#include<stdio.h>

#include<stdlib.h>

struct LinkedList{

int data;

struct LinkedList \*next;

};

typedef struct LinkedList \*node;

node CreateNode(int value){

node temp;

temp = (node)malloc(sizeof(struct LinkedList));

temp->next = NULL;

temp->data = value;

return temp;

}

node AddTail(node head, int value){

node temp,p;

temp = CreateNode(value);

if(head == NULL){

head = temp;

}

else{

p = head;

while(p->next != NULL){

p = p->next;

}

p->next = temp;

}

return head;

}

node AddHead(node head, int value){

node temp = CreateNode(value);

if(head == NULL){

head = temp;

}else{

temp->next = head;

head = temp;

}

return head;

}

node AddAt(node head, int value, int position){

if(position == 0 || head == NULL){

head = AddHead(head, value);

}else{

int k = 1;

node p = head;

while(p != NULL && k != position){

p = p->next;

++k;

}

if(k != position){

head = AddTail(head, value);

}else{

node temp = CreateNode(value);

temp->next = p->next;

p->next = temp;

}

}

return head;

}

node DelHead(node head){

if(head == NULL){

printf("\nCha co gi de xoa het!");

}else{

head = head->next;

}

return head;

}

node DelTail(node head){

node p = head;

while(p->next->next != NULL){

p = p->next;

}

p->next = p->next->next;

return head;

}

node DelAt(node head, int position){

if(position == 0 || head == NULL){

head = DelHead(head);

}else{

int k = 1;

node p = head;

while(p->next->next != NULL && k != position){

p = p->next;

++k;

}

if(k != position){

head = DelTail(head);

}else{

p->next = p->next->next;

}

}

return head;

}

int Get(node head, int index){

int k = 0;

node p = head;

while(p != NULL && k != index){

p = p->next;

}

return p->data;

}

int Search(node head, int value){

int position = 0;

for(node p = head; p != NULL; p = p->next){

if(p->data == value){

return position;

}

++position;

}

return -1;

}

node DelByVal(node head, int value){

int position = Search(head, value);

while(position != -1){

DelAt(head, position);

position = Search(head, value);

}

return head;

}

void Traverser(node head){

printf("\n");

for(node p = head; p != NULL; p = p->next){

printf("%5d", p->data);

}

}

node InitHead(){

node head;

head = NULL;

return head;

}

int Length(node head){

int length = 0;

for(node p = head; p != NULL; p = p->next){

++length;

}

return length;

}

node Input(){

node head = InitHead();

int n, value;

do{

printf("\nNhap so luong phan tu n = ");

scanf("%d", &n);

}while(n <= 0);

for(int i = 0; i < n; ++i){

printf("\nNhap gia tri can them: ");

scanf("%d", &value);

head = AddTail(head, value);

}

return head;

}

int main(){

printf("\n==Tao 1 danh sach lien ket==");

node head = Input();

Traverser(head);

printf("\n==Them phan tu vao linked list==");

head = AddAt(head, 100, 2);

Traverser(head);

printf("\n==Xoa phan tu khoi linked list==");

head = DelAt(head, 1);

Traverser(head);

printf("\n==Tim kiem phan tu trong linked list==");

int index = Search(head, 5);

}