

〈고급 C프로그래밍 및 실습〉 3차 과제 (11장 구조체)

※ 문제지의 무단 배포 및 사용을 원칙적으로 금지합니다.

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

※ 문제에 대한 안내

- 입출력 예시에서 ↳ 이 후는 각 입력과 출력에 대한 설명이다.

- ex-oj에서 Sample Submit 기능사용 가능하다.

※ [문제 1-1]에서 [문제 1-2]까지 연관된 문제이다. [문제 1-1] 만 풀면 50점, [문제 1-2] 까지 모두 풀면 100점이다. 가장 높은 점수 하나만 반영한다.

[문제 1-1] [레벨 1] (50점) 5명 학생의 이름과 각 학생의 학번 및 지난학기 평균학점을 입력받는다. 그다음 찾고자 하는 학생의 이름을 입력하면, 이 학생의 학번과 지난학기 평균학점을 출력하시오.

- 한 학생의 정보는 다음과 같다. (struct student)
 - 이름 : 공백을 포함하지 않고 길이가 최대 9인 문자열 (널 문자까지 10 바이트만 선언하기)
 - 학번 : 공백을 포함하지 않고 길이가 10인 문자열이고, 첫 4자리는 입학년도를 의미
 - 지난학기 평균학점 : A, B, F 중 한 개의 학점 (문자)
- 학생들의 이름은 모두 다르고, 대소문자를 구분한다. (예: Lee와 lee는 다른 학생이다)
- 학생들의 입학년도는 2017년에서 2019년 사이로 가정한다.

입력 예시1

출력 예시1

lee1 2017102656 A lee2 2018111823 A park 2018059961 A choi 2018010101 F lee3 2019565656 B park	<input type="checkbox"/> 2018059961 A
---	---------------------------------------

[문제 1-2] [레벨 1] (100점) 5명 학생의 이름과 각 학생의 학번 및 지난학기 평균학점을 입력받는다. 그다음 찾고자 하는 학생의 이름을 입력하면, 이 학생과 같은 학점을 받은 입학년도가 같은 다른 학생들의 이름을 (입력 순서대로) 첫 줄에 모두 출력하시오. 조건에 맞는 학생이 없다면 0을 출력하시오. 그다음, 찾고자 하는 학생의 입학년도와 입학년도에 131을 나눈 나머지를 출력하시오.

Hint: `strncpy(str_year, str_id, n);` // `char str_year[5];`

`int num = atoi(str_year) % 131 ;` // `#include <stdlib.h>` 필요

입력 예시1

출력 예시1

lee1 2017102656 A <u>lee2 2018111823 A</u> <u>kwon 2018111888 A</u> <u>park 2018059961 A</u> choi 2018010101 B <u>park</u>	lee2 kwon ↳ 첫 줄에 이름 모두 출력 2018 53
---	--------------------------------------

8308

입력 예시2

```
lee1 2019102656 B
lee2 2019111823 B
park 2019123456 A
choi 2019010101 F
lee3 2019565656 B
park
```

출력 예시2

```
0
2019 54
```

※ [문제 2-1]에서 [문제 2-2]까지 연관된 문제이며, 배점이 각각 50, 100점이다. [문제 2-1]을 안 풀고 [문제 2-2]만 풀어도 100점을 부여한다. 가장 높은 배점 하나만 반영한다. 합산하지 않는다.

※ [문제 2-1]과 [문제 2-2] 모두 정의된 함수 내에서는 구조체 배열에 대한 포인터 표현을 사용한다. 단, main 함수에서는 배열 표기 [] 사용이 가능하다.

[문제 2-1] [레벨 1] (50점) 다음 작업을 하는 프로그램을 구조체 배열과 함수를 사용하여 작성하시오.

- (1) 정수 N을 입력받고, N명 학생의 이름, 국어, 영어, 수학 성적을 입력받아 구조체 배열에 저장한다. ($1 \leq N \leq 50$)
- (2) 과목별 N명 학생의 평균 점수를 계산하여 소수점 이하 첫째 자리까지 출력한다. 과목별 평균 점수 계산을 위해 calculate_average 함수를 사용한다. 계산된 과목별 평균값은 main 함수에서 출력한다.
- (3) main 함수에서 각 과목의 평균 점수보다 낮은 점수를 받은 학생 수를 계산하여 과목별로 출력한다.

