WAP to Implement Singly Linked List with following operations a) Create a linked list. b) Deletion of first element, specified element and last element in the list. c) Display the contents of the linked list.

```
#include <stdio.h>
#include <stdlib.h>

/* Node structure */
struct Node {
    int data;
    struct Node *next;
};

struct Node *head = NULL;

/* Create linked list */
void createList(int n) {
    struct Node *newNode, *temp;
    int data;

    for (int i = 0; i < n; i++) {
        newNode = (struct Node *)malloc(sizeof(struct Node));
        printf("Enter data for node %d: ", i + 1);
        scanf("%d", &data);

        newNode->data = data;
        newNode->next = NULL;

        if (head == NULL) {
            head = newNode;
```

```
            temp = head;
        } else {
            temp->next = newNode;
            temp = newNode;
        }
    }
}

/* Delete first node */
void deleteFirst() {
    struct Node *temp;
    if (head == NULL) {
        printf("List is empty!\n");
        return;
    }
    temp = head;
    head = head->next;
    free(temp);
}

/* Delete last node */
void deleteLast() {
    struct Node *temp = head, *prev = NULL;

    if (head == NULL) {
        printf("List is empty!\n");
        return;
    }
```

```
    if (head->next == NULL) {
        free(head);
        head = NULL;
        return;
    }

    while (temp->next != NULL) {
        prev = temp;
        temp = temp->next;
    }
    prev->next = NULL;
    free(temp);
}
/* Delete specified element */
void deleteSpecified(int key) {
    struct Node *temp = head, *prev = NULL;
    if (head == NULL) {
        printf("List is empty!\n");
        return;
    }

    if (head->data == key) {
        deleteFirst();
        return;
    }

    while (temp != NULL && temp->data != key) {
        prev = temp;
        temp = temp->next; }
```

```
    if (temp == NULL) {
        printf("Element not found!\n");
        return;
    }
    prev->next = temp->next;
    free(temp);
}
/* Display linked list */
void displayList() {
    struct Node *temp = head;
    if (head == NULL) {
        printf("List is empty.\n");
        return;
    }
    printf("Linked List: ");
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}
/* Main function */
int main() {
    int n, choice, key;
    printf("Enter number of nodes to create: ");
    scanf("%d", &n);
    createList(n);
    while (1) {
        printf("\nMenu:\n");
```

```
        printf("1. Delete first element\n");
        printf("2. Delete specified element\n");
        printf("3. Delete last element\n");
        printf("4. Display list\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                deleteFirst();
                break;
            case 2:
                printf("Enter element to delete: ");
                scanf("%d", &key);
                deleteSpecified(key);
                break;
            case 3:
                deleteLast();
                break;
            case 4:
                displayList();
                break;
            case 5:
                exit(0);
            default:
                printf("Invalid choice!\n");
        }
    }
}
```

```
Enter number of nodes to create: 3
Enter data for node 1: 10
Enter data for node 2: 20
Enter data for node 3: 30

Menu:
1. Delete first element
2. Delete specified element
3. Delete last element
4. Display list
5. Exit
Enter your choice: 1

Menu:
1. Delete first element
2. Delete specified element
3. Delete last element
4. Display list
5. Exit
Enter your choice: 4
Linked List: 20 -> 30 -> NULL

Menu:
1. Delete first element
2. Delete specified element
3. Delete last element
4. Display list
5. Exit
Enter your choice: 3

Menu:
1. Delete first element
2. Delete specified element
3. Delete last element
4. Display list
5. Exit
Enter your choice: 4
Linked List: 20 -> NULL

Menu:
1. Delete first element
2. Delete specified element
3. Delete last element
4. Display list
5. Exit
Enter your choice: 2
Enter element to delete: 11
Element not found!

Menu:
1. Delete first element
2. Delete specified element
3. Delete last element
4. Display list
5. Exit
Enter your choice: 5

Process returned 0 (0x0)   execution time : 38.205 s
Press any key to continue.
```