#include <stdio.h>

#include <stdlib.h>

// Node structure

struct Node {

int data;

struct Node\* next;

};

// Head pointer

struct Node\* head = NULL;

// Function to insert at the end

void insert(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;

}

printf("Inserted %d successfully.\n", value);

}

// Function to delete a node by value

void delete(int value) {

struct Node \*temp = head, \*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;

} else {

prev->next = temp->next;

}

free(temp);

printf("Deleted %d successfully.\n", value);

}

// Function to 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("\n--- Linked List Operations ---\n");

printf("1. Insert\n");

printf("2. Delete\n");

printf("3. Display\n");

printf("4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter value to insert: ");

scanf("%d", &value);

insert(value);

break;

case 2:

printf("Enter value to delete: ");

scanf("%d", &value);

delete(value);

break;

case 3:

display();

break;

case 4:

printf("Exiting program.\n");

return 0;

default:

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

}

}

}