- 구조체 student를 사용한다. 구조체 student의 멤버는 아래와 같다.

- name : 학생 이름을 저장할 문자 배열. 이름은 공백을 포함하지 않고, 최대 길이는 20이다.
- kor, eng, math : 국어, 영어, 수학 성적을 저장할 int형 변수 3개

- calculate_average 함수의 반환 값으로 구조체 average를 사용한다. 구조체 average의 멤버는 아래와 같다.

- kor, eng, math : 국어, 영어, 수학의 평균 점수를 저장할 double형 변수 3개

- **calculate_avg** 함수를 정의해서 사용한다.

- 함수 원형: struct average calculate_avg(struct student *st, int N);
- 인자: student 구조체에 대한 포인터 1개와 정수 N
- 구조체 배열의 시작 주소와 정수 N을 인자로 받아, N명 학생의 국어 평균, 영어 평균, 수학 평균을 계산하여, average 구조체 변수에 저장한다. 평균값의 출력은 하지 않는다.
- 반환 값: average 구조체 변수의 값

- 다음 조건들을 지켜 프로그램을 작성하시오.

- ▶ student 구조체 타입의 배열을 사용하지 않거나 멤버를 다르게 정의하면 (100% 감점)
- ▶ 함수를 목적에 맞게 사용하지 않거나, 지정된 함수 원형을 지키지 않으면 (100% 감점)
- ▶ 과목별 평균을 average 구조체 타입 변수에 저장하지 않거나, 전역변수 사용 시 (100% 감점)
- ▶ calculate_avg 함수 내에서 구조체 포인터를 사용하지 않거나, 구조체 배열을 반복문으로 순회할 경우 주소를 이용하여 포인터를 이동시키며, 반복문 구현을 하지 않은 경우 (30% 감점) 단, main 함수에서는 배열 표기 [] 사용 가능하다.

입력 예시 1

```
5
Kim 100 90 80
Lee 88 94 92
Park 92 100 100
Hong 84 82 85
Nam 80 84 72
```

출력 예시 1

```
88.8 90.0 85.8   ↳ 국어 영어 수학 평균
3 2 3           ↳ 국어 영어 수학 순으로 평균 미달 학생 수
```

[문제 2-2] [레벨 2] (100점) 다음 작업을 하는 프로그램을 구조체 배열과 함수를 사용하여 작성하시오.

- (1) 정수 N을 입력받고, N명 학생의 이름, 국어, 영어, 수학 성적을 입력받아, 세 과목의 평균 점수와 함께 구조체 배열 st1에 저장한다. 다시 한번, N명 학생의 이름 국어, 영어, 수학 성적을 입력받아, 세 과목의 평균 점수와 함께 구조체 배열 st2에 저장한다. ($1 \leq N \leq 49$ 이며, N은 홀수이다.) 입력을 받기 위해 **read_data 함수**를 사용한다.
- (2) 두 배열에 저장된 학생들의 시험 정보를 평균 점수가 높은 학생부터 낮은 학생 순서로 정렬한다. 단, 평균 점수가 같은 학생은 국어 성적이 높은 순으로, 평균 점수가 같고, 국어 성적도 같은 경우에는 영어 성적이 높은 순으로 정렬한다. 모든 성적이 같은 학생은 없다고 가정한다. 정렬을 위해 **sort_score 함수**를 사용한다.
- (3) 정렬된 두 배열의 중간값을 비교하여, 중간값이 더 큰 배열에 저장된 학생들의 이름과 국어, 영어, 수학 성적 그리고 평균 점수를 출력한다. **평균 점수는 소수점 이하 첫째 자리까지** 출력한다. 중간값이란, 정렬된 배열의 중간 등수에 해당하는 학생의 평균 점수로 입출력 예시와 같이 N이 5인 경우, 3등 학생의 성적을 말한다. 중간 등수 학생의 평균 점수가 같은 경우는 없다고 가정한다. 중간값이 더 큰 배열을 선택하기 위해 select_out 함수를 사용하고, main 함수에서 선택된 배열의 정보를 출력한다.

