

최범수 5일차 과제

HW_001

```
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>

//구조체
typedef struct Student
{
    int number;
    char name[10];
    double grade;
} student;

int main()
{
    //구조체 불러오는 부분
    student s[5];
    //임시 저장 공간
    student tmp;

    printf("입력 : Wn");

    //5번 반복
    for (int i = 0; i < 5; i++)
    {
        //각각의 학년 성적 이름 입력받기
        scanf("%d %lf %s", &s[i].number, &s[i].grade, s[i].name);
    }

    //어떤 값이 크고 작은지 비교하는 과정
    for (int i = 0; i < 4; i++)
    {
        for (int j = i + 1; j < 5; j++)
        {
            //학년이 다르면 큰 값을 뒤에
            if (s[i].number > s[j].number)
            {
                tmp = s[i];
                s[i] = s[j];
                s[j] = tmp;
            }
            //학년이 같다면
            else if (s[i].number == s[j].number)
            {
                //성적이 다르면 큰 값을 뒤에
                if (s[i].grade > s[j].grade)
                {
                    tmp = s[i];
                    s[i] = s[j];
                    s[j] = tmp;
                }
            }
        }
    }
}
```

```

        //성적이 같다면
        else if (s[i].grade == s[j].grade)
        {
            for (int k = 0; k < 10; k++)
            {
                //이름의 총 자리수를 10칸으로 제한하고 각 자리수의 알파벳 순서대로
                //첫번째 알파벳이 같으면 두번째 알파벳 비교
                if ((int)s[i].name[k] > (int)s[j].name[k])
                {
                    tmp = s[i];
                    s[i] = s[j];
                    s[j] = tmp;
                }
            }
        }
    }
}

printf("출력 : Wn");
//최종 결과대로 학년 순 성적 순 이름 순으로 출력
for (int i = 0; i < 5; i++)
{
    printf("%d %.1lf %sWn", s[i].number, s[i].grade, s[i].name);
}

return 0;
}

```

The screenshot shows the Microsoft Visual Studio IDE. The main window displays the source code for a C program named `HW_07_05.c`. The code defines a `Student` struct with fields `number`, `name`, and `grade`, and implements a `main` function that reads student data and prints it. The code is partially commented in Korean. The output window on the right shows the program's execution results, displaying the input and output of the `main` function. The output shows the input data and the sorted output data.

```

1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <stdio.h>
4
5  //구조체
6  typedef struct Student
7  {
8      int number;
9      char name[10];
10     double grade;
11 } student;
12
13 int main()
14 {
15     //구조체 불러오는 부분
16     student s[5];
17     //임시 저장 공간
18     student tmp;
19
20     printf("입력 : Wn");
21

```

Microsoft Visual Studio 디버그

입력 :
4 4.5 kwon
1 3.2 jo
3 2.3 yoon
3 1.6 koh
3 2.3 park
출력 :
1 3.2 jo
3 1.6 koh
3 2.3 park
3 2.3 yoon
4 4.5 kwon

C:\Users\chlqj\Desktop\HW_07_05\x64\Debug\HW_07_05.exe 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] 하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요 ...|

입력 :

```
1 3.2 jo
4 3.2 kwon
4 3.7 kim
5 3.1 kiin
5 4.2 lom
```

출력 :

```
1 3.2 jo
4 3.2 kwon
4 3.7 kim
5 3.1 kiin
5 4.2 lom
```

C:\Users\chlqj\Desktop\HW__07_05\x64\Debug\HW_...
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...]

C언어 - 5일차.pdf

파일 | C:/Users/chlqj/Documents...

즐거찾기 가져오기 | 과고생의 물리1 수... | Adobe Acrobat Pro... | Adobe Premiere Pr...

19

```
}
int isEqualGrade(Student s1, Student s2)
{
    if(s1.grade == s2.grade)
        return 1;
    else
        return 0;
}

```

BIT 수습단원 교육자료

과제 1

5명의 학년, 성적, 이름을 받는 구조체를 만들고, 학년이 같으면
3가지 요소를 모두 고려하여 오름차순으로 정렬하시오.

