EXPERIMENT [10](linked list operations)

CODE:

#include <stdio.h>

#include <stdlib.h>

// Node structure

struct Node {

int data;

struct Node\* next;

};

struct Node\* head = NULL;

// 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);

}

// Insert at the end

void insertAtEnd(int value) {

struct Node\* newNode = (struct Node\*) malloc(sizeof(struct Node));

struct Node\* temp = head;

newNode->data = value;

newNode->next = NULL;

if (head == NULL) {

head = newNode;

} else {

while (temp->next != NULL)

temp = temp->next;

temp->next = newNode;

}

printf("%d inserted at the end.\n", value);

}

// Delete a node by value

void deleteNode(int value) {

struct Node\* temp = head;

struct Node\* prev = NULL;

// If head node holds the value

if (temp != NULL && temp->data == value) {

head = temp->next;

free(temp);

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

return;

}

// Search for the value

while (temp != NULL && temp->data != value) {

prev = temp;

temp = temp->next;

}

// Not found

if (temp == NULL) {

printf("%d not found in the list.\n", value);

return;

}

// Delete the node

prev->next = temp->next;

free(temp);

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

}

// Display the list

void display() {

struct Node\* temp = head;

if (temp == NULL) {

printf("List is empty.\n");

return;

}

printf("Linked List: ");

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("\nLinked List Operations:\n");

printf("1. Insert at Beginning\n");

printf("2. Insert at End\n");

printf("3. Delete Node\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);

deleteNode(value);

break;

case 4:

display();

break;

case 5:

printf("Exiting program.\n");

return 0;

default:

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

}

}

return 0;

}

OUTPUT:

