Name:

Wajiha Zahid

Roll No:

S24-040

Subject:

“DSA LAB ”

Section

BSSE-3A

Resource Person:

Sir Rasikh Ali

(Lab Task 6)

Q1: Singly Linked List (Delete Nodes)

Task: Implement functions to delete the first node, last node, Nth node, and centre node of a singly linked list.

Answer:

```
#include <iostream>

using namespace std;

// Node structure

struct Node {
    int data;
    Node* next;
};
void insertAtBeginning(Node** head, int newData) {
    Node* newNode = new Node();
    newNode->data = newData;
    newNode->next = (*head);
    (*head) = newNode;
```

```
}
void insertAtEnd(Node** head, int newData) {
    Node* newNode = new Node();
    Node* last = *head;
    newNode->data = newData;
    newNode->next = NULL;

    if (*head == NULL) {
        *head = newNode;
        return;
    }
    while (last->next != NULL)
        last = last->next;

    last->next = newNode;
    return;
}
void deleteFirstNode(Node** head) {
    if (*head == NULL)
        return;
    Node* temp = *head;
    *head = (*head)->next;
    delete temp;
}
void deleteLastNode(Node** head) {
```

```
    if (*head == NULL)
        return;

    Node* temp = *head;
    Node* prev;

    while (temp->next != NULL) {
        prev = temp;
        temp = temp->next;
    }
    if (temp == *head)
        *head = NULL;
    else
        prev->next = NULL;

    delete temp;
}
void deleteNthNode(Node** head, int position) {
    if (*head == NULL || position <= 0)
        return;

    Node* temp = *head;
    Node* prev;

    if (position == 1) {
```

```
        *head = (*head)->next;
        delete temp;
        return;
    }
    for (int i = 1; temp != NULL && i < position; i++) {
        prev = temp;
        temp = temp->next;
    }
    if (temp == NULL)
        return;
    prev->next = temp->next;
    delete temp;
}
void deleteCentreNode(Node** head) {
    if (*head == NULL)
        return;
    Node* slow = *head;
    Node* fast = *head;

    while (fast != NULL && fast->next != NULL) {
        slow = slow->next;
        fast = fast->next->next;
    }

    Node* temp = slow->next;
```

```
    slow->next = slow->next->next;
    delete temp;
}
void printList(Node* node) {
    while (node != NULL) {
        cout << node->data << " ";
        node = node->next;
    }
    cout << endl;
}

int main() {
    Node* head = NULL;
    insertAtBeginning(&head, 7);
    insertAtBeginning(&head, 5);
    insertAtBeginning(&head, 3);
    insertAtBeginning(&head, 1);
    printList(head);
    deleteFirstNode(&head);
    printList(head);
    deleteLastNode(&head);
    printList(head);
    deleteNthNode(&head, 2);
    printList(head);
    deleteCentreNode(&head);
```

```
    printList(head);
    return 0;

    return 0;
}
```

Output: