#include <stdio.h>

#include <stdlib.h>

#include <time.h>

typedef int infotype;

typedef struct item{

infotype info;

struct item \*next;

} \*pointer;

int count\_list(pointer p){

int cnt = 0;

while(p != NULL){

cnt++;

p = p -> next;

}

return cnt;

}

//insert\_top

pointer add\_list(infotype x, pointer p){

pointer q;

q = malloc(sizeof \*q);

if(q == NULL){

printf("メモリ不足\n");

}

q -> info = x;

q -> next = p;

return q;

}

void insert\_last(infotype x, pointer p){

pointer q;

q = malloc(sizeof \*q);

if(q == NULL){

printf("メモリ不足\n");

}

while(p -> next != NULL){

p = p -> next;

}

q -> info = x;

p -> next = q;

}

void insert\_anyIndex(infotype x, int i, pointer p){

int cnt = count\_list(p);

int k = cnt - i;

while(k != 2){

p = p -> next;

k--;

}

pointer q, r, s;

r = p -> next;

s = r -> next;

q = malloc(sizeof \*q);

if(q == NULL){

printf("メモリ不足\n");

}

q -> info = x;

q -> next = s;

r -> next = q;

}

void show\_list(pointer p){

while(p != NULL){

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

p = p -> next;

}

printf("\n");

}

pointer reverse\_list(pointer p){

pointer q, t;

q = NULL;

while(p != NULL){

t = q;

q = p;

p = p -> next;

q -> next = t;

}

return q;

}

pointer delete(int i, pointer p){

pointer q = p;

//int cnt = count\_list(p);

int k = i-1;

if(k == 0){

q = p -> next;

free(p);

return q;

}else{

while(k != 1){

p = p -> next;

k--;

}

pointer r, s;

r = p -> next;

s = r -> next;

free(r);

p -> next = s;

return q;

}

}

int main(){

srand(time(NULL));

infotype x;

pointer head;

head = NULL;

// for(x = 1; x < 11; x++){

// int random = (int)(rand() % 50 + 1);

// head = add\_list(random, head);

// }

head = add\_list(1, head);

head = add\_list(2, head);

head = add\_list(3, head);

head = add\_list(4, head);

head = add\_list(5, head);

show\_list(head);

//insert\_last(6, head); #リストの最後に6を加える

//show\_list(head);

//insert\_anyIndex(7, 3, head); //#リストの任意の位置（第二引数）に7を加える

//printf("\n");

//show\_list(head);

head = delete(1, head); //#任意の位置を削除

printf("\n");

show\_list(head);

//head = reverse\_list(head); //#リストを反転する

//show\_list(head);

return EXIT\_SUCCESS;

}