#include <stdio.h>

#include<conio.h>

#include <stdlib.h>

typedef struct node {

int info;

struct node \*next;

} node;

typedef struct list {

node \*first;

} list;

list\* makelist(void) {

list \*l;

l = (list\*) malloc(sizeof(list));

l->first = NULL;

return l;

}

void createlist(list \*l, int v) {

node \*n, \*p;

n = (node\*) malloc(sizeof(node));

n->info = v;

if (l->first == NULL) {

n->next = NULL;

l->first = n;

} else {

p = l->first;

while (p->next != NULL) {

p = p->next;

}

p->next = n;

n->next = NULL;

}

}

void firstnode(list \*l, int v) {

node \*n;

n = (node\*) malloc(sizeof(node));

n->info = v;

n->next = NULL;

if (l->first == NULL) {

l->first = n;

} else {

n->next = l->first;

l->first = n;

}

}

void maddnode(list \*l, int p, int v) {

node \*n, \*t;

n = (node\*) malloc(sizeof(node));

n->info = v;

t = l->first;

while (t->info != p) {

t = t->next;

}

n->next = t->next;

t->next = n;

}

void laddnode(list \*l, int v) {

node \*n, \*p;

n = (node\*) malloc(sizeof(node));

n->info = v;

p = l->first;

while (p->next != NULL) {

p = p->next;

}

n->next = NULL;

p->next = n;

}

void printlist(list \*l) {

node \*t;

t = l->first;

while (t != NULL) {

printf("%d ", t->info);

t = t->next;

}

}

void dellist(list \*l, int v) {

node \*t, \*s;

t = l->first;

while (t->info != v) {

s = t;

t = t->next;

}

s->next = t->next;

free(t);

}

int main() {

list \*ls;

int a, i, v, p;

clrscr(); // If using Turbo C, otherwise ignore this line.

ls = makelist();

printf("NTR VAL IN LIST: ");

for (i = 0; i < 5; i++) {

scanf("%d", &a);

createlist(ls, a);

}

printlist(ls);

printf("\nNTR FIRST NODE IN LIST: ");

scanf("%d", &v);

firstnode(ls, v);

printlist(ls);

printf("\nNTR NODE IN MIDDLE:\n");

printf("ntr prev node: ");

scanf("%d", &p);

printf("ntr new node: ");

scanf("%d", &v);

maddnode(ls, p, v);

printlist(ls);

printf("\nNTR LAST NODE: ");

scanf("%d", &v);

laddnode(ls, v);

printlist(ls);

printf("\n\nntr val 2 del= ");

scanf("%d", &v);

dellist(ls, v);

printlist(ls);

getch(); // If using Turbo C, otherwise remove or replace

return 0;

}