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

// Define a structure for a node in the linked list
struct Node {
    int data;
    struct Node* next;
};

// Function to create a new node
struct Node* createNode(int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    if (newNode == NULL) {
        printf("Memory allocation failed.\n");
        exit(1);
    }
    newNode->data = value;
    newNode->next = NULL;
    return newNode;
}

// Function to insert a node at the beginning of the linked list
struct Node* insertAtBeginning(struct Node* head, int value) {
    struct Node* newNode = createNode(value);
    newNode->next = head;
    return newNode;
}

// Function to insert a node at any position in the linked list
struct Node* insertAtPosition(struct Node* head, int value, int position) {
    struct Node* newNode = createNode(value);
    if (position == 1) {
        newNode->next = head;
        return newNode;
    }

    struct Node* current = head;
    for (int i = 1; i < position - 1 && current != NULL; i++) {
        current = current->next;
    }

    if (current == NULL) {
        printf("Invalid position.\n");
        return head;
    }

    newNode->next = current->next;
    current->next = newNode;
    return head;
}
```

```
// Function to insert a node at the end of the linked list
struct Node* insertAtEnd(struct Node* head, int value) {
    struct Node* newNode = createNode(value);
    if (head == NULL) {
        return newNode;
    }

    struct Node* current = head;
    while (current->next != NULL) {
        current = current->next;
    }

    current->next = newNode;
    return head;
}

// Function to delete the first element in the linked list
struct Node* deleteFirst(struct Node* head) {
    if (head == NULL) {
        printf("List is empty. Cannot delete.\n");
        return NULL;
    }

    struct Node* temp = head;
    head = head->next;
    free(temp);
    return head;
}

// Function to delete a specified element in the linked list
struct Node* deleteElement(struct Node* head, int value) {
    if (head == NULL) {
        printf("List is empty. Cannot delete.\n");
        return NULL;
    }

    if (head->data == value) {
        struct Node* temp = head;
        head = head->next;
        free(temp);
        return head;
    }

    struct Node* current = head;
    while (current->next != NULL && current->next->data != value) {
        current = current->next;
    }

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

```
        printf("Element not found in the list.\n");
        return head;
    }

    struct Node* temp = current->next;
    current->next = current->next->next;
    free(temp);
    return head;
}

// Function to delete the last element in the linked list
struct Node* deleteLast(struct Node* head) {
    if (head == NULL) {
        printf("List is empty. Cannot delete.\n");
        return NULL;
    }

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

    struct Node* current = head;
    while (current->next->next != NULL) {
        current = current->next;
    }

    free(current->next);
    current->next = NULL;
    return head;
}

// Function to display the contents of the linked list
void displayLinkedList(struct Node* head) {
    printf("Linked List: ");
    while (head != NULL) {
        printf("%d -> ", head->data);
        head = head->next;
    }
    printf("NULL\n");
}

// Main function to test the linked list operations
int main() {
    struct Node* head = NULL;

    // Creating a linked list
    head = insertAtBeginning(head, 3);
    head = insertAtBeginning(head, 2);
    head = insertAtBeginning(head, 1);
```

```
    head = insertAtEnd(head, 4);
    head = insertAtEnd(head, 5);

    // Displaying the original linked list
    printf("Original ");
    displayLinkedList(head);

    // Deleting elements from the linked list
    head = deleteFirst(head);
    printf("After deleting the first element: ");
    displayLinkedList(head);

    head = deleteElement(head, 3);
    printf("After deleting element '3': ");
    displayLinkedList(head);

    head = deleteLast(head);
    printf("After deleting the last element: ");
    displayLinkedList(head);

    return 0;
}
```

```
Original Linked List: 1 -> 2 -> 3 -> 4 -> 5 -> NULL
After deleting the first element: Linked List: 2 -> 3 -> 4 -> 5 -> NULL
After deleting element '3': Linked List: 2 -> 4 -> 5 -> NULL
After deleting the last element: Linked List: 2 -> 4 -> NULL
Shreyash Sinha 1BM22CS273
```