/\*

240111 1번문제 이진 검색 트리 구축과 응용

10-11주차 pdf 참조, 381p 9-16

학생정보 트리 탐색 예제 프로그램

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//////////////////////////////////////////////////////////////////////////////////////////////////

//2. 이진 트리의 노드 구조 정의를 위한 선언

typedef struct node {

struct node \*left;

int eid;

int syear;

char grade;

struct node \*right;

}ENODE;

int insert\_node(ENODE \*root, int num, int year, char score);

ENODE \*search(ENODE \*root, int key);

void inorder(ENODE \*root);

void year\_search(ENODE \*root, int year);

void main(int argc, char\* argv[])

{

ENODE\* tree = NULL;

ENODE\* ptr;

int id, year;

char score;

FILE\* input;

if ((input = fopen(argv[1], "r")) == NULL) {

printf("Invaild Open input file...\n");

exit(1);

}

while (fscanf(input, "%d %d %c", &id, &year, &score) != EOF) {

if (!tree)

{

tree = (ENODE\*)malloc(sizeof(ENODE));

tree->eid = id;

tree->syear = year;

tree->grade = score;

tree->left = tree->right = NULL;

}

else

insert\_node(tree, id, year, score);

}

printf("\n-----트리에 구축된 사원정보-----\n");

inorder(tree);

printf("\n 입사년도: ");

scanf("%d", &year);

year\_search(tree, year);

printf("\n 사원 번호: ");

scanf("%d", &id);

ptr = search(tree, id);

if (ptr)

printf("%d번 사원의 정보 : %d\t%c\n", ptr->eid, ptr->syear, ptr->grade);

else

printf("%d번 사원에 대한 정보는 없습니다.\n", id);

}

ENODE \*search(ENODE\* root, int key)

{

ENODE \*tptr = root;

while (tptr) {

if (key < tptr -> eid)

tptr = tptr->left;

else if (key > tptr->eid)

tptr = tptr->right;

else // found

return tptr;

}

return NULL; // not found

}

int insert\_node(ENODE \*root, int num, int year, char score) {

ENODE \*tptr = root, \*before = NULL;

while (tptr) {

if (num < tptr->eid) {

before = tptr;

tptr = tptr->left;

}

else if (num > tptr->eid) {

before = tptr;

tptr = tptr->right;

}

else {

return 0;

}

}

tptr = (ENODE \*)malloc(sizeof(ENODE));

tptr->eid = num;

tptr->syear = year;

tptr->grade = score;

tptr->left = tptr->right = NULL;

if (num < before->eid) before->left = tptr;

else before->right = tptr;

return 1;

}

//이진검색트리의 운행 함수

void inorder(ENODE \*ptr) {

if (ptr) {

inorder(ptr->left);

printf("%d\t%d\t%c\n",ptr->eid, ptr->syear, ptr->grade);

inorder(ptr->right);

}

}

void year\_search(ENODE\* ptr, int year) {

if (ptr) {

year\_search(ptr->left, year);

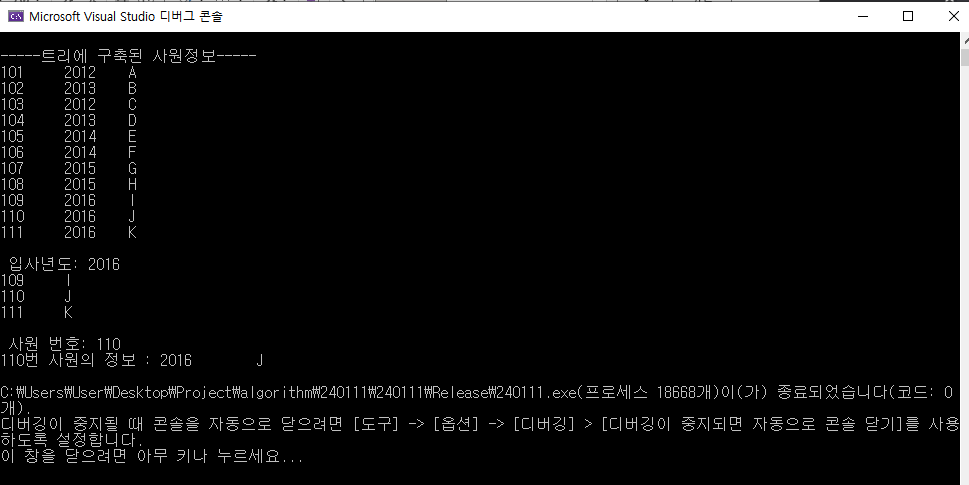
if ((ptr->syear) == year)

printf("%d\t%c\n", ptr->eid, ptr->grade);

year\_search(ptr->right, year);

}

}



/\*

240111 2번문제 정렬 알고리즘 설계 및 구현

버블 정렬(bubble sort) 398p

1) 1회전에 마지막 인덱스의 값이 결정된다

2) 2회전에 끝에서 두번째 인덱스의 값이 결정된다

3) 3회전에 끝에서 세번째 인덱스의 값이 결정된다.

삽입 정렬(insertion sort)

퀵 정렬(quick sort)

이진 합병 정렬(2-way merge sort)

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//////////////////////////////////////////////////////////////////////////////////////////////////

void bubble(int a[], int n);

main()

{

int list[] = { 12, 17, 25, 33, 48, 52, 86, 37, 100 };

int i, n;

n = sizeof(list) / sizeof(int);

bubble(list, n);

printf("정렬된 데이터 리스트 : \n");

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

printf("%d ", list[i]);

}

void bubble(int a[], int n)

{

int i = n - 2, j, tmp;

while (i != 0) {

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

if (a[j] > a[j + 1]) {

tmp = a[j];

a[j] = a[j + 1];

a[j + 1] = tmp;

}

}

i--;

} //while

}



void bubble\_flag(int a[], int n) //플래그 추가 함수 -> 한번의 pass동안 교환이 이루어지지 않으면 끝내도록 개선

{

int i = n-2, j, tmp, flag = 1;

while (flag && i != 0) {

flag = 0;

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

if (a[j] > a[j+1]) {

flag = 1;

tmp = a[j];

a[j] = a[j + 1];

a[j + 1] = tmp;

} //if

} //for

i--;

} //while

}

/\*

240111 2번문제 정렬 알고리즘 설계 및 구현

버블 정렬(bubble sort) 398p

1) 1회전에 마지막 인덱스의 값이 결정된다

2) 2회전에 끝에서 두번째 인덱스의 값이 결정된다

3) 3회전에 끝에서 세번째 인덱스의 값이 결정된다.

삽입 정렬(insertion sort)

퀵 정렬(quick sort)

이진 합병 정렬(2-way merge sort)

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define INUM 100

//////////////////////////////////////////////////////////////////////////////////////////////////

int bsearch\_stock(unsigned a[][2], int n, unsigned key);

void print\_item(char\* heading, unsigned a[][2], int n);

void bubble(unsigned a[][2], int n);

void main(int argc, char \*argv[])

{

FILE\* stockdb;

unsigned stock[INUM][2], item\_code;

int k = 0, s;

if ((stockdb = fopen(argv[1], "r")) == NULL) {

printf("Cannot open read file...\n"); exit(1);

}

while ((fscanf(stockdb, "%u %u", &stock[k][0], &stock[k][1])) != EOF) k++;

print\_item("임의로 저장된 물품코드: \n", stock, k);

//물품코드의 내림차순으로 정렬

bubble(stock, k);

print\_item("내림차순으로 정렬된 물품코드: \n", stock, k);

//검색

printf("검색할 item number 입력: ");

scanf("%u", &item\_code);

s = bsearch\_stock(stock, k, item\_code);

if (s == -1) printf("재고물품이 없습니다\n");

else printf("%u의 재고개수 = %u\n", stock[s][0], stock[s][1]);

}

void print\_item(char\* heading, unsigned a[][2], int n)

{

int i;

printf(heading);

for (i = 0; i < n; i++) printf("%u\n", a[i][0]);

}

int bsearch\_stock(unsigned a[][2], int n, unsigned key)

{

int left = 0, right = n - 1, mid;

while (left <= right) {

mid = (left + right) / 2;

if (key < a[mid][0]) left = mid + 1;

else return mid;

} //while

return -1;

} //search

void bubble(unsigned a[][2], int n)

{

int i = n - 1, j, flag = 1;

unsigned tmp0, tmp1;

while (flag && i != 0) {

flag = 0;

for (j = 0; j <= i-1; j++) {

if (a[j][0] > a[j + 1][0]) {

flag = 1;

tmp0 = a[j][0]; a[j][0] = a[j + 1][0]; a[j + 1][0] = tmp0;

tmp1 = a[j][1]; a[j][1] = a[j + 1][1]; a[j + 1][1] = tmp1;

} //if

} //for

i--;

} //while

}



/\*

240111 4번문제 정렬 알고리즘 설계 및 구현

버블 정렬(bubble sort) PPT 내림차순으로 바꾸어 실행해보기

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//////////////////////////////////////////////////////////////////////////////////////////////////

void bubble(int a[], int n);

main()

{

int list[] = { 12, 17, 25, 33, 48, 52, 86, 37, 100 };

int i, n;

n = sizeof(list) / sizeof(int);

bubble(list, n);

printf("정렬된 데이터 리스트 : \n");

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

printf("%d ", list[i]);

}

void bubble(int a[], int n)

{

int i = n - 2, j, tmp;

while (i != 0) {

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

if (a[j] < a[j + 1]) { // a[0] < a[1]는 내림차순, a[0] > a[1]는 오름차순

tmp = a[j];

a[j] = a[j + 1];

a[j + 1] = tmp;

