#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <stdbool.h>

struct node

{

int data;

int key;

struct node \*next;

};

struct node \*head = NULL;

struct node \*current = NULL;

//hien thi danh sach

void printList()

{

struct node \*ptr = head;

printf("\n[ ");

//bat dau tu phan dau danh sach

while(ptr != NULL)

{

printf("(%d,%d) ",ptr->key,ptr->data);

ptr = ptr->next;

}

printf(" ]");

}

//chen link tai vi tri dau tien

void insertFirst(int key, int data)

{

//tao mot link

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

link->key = key;

link->data = data;

//tro link nay toi first node cu

link->next = head;

//tro first toi first node moi

head = link;

}

//xoa phan tu dau tien

struct node\* deleteFirst()

{

//luu tham chieu toi first link

struct node \*tempLink = head;

//danh dau next toi first link la first

head = head->next;

//tra ve link bi xoa

return tempLink;

}

//kiem tra list co trong hay khong

bool isEmpty()

{

return head == NULL;

}

int length()

{

int length = 0;

struct node \*current;

for(current = head; current != NULL; current = current->next)

{

length++;

}

return length;

}

//tim mot link voi key da cho

struct node\* find(int key){

//bat dau tim tu first link

struct node\* current = head;

//neu list la trong

if(head == NULL)

{

return NULL;

}

//duyet qua list

while(current->key != key){

//neu day la last node

if(current->next == NULL){

return NULL;

}else {

//di chuyen toi next link

current = current->next;

}

}

//neu tim thay du lieu, tra ve link hien tai

return current;

}

//xoa mot link voi key da cho

struct node\* deleteKey(int key){

//bat dau tu first link

struct node\* current = head;

struct node\* previous = NULL;

//neu list la trong

if(head == NULL){

return NULL;

}

//duyet qua list

while(current->key != key){

//neu day la last node

if(current->next == NULL){

return NULL;

}else {

//luu tham chieu toi link hien tai

previous = current;

//di chuyen toi next link

current = current->next;

}

}

//cap nhat link

if(current == head) {

//thay doi first de tro toi next link

head = head->next;

}else {

//bo qua link hien tai

previous->next = current->next;

}

return current;

}

// ham sap xep

void sort(){

int i, j, k, tempKey, tempData ;

struct node \*current;

struct node \*next;

int size = length();

k = size ;

for ( i = 0 ; i < size - 1 ; i++, k-- ) {

current = head ;

next = head->next ;

for ( j = 1 ; j < k ; j++ ) {

if ( current->data > next->data ) {

tempData = current->data ;

current->data = next->data;

next->data = tempData ;

tempKey = current->key;

current->key = next->key;

next->key = tempKey;

}

current = current->next;

next = next->next;

}

}

}

// ham dao nguoc list

void reverse(struct node\*\* head\_ref) {

struct node\* prev = NULL;

struct node\* current = \*head\_ref;

struct node\* next;

while (current != NULL) {

next = current->next;

current->next = prev;

prev = current;

current = next;

}

\*head\_ref = prev;

}

main() {

insertFirst(1,10);

insertFirst(2,20);

insertFirst(3,30);

insertFirst(4,1);

insertFirst(5,40);

insertFirst(6,56);

printf("Danh sach ban dau: ");

//in danh sach

printList();

while(!isEmpty()){

struct node \*temp = deleteFirst();

printf("\nGia tri bi xoa:");

printf("(%d,%d) ",temp->key,temp->data);

}

printf("\nDanh sach sau khi da xoa gia tri: ");

printList();

insertFirst(1,10);

insertFirst(2,20);

insertFirst(3,30);

insertFirst(4,1);

insertFirst(5,40);

insertFirst(6,56);

printf("\nPhuc hoi danh sach: ");

printList();

printf("\n");

struct node \*foundLink = find(4);

if(foundLink != NULL){

printf("Tim thay phan tu: ");

printf("(%d,%d) ",foundLink->key,foundLink->data);

printf("\n");

}else {

printf("Khong tim thay phan tu.");

}

deleteKey(4);

printf("Danh sach, sau khi xoa mot phan tu: ");

printList();

printf("\n");

foundLink = find(4);

if(foundLink != NULL){

printf("Tim thay phan tu: ");

printf("(%d,%d) ",foundLink->key,foundLink->data);

printf("\n");

}else {

printf("Khong tim thay phan tu.");

}

printf("\n");

sort();

printf("Danh sach sau khi duoc sap xep: ");

printList();

reverse(&head);

printf("\nDanh sach sau khi bi dao nguoc: ");

printList();

}