

## **Experiment No.6:**

### **Singly Linked List Operations**

**You are building a text editor where lines of text are stored dynamically. You need to allow insertion and deletion of lines at any position, and display text both normally and in reverse. Use a singly linked list to implement: display, insert (front/end/middle), delete (front/end/middle), display in reverse, and reverse the list.**

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

struct Node {
    char line[100];
    struct Node* next;
};

struct Node* head = NULL;

// Insert at end
void insertEnd(char text[]) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    strcpy(newNode->line, text);
    newNode->next = NULL;

    if (head == NULL) {
        head = newNode;
    } else {
        struct Node* temp = head;
        while (temp->next != NULL)
            temp = temp->next;
        temp->next = newNode;
    }
    printf("Line added at end.\n");
}

// Display all lines
void display() {
    struct Node* temp = head;
    if (temp == NULL) {
        printf("List is empty.\n");
        return;
    }

    printf("Text Lines:\n");
    while (temp != NULL) {
```

```

        printf("%s\n", temp->line);
        temp = temp->next;
    }
}

// Display lines in reverse using recursion
void displayReverse(struct Node* node) {
    if (node == NULL) return;
    displayReverse(node->next);
    printf("%s\n", node->line);
}

// Reverse the list
void reverseList() {
    struct Node *prev = NULL, *curr = head, *next = NULL;
    while (curr != NULL) {
        next = curr->next;
        curr->next = prev;
        prev = curr;
        curr = next;
    }
    head = prev;
    printf("List reversed.\n");
}

// Delete from front
void deleteFront() {
    if (head == NULL) {
        printf("List is empty.\n");
        return;
    }
    struct Node* temp = head;
    head = head->next;
    free(temp);
    printf("First line deleted.\n");
}

int main() {
    int choice;
    char text[100];

    do {
        printf("\n--- Simple Text Editor ---\n");
        printf("1. Add Line at End\n");

```

```

printf("2. Display All Lines\n");
printf("3. Display in Reverse\n");
printf("4. Reverse the List\n");
printf("5. Delete First Line\n");
printf("6. Exit\n");
printf("Enter choice: ");
scanf("%d", &choice);
getchar(); // clear newline

switch (choice) {
    case 1:
        printf("Enter line: ");
        fgets(text, sizeof(text), stdin);
        text[strcspn(text, "\n")] = 0; // remove newline
        insertEnd(text);
        break;
    case 2:
        display();
        break;
    case 3:
        printf("Lines in reverse:\n");
        displayReverse(head);
        break;
    case 4:
        reverseList();
        break;
    case 5:
        deleteFront();
        break;
    case 6:
        printf("Exiting...\n");
        break;
    default:
        printf("Invalid choice.\n");
}
} while (choice != 6);

return 0;
}

```