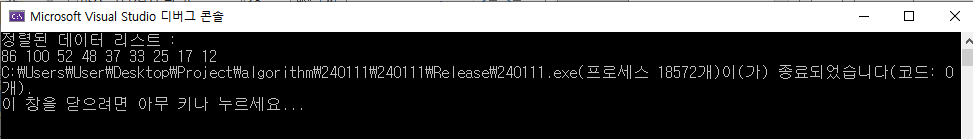
}

}

i--;

} //while

}



/\*

240111 1번문제 이진 검색 트리 구축과 응용

10-11주차 pdf 참조, 381p 9-16

학생정보 트리 탐색 예제 프로그램

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//////////////////////////////////////////////////////////////////////////////////////////////////

//2. 이진 트리의 노드 구조 정의를 위한 선언

typedef struct student {

long id;

char name[20];

double score;

}sdatatype;

void insert\_sort1(sdatatype a[], int n);

void insert\_sort2(sdatatype a[], int n);

void main(int argc, char \*argv[])

{

sdatatype s[50];

FILE \*infile, \*out;

int i = 0, j;

char choice;

if (argc != 3) {

printf("실행인수를 잘못 주었습니다. \n");

exit(1);

}

if ((infile = fopen(argv[1], "r")) == NULL) {

printf("입력파일을 열 수 없습니다. \n");

exit(1);

}

if ((out = fopen(argv[2], "w")) == NULL) {

printf("결과 파일을 열 수 없습니다. \n");

exit(1);

}

while (fscanf(infile, "%ld %s %lf", &(s[i].id), s[i].name, &(s[i].score)) != EOF) i++;

printf("학번순으로 저장하려면 1을 \n");

printf("성적순으로 저장하려면 2를 선택하시오: ");

scanf("%c", &choice);

if (choice == '1') insert\_sort1(s, i);

else if (choice == '2') insert\_sort2(s, i);

else {

printf("선택번호를 잘못 입력하였습니다. \n");

exit(1);

}

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

fprintf(out, "%ld\t%s\t%.2f\n", (s[j].id, s[j].name, (s[j].score)));

fprintf(out, "%ld\t%s\t%.2f\n", (s[j].id, s[j].name, (s[j].score)));

}

printf("Program terminated... \n");

}

void insert\_sort1(sdatatype a[], int n)

{

int i, j;

sdatatype temp;

for (i = 1; i <= n - 1; i++) {

temp = a[i];

j = i - 1;

while (a[j].id > temp.id && j >= 0) {

a[j + 1] = a[j];

j--;

}

a[j + 1] = temp;

}

}

void insert\_sort2(sdatatype a[], int n)

{

int i, j;

sdatatype temp;

for (i = 1; i <= n - 1; i++) {

temp = a[i];

j = i - 1;

while (a[j].id < temp.id && j >= 0) {

a[j + 1] = a[j];

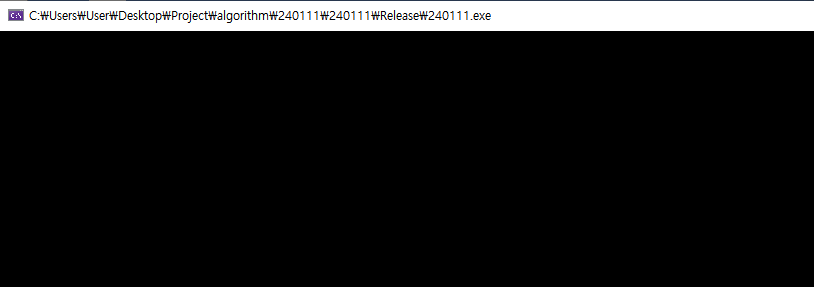
j--;

}

a[j + 1] = temp;

}

}



/\*

240111 1번문제 이진 검색 트리 구축과 응용

10-11주차 pdf 참조, 381p 9-16

학생정보 트리 탐색 예제 프로그램

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <string.h>

void insertion\_string(char a[][10], int n);

//////////////////////////////////////////////////////////////////////////////////////////////////

main()

{

char list[][10] = { "name", "is", "woo", "hyuk","my" };

int i, n;

n = sizeof(list) / sizeof(list[0]);

insertion\_string(list, n);

printf("정렬된 데이터 리스트 : \n");

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

printf("%s ", list[i]);

}

}

void insertion\_string(char a[][10], int n) {

int i, j;

char idata[10];

for (i = 1; i <= n - 1; i++) {

strcpy(idata, a[i]);

j = i - 1;

while (strcmp(a[j], idata) > 0 && j >= 0) {

strcpy(a[j + 1], a[j]); j--;

}

strcpy(a[j + 1], idata);

}

}