입력 : 4 4.5 kwon
1 3.2 jo
3 2.3 yoon
3 1.6 koh
3 2.3 park

Microsoft Visual Studio 디버그

입력 :

```
2 3.7 jason
1 4.2 bs
3 5.1 kin
5 4.2 joon
4 1.1 john
```

출력 :

```
1 4.2 bs
2 3.7 jason
3 5.1 kin
4 1.1 john
5 4.2 joon
```

C:\Users\chlqj\Desktop\HW__07_05\x64\Debug\HW_...
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...]

HW_002

```
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>
#include <stdlib.h>
#include <math.h> //제공된 구하는 함수 - 5일차 실습예제 3번

//구조체 선언
typedef struct Loc
{
    int x, y;
    double distance;
} loc;

int main()
{
    //x거리 차이, y거리 차이 선언
    int x_dis;
    int y_dis;

    //구조체 포인터 선언
    loc* location;

    int num = 0;
    int tmp_num = 0;
    printf("입력 : ");
    scanf("%d", &num);
    //구조체 형의 포인터를 사용해서 구조체 사이즈 * 입력받은 숫자 만큼의 크기 할당
    location = (loc*)malloc(sizeof(loc) * num);

    //첫번째 위치 사이의 거리를 0으로 초기화해주는 과정
    for (int i = 0; i < num; i++)
    {
        location[i].distance = 0;
    }
    //그 후 나머지 값들 입력받기
    for (int i = 0; i < num; i++)
    {
        scanf("%d %d", &location[i].x, &location[i].y);
    }
    //x끼리의 거리 차이와 y끼리의 거리 차이를 변수에 저장, 두 좌표 사이의 거리만 배열에
    저장
    for (int i = 0; i < num; i++)
    {
        for (int j = 0; j < num; j++)
        {
            x_dis = location[i].x - location[j].x;
            y_dis = location[i].y - location[j].y;
            location[i].distance += sqrt((double)(x_dis * x_dis) + (y_dis *
y_dis));
        }
    }
}
```

```

    }
    //처음 초기화했던 거리를 max에 입력
    double max = location[0].distance;
    int max_num = 0;
    //최대값 숫자

    //5개의 점 사이의 거리는 4개이므로 입력받은 숫자보다 한칸 적게
    for (int i = 0; i < num-1; i++)
    {
        //계속 합산해주고 max_num은 1을 더해줘야 최대값임을 알 수 있다.
        if (location[i].distance < location[i + 1].distance)
        {
            max = location[i + 1].distance;
            max_num = i + 1;
        }
    }
    //먼 좌표와 거리 총합 구함
    printf("출력 : 가장 거리가 먼 좌표는 (%d, %d)이며, 다른 좌표의 거리 총합은
    약%.1lf입니다.", location[max_num].x, location[max_num].y, location[max_num].distance);

    free(location);
}

```

Microsoft Visual Studio 디버그

```

입력 : 5
0 0
1 1
2 3
2 1
7 7
출력 : 가장 거리가 먼 좌표는 (7, 7)이며, 다른 좌표의 거리 총합은 약32.6입니다.
C:\Users\chlqj\Desktop\HW_07_05\x64\Debug\HW_07_05.exe(프로세스 15384개)이(가) 종료되
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...|

```

Microsoft Visual Studio 디버그

```

입력 : 4
7 7
6 6
3 3
4 4
출력 : 가장 거리가 먼 좌표는 (3, 3)이며, 다른 좌표의 거리 총합은 약11.3입니다.
C:\Users\chlqj\Desktop\HW_07_05\x64\Debug\HW_07_05.exe(프로세스 15644개)이(가) 종료
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...|

```

```
Microsoft Visual Studio 디버그 × + ▾
입력 : 4
3 3
4 4
6 6
7 7
출력 : 가장 거리가 먼 좌표는 (7, 7)이며, 다른 좌표의 거리 총합은 약11.3입니다.
C:\Users\chlj\j\Desktop\HW_07_05\x64\Debug\HW_07_05.exe(프로세스 9080개)이(가) 종료
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅 이
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요 ...|
```

```
Microsoft Visual Studio 디버그 × + ▾
입력 : 1
5 7
출력 : 가장 거리가 먼 좌표는 (5, 7)이며, 다른 좌표의 거리 총합은 약0.0입니다.
C:\Users\chlj\j\Desktop\HW_07_05\x64\Debug\HW_07_05.exe(프로세스 22920개)이(가)
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요 ...|
```