- 구조체 student를 사용한다. 구조체 student의 멤버는 아래와 같다.
 - name : 학생 이름을 저장할 문자 배열. 이름은 공백을 포함하지 않고, 최대 길이는 20이다.
 - kor, eng, math : 국어, 영어, 수학 성적을 저장할 int형 변수 3개
 - avg : 세 과목의 평균 점수를 저장할 double형 변수 1개

- 다음 함수를 정의해서 사용한다.

(1) **read_data** 함수

- 인자: student 구조체에 대한 포인터 1개와 정수 N
- 구조체 배열의 시작 주소와 정수 N을 인자로 받아, N명 학생의 국어, 영어, 수학 성적을 입력받고, 각 학생의 평균 점수를 계산한다.

○ 반환 값: 없음

(2) **sort_score** 함수

- 인자: student 구조체에 대한 포인터 1개와 정수 N
- 구조체 배열의 시작 주소와 정수 N을 인자로 받아, 평균 점수가 높은 순으로, 평균 점수가 같으면 국어 성적이 높은 순으로, 평균 점수와 국어 성적이 같으면 영어 성적이 높은 순으로 정렬한다.

○ 반환 값: 없음

(3) **select_out** 함수

- 인자: student 구조체에 대한 포인터 2개와 정수 N
- 두 구조체 배열의 시작 주소와 정수 N을 인자로 받아, 중간 등수 학생의 성적이 더 높은 배열의 주소를 반환한다.
- 반환 값: student 구조체 배열의 시작 주소

- 다음 조건들을 지켜 프로그램을 작성하시오.

- ▶ student 구조체 타입의 배열을 사용하지 않거나 멤버를 다르게 정의하면 (100% 감점)
- ▶ 지정된 함수를 목적에 맞게 사용하지 않거나, 지정된 함수 원형을 지키지 않으면 (read_data, select_out 함수는 각 25% 감점, sort_score 함수는 50% 감점)
- ▶ 함수 내에서 구조체 포인터를 사용하지 않거나, 구조체 배열을 반복문으로 순회할 경우 주소를 이용하여 포인터를 이동시키며, 반복문 구현을 하지 않은 경우 (read_data, select_out 함수는 각 10% 감점, sort_score 함수는 20% 감점) 단, 정렬에서 단순 반복 횟수를 나타내는 for (i=0; i<N-1; i++)은 사용 가능하다. 또한, main 함수에서는 배열 표기 [] 사용 가능하다.
- ▶ sort_score 함수의 정렬에서 구조체 배열의 원소 교환 시 구조체 대입을 하지 않고 멤버 변수 값을 각각 이동시킨 경우 (20% 감점)
- ▶ 함수 원형 선언을 main 함수 전에 하고 함수를 main 함수 뒤에 정의하지 않는 경우 (10% 감점)

입력 예시 1

```
5
Kim 100 85 80    ↳ 첫 번째 배열에 저장될 학생 정보
Lee 88 94 92
Park 92 100 100
Hong 84 82 85
Nam 80 84 72
Song 100 90 97   ↳ 두 번째 배열에 저장될 학생 정보
Lee 92 82 93
Park 80 84 72
Choi 97 80 90
Chung 100 94 93
```

출력 예시 1

```
Chung 100 94 93 95.7
Song 100 90 97 95.7
Choi 97 80 90 89.0
Lee 92 82 93 89.0
Park 80 84 72 78.7

↳ 중간값인 3등 학생의 평균 성적이 더 높은 두 번째
배열의 학생 정보를 출력 (3등인 학생의 정보는 배열
[2]번 원소에 저장 되어 있음)
```

※ [문제 3-1]에서 [문제 3-2]까지 연관된 문제이다. [문제 3-1] 만 풀면 50점, [문제 3-2] 까지 모두 풀면 100점이다. 가장 높은 점수 하나만 반영한다.

[문제 3-1] [레벨 1] (50점) 어느 도시의 호텔들에 대한 정보를 입력받고, 원하는 호텔 등급 G와 도시 중심으로부터 호텔까지의 거리 D를 입력하면, 입력된 호텔 중, 등급이 G 이상이고, 호텔까지의 거리가 D 이하인 호텔 중, 평판도가 가장 높은 호텔의 정보를 출력하는 프로그램을 아래의 in hotel info와 out hotel info 두 함수를 사용하여 작성 하시오. 만약, 해당 조건을 만족하고 평판도가 가장 높은 호텔이 두 개 이상 있는 경우에는 호텔 이름이 사전 순서 상 가장 빠른 호텔의 정보를 출력한다. 단, 조건을 만족하는 호텔이 없는 경우는 없다고 가정한다.

