최범수 9일차 과제

1. HW_001

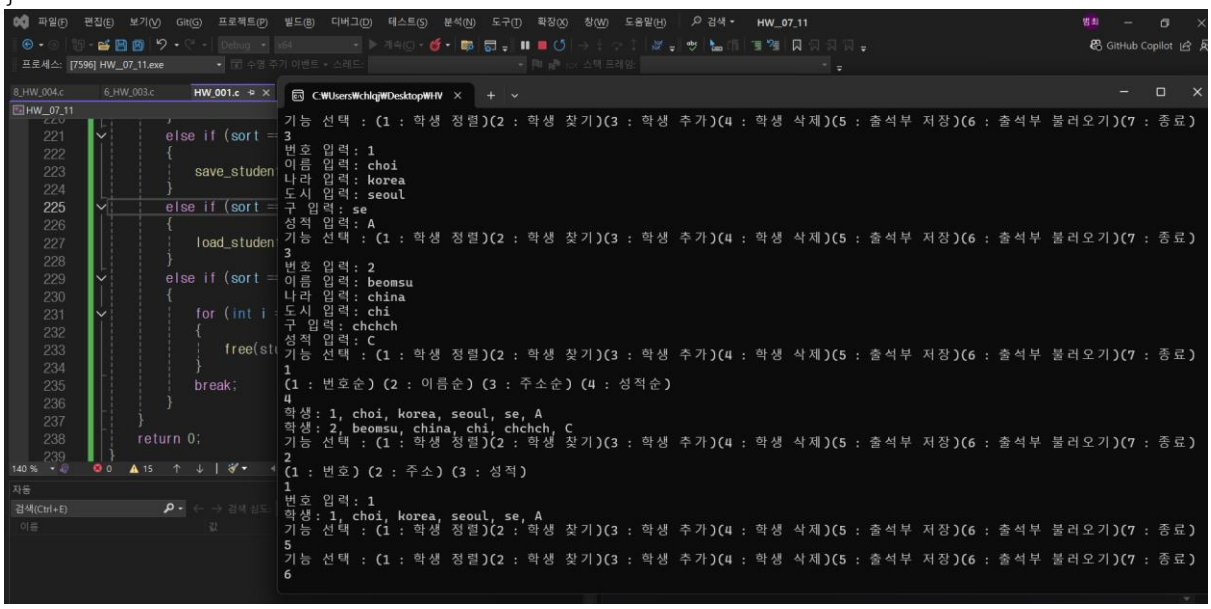
```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct Student{
    int number;
    char name[30];
    char country[30];
    char city[30];
    char gu[30];
    char grade;
} student;

student* students[50];
int student_count = 0;

void print_students()
{
    for (int i = 0; i < student_count; i++)
    {
        printf("학생: %d, %s, %s, %s, %s, %c₩n", students[i]->number, students[i]->name,
students[i]->country, students[i]->city, students[i]->gu, students[i]->grade);
    }
}

void sort_student(int sort)
{
    int i, j;
    student* tmp_sort;
    for (i = 0; i < student_count - 1; i++)
    {
        for (j = i + 1; j < student_count; j++)
        {
            int tmp = 0;
            switch (sort)
            {
                case 1: tmp = students[i]->number - students[j]->number; break;
                case 2: tmp = strcmp(students[i]->name, students[j]->name); break;
                case 3:tmp = strcmp(students[i]->country, students[j]->country);

                if (tmp == 0)
                {
                    tmp = strcmp(students[i]->city, students[j]->city);
                    if (tmp == 0) {
                        tmp = strcmp(students[i]->gu, students[j]->gu);
                    }
                }
                break;
```