HW_003

```
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct Counter{
    char name[20];
    int money;
}counter;

int main()
{
    //구조체 포인터 선언
    counter* count;
    int num = 0;
    char name_name[20];
    int sum = 0;

    printf("입력 : ");
    scanf("%d", &num);

    //구조체 형의 포인터를 사용해서 구조체 사이즈 * 입력받은 숫자 만큼의 크기 할당
    count = (counter*)malloc(sizeof(counter) * num);

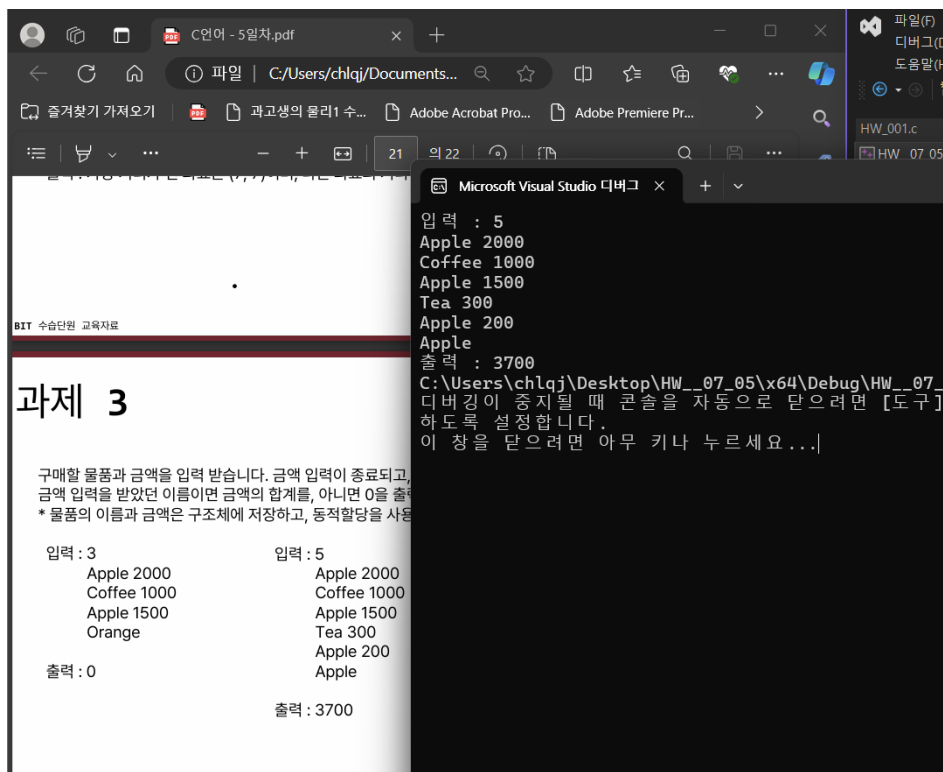
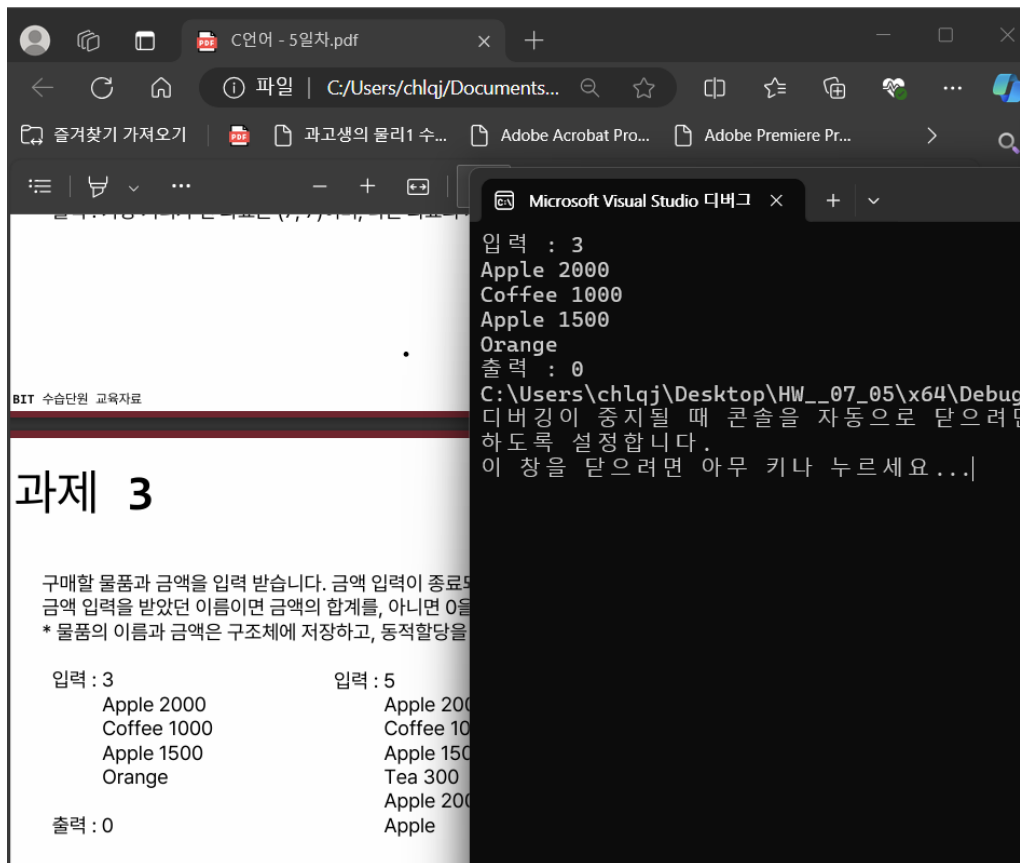
    //scanf로 이름과 가격 입력받기
    for (int i = 0; i < num; i++)
    {
        scanf("%s %d", &count[i].name, &count[i].money);
    }

    //맨 마지막에 이름 입력받기
    scanf("%s", &name_name);

    //가격 총합 계산해주기
    for (int i = 0; i < num; i++)
    {
        //string함수를 이용해서 이름이 같은지 비교
        if (strcmp(name_name, count[i].name) == 0)
        {
            sum += count[i].money;
        }
    }

    //최대 sum 출력
    //이름이 다르면 자연스럽게 sum은 0으로 초기화 되어있으므로 0 출력
    printf("출력 : %d", sum);

    free(count);
}
```



C언어 - 5일차.pdf

← ↻ 🏠 ⓘ 파일 | C:/Users/chlqj/Documents... 🔍 ☆ 📄 📌 📁 📂 📅 📆 📇 📈 📉 📊 📋 📌 📍 📎 📏 📐 📑 📒 📓 📔 📕 📖 📗 📙 📚 📛 📜 📝 📞 📟 📠 📡 📢 📣 📤 📥 📦 📧 📨 📩 📪 📫 📬 📭 📮 📯 📰 📱 📲 📳 📴 📵 📶 📷 📸 📹 📺 📻 📼 📽 📾 📿 📠 📡 📢 📣 📤 📥 📦 📧 📨 📩 📪 📫 📬 📭 📮 📯 📰 📱 📲 📳 📴 📵 📶 📷 📸 📹 📺 📻 📼 📿

🔍 즐겨찾기 가져오기 📄 📌 과고생의 물리1 수... 📄 Adobe Acrobat Pro... 📄 Adobe Premiere Pr...

📄 21 의 22 🔍 📄 📌 📁 📂 📅 📆 📇 📈 📉 📊 📋 📌 📍 📎 📏 📐 📑 📒 📓 📔 📕 📖 📗 📙 📚 📛 📜 📝 📞 📟 📠 📡 📢 📣 📤 📥 📦 📧 📨 📩 📪 📫 📬 📭 📮 📯 📰 📱 📲 📳 📴 📵 📶 📷 📸 📹 📺 📻 📼 📿

BIT 수습단원 교육자료

과제 3

구매할 물품과 금액을 입력 받습니다. 금액 입력이 종료되고, 물품의 금액 입력을 받았던 이름이면 금액의 합계를, 아니면 0을 출력하시오.
* 물품의 이름과 금액은 구조체에 저장하고, 동적할당을 사용하여 메모리 관리

입력 : 3	입력 : 5
Apple 2000	Apple 2000
Coffee 1000	Coffee 1000
Apple 1500	Apple 1500
Orange	Tea 300
	Apple 200
출력 : 0	Apple
	출력 : 3700

Microsoft Visual Studio 디버그

입력 : 4
Apple 2000
Apple 1000
Apple 0
Apple 500
Apple
출력 : 3500
C:\Users\chlqj\Desktop\HW_07_05\x64
디버깅이 중지될 때 콘솔을 자동으로 닫
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요.

