

```

};

struct Node* head = NULL;

void createList(int n) {
    struct Node *newNode, *temp;
    int data, i;

    head = (struct Node*)malloc(sizeof(struct Node));
    if (head == NULL) {
        printf("Memory not allocated!\n");
        return;
    }

    printf("Enter data for node 1: ");
    scanf("%d", &data);
    head->data = data;
    head->next = NULL;
    temp = head;

    for (i = 2; i <= n; i++) {
        newNode = (struct Node*)malloc(sizeof(struct Node));
        if (newNode == NULL) {
            printf("Memory not allocated!\n");
            break;
        }
        printf("Enter data for node %d: ", i);
        scanf("%d", &data);
        newNode->data = data;
        newNode->next = NULL;
        temp->next = newNode;
        temp = newNode;
    }
    printf("Linked list created successfully!\n");
}

```

```

void deleteFirst() {
    if (head == NULL) {

```

```

void deleteLast() {
    if (head == NULL) {
        printf("List is empty.\n");
        return;
    }
    struct Node *temp = head, *prev;
    if (head->next == NULL) {
        printf("Deleted last element: %d\n", head->data);
        free(head);
        head = NULL;
        return;
    }
    while (temp->next != NULL) {
        prev = temp;
        temp = temp->next;
    }
    printf("Deleted last element: %d\n", temp->data);
    prev->next = NULL;
    free(temp);
}

```

```

void deleteSpecific(int value) {
    if (head == NULL) {
        printf("List is empty.\n");
        return;
    }

    struct Node *temp = head, *prev = NULL;

    if (head->data == value) {
        head = head->next;
        printf("Deleted element: %d\n", temp->data);
        free(temp);
        return;
    }

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

```

```

    if (temp == NULL) {

```

```

    }

void displayList() {
    struct Node* temp = head;
    if (temp == NULL) {
        printf("List is empty.\n");
        return;
    }
    printf("Linked List: ");
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

int main() {
    int n, choice, value;

    while (1) {
        printf("1. Create Linked List\n");
        printf("2. Delete First Element\n");
        printf("3. Delete Specific Element\n");
        printf("4. Delete Last Element\n");
        printf("5. Display List\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter number of nodes: ");
                scanf("%d", &n);
                createList(n);
                break;
            case 2:
                deleteFirst();
                break;
            case 3:
                printf("Enter element to delete: ");
                scanf("%d", &value);
                deleteSpecific(value);
                break;

```

```
ce! Try again.\n");
```

--- Singly Linked List Operations ---

1. Create Linked List
2. Delete First Element
3. Delete Specific Element
4. Delete Last Element
5. Display List
6. Exit

Enter your choice: 1

Enter number of nodes: 5

Enter data for node 1: 56

Enter data for node 2: 89

Enter data for node 3: 53

Enter data for node 4: 24

Enter data for node 5: 56

Linked list created successfully!

--- Singly Linked List Operations ---

1. Create Linked List
2. Delete First Element
3. Delete Specific Element
4. Delete Last Element
5. Display List
6. Exit

Enter your choice: 5

Linked List: 56 -> 89 -> 53 -> 24 -> 56 -> NULL

--- Singly Linked List Operations ---

1. Create Linked List
2. Delete First Element
3. Delete Specific Element
4. Delete Last Element
5. Display List
6. Exit

Enter your choice: 3

Enter element to delete: 53

Deleted element: 53

--- Singly Linked List Operations ---

1. Create Linked List
2. Delete First Element
3. Delete Specific Element
4. Delete Last Element
5. Display List
6. Exit

Enter your choice: 5

Linked List: 56 -> 89 -> 24 -> 56 -> NULL

--- Singly Linked List Operations ---

1. Create Linked List
2. Delete First Element

```
. Delete First Element
. Delete Specific Element
. Delete Last Element
. Display List
. Exit
```

Enter your choice: 2

Deleted first element: 56

--- Singly Linked List Operations ---

```
. Create Linked List
. Delete First Element
. Delete Specific Element
. Delete Last Element
. Display List
. Exit
```

Enter your choice: 5

Linked List: 89 -> 24 -> NULL

--- Singly Linked List Operations ---

```
. Create Linked List
. Delete First Element
. Delete Specific Element
. Delete Last Element
. Display List
. Exit
```

Enter your choice: 6

Process returned 0 (0x0) execution time : 47.413 s

Press any key to continue.