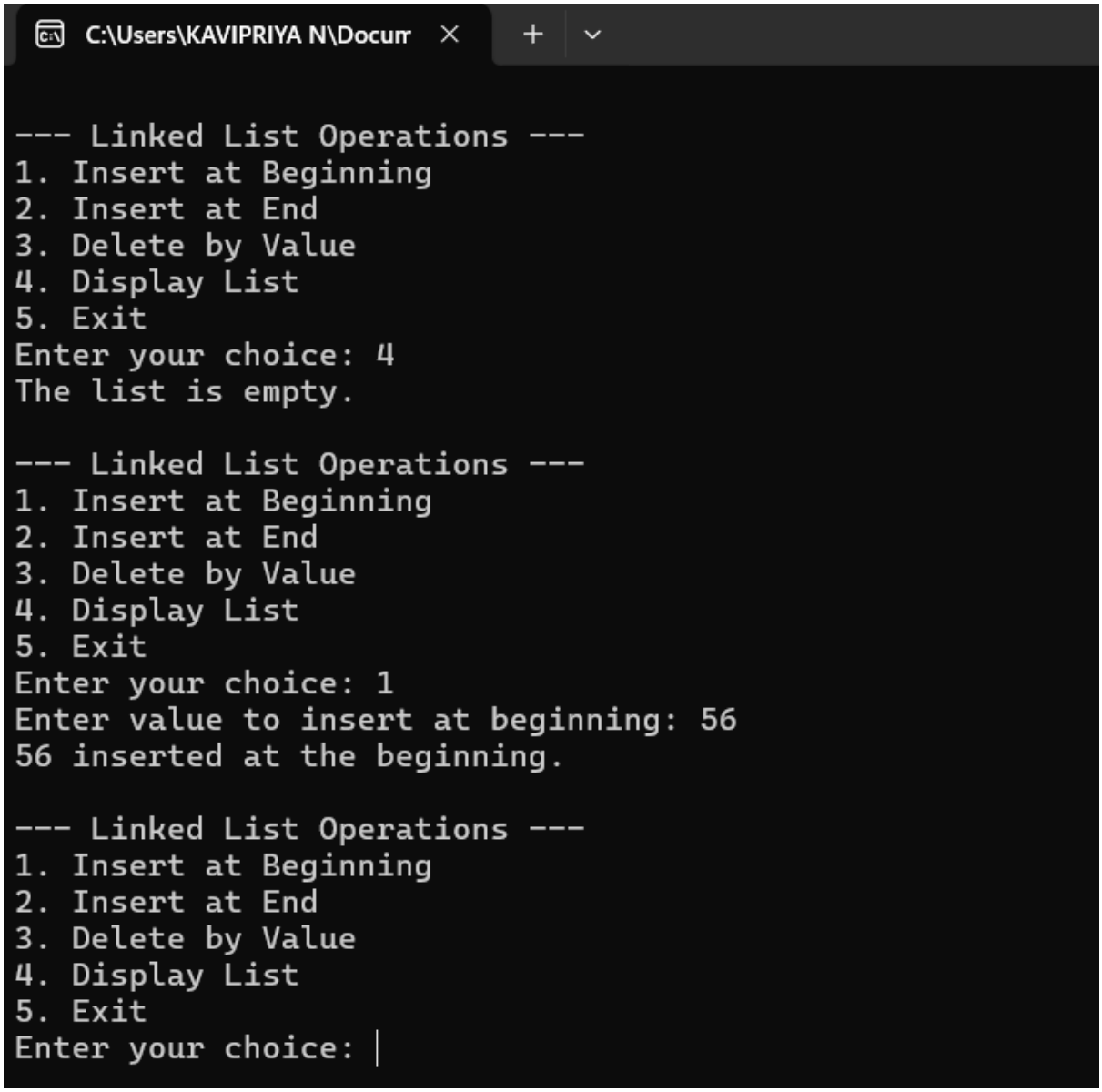
```
#include <stdio.h>
#include <stdlib.h>

// Define the node structure
struct Node {
    int data;
    struct Node* next;
};

struct Node* head = NULL;

// Function to insert at the beginning
void insertAtBeginning(int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->next = head;
    head = newNode;
    printf("%d inserted at the beginning.\n", value);
}

// Function to insert at the end
void insertAtEnd(int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->next = NULL;

    if (head == NULL) {
        head = newNode;
        printf("%d inserted as the first node.\n", value);
        return;
    }
```

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

    temp->next = newNode;
    printf("%d inserted at the end.\n", value);
}

// Function to delete a node by value
void deleteByValue(int value) {
    struct Node* temp = head;
    struct Node* prev = NULL;

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

    if (temp == NULL) {
        printf("Value %d not found in the list.\n", value);
        return;
    }

    if (prev == NULL)
        head = temp->next;  // Deleting the first node
    else
        prev->next = temp->next;

    free(temp);
    printf("Value %d deleted from the list.\n", value);
```

```
}

// Function to display the linked list
void displayList() {
    struct Node* temp = head;

    if (temp == NULL) {
        printf("The list is empty.\n");
        return;
    }

    printf("Linked list elements: ");
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

// Main function with menu
int main() {
    int choice, value;

    while (1) {
        printf("\n--- Linked List Operations ---\n");
        printf("1. Insert at Beginning\n");
        printf("2. Insert at End\n");
        printf("3. Delete by Value\n");
        printf("4. Display List\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
```

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

        switch (choice) {
        case 1:
            printf("Enter value to insert at beginning: ");
            scanf("%d", &value);
            insertAtBeginning(value);
            break;
        case 2:
            printf("Enter value to insert at end: ");
            scanf("%d", &value);
            insertAtEnd(value);
            break;
        case 3:
            printf("Enter value to delete: ");
            scanf("%d", &value);
            deleteByValue(value);
            break;
        case 4:
            displayList();
            break;
        case 5:
            printf("Exiting program.\n");
            exit(0);
        default:
            printf("Invalid choice! Try again.\n");
        }
    }

    return 0;
}
```


C:\Users\KAVIPRIYA N\Docum
--- Linked List Operations ---
1. Insert at Beginning
2. Insert at End
3. Delete by Value
4. Display List
5. Exit
Enter your choice: 4
The list is empty.
--- Linked List Operations ---
1. Insert at Beginning
2. Insert at End
3. Delete by Value
4. Display List
5. Exit
Enter your choice: 1
Enter value to insert at beginning: 56
56 inserted at the beginning.
--- Linked List Operations ---
1. Insert at Beginning
2. Insert at End
3. Delete by Value
4. Display List
5. Exit
Enter your choice: