

## <고급 C프로그래밍 및 실습> 4차 과제 (12장 동적할당)

※ 문제지의 무단 배포 및 사용을 원칙적으로 금지합니다.

- 특히, 커뮤니티, 개인 블로그 등 인터넷 사이트 게시를 절대 금지합니다.

※ 문제에 대한 안내

- 입출력 예시에서  $\mapsto$  이 후는 각 입력과 출력에 대한 설명이다.

- ex-oj에서 Sample Submit 기능 사용가능 합니다.

※ [ 문제 3-1 ]에서 [ 문제 3-2 ]까지 연관된 문제이다. [ 문제 3-1 ] 만 풀면 50점, [ 문제 3-2 ] 만 풀어도 100점이다.

[ 문제 3-1 ] [레벨 1] (50점) 총 N명 학생들의 과목 점수를 관리하는 프로그램이다. n 번째 학생에 대해 이름, 학번, 과목 이름 및 점수를 입력 받아 메모리를 동적으로 할당한 구조체에 저장한다. 학생의 과목 점수가 가장 높은 학생부터 내림차순으로 정렬 한 후 K를 입력 받아 K 번째 학생의 성명, 학번, 과목 이름, 과목 점수를 출력하는 프로그램을 작성하시오.

- 첫 줄에서 N을 입력 받고, 각 학생의 정보를 공백이 있는 N개의 문자열로 입력 받는다.
- 각 문자열의 최대 길이는 100이며, 문자열 내 자료는 공백 한 개로 구분되어 있다.
- 학번은 총 4개의 숫자로 이루어져 있다.
- 과목 점수는 0~100 범위의 정수이다.
- 과목 점수가 같은 학생이 있는 경우에는 먼저 입력된 1명만 출력 하시오.
- 과목 점수를 출력할 때는 서식 지정자로 "%2lf"를 사용하시오.
- 다음과 같은 구조체를 정의해서 사용하고, 구조체 배열과 문자열을 저장 할 배열은 동적으로 할당 받아 사용하고, 프로그램 종료 시 할당된 메모리는 반드시 해제한다.
- 구조체의 멤버 변수명을 수정하는 것 이외의 변경은 할 수 없다. (즉, 멤버 변수의 자료형 변경이나 멤버 변수 추가도 불가)

```
struct student{
    char *studentName;    // 학생 이름
    char id[5];           // 학생 학번
    char *subjectName;    // 과목 이름
    double subjectScore;  // 과목 점수
};
```

입력 예시 1

```
5
HongGilDong 1003 Math 78
JeonWooChi 1001 English 60
HeungBoo 1002 Economics 80
NolBoo 1004 Ethics 50
Euler 1005 Math 100
2
```

$\mapsto$  N

$\mapsto$  K

출력 예시 1

```
HeungBoo 1002 Economics 80.00
```

- ▶ 구조체 배열을 동적으로 할당 받지 않으면 (100% 감점)
- ▶ 구조체 멤버 변수에 문자열을 저장할 때, 저장에 필요한 공간을 동적으로 할당 받지 않거나 문자열 보다 큰 공간을 할당 받으면 (50% 감점)
- ▶ 동적으로 할당 받은 모든 메모리를 해제 하지 않으면 (20% 감점)

[ 문제 3-2 ] [레벨 2] (100점) 총 N명 학생들의 과목별 점수를 관리하는 프로그램이다. n 번째 학생에 대해 이름, 학번, 과목 1의 이름 및 점수, 과목 2의 이름 및 점수, ..., 과목 M<sub>n</sub>의 이름 및 점수를 입력 받아 메모리를 동적으로 할당한 구조체에 저장한다. M<sub>n</sub>은 n 번째 학생이 수강한 과목의 개수이다. 학생이 수강한 모든 과목 점수에 대한 평균을 구조체 멤버 변수 avg에 저장한다. **평균 점수**가 가장 높은 학생부터 **내림차순**으로 정렬 한 후 K를 입력 받아 K 번째 학생의 이름, 학번, 평균 점수를 출력하고 K 번째 학생의 과목 중 가장 점수가 높은 과목의 이름과 점수를 출력하는 프로그램을 작성하시오.

- 첫 줄에서 N을 입력 받고, 각 학생의 정보를 N개의 **공백이 있는 문자열**로 입력 받는다.
- 각 문자열의 최대 길이는 100이며, 문자열 내 자료는 **공백 한 개**로 구분되어 있다.
- 학번은 총 4개의 숫자로 이루어져 있다.
- 과목 점수는 0~100 범위의 정수이다.
- 평균 점수가 같은 학생의 경우에는 학번 순으로 정렬하시오.
- 점수를 출력할 때는 서식 지정자로 "%.2lf"를 사용하시오.
- 한 학생의 **가장 점수가 높은 과목이 2개 이상이면 제일 먼저 입력된 과목** 정보를 출력하시오.
- 다음과 같은 두 구조체를 정의해서 사용하고, 각 구조체 배열과 문자열을 저장 할 배열은 동적으로 할당 받아 사용하고, 프로그램 종료 시 할당된 메모리는 반드시 해제한다.
- 구조체의 멤버 변수명을 수정하는 것 이외의 변경은 할 수 없다. (즉, 멤버 변수의 자료형 변경이나 멤버 변수 추가도 불가.)

```
struct subject{
    char *name;    // 과목 이름
    double score;  // 과목 점수
};
```

```
struct student{
    char *name;    // 학생 이름
    char id[5];    // 학생 학번
    int numSubject; // 과목 개수
    struct subject *psub; // 과목 구조체 포인터
    double avg;    // 모든 과목의 평균 점수
};
```

