Name:

Wajiha Zahid

Roll No:

S24-040

Subject:

“DSA LAB ”

Section

BSSE-3A

Resource Person:

Sir Rasikh Ali

(Lab Task 8)

Q1: Circular Linked List (Insert & Display Nodes)

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.

Answer:

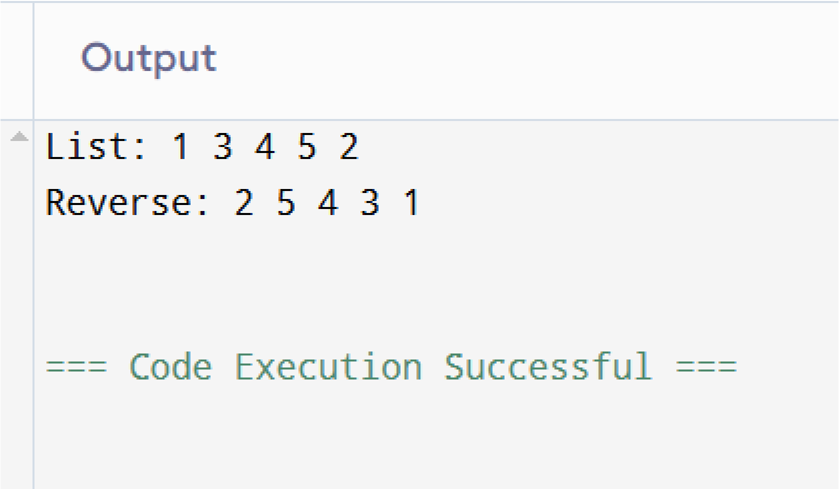
```
#include <iostream>
using namespace std;

struct Node {
    int data;
    Node* next;
    Node(int val) : data(val), next(this) {}
};
Node* insertAtFirst(Node* head, int data) {
    Node* newNode = new Node(data);
    if (!head) return newNode;
    Node* temp = head;
    while (temp->next != head) temp = temp->next;
    temp->next = newNode;
    newNode->next = head;
    return newNode;
}
Node* insertAtLast(Node* head, int data) {
    if (!head) return new Node(data);
    Node* temp = head;
    while (temp->next != head) temp = temp->next;
    temp->next = new Node(data);
    temp->next->next = head;
    return head;
}

Node* insertAtNth(Node* head, int data, int pos) {
    if (!head || pos == 0) return insertAtFirst(head, data);
    Node* temp = head;
    for (int i = 1; i < pos && temp->next != head; i++) temp = temp->next;
    Node* newNode = new Node(data);
    newNode->next = temp->next;
    temp->next = newNode;
    return head;
}
void display(Node* head) {
    if (!head) return;
    Node* temp = head;
    do {
        cout << temp->data << " ";
        temp = temp->next;
    } while (temp != head);
    cout << endl;
}
void displayReverse(Node* head) {
    if (!head) return;
    Node* temp = head;
    string rev = "";
    do {
        rev = to_string(temp->data) + " " + rev;
        temp = temp->next;
    } while (temp != head);
    cout << rev << endl;
}
int main() {
    Node* head = NULL;
    head = insertAtFirst(head, 1);
    head = insertAtLast(head, 2);
    head = insertAtNth(head, 3, 1);
    head = insertAtNth(head, 4, 2);
    head = insertAtNth(head, 5, 3);

    cout << "List: "; display(head);
    cout << "Reverse: "; displayReverse(head);
    return 0;
}
```

Output:



```
List: 1 3 4 5 2
Reverse: 2 5 4 3 1


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