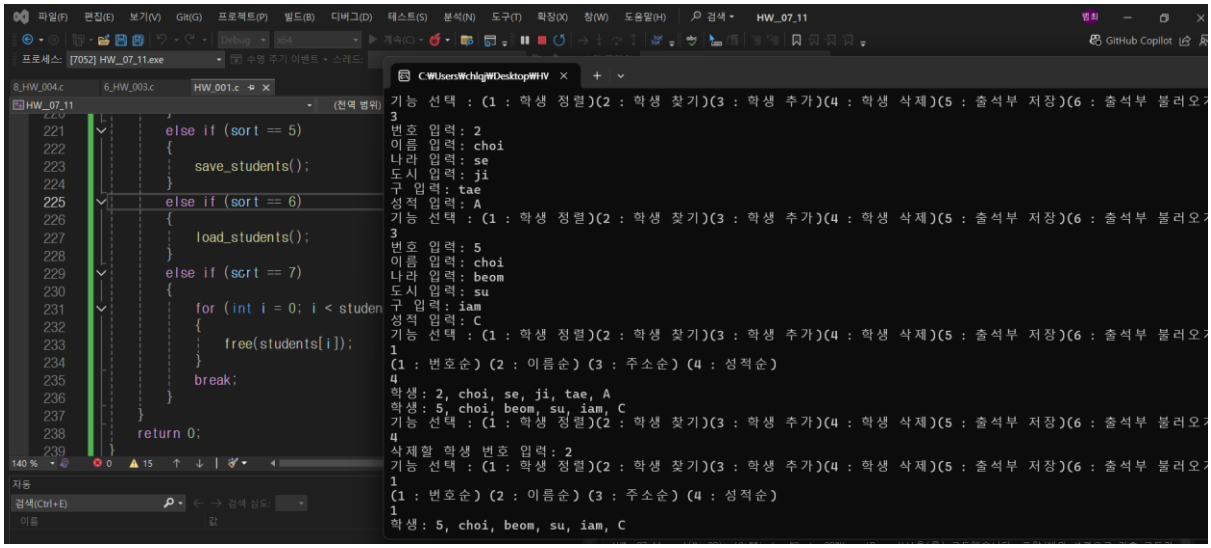
```
                case 4: tmp = students[i]->grade - students[j]->grade; break;
            }
            if (tmp > 0)
            {
                tmp_sort = students[i];
                students[i] = students[j];
                students[j] = tmp_sort;
            }
        }
    }
    print_students();
}

void find_student()
{
    int sort, num = 0;
    char sentence[100];
    printf("(1 : 번호) (2 : 주소) (3 : 성적)₩n");
    scanf("%d", &sort);

    if (sort == 1)
    {
        printf("번호 입력: ");
        scanf("%d", &num);
        for (int i = 0; i < student_count; i++)
        {
            if (students[i]->number == num)
            {
                printf("학생: %d, %s, %s, %s, %s, %c₩n", students[i]->number, students[i]-
>name, students[i]->country, students[i]->city, students[i]->gu, students[i]->grade);
            }
        }
    }
    else if (sort == 2)
    {
        printf("주소 입력: ");
        scanf("%s", sentence);
        for (int i = 0; i < student_count; i++)
        {
            if (strstr(students[i]->country, sentence) != NULL || strstr(students[i]->city,
sentence) != NULL || strstr(students[i]->gu, sentence) != NULL) {
                printf("학생: %d, %s, %s, %s, %s, %c₩n", students[i]->number, students[i]-
>name, students[i]->country, students[i]->city, students[i]->gu, students[i]->grade);
            }
        }
    }
    else if (sort == 3)
    {
        printf("성적 입력: ");
        scanf(" %c", sentence);
        for (int i = 0; i < student_count; i++)
        {
            if (students[i]->grade == sentence[0])
            {
                printf("학생: %d, %s, %s, %s, %s, %c₩n", students[i]->number, students[i]-
```