입력 예시 1

5	↪ N
HongGilDong 1003 Math 85 C 80 AdvC 90	↪ 평균 점수: 85.00
JeonWooChi 1001 English 60 C 75 Physics 80 Math 70	↪ 평균 점수: 71.25
HeungBoo 1002 Economics 80 AdvC 90 Math 80 C 90	↪ 평균 점수: 85.00
NolBoo 1004 Ethics 50 C 62	↪ 평균 점수: 56.00
Euler 1005 Math 100 Physics 90 AdvC 85	↪ 평균 점수: 91.67
2	↪ K

출력 예시 1

HeungBoo 1002 85.00 AdvC 90.00  
 ↪ 학생 이름, 학번, 평균 점수, 가장 점수가 높은 과목 이름, 가장 높은 과목 점수

입력 예시 2

5	↪ N
HongGilDong 1003 Math 85 C 80 AdvC 90	↪ 평균 점수: 85.00
JeonWooChi 1001 English 60 C 75 Physics 80 Math 70	↪ 평균 점수: 71.25
HeungBoo 1002 Economics 80 AdvC 90 Math 80 C 90	↪ 평균 점수: 85.00
NolBoo 1004 Ethics 50 C 62	↪ 평균 점수: 56.00
Euler 1005 Math 100 Physics 90 AdvC 85	↪ 평균 점수: 91.67
4	↪ K

출력 예시 2

JeonWooChi 1001 71.25 Physics 80.00  
 ↪ 학생 이름, 학번, 평균 점수, 가장 점수가 높은 과목 이름, 가장 높은 과목 점수

- ▶ 구조체 배열을 동적으로 할당 받지 않으면 (100% 감점)
- ▶ 구조체 멤버 변수에 문자열을 저장할 때, 저장에 필요한 공간을 동적으로 할당 받지 않거나 문자열 보다 큰 공간을 할당 받으면 (50% 감점)
- ▶ 동적으로 할당 받은 모든 메모리를 해제 하지 않으면 (20% 감점)

※ [ 문제 4-1 ]에서 [ 문제 4-2 ]까지 연관된 문제이다. [ 문제 4-1 ] 만 풀면 50점, [ 문제 4-2 ] 만 풀어도 100점이다.

**[ 문제 4-1 ] (50점)** 세종대 수시모집 전형은 2단계로 구성되어 있고, 이 중 2단계 전형에서는 1단계 점수 70%와 면접 점수 30%를 합산하여 총점이 높은 학생 순으로 신입생을 선발한다. 1단계 합격자는 N명이고, 최종 M명의 신입생을 선발하며, 각 지원자의 이름과 1단계 성적, 면접 점수가 아래 예시와 같이 주어질 때 최종 합격자의 이름과 총점을 출력하는 프로그램을 작성하시오.

- 1단계 합격자의 수 N과 최종 합격자의 수 M이 입력된다.
- N명 지원자의 이름, 1단계 점수, 면접 점수가 순서대로 입력된다.
- 이름은 공백을 포함하지 않는 최대 10자의 문자열이고, 문자열을 저장할 배열의 크기는 문자개

수+1로 설정한다.

- 1단계 점수는 0~100점, 면접 점수는 0~100점의 정수로 입력된다.
- 최종 합격자의 이름, 1단계 점수 70%와 면접 점수 30%가 반영된 총점을 소수점 첫째 자리까지 출력한다. 총점은 부동소수 자료형 double을 사용시오.
- 총점이 높은 합격자부터 낮은 합격자 순으로 출력한다.
- 총점이 같은 동점자는 없다고 가정한다.
- 다음과 같은 구조체를 정의해서 사용하고, 구조체 배열과 문자열을 저장할 배열은 동적으로 할당받아 사용하고, 프로그램 종료 시 할당된 메모리는 반드시 해제한다.
- 아래 주어진 구조체의 멤버 변수명을 수정하는 것 이외의 변경은 할 수 없다. (즉, 멤버 변수의 자료형 변경이나 멤버 변수 추가도 불가)

```
struct student {
    char *name;      // 이름 저장
    int firstStage;  // 1단계 점수 저장
    int interview;   // 면접 점수 저장
    double sum;      // 총점 저장
};
```