- 호텔의 정보는 다음과 같다. (struct hotel_info)
 - 이름 : 공백을 포함하지 않고 길이가 최대 30인 문자열
 - 등급 : 1부터 5 사이의 정수
 - 평판도 : 1부터 5 사이의 실수
 - 거리 : 시의 중심으로부터 호텔까지의 거리, 실수
 - 조식포함여부 : 'Y' 또는 'N'의 문자
- 호텔 이름은 영문 소문자로만 이루어져있다.
- 호텔은 최대 100개까지 사용자로부터 입력 받는다.
- 다음 두 함수를 사용한다.
 - 1) in_hotel_info 함수
 - 함수 원형: `int in_hotel_info(struct hotel_info *p)`
 - 인자: 호텔 정보를 저장할 구조체 배열에 대한 포인터 p
 - 호텔의 이름, 등급, 평판도, 거리, 조식포함여부를 사용자로부터 입력 받아 배열에 저장한다.
 - 호텔 이름으로 "0"이 입력되면, 입력을 종료한다.
 - 반환 값 : 정보가 입력된 호텔의 수
 - 2) out_hotel_info 함수

- 함수 원형: `void out_hotel_info(struct hotel_info *p, int N, int G, double D)`
- 인자: 호텔 정보가 저장된 구조체 배열에 대한 포인터 p, 배열의 크기 N, 원하는 호텔 등급 G, 원하는 호텔의 거리 D
- 배열에 저장된 호텔 중 G 이상의 등급과 D 이하의 거리를 갖는 호텔을 검색하여, 이 중 평판도가 가장 높은 호텔의 정보를 화면에 출력한다. 평판과 거리는 소수점 아래 첫째자리까지 출력한다.
- 반환 값 : 없음

3) main 함수

- in_hotel_info 함수를 호출하여 호텔 정보 입력 받음
- 원하는 호텔의 G 값과 D 값 입력 받음
- out_hotel_info 함수를 호출하여 원하는 호텔 중 평판도가 가장 높은 호텔 정보 출력

- ▶ 배열 선언 이후, 배열 표기 [] 는 사용 금지 (구조체에 대한 포인터 표기 사용) (50% 감점)
 - ▶ 반복문으로 배열 훑어볼 시, 주소를 이용하여 반복문 구현 (50% 감점)
- [예: `for(p = ar ; p < ar+5 ; p++)` : p는 구조체 포인터, ar은 구조체 배열의 이름]

입력 예시1

```
marriott 4 4.7 12.5 Y // 호텔의 이름, 등급, 평판도,
ibis 2 3.5 5.6 N      // 거리, 조식포함여부 순으로 입력
novotel 3 3.0 2.7 N
renaissance 3 3.5 4.8 N
hyatt 5 3.5 7.4 Y
bestwestern 2 2.5 3.8 Y
0 // "0" 입력 시 호텔 정보 입력 종료
3 7.5 // 원하는 호텔 등급, 거리
```

출력 예시1

```
hyatt 5 3.5 7.4 Y
// 3등급 이상, 7.5km 이내 호텔 중
// 평판도가 가장 높은 두 호텔 hyatt와
// renaissance 중 이름이 사전 순서 상
// 빠른 hyatt 호텔 정보 출력
```

[문제 3-2] [레벨 2] (100점) 어느 도시의 호텔들에 대한 정보를 입력받고, 원하는 호텔 등급 G 와 도시 중심으로부터 호텔까지의 거리 D를 입력하면, 입력된 호텔 중, 등급이 G 이상이고, 호텔까지의 거리가 D 이하인 모든 호텔의 호텔 정보를 출력하는 프로그램을 in_hotel_info와 out_hotel_info 두 함수를 사용하여 작성 하시오. 단, 호텔 정보는 평판도가 가장 높은 호텔부터 출력한다. 평판도가 동일한 호텔의 경우, 호텔 이름이 사전 순서 상 빠른 호텔의 정보를 먼저 출력한다. 조건을 만족하는 호텔이 없는 경우는 없다고 가정한다.

