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

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

struct Node *head = NULL;


void create(int value) {
    struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->next = NULL;

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


void deleteFirst() {
    if (head == NULL) {
        printf("List is empty.\n");
        return;
    }
    struct Node *temp = head;
    head = head->next;
    free(temp);
    printf("First element deleted.\n");
}

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

```c
    if (head->next == NULL) {
        free(head);
        head = NULL;
        printf("Last element deleted.\n");
        return;
    }

    struct Node *temp = head;
    while (temp->next->next != NULL)
        temp = temp->next;

    free(temp->next);
    temp->next = NULL;
    printf("Last element deleted.\n");
}


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

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


    if (temp != NULL && temp->data == value) {
        head = temp->next;
        free(temp);
        printf("Element %d deleted.\n", value);
        return;
    }


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

    if (temp == NULL) {
        printf("Element %d not found.\n", value);
        return;
    }
```

```c
        prev->next = temp->next;
        free(temp);
        printf("Element %d deleted.\n", value);
}


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

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

int main() {
    int choice, value;

    while (1) {
        printf("\n--- Singly Linked List Menu ---\n");
        printf("1. Insert element\n");
        printf("2. Delete first element\n");
        printf("3. Delete last element\n");
        printf("4. Delete specific element\n");
        printf("5. Display list\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
        case 1:
            printf("Enter value to insert: ");
            scanf("%d", &value);
            create(value);
            break;

        case 2:
            deleteFirst();
```

```c
            break;

        case 3:
            deleteLast();
            break;

        case 4:
            printf("Enter value to delete: ");
            scanf("%d", &value);
            deleteSpecific(value);
            break;

        case 5:
            display();
            break;

        case 6:
            exit(0);

        default:
            printf("Invalid choice! Try again.\n");
        }
    }

    return 0;
}
```

Output

```
--- Singly Linked List Menu ---
1. Insert element
2. Delete first element
3. Delete last element
4. Delete specific element
5. Display list
6. Exit
Enter your choice: 1
Enter value to insert: 10

--- Singly Linked List Menu ---
1. Insert element
2. Delete first element
3. Delete last element
4. Delete specific element
5. Display list
6. Exit
Enter your choice: 1
Enter value to insert: 20

--- Singly Linked List Menu ---
1. Insert element
2. Delete first element
3. Delete last element
4. Delete specific element
5. Display list
6. Exit
Enter your choice: 1
Enter value to insert: 30

--- Singly Linked List Menu ---
1. Insert element
2. Delete first element
3. Delete last element
4. Delete specific element
5. Display list
6. Exit
Enter your choice: 5
Linked List: 10 20 30

--- Singly Linked List Menu ---
1. Insert element
2. Delete first element
3. Delete last element
4. Delete specific element
5. Display list
6. Exit
Enter your choice: 2
First element deleted.

--- Singly Linked List Menu ---
1. Insert element
2. Delete first element
3. Delete last element
4. Delete specific element
5. Display list
6. Exit
Enter your choice: 5
Linked List: 20 30

--- Singly Linked List Menu ---
1. Insert element
```

```
--- Singly Linked List Menu ---
1. Insert element
2. Delete first element
3. Delete last element
4. Delete specific element
5. Display list
6. Exit
Enter your choice: 3
Last element deleted.

--- Singly Linked List Menu ---
1. Insert element
2. Delete first element
3. Delete last element
4. Delete specific element
5. Display list
6. Exit
Enter your choice: 5
Linked List: 20

--- Singly Linked List Menu ---
1. Insert element
2. Delete first element
3. Delete last element
4. Delete specific element
5. Display list
6. Exit
Enter your choice: 1
Enter value to insert: 60

--- Singly Linked List Menu ---
1. Insert element
2. Delete first element
3. Delete last element
4. Delete specific element
5. Display list
6. Exit
Enter your choice: 5
Linked List: 20 60

--- Singly Linked List Menu ---
1. Insert element
2. Delete first element
3. Delete last element
4. Delete specific element
5. Display list
6. Exit
Enter your choice: 4
Enter value to delete: 20
Element 20 deleted.

--- Singly Linked List Menu ---
1. Insert element
2. Delete first element
3. Delete last element
4. Delete specific element
5. Display list
6. Exit
Enter your choice: 5
Linked List: 60

--- Singly Linked List Menu ---
```

```
1. Insert element
2. Delete first element
3. Delete last element
4. Delete specific element
5. Display list
6. Exit
Enter your choice: 4
Enter value to delete: 20
Element 20 deleted.

--- Singly Linked List Menu ---
1. Insert element
2. Delete first element
3. Delete last element
4. Delete specific element
5. Display list
6. Exit
Enter your choice: 5
Linked List: 60

--- Singly Linked List Menu ---
1. Insert element
2. Delete first element
3. Delete last element
4. Delete specific element
5. Display list
6. Exit
Enter your choice:
```