------------------------------
Process exited after 0.6244 seconds with return value 0
Press any key to continue . . .
```