- ▶ 배열 선언 이후, 배열 표기 [] 는 사용 금지 (구조체에 대한 포인터 표기 사용) (50% 감점)
 - ▶ 반복문으로 배열 훑어볼 시, 주소를 이용하여 반복문 구현 (50% 감점)
- [예: `for(p = ar ; p < ar+5 ; p++)` : p는 구조체 포인터, ar은 구조체 배열의 이름]
버블 정렬하는 경우, 외부 for문에서는 int i 사용해도 되며 내부 for 문은 구조체 포인터 사용한다.

입력 예시1

```
marriott 4 4.7 12.5 Y // 호텔의 이름, 등급, 평판도,
ibis 2 3.5 5.6 N // 거리, 조식포함여부 순으로 입력
novotel 3 3.0 2.7 N
renaissance 3 3.5 4.8 N
hyatt 5 3.5 7.4 Y
bestwestern 2 2.5 3.8 Y
0 // "0" 입력 시 호텔 정보 입력 종료
3 7.5 // 원하는 호텔 등급, 거리
```

출력 예시1

```
hyatt 5 3.5 7.4 Y
renaissance 3 3.5 4.8 N
novotel 3 3.0 2.7 N

// 3등급 이상, 7.5km 이내 호텔 중
// 평판도가 높은 호텔부터 출력,
// 평판도가 동일한 두 호텔 hyatt와
// renaissance 중 이름이 사전 순서 상
// 빠른 hyatt 호텔 정보 먼저 출력
```

※ [문제 4-1]에서 [문제 4-2]까지 연관된 문제이다. [문제 4-1] 만 풀면 50점, [문제 4-2] 까지 모두 풀면 100점이다. 가장 높은 점수 하나만 반영한다.

[문제 4-1] [레벨 1] (50점) 주차 빌딩은 100대가 주차할 수 있는 주차장을 보유하고 있다. 다음 [표]를 참고하여 주차장 사용자의 요금을 계산하시오. 주차장은 정기권 S와 일일권 D로 구분한다. 정기권 고객에게 60분을 넘지 않으면 추가요금은 없다. 정기권 구매를 원하면, 요금에 정기권 비용이 추가 된다 (입력예시 1 참고). 일일권 사용자는 별도의 등록비 없이 주차장 이용시간을 근거 (표)로 요금이 계산된다. (단, 구조체를 이용하여 코딩을 완성할 것)

구분	정기권 (S)		일일권 (D)
상품	종류	기본 4시간	기본 4시간
	7일	30,000원	5,000원
	30일	100,000원	
	180일	500,000원	
추가요금	4시간 초과 시 10분당 200원씩 추가 (4시간 10분 200원 추가, 4시간 20분 400원 추가) (정기권)(일일권)은 기본 주차시간이 4시간 주차시간은 24시간을 초과하지 않는다고 간주		

- 주차장 이용 정보는 다음과 같다. (struct parking_info)

- 차량 번호: 정수형 (4자리)
- 고객 전화번호: 정수형 (8자리)
- 상품: 문자형 (S 혹은 D)
- 정기권 가입유무: 문자형 (Y 혹은 N)
일일권은 해당사항 없음
- 종류: 정기권(S) -> 정수형 { 7 / 30 / 180 } 일(day);
일일권은 해당사항 없음
- 주차시간 (분): 정수형
- 요금 (원): 실수형(double)

- 다음 3개의 함수를 사용함

1) input 함수

- 함수 원형: void input(struct parking_info *p, int N)
- 인자: 주차장 이용 정보를 저장할 구조체 배열에 대한 포인터 p, 데이터의 개수 N
- 기능: N개의 데이터
(차량 번호, 고객 전화번호, 상품, 정기관 가입유무, 종류, 주차시간)을 읽는다.
- 반환 값: 없음

