#include<stdio.h>

#include<stdlib.h>

typedef struct Node{

int data;

struct Node \*next;

}Node;

int main(){

Node \*head = (Node\*)malloc(sizeof(Node));

head -> next = NULL;

createLink(head,10);//创建链表

travelLink(head);//遍历链表

insertForward(head,100);//头插法

insertBack(head,200);//尾插法

travelLink(head);//遍历链表

deleteSame(head);//删除相同元素

travelLink(head);//遍历链表

reverseLink(head);//翻转链表元素

travelLink(head);//遍历链表

deleteLink(head);//删除链表

isEmpty(head);//判断链表是否为空

return 0;

}

//创建链表

void createLink(Node \*head,int size){

Node \*rear = head;

int i;

for(i = 0;i < size;++i){

Node \*newnode = (Node\*)malloc(sizeof(Node));

newnode -> next = NULL;

scanf("%d",&newnode->data);

rear -> next = newnode;

rear = newnode;

}

}

//遍历链表

void travelLink(Node \*head){

Node \*p = head -> next;

while(p != NULL){

printf("%d\t",p->data);

p = p -> next;

}

putchar('\n');

}

//头插法

void insertForward(Node \*head,int data){

Node \*newnode = (Node\*)malloc(sizeof(Node));

newnode -> next = NULL;

newnode -> data = data;

newnode -> next = head -> next;

head -> next = newnode;

}

//尾插法

void insertBack(Node \*head,int data){

Node \*newnode = (Node\*)malloc(sizeof(Node));

newnode -> next = NULL;

newnode -> data = data;

Node \*p = head;

while(p->next != NULL){

p = p -> next;

}

p -> next = newnode;

p = newnode;

}

//删除相同元素

void deleteSame(Node \*head){

Node \*curr = head -> next;

while(curr != NULL){

Node \*pre = curr;

Node \*p = curr -> next;

while(p != NULL){

//若有相同的元素，则删除；否则两个指针继续向下走

if(curr->data == p->data){

pre -> next = p -> next;

free(p);

p = pre -> next;

}else{

pre = pre -> next;

p = p -> next;

}

}

curr = curr -> next;

}

}

//翻转链表

void reverseLink(Node \*head){

Node \*curr;

Node \*pre = NULL;

while(head -> next != NULL){

curr = head -> next;

head -> next = curr -> next;

curr -> next = pre;

pre = curr;

}

head -> next = pre;

}

//删除链表

void deleteLink(Node \*head){

Node \*curr;

while(head -> next != NULL){

curr = head -> next;

head -> next = curr -> next;

free(curr);

}

}

//判断链表是否为空

void isEmpty(Node \*head){

if(head -> next == NULL){

printf("链表为空！\n");

}else{

printf("链表不为空！\n");

}

}