

Lab program 5a:-

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, value;
```

```
    struct Node *newNode, *temp;
```

```
    printf("Enter number of nodes: ");
```

```
    scanf("%d", &n);
```

```
    for (int i = 0; i < n; i++) {
```

```
        newNode = (struct Node *)malloc(sizeof(struct Node));
```

```
        printf("Enter data for node %d: ", i + 1);
```

```
        scanf("%d", &value);
```

```
        newNode->data = value;
```

```
        newNode->next = NULL;
```

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

```
            head = newNode;
```

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

// Delete first node

```
void deleteFirst() {  
    struct Node *temp;  
  
    if (head == NULL) {  
        printf("List is empty. Deletion not possible.\n");  
        return;  
    }  
  
    temp = head;  
    head = head->next;  
    free(temp);  
  
    printf("First element deleted successfully.\n");  
}
```

// Delete last node

```
void deleteLast() {  
    struct Node *temp, *prev;  
  
    if (head == NULL) {  
        printf("List is empty. Deletion not possible.\n");  
    }
```

```
    return;
```

```
}
```

```
if (head->next == NULL) {
```

```
    free(head);
```

```
    head = NULL;
```

```
    printf("Last element deleted successfully.\n");
```

```
    return;
```

```
}
```

```
temp = head;
```

```
while (temp->next != NULL) {
```

```
    prev = temp;
```

```
    temp = temp->next;
```

```
}
```

```
prev->next = NULL;
```

```
free(temp);
```

```
printf("Last element deleted successfully.\n");
```

```
}
```

```
// Delete specified element
```

```
void deleteSpecified() {
```

```
    struct Node *temp, *prev;
```

```
    int key;
```

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

```
    printf("List is empty. Deletion not possible.\n");
```

```
    return;
```

```
}
```

```

printf("Enter element to delete: ");

scanf("%d", &key);


// If first node contains the key
if (head->data == key) {
    temp = head;
    head = head->next;
    free(temp);
    printf("Element %d deleted successfully.\n", key);
    return;
}


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


if (temp == NULL) {
    printf("Element not found in the list.\n");
} else {
    prev->next = temp->next;
    free(temp);
    printf("Element %d deleted successfully.\n", key);
}
}


// Display list
void display() {
    struct Node *temp;

```

```

if (head == NULL) {

    printf("Linked List is empty.\n");

    return;

}


temp = head;

printf("Linked List: ");

while (temp != NULL) {

    printf("%d -> ", temp->data);

    temp = temp->next;

}

printf("NULL\n");

}


// Main function

int main() {

    int choice;


    while (1) {

        printf("\n--- Singly Linked List Menu ---\n");

        printf("1. Create Linked List\n");

        printf("2. Delete First Element\n");

        printf("3. Delete Specified Element\n");

        printf("4. Delete Last Element\n");

        printf("5. Display Linked List\n");

        printf("6. Exit\n");

        printf("Enter your choice: ");

        scanf("%d", &choice);


        switch (choice) {

```

```

    case 1: createList(); break;

    case 2: deleteFirst(); break;

    case 3: deleteSpecified(); break;

    case 4: deleteLast(); break;

    case 5: display(); break;

    case 6: exit(0);

    default: printf("Invalid choice!\n");

}

}

return 0;

}

```

```

--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display Linked List
6. Exit
Enter your choice: 1
Enter number of nodes: 4
Enter data for node 1: 12
Enter data for node 2: 13
Enter data for node 3: 14
Enter data for node 4: 15

--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display Linked List
6. Exit
Enter your choice: 5
Linked List: 12 -> 13 -> 14 -> 15 -> NULL

--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display Linked List
6. Exit
Enter your choice: 2
First element deleted successfully.

```

```
--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display Linked List
6. Exit
Enter your choice: 5
Linked List: 13 -> 14 -> 15 -> NULL
```

```
--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display Linked List
6. Exit
Enter your choice: 3
Enter element to delete: 14
Element 14 deleted successfully.
```

```
--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display Linked List
6. Exit
Enter your choice: 5
Linked List: 13 -> 15 -> NULL
```

```
--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display Linked List
6. Exit
Enter your choice: 4
Last element deleted successfully.
```

```
--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display Linked List
6. Exit
Enter your choice: 5
Linked List: 13 -> NULL
```

```
--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display Linked List
6. Exit
Enter your choice: 6
```

```
Process returned 0 (0x0)   execution time : 43.378 s
Press any key to continue.
```