2) compute 함수

- 함수 원형: void compute(struct parking_info *p)
- 인자: parking_info 구조체 포인터 p
- 기능: 차량 1대의 주차요금을 계산한다. 상품이 S이고 정기관에 가입이 안 된 경우, 종류별 (7일, 30일, 180일)에 따른 가입비용을 대여비용에 합산한다. 상품이 S이고 정기관에 가입된 경우 종류에 따른 일자를 초과하지 않으면서 4시간을 넘지 않으면 비용이 청구되지 않는다. D에 대해서 4시간당 5000원의 요금이 발생한다. S 또는 D에 대해서 4시간을 초과 하면 10분당 200원을 추가
- 반환 값: 없음

3) display 함수

- 함수 원형: void display(struct parking_info *p, int N)
- 인자: 주차장 이용 정보를 저장할 구조체 배열에 대한 포인터 p, 데이터의 개수 N
- 기능: 계산된 주차장 이용요금을 출력한다.
(차량_번호, 고객전화, 상품(S/D), 이용요금)
- 반환 값: 없음

4) main 함수

- input 함수를 호출하여 주차장 이용정보를 입력 받음
- compute 함수를 호출하여 주차장 이용료 계산
- display 함수를 호출하여 계산한 요금을 출력한다. (출력예시와 같이 '차량번호', '고객전화', 'S/D', '주차료'를 출력) (단, 요금은 소수점이하는 표시하지 않음)

▶ 배열 선언 이후, 배열 표기 [] 는 사용 금지 (구조체에 대한 포인터 표기 사용) (50% 감점)

▶ 반복문으로 배열 훑어볼 시, 주소를 이용하여 반복문 구현 (위반 시 50% 감점)

[예: for(p = ar ; p < ar+5 ; p++) : p는 구조체 포인터, ar은 구조체 배열의 이름]

입력 예시1	출력 예시1
4 → N 1111 12345678 S N 7 230 2222 22223333 D 240 3333 44445555 D 255 4444 66667777 S Y 30 200 ↳ 차량번호, 고객전화번호, 상품(S/D), 정기권유무(Y/N), 가입종류, 대여시간(분) ↳ 일일권인 경우 정기권유무, 가입종류는 입력받지 않음.	1111 12345678 S 30000 ↳ 7일 정기권 신청자 30,000원 2222 22223333 D 5000 ↳ 일일권 5,000원 (4시간 이하) 3333 44445555 D 5200 ↳ 일일권 5,000원 (초과 10분) 4444 66667777 S 0 ↳ 정기권 가입자 (4시간 이하) ↳ 차량 번호, 고객전화번호, 상품(S/D), 대여비용

입력 예시2	출력 예시2
6 → N 3777 28171652 S Y 30 348 1989 97027706 D 429 7757 25158262 S Y 30 392 2714 15392245 D 560 8506 12389677 D 47 5113 59262609 S N 30 967	3777 28171652 S 2000 ↳ 정기권 소지자 (4시간 초과) 1989 97027706 D 8600 7757 25158262 S 3000 2714 15392245 D 11400 8506 12389677 D 5000 5113 59262609 S 114400 ↳ 정기권 구매 (4시간 초과)

[문제 4-2] [레벨 2] (100점) 주차 빌딩은 100대가 주차할 수 있는 주차장을 보유하고 있다. [문제 4-1]의 [표]를 참고하여 주차장 사용자의 요금을 계산하시오. 프로그램 실행 후 출력 내역은 첫째, 요금 리스트, 둘째, 요금 총액이다. 세부 설명은 아래 함수 설명 부분을 참고할 것 (단, 구조체를 이용하여 코딩을 완성할 것)

- 다음의 함수를 사용함 (데이터의 입력과 요금 계산을 위한 함수는 [문제 4-1]을 참고 할 것.)

1) bike_info_sort 함수

- 함수 원형: void bike_info_sort(struct parking_info *p, int N)
- 인자: 주차장 이용 정보를 저장할 구조체 배열에 대한 포인터 p, 데이터의 개수 N
- 기능: (1) 상품을 기준으로 오름차순 정렬
(2) 상품별 차량_번호에 대해서 오름차순 정렬한다.
- 반환 값: 없음
- parking_info_sort 함수 구현 시 배열 표기 사용을 허용함 ([] 가능)

- 화면에 출력 :

- 1) 요금계산 후 정렬된 결과를 출력 함 (출력예시1을 보시오)
정렬 방법은 parking_info_sort에 따르며, 출력예시를 참고할 것
- 2) 요금 총액 출력: 'D'의 총액; 'S' 총액; 전체총액

입력 예시1

4	↳ N
1111 12345678 S N 7 230	
2222 22223333 D 240	
3333 44445555 D 255	
4444 66667777 S Y 30 200	
↳ 차량 번호, 고객전화번호, 상품(S/D), 정기권유무(Y/N), 가입종류, 주차시간(분)	
↳ 일일권인 경우 정기권유무, 가입종류는 입력받지 않음.	

