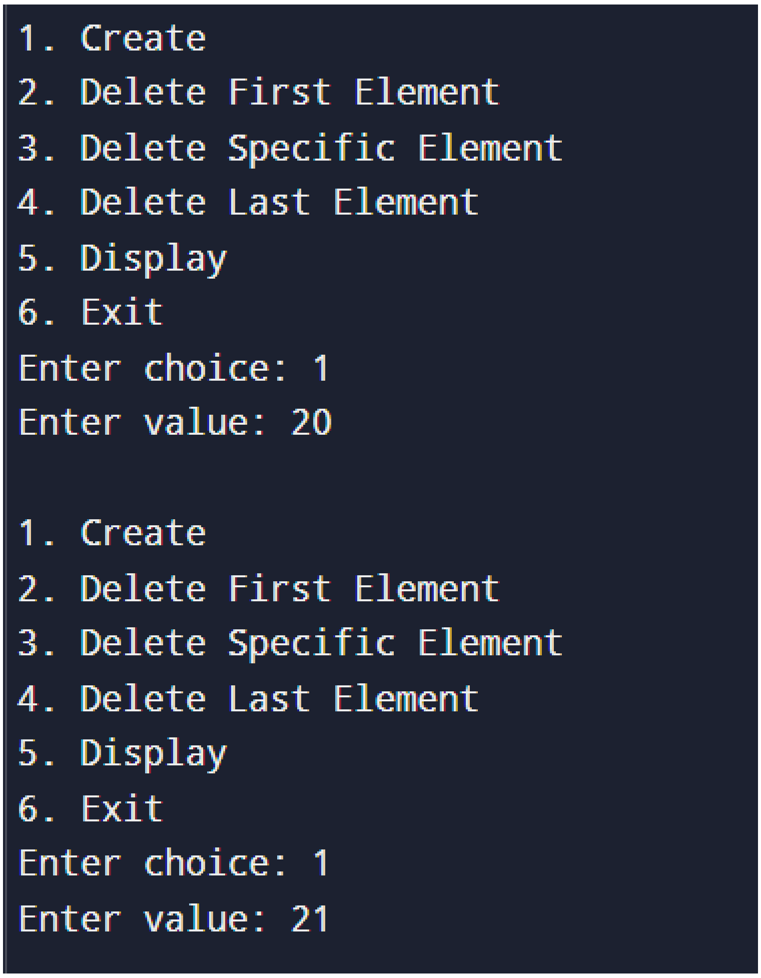
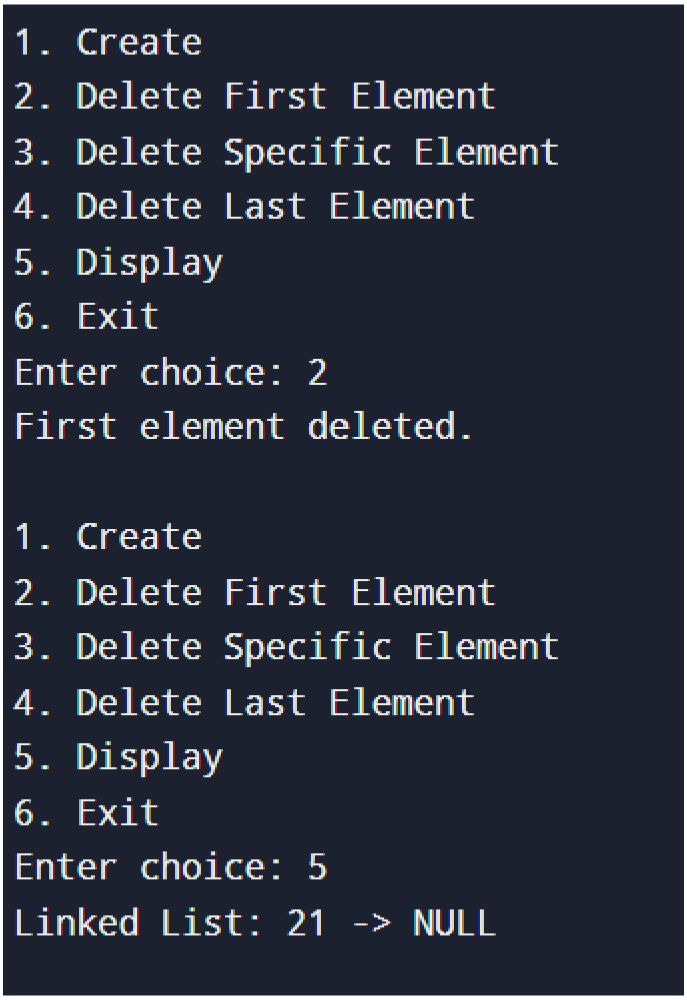
```
#include <stdio.h>
#include <stdlib.h>

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

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

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

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

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

    free(temp);
    printf("Element %d deleted.\n", value);
}

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

    if (head->next == NULL) {
        free(head);
        head = NULL;
    } else {
        struct Node* temp = head;
        struct Node* prev = NULL;

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

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

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("NULL\n");
```

```
}

int main() {
    int choice, value;

    while (1) {
        printf("\n1. Create\n");
        printf("2. Delete First Element\n");
        printf("3. Delete Specific Element\n");
        printf("4. Delete Last Element\n");
        printf("5. Display\n");
        printf("6. Exit\n");
        printf("Enter choice: ");
        scanf("%d", &choice);

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

            case 2:
                deleteFirst();
                break;

            case 3:
                printf("Enter value to delete: ");
                scanf("%d", &value);
                deleteSpecific(value);
                break;

            case 4:
                deleteLast();
                break;

            case 5:
                display();
                break;

            case 6:
                exit(0);

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

    return 0;
}
```



```
1. Create
2. Delete First Element
3. Delete Specific Element
4. Delete Last Element
5. Display
6. Exit
Enter choice: 1
Enter value: 20

1. Create
2. Delete First Element
3. Delete Specific Element
4. Delete Last Element
5. Display
6. Exit
Enter choice: 1
Enter value: 21
```



```
1. Create
2. Delete First Element
3. Delete Specific Element
4. Delete Last Element
5. Display
6. Exit
Enter choice: 2
First element deleted.

1. Create
2. Delete First Element
3. Delete Specific Element
4. Delete Last Element
5. Display
6. Exit
Enter choice: 5
Linked List: 21 -> NULL
```