HW_004

```
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>

// 각 달의 일 수 우선 윤년 계산 X
int month_day[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

// 윤년인지 확인하는 함수
int year_odd(int year) {
    if (year % 4 == 0) {
        if (year % 100 == 0) {
            if (year % 400 == 0) {
                return 1;
            }
            else {
                return 0;
            }
        }
        else {
            return 1;
        }
    }
    else {
        return 0;
    }
}

// 구조체 선언
typedef struct Time
{
    int year;
    int month;
    int day;
    int hour;
    int minute;
    int second;
} time;

// 윤년이면 366 아니면 365
int day_year(time day) {
    if (year_odd(day.year)) {
        return 366;
    }
    else {
        return 365;
    }
}

// 입력받은 날 사이 계산
int day_calcul(time first, time second) {
    int days = 0;
```

```

// 첫번째 입력받은 연도와 두번째 입력받은 연도가 같으면
if (first.year == second.year) {
    //두 날짜 사이의 달 수 계산
    for (int i = first.month - 1; i < second.month - 1; ++i) {
        days += month_day[i];
    }
    //일 수 계산
    days += second.day - first.day;
}
//첫번째 입력받은 연도와 두번째 입력받은 연도가 다르면
else {
    // 시작 날짜의 해당 년도 남은 일 수 계산
    for (int i = first.month - 1; i < 12; ++i) {
        days += month_day[i];
    } //첫번째 달 지난 일수 빼기
    days -= first.day;
    //윤년이고 2월 안지났으면 1일 추가
    if (year_odd(first.year) && first.month <= 2) {
        days += 1;
    }
    //아니면 무시

    // 입력받은 두 연도 사이 연도의 일 수 계산
    for (int i = first.year + 1; i < second.year; ++i)
    {
        //잠시 저장했다가 일수에 더해주기
        time temp_day = { i, 1, 1 };
        days += day_year(temp_day);
    }

    // 끝 날짜의 해당 년도 시작부터 일 수 계산
    for (int i = 0; i < second.month - 1; ++i)
    {
        //지난 달만큼 더해주기
        days += month_day[i];
    }
    days += second.day;
    //윤년이고 2월보다 크면 1일 더해주기
    if (year_odd(second.year) && second.month > 2)
    {
        days += 1;
    }
    //아니면 무시
}

//최종으로 더해진 일을 반환
return days;
}

// 두 날짜 사이의 총 시간 차이를 계산하는 함수
void time_differ(time first, time second, int* total_hours, int* minutes, int* seconds)
{
    //일 수 계산 함수
    int total_days = day_calcul(first, second);

```

```

// 초 단위 계산
//두번째 초 - 첫번째 초
*seconds = second.second - first.second;
if (*seconds < 0) //계산해서 0보다 작으면 60더하고 분에서 -1
{
    *seconds += 60;
    second.minute--;
}

// 분 단위 계산
*minutes = second.minute - first.minute;
if (*minutes < 0) //계산해서 0보다 작으면 60더하고 시간에서 -1
{
    *minutes += 60;
    second.hour--;
}

// 시간 차이 계산
*total_hours = ((total_days + 1) * 24) + (second.hour - first.hour);
if (*total_hours < 0) //계산해서 0보다 작으면 24더하고 일 수 에서 -1
{
    *total_hours += 24;
    total_days--;
}
}

int main()
{
    time day1, day2;
    int total_hours, minutes, seconds;

    printf("입력 : Wn");
    scanf("%d %d %d %d %d %d", &day1.year, &day1.month, &day1.day, &day1.hour, &day1.minute,
    &day1.second);
    scanf("%d %d %d %d %d %d", &day2.year, &day2.month, &day2.day, &day2.hour, &day2.minute,
    &day2.second);

    time_differ(day1, day2, &total_hours, &minutes, &seconds);

    //최종 출력
    printf("출력 : %d시 %d분 %d초", total_hours, minutes, seconds);

    return 0;
}

```

