// HW4\_3

#include <stdio.h>

#include <stdlib.h>

#include <memory.h>

#include <string.h>

#define TRUE 1

#define FALSE 0

#define MAX\_STRING 100

typedef struct {

int id;

char name[MAX\_STRING];

char tel[MAX\_STRING];

char dept[MAX\_STRING];

} element;

typedef struct TreeNode {

element data;

struct TreeNode \*left, \*right;

} TreeNode;

TreeNode \* search(TreeNode \* node, int key) // 조금 고쳐야

{ // 코드 }

TreeNode \* new\_node(element data) // 조금 고쳐야

{ // 코드 }

TreeNode \* insert\_node(TreeNode \* node, element data) // 조금 고쳐야

{ // 코드 }

TreeNode \* min\_value\_node(TreeNode \* node) // 그대로

{

TreeNode \* current = node;

// 맨 왼쪽 단말 노드를 찾으러 내려감

while (current->left != NULL)

current = current->left;

return current;

}

TreeNode \* delete\_node(TreeNode \* root, int key) // 조금 고쳐야

{ // 코드 }

//한 학생 정보 출력

void print\_data(element \* e)

{

printf("-----------------------------------\n");

printf("학번: %d\n", e->id);

printf("이름: %s\n", e->name);

printf("전화번호: %s\n", e->tel);

printf("학과: %s\n", e->dept);

}

// 중위 순회

void inorder(TreeNode \* root) // 조금 고쳐야

{ // 코드 }

int get\_node\_count(TreeNode \*node)

{

int count = 0;

if (node != NULL)

count = 1 + get\_node\_count(node->left) + get\_node\_count(node->right);

return count;

}

int main(void)

{

TreeNode \*root = NULL;

TreeNode \*node;

element data;

char choice;

int id;

char name[MAX\_STRING];

char tel[MAX\_STRING];

char dept[MAX\_STRING];

printf("Enter i(nsert), d(elete), s(earch), p(rint), c(ount), q(uit):");

scanf("%c", &choice);

while (choice != 'q') {

switch (choice) {

case 'i':

printf("학번 입력:");

scanf("%d", &id);

printf("이름 입력:");

scanf("%s", name);;

printf("전화번호 입력:");

scanf("%s", tel);;

printf("학과 입력:");

scanf("%s", dept);;

data.id = id;

strcpy(data.name, name);

strcpy(data.tel, tel);

strcpy(data.dept, dept);

root = insert\_node(root, data);

break;

case 'p':

printf("학생 정보 학번 순 출력\n");

inorder(root);

printf("\n");

break;

case 'd':

printf("삭제할 학번 입력:");

scanf("%d", &id);

root = delete\_node(root, id);

break;

case 's':

printf("탐색할 학번 입력:");

scanf("%d", &id);

node = search(root, id);

if (node != NULL)

print\_data(&(node->data));

else

printf("id가 %d인 학생은 없읍니다.\n", id);

break;

case 'c':

printf("현재 저장된 학생의 총 수는 %d\n", get\_node\_count(root));

break;

}

while (getchar() != '\n'); // 버퍼 비움

printf("Enter i(nsert), d(elete), s(earch), p(rint), c(ount), q(uit):");

scanf("%c", &choice);

}

}