#include <stdio.h>

#include <stdlib.h>

#include <time.h>

typedef int infotype;

typedef struct item{

infotype info;

struct item \*next;

} \*pointer;

int count\_list();

infotype get();

void set();

void swap();

pointer add\_list();

void insert\_last();

void insert\_anyIndex();

void show\_list();

pointer reverse\_list();

pointer delete();

void selectionSort();

int main(){

srand((unsigned)time(NULL));

pointer head;

head = NULL;

for(int i = 1; i < 11; i++){

int random = (int)(rand() % 100 + 1);

head = add\_list(random, head);

}

show\_list(head);

selectionSort(head);

show\_list(head);

return EXIT\_SUCCESS;

}

int count\_list(pointer p){

int cnt = 0;

while(p != NULL){

cnt++;

p = p -> next;

}

return cnt;

}

infotype get(int i, pointer p){

pointer q = p;

int k = i;

if(k == 0){

q = p -> next;

return (q -> info);

}else{

while(k != 1){

p = p -> next;

k--;

}

}

return (p -> info);

}

void set(int i, infotype x, pointer p){

pointer q = p;

int k = i;

if(k == 0){

q = p -> next;

q -> info = x;

}else{

while(k != 1){

p = p -> next;

k--;

}

}

p -> info = x;

}

void swap(int i, int j, pointer p){

infotype tmp = get(i, p);

set(i, get(j, p), p);

set(j, tmp, p);

}

//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;

}

}

void selectionSort(pointer p){

int i, j;

for(i = 1; i < 11; i++){

int min\_pos = i;

for(j = i+1; j < 11; j++){

if(get(j, p) < get(min\_pos, p))

min\_pos = j;

}

swap(i, min\_pos, p);

}

}