출력 예시1

↳ 상품별 오름차순 정렬 후, 같은 상품에서 차량 번호에 대해 오름차순 정렬한 결과를 출력	
2222	22223333 D 5000
3333	44445555 D 5200
1111	12345678 S 30000
4444	66667777 S 0
10200	30000 40200
↳ 주차료합계: 'D'합, 'S'합, 총합	

입력 예시2

6	↳ N
3777 28171652 S Y 30 348	
1989 97027706 D 429	
7757 25158262 S Y 30 392	
2714 15392245 D 560	
8506 12389677 D 47	
5113 59262609 S N 30 967	

출력 예시2

1989	97027706 D 8600
2714	15392245 D 11400
8506	12389677 D 5000
3777	28171652 S 2000
5113	59262609 S 114400
7757	25158262 S 3000
25000	119400 144400

[문제 5] [레벨 3] (100점) 학사정보시스템을 개발하려고 한다. 학생들의 정보로는, 이름(최대영문 30자), 학번(숫자로8자), 과목수, 과목명(최대영문30자, 최대10개), 취득학점(실수) 등을 저장한다. 다음의 작업명령에 따라 작업하고, 필요한 출력을 수행한다. 학생 수는 최대 100명이다. 학생의 정보를 저장할 때 구조체를 사용하여야 하고 각 명령을 수행하기 위한 함수를 작성하여 호출하여 사용한다. 편의상 동점자, 동명이인은 없다고 가정한다. 이름, 학번, 과목명은 공백이 없는 문자열이다.

명령 :

'I' : 1명의 학생정보 입력 (다음 줄에 아래 정보가 차례로 공백을 사이에 두고 입력)

이름 학번 과목수 과목1 과목1_학점 과목2 과목2_학점 과목5 과목5_학점

입력된 순서대로 저장한다.

'S' : 이름 순으로 오름차순 정렬 후 '이름 학번 과목수 전체_평점(소수점이하 둘째자리)' 출력
정렬 후 출력만 하고, 원래 학생정보 배열에 저장된 값은 변경하지 않고, 정렬하기 전 값을 유지한다.

힌트: sort_s() 함수 안에서 print_s() 함수를 호출한다.

void sort_s(Stud S[], int cnt)

void print_s(Stud S[], int cnt)

'D' : 학생정보 삭제

D 다음에 나오는 이름의 학생정보를 삭제한다.

주의 : 학생이름이 없으면 아무 작업을 하지 않는다.

'P' : 남아있는 모든 학생 정보 출력. (입력되었던 순서대로)

'이름 학번 과목수 전체_평균(소수점이하 둘째자리)

주의 : 남아있는 학생이 없는 경우 출력은 없다.

'Q' : 작업 종료

주의 : 각 줄의 마지막에는 눈에 보이지 않는 'wn' 문자가 있으므로 명령을 처리할 때 이에 유의하여야 한다.

입력 예시 1

```
I
Kim 18011001 3 Korean 3.5 English 4.0 Math 2.0
I
Park 18011003 1 Korean 4.0
I
Lee 17012031 4 English 2.0 Math 4.5 Programming 3.8 Korean 3.2
P
S
D Lee
P
Q
```

출력예시 1

```
Kim 18011001 3 3.17
Park 18011003 1 4.00
Lee 17012031 4 3.38      ↳ P 결과
Kim 18011001 3 3.17
Lee 17012031 4 3.38
Park 18011003 1 4.00      ↳ S 결과
Kim 18011001 3 3.17
Park 18011003 1 4.00      ↳ P 결과
```

제출기한 및 방법

- * OJ시스템(<https://ex-oj.sejong.ac.kr/index.php/auth/login>) 내의 **과제3**을 이용하여 제출
- * 제출 마감: **x월 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