//Ankit Kumar 2K20/C0/72

#include <stdio.h>

#include <stdlib.h>

struct node {

    int data;

    struct node \*next;

};

struct node \*head = NULL;

void display() {

    struct node \*ptr = head;

    if (!ptr) {

        printf("list is empty\n");

        return;

    }

    printf("List elements are:\n");

    while (ptr) {

        printf("%d ", ptr->data);

        ptr = ptr->next;

    }

    printf("\n");

}

void insertAtFront() {

    int x;

    printf("Enter data to be inserted\n");

    scanf("%d", &x);

    struct node \*ptr = (struct node \*)malloc(sizeof(struct node));

    ptr->data = x;

    ptr->next = NULL;

    if (!head) {

        head = ptr;

        return;

    }

    ptr->next = head;

    head = ptr;

}

void insertAtEnd() {

    int x;

    printf("Enter data to be inserted\n");

    scanf("%d", &x);

    struct node \*ptr = (struct node \*)malloc(sizeof(struct node));

    ptr->data = x;

    ptr->next = NULL;

    if (!head) {

        head = ptr;

        return;

    }

    struct node \*p = head;

    while (p->next)

        p = p->next;

    p->next = ptr;

}

void insertAtPosition() {

    int x, pos;

    printf("Enter node after which insertion is to be done:\n");

    scanf("%d", &pos);

    printf("Enter data to be inserted\n");

    scanf("%d", &x);

    struct node \*ptr = (struct node \*)malloc(sizeof(struct node));

    ptr->data = x;

    ptr->next = NULL;

    if (!head) {

        head = ptr;

        return;

    }

    struct node \*p = head;

    while (p->next && p->data != pos)

        p = p->next;

    ptr->next = p->next;

    p->next = ptr;

}

void deleteNode() {

    int x;

    printf("Enter node to be deleted:\n");

    scanf("%d", &x);

    struct node \*ptr = head;

    struct node \*tail = NULL;

    if (head->data == x) {

        head = head->next;

        free(ptr);

    }

    while (ptr && ptr->data != x) {

        tail = ptr;

        ptr = ptr->next;

    }

    if (!ptr) {

        printf("Element not present in list!\n");

        return;

    }

    tail->next = ptr->next;

    free(ptr);

}

void search() {

    int x;

    printf("Enter data to be searched\n");

    scanf("%d", &x);

    struct node \*p = head;

    while (p) {

        if (p->data == x) {

            printf("Element found in list!\n");

            return;

        }

        p = p->next;

    }

    printf("Element not found!!\n");

}

int main() {

    int choice;

    while (1) {

        printf("\n1 To see list\n");

        printf("2 For insertion at starting\n");

        printf("3 For insertion at end\n");

        printf("4 For insertion after given node\n");

        printf("5 For deletion of node\n");

        printf("6 To search element\n");

        printf("7 To exit\n");

        printf("\nEnter Choice :\n");

        scanf("%d", &choice);

        switch (choice) {

        case 1:

            display();

            break;

        case 2:

            insertAtFront();

            break;

        case 3:

            insertAtEnd();

            break;

        case 4:

            insertAtPosition();

            break;

        case 5:

            deleteNode();

            break;

        case 6:

            search();

            break;

        case 7:

            exit(1);

            break;

        default:

            printf("Enter valid choice!\n");

            continue;

        }

    }

    return 0;

}