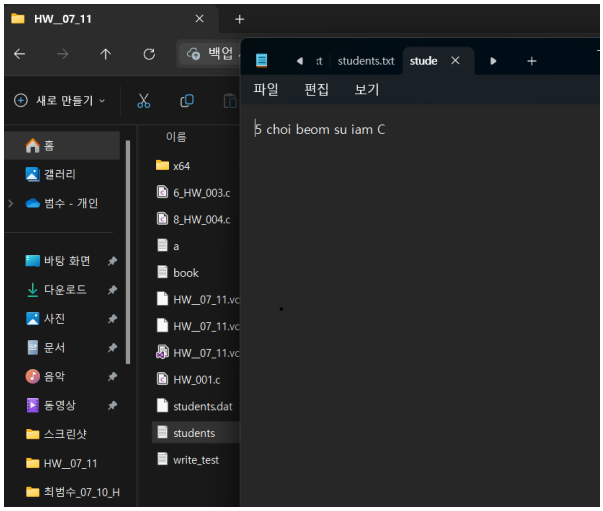
```
>name, students[i]->country, students[i]->city, students[i]->gu, students[i]->grade);
            }
        }
    }
}

void add_student()
{
    if (student_count < 50)
    {
        students[student_count] = (student*)malloc(sizeof(student));
        if (students[student_count] != NULL)
        {
            printf("번호 입력: ");
            scanf("%d", &students[student_count]->number);
            printf("이름 입력: ");
            scanf("%s", students[student_count]->name);
            printf("나라 입력: ");
            scanf("%s", students[student_count]->country);
            printf("도시 입력: ");
            scanf("%s", students[student_count]->city);
            printf("구 입력: ");
            scanf("%s", students[student_count]->gu);
            printf("성적 입력: ");
            scanf(" %c", &students[student_count]->grade);
            student_count++;
        }
    }
    else
    {
        printf("학생 수가 가득 찼습니다.₩n");
    }
}

void delete_student()
{
    int num;
    printf("삭제할 학생 번호 입력: ");
    scanf("%d", &num);
    for (int i = 0; i < student_count; i++)
    {
        if (students[i]->number == num)
        {
            free(students[i]);
            for (int j = i; j < student_count - 1; j++)
            {
                students[j] = students[j + 1];
            }
            student_count--;
            break;
        }
    }
}

void save_students()
```

```
{
    FILE* file = fopen("students.txt", "w");
    if (file != NULL)
    {
        for (int i = 0; i < student_count; i++)
        {
            fprintf(file, "%d %s %s %s %s %c₩n", students[i]->number, students[i]->name,
students[i]->country, students[i]->city, students[i]->gu, students[i]->grade);
        }
        fclose(file);
    }
}

void load_students()
{
    FILE* save = fopen("students.txt", "r");
    if (save != NULL)
    {
        student_count = 0;
        while (1) {
            students[student_count] = (student*)malloc(sizeof(student));
            if (fscanf(save, "%d %s %s %s %s %c", &students[student_count]->number,
students[student_count]->name, students[student_count]->country, students[student_count]->city,
students[student_count]->gu, &students[student_count]->grade) != -1)
            {
                student_count++;
            }
            else
            {
                free(students[student_count]);
                break;
            }
        }
        fclose(save);
    }
    else
    {
        printf("불가₩n");
    }
}

int main()
{
    while (1)
    {
        int sort;
        printf("기능 선택 : (1 : 학생 정렬)(2 : 학생 찾기)(3 : 학생 추가)(4 : 학생 삭제)(5 :
출석부 저장)(6 : 출석부 불러오기)(7 : 종료)₩n");
        scanf("%d", &sort);

        if (sort == 1)
        {
            int num;
            printf("(1 : 번호순) (2 : 이름순) (3 : 주소순) (4 : 성적순)₩n");
            scanf("%d", &num);
```

```
            sort_student(num);
        }
        else if (sort == 2)
        {
            find_student();
        }
        else if (sort == 3)
        {
            add_student();
        }
        else if (sort == 4)
        {
            delete_student();
        }
        else if (sort == 5)
        {
            save_students();
        }
        else if (sort == 6)
        {
            load_students();
        }
        else if (sort == 7)
        {
            for (int i = 0; i < student_count; i++)
            {
                free(students[i]);
            }
            break;
        }
    }
    return 0;
}
```




HW_07_11
else if (sort == 5)
{
save_students();
}
else if (sort == 6)
{
load_students();
}
else if (sort == 7)
{
for (int i = 0; i < studen
{
free(students[i]);
}
break;
}
}
return 0;
}
기능 선택 : (1 : 학생 정렬)(2 : 학생 찾기)(3 : 학생 추가)(4 : 학생 삭제)(5 : 출석부 저장)(6 : 출석부 불러오
3
번호 입력: 2
이름 입력: choi
나라 입력: se
도시 입력: ji
구 입력: tae
성적 입력: A
기능 선택 : (1 : 학생 정렬)(2 : 학생 찾기)(3 : 학생 추가)(4 : 학생 삭제)(5 : 출석부 저장)(6 : 출석부 불러오
3
번호 입력: 5
이름 입력: choi
나라 입력: beom
도시 입력: su
구 입력: iam
성적 입력: C
기능 선택 : (1 : 학생 정렬)(2 : 학생 찾기)(3 : 학생 추가)(4 : 학생 삭제)(5 : 출석부 저장)(6 : 출석부 불러오
1
(1 : 번호순) (2 : 이름순) (3 : 주소순) (4 : 성적순)
4
학생: 2, choi, se, ji, tae, A
학생: 5, choi, beom, su, iam, C
기능 선택 : (1 : 학생 정렬)(2 : 학생 찾기)(3 : 학생 추가)(4 : 학생 삭제)(5 : 출석부 저장)(6 : 출석부 불러오
4
삭제할 학생 번호 입력: 2
기능 선택 : (1 : 학생 정렬)(2 : 학생 찾기)(3 : 학생 추가)(4 : 학생 삭제)(5 : 출석부 저장)(6 : 출석부 불러오
1
(1 : 번호순) (2 : 이름순) (3 : 주소순) (4 : 성적순)
1
학생: 5, choi, beom, su, iam, C


HW_07_11
students.txt
stude
파일 편집 보기
5 choi beom su iam C
새로 만들기
홈
갤러리
범수 - 개인
바탕 화면
다운로드
사진
문서
음악
동영상
스크린샷
HW_07_11
이름
x64
6_HW_003.c
8_HW_004.c
a
book
HW_001.c
students.dat
students
write_test