#include <stdio.h>

#include <stdlib.h>

// Define the structure of a doubly linked list node

struct Node {

int data;

struct Node\* next;

struct Node\* prev;

};

// Function to create a new node with given data

struct Node\* createNode(int data) {

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

newNode->data = data;

newNode->next = NULL;

newNode->prev = NULL;

return newNode;

}

// Insert a node at the front of the list

void insertAtFront(struct Node\*\* head, int data) {

struct Node\* newNode = createNode(data);

newNode->next = \*head;

if (\*head != NULL)

(\*head)->prev = newNode;

\*head = newNode;

}

// Insert a new node to the left of the node with the given key

void insertLeftOfKey(struct Node\*\* head, int key, int data) {

struct Node\* current = \*head;

while (current != NULL && current->data != key) {

current = current->next;

}

if (current == NULL) {

printf("Node with data %d not found.\n", key);

return;

}

struct Node\* newNode = createNode(data);

newNode->next = current;

newNode->prev = current->prev;

if (current->prev != NULL) {

current->prev->next = newNode;

} else {

\*head = newNode;

}

current->prev = newNode;

}

// Delete the node with the given data

void deleteNode(struct Node\*\* head, int data) {

struct Node\* current = \*head;

while (current != NULL && current->data != data) {

current = current->next;

}

if (current == NULL) {

printf("Node with data %d not found.\n", data);

return;

}

if (current->prev != NULL) {

current->prev->next = current->next;

} else {

\*head = current->next;

}

if (current->next != NULL) {

current->next->prev = current->prev;

}

free(current);

printf("Node with data %d deleted.\n", data);

}

// Display the contents of the list

void displayList(struct Node\* head) {

struct Node\* current = head;

printf("Doubly linked list contents: ");

while (current != NULL) {

printf("%d ", current->data);

current = current->next;

}

printf("\n");

}

// Main function to test the operations

int main() {

struct Node\* head = NULL;

int choice, data, key;

while (1) {

printf("\n1. Insert at front\n2. Insert left of key\n3. Delete node\n4. Display list\n5. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter data to insert at front: ");

scanf("%d", &data);

insertAtFront(&head, data);

break;

case 2:

printf("Enter key to insert left of: ");

scanf("%d", &key);

printf("Enter data to insert: ");

scanf("%d", &data);

insertLeftOfKey(&head, key, data);

break;

case 3:

printf("Enter data to delete: ");

scanf("%d", &data);

deleteNode(&head, data);

break;

case 4:

displayList(head);

break;

case 5:

exit(0);

break;

default:

printf("Invalid choice. Please try again.\n");

}

}

return 0;

}