

〈고급 C프로그래밍 및 실습〉 4차 과제 (11장 구조체)

※ 문제에 대한 안내

- 특별한 언급이 없으면 문제의 조건에 맞지 않는 입력은 입력되지 않는다고 가정하라.
- 특별한 언급이 없으면, 각 줄의 맨 앞과 맨 뒤에는 공백을 출력하지 않는다.
- 출력 예시에서 □는 각 줄의 맨 앞과 맨 뒤에 출력되는 공백을 의미한다.
- 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.
- typedef 를 사용하시오.

입력 예시1

출력 예시1

lee1 2017102656 A lee2 2018111823 A park 2018059961 A choi 2018010101 F lee3 2019565656 B park	□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) 출력하시오. 조건에 맞는 학생이 없다면 0을 출력하시오.

입력 예시1

출력 예시1

lee1 2017102656 A lee2 2018111823 A park 2018059961 A choi 2018010101 F lee3 2019565656 B park	□lee2
---	-------

입력 예시2

```
lee1 2018102656 B
lee2 2018111823 B
park 2018123456 A
choi 2018010101 F
lee3 2018565656 B
park
```

출력 예시2

0

[문제 2] [레벨 2] (100점) 은행에는 창구가 하나가 있고, 이 창구에 N명의 고객이 자기순서(순번)를 기다리며 대기하고 있다. N명에 대한 평균 대기시간(waiting time)을 구하는 프로그램을 작성하시오. (단 일단 모든 업무가 완료되기 전까지 대기행렬의 중간에 누구든 끼어들 수 없다.)

예)

다음의 표와 같이 T1번 대기자가 0시에 도착해서 7시간의 서비스를 받는다. T1번 대기자는 따라서 대기시간이 0이다. T2번 대기자는 2시에 도착했으나 T1번 대기자가 7시간이 서비스를 마치기 전까지 대기해야한다. 즉, T2의 대기시간은 5시간 (=7-2)이다. T3번 대기자는 1번과 2번의 서비스 시간을 합한 시간에서 자신의 도착 시간인 4를 제한(뺀) 7시간을 대기한다.

대기자 (ID)	도착 시간	서비스 시간
T1	0	7
T2	2	4
T3	4	1
T4	5	4

그 결과 다음의 수식에 의해서 평균대기 시간은 4.75시간이 된다.

평균 대기시간 계산:

$T1 = 0; T2 = 5 (=7-2); T3 = 7 (=11-4); T4 = 7 (=12-5)$

$(T1+T2+T3+T4)/4 = (0+5+7+7)/4 = 19/4 = 4.75$

1) 이 문제를 해결하기 위해서 구조체를 사용하며, 구조체의 정의는 다음과 같다.

```
struct _bank {
    int id; // 아이디, T1, T2, T3 대신, 1, 2, 3, ... 으로 한다.
    int arrival_time; // 도착시간
    int service_time; // 서비스시간
    int waiting_time; // 대기시간
};
```

2) main 함수

- 입력형식: 첫줄에 N이 주어진다. N은 리스트 줄 수이다. ($1 \leq N \leq 100$)

둘째 줄부터 각 줄에 id, 도착시간과 서비스시간을 입력한다.

- 출력형식: N명에 대한 평균 대기시간을 프린트 한다. (단 소수점 이하 둘째 자리까지 표현)

3) void waitingtime 함수

- 함수 원형: **void waitingtime(struct _bank guest[], int N)**
- 인자(guest): 도착시간과 서비스시간이 저장되어 있는 구조체 배열의 이름, N은 대기자 수
- 이 함수에서는 각 id에 대한 대기시간을 구하여 각 배열의 멤버인 waiting_time에 저장한다.
- 반환 값 : 없음

4) double avgtime 함수

- 함수 원형: **double avgtime(struct _bank guest[], int N)**
- 인자(guest): 숫자 (도착시간, 서비스시간, 대기시간)이 저장되어 있는 구조체 배열의 이름, N은 대기자 수
- 이 함수에서는 각 배열에 저장된 값을 이용해서 평균대기시간을 계산하여 main함수에 반환 한다. (이 함수에서는 평균대기 시간을 프린트하지 않는다.)
- 반환값: 평균 대기시간

- ▶ 함수 (main 함수를 제외하고)에서 printf를 사용을 금지 (20% 감점)
- ▶ 구조체의 멤버인 waiting_time에는 반드시 계산된 대기시간이 저장되어야 함 (20% 감점)
- ▶ 함수 사용하지 않거나, 함수 원형 수정 시 (함수 당 30% 감점)
- ▶ 전역변수 사용금지 (100% 감점)

입력 예시 1

```
4      ↳ 대기자 수 N
1 0 7   ↳ 아이디, 도착시간, 서비스 시간
2 2 4
3 4 1
4 5 4
```

출력 예시 1

```
4.75    ↳ 평균 대기시간
```

※ [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_info**와 **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 2] (50점) 교내 주차 전산시스템을 구조체를 활용하여 구현해보고자 한다. 주차 센서는 차량의 번호를 체크하고, 교내 주차 전산시스템의 DB와 연동되어 해당 차량의 등록된 소유자의 학번을 기록하고, 입차 시간과 출차 시간을 YYYY-MM-DD HH:MM (예, 2017-09-03 23:59)의 형태로 기록을 한다. 이 때 부정 사용자의 학번을 오름차순으로 출력하자. 부정사용자란 같은 학번을 이용하여 차량을 2대 이상으로 등록한 사용자이다.

- 입력되는 주차의 개수는 최대 100 이다.
- 첫 입력 값은 총 주차의 개수이다.
- 입력되는 값은 차량번호 (숫자 4자리), 학생의 학번 (6자리), 입차 시간 및 출차 시간이다.

입력 예시 1

```
7
5321□160123□2017-09-03□07:00□2017-09-03□15:12
1234□151237□2017-09-03□08:00□2017-09-03□13:42
8623□158762□2017-09-03□06:00□2017-09-03□23:59
9752□171893□2017-09-03□13:55□2017-09-03□15:22
5715□151237□2017-09-03□15:43□2017-09-03□17:31
7473□160123□2017-09-03□16:41□2017-09-03□19:00
6309□174099□2017-09-03□15:33□2017-09-03□14:12
```

출력 예시 1

```
151237
160123
```

[문제 4-2] [레벨 3] (100점) 앞에서 사용된 입력값을 이용하여, 해당 차량 번호를 입력하면, 출금시 요금을 계산하시오. 주차료는 다음과 같은 규칙으로 산정된다.

- 편의상 주차 전산시스템은 1일 기록만 기록한다. (아래 예시 1은 9월 3일)
- 최초 1시간은 3,000원
- 최초 1시간 후 10분당 1000원 (1분이 넘어도 10분으로 계산되어 1000원)
- 하루 최대 주차시간은 3만원을 넘지 않는다.
- [문제 4-2]에서 얻은 부정이용자에게는 100000원의 벌금을 부과한다.
- 기록 입력 후 입력되는 숫자는 요금 산정 시 차량의 대수이다.
- 요금은 입력된 차량 순서대로 출력한다.

입력 예시 1

```
7
5321□160123□2017-09-03□07:00□2017-09-03□15:12
1234□151237□2017-09-03□08:00□2017-09-03□13:42
8623□158762□2017-09-03□06:00□2017-09-03□23:59
9752□171893□2017-09-03□13:55□2017-09-03□15:22
5715□151237□2017-09-03□15:43□2017-09-03□17:31
7473□160123□2017-09-03□16:41□2017-09-03□19:00
6309□174099□2017-09-03□15:33□2017-09-03□16:12
3
6309
5715
8623
```

출력 예시 1

```
3000
100000
30000
```

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