- ▶ 구조체 배열을 동적으로 할당받지 않는 경우 (100% 감점)
- ▶ 구조체 멤버 변수에 문자열을 저장할 때, 저장에 필요한 공간을 동적으로 할당받지 않거나 문자열보다 큰 공간을 할당받는 경우 (50% 감점)
- ▶ 동적으로 할당받은 메모리를 해제하지 않는 경우 (20% 감점)

입력 예시 1

출력 예시 1

7 3      ↳ N=7, M=3	
Kim 75 85	Park 90.4
Lee 86 77	Kang 85.1
Park 94 82	Cho 84.0
Choi 78 70	
Jung 65 85	
Kang 83 90	
Cho 90 70	

**[ 문제 4-2 ] (100점)** 세종대 수시모집 전형은 2단계로 구성되어 있고, 이 중 2단계 전형에서는 1단계 점수 70%와 면접 점수 30%를 합산하여 총점이 높은 학생 순으로 신입생을 선발한다. 1단계 합격자는 N명이고, 최종 M명의 신입생을 선발하며, 각 지원자의 이름과 1단계 성적, 면접 점수가 아래 예시와 같이 주어질 때 최종 합격자의 이름과 1단계 점수, 면접 점수, 총점을 출력하고, 최종 합격자의 수도 출력하는 프로그램을 작성하시오. 다른 조건은 [ 문제 1-1 ]과 같으며, 동점자에 대한 선발 기준만 다음과 같이 다르다고 가정한다.

- 최종 합격자의 이름과 총점뿐만 아니라, 1단계 점수, 면접 점수를 모두 출력한다.
- 합격자 명단 아래에 최종 합격자의 수도 함께 출력한다.
- 총점이 같은 동점자가 있으면, 1단계 점수가 더 높은 사람을 선발하고, 1단계 점수가 높은 사

람을 먼저 출력한다.

- **M명 이내의 합격자와 총점 및 1단계 점수가 모두 같은 동점자도 합격자로 선발하고, 이름이 사전 순서상 빠른 사람을 먼저 출력한다.** (이름까지 같은 동점자는 없으며, 대문자 Z가 소문자 a보다 먼저 나온다고 가정)
- **double형 자료의 대소 비교 시, 부동소수점으로 표현된 자료의 오차를 고려하여 비교해야 한다.** (double형 자료 7\*0.3과 3\*0.7을 비교하면 같지 않음에 유의해서 비교할 것)

(힌트: 다음과 같이 계산하시오)

```
p[i].sum = p[i].firstStage * 7 + p[i].interview * 3;
printf("..... %.1fWn", p[i].name, ..... , p[i].sum/10);
```

- ▶ 구조체 배열을 동적으로 할당받지 않는 경우 **(100% 감점)**
- ▶ 구조체 멤버 변수에 문자열을 저장할 때, 저장에 필요한 공간을 동적으로 할당받지 않거나 문자열보다 큰 공간을 할당받는 경우 **(50% 감점)**
- ▶ 동적으로 할당받은 메모리를 해제하지 않는 경우 **(20% 감점)**

입력 예시 1

출력 예시 1

7 3      ↳ N=7, M=3	Park 94 82 90.4
Kim 75 85	Kang 83 90 85.1
<u>Lee 93 63</u>	Lee 93 63 84.0    ↳ 총점이 같은 Lee, Cho 중에서
Park 94 82	3                    1단계 점수가 높은 Lee만 합격하고,
Choi 78 70	Cho는 탈락
Jung 65 85	
Kang 83 90	
<u>Cho 90 70</u>	

입력 예시 2

출력 예시 2

7 3      ↳ N=7, M=3	Park 94 82 90.4
Kim 75 85	Lee 96 56 84.0
<u>Lee 96 56</u>	Cho 93 63 84.0
Park 94 82	Kang 93 63 84.0
Choi 78 70	4      ↳ 총점이 같은 Lee, Kang, Cho 중에서
Jung 65 85	1단계 점수가 높은 Lee부터 합격하고,
<u>Kang 93 63</u>	총점 및 1단계 점수가 모두 같은 Kang, Cho
<u>Cho 93 63</u>	는 모두 합격,
	이 중에서 이름이 빠른 Cho부터 출력,
	총 합격자 수는 동점자를 포함한 4명

## 제출기한 및 방법

- \* OJ시스템(<https://ex-oj.sejong.ac.kr/index.php/auth/login>) 내의 **과제4**를 이용하여 제출
- \* 제출 마감: **11월 X일(X요일)** 밤 12시 까지 제출
- \* 과제 점수는 위 마감일 전에 제출된 가장 마지막 코드를 기준으로 부여

(마감일 이후에 제출되는 코드는 채점에서 제외됨)

\* OJ시스템의 점수는 참고로만 사용

- 문제의 조건을 만족시켜서 코딩했는지 조사 후 최종 점수 부여
- 코드 2줄당 1줄의 비율로 반드시 주석을 적을 것(주석이 없을 경우 감점)
- 보고서 등 기타 제출물 없음