

<고급 C프로그래밍 및 실습> 5차 과제 (12장 동적할당)

※ 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.

[문제 1] [레벨 0] (100점) 공백을 포함하는 문자열 두 개를 각각 동적할당 방식을 사용하여 입력받고, 입력 받은 두 문자열이 같은 경우 "identical"을, 다를 경우 "different"를 출력하는 프로그램을 작성하시오.

- compareString() 함수를 정의하여 사용하시오.
 - 인자: 입력 받은 두 개의 문자열
 - 두 문자열을 비교하여, 문자열이 같으면 "identical"을, 다르면 "different"를 출력
 - 반환값: 없음
- 문자열은 대소문자를 구별하며, 영문자로만 구성된다.
- 문자열의 길이는 최대 50 이다.
- 문자열 처리 표준 함수를 사용해도 된다.
- 프로그램 종료 시, 할당한 메모리 해제한다.

입력 예시 1

출력 예시 1

This is Sejong This is Sejong	identical
----------------------------------	-----------

[문제 2] [레벨 1] (100점) 정수 N을 입력 받고, 공백을 포함하지 않은 N개의 문자열을 입력 받아, 각 문자열에 포함된 자음의 수를 세어, 자음의 수가 가장 많은 문자열부터 내림차순으로 정렬한 후 순서대로 문자열을 출력하는 프로그램을 작성 하시오.

- 자음의 수가 같은 문자열이 있는 경우에는 입력된 순서대로 출력 하시오.
- 문자열의 최대 길이는 100 이며, 문자열은 영문 대문자와 소문자로 이루어진다.
- 다음과 같은 구조체를 정의해서 사용하고, 구조체 배열과 문자열을 저장 할 배열은 동적으로 할당 받아 사용하고 , 프로그램 종료 시 할당된 메모리는 반드시 해제한다.
- 문자열을 저장 할 배열의 크기는 문자개수+1 (위반 시 30% 감점)

```
struct st{
    char *str;    // 문자열 저장
    int cnt;      // 자음 수 저장
};
```

입력 예시 1

```
5           ↦ N
History
Politics
DonQuixote
LaPeste
Chaos
```

출력 예시 1

```
History           // 자음의 수가 같으면
Politics          // 입력된 순서대로 출력
DonQuixote
LaPeste
Chaos
```

※ [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 점수**가 가장 높은 학생부터 **내림차순**으로 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 입력된 순서대로 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           ↦ N
HongGilDong 1003 Math 78
JeonWooChi 1001 English 60
HeungBoo 1002 Economics 80
NolBoo 1004 Ethics 50
Euler 1005 Math 100
2           ↦ 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_n의 이름 및 점수를 입력 받아 메모리를 동적으로 할당한 구조체에 저장한다. M_n은 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 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.
- 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] [레벨 2] 여러 도시의 관광지에 대한 정보를 동적할당을 이용하여 저장하려고 한다. 도시의 개수, 각 도시의 관광지 정보를 차례로 입력받아 저장하고 다음의 작업을 실행한다.

입력은 다음과 같은 순서이다.

도시 의 개수 N : 양의 정수로 크기 제한 없음.

도시 명 N개 : 각각 최대 영문 20자

각 도시의 관광지 개수 M_0, M_1, \dots, M_{n-1} : 각 N개 도시의 관광지 수

(양의 정수로 크기 제한없음)

각 관광지의 정보 (tourcity)

1. 관광지명 : 공백을 포함하지 않는 영문자 20자리
2. 입장료 : 정수
3. 야외/실내 구분 : 문자 1개 ('I' : 실내 , 'O' : 야외 , 'B' : 둘다)
4. 관광지 시작 일 : YYYYMMDD (문자로 8자리)
5. (필요하면 도시명을 저장)

주의사항

1. 관광지의 정보는 위의 4(또는 5)가지 정보를 포함하는 구조체 타입을 선언하여 사용한다.
2. 도시의 개수에 따라 구조체 포인터의 배열을 동적할당하여야 한다.
예) struct tourcity **ptr; // 이중포인터
3. 각 도시의 관광지 개수에 따라 구조체 배열을 동적할당 한다.
4. 도시는 중복되지 않는다.
5. 관광지 명은 중복될 수 있다.
6. 문자열을 저장하기 위한 메모리 절약을 하지 않아도 된다.
7. 도시명은 struct tourcity 구조체 안에 저장하지 않고 따로 저장해도 되고, 멤버로 저장해도 된다.
8. 메모리 동적할당이 실패하는 경우 Not Enough Memory.를 출력하고 종료.

위와 같은 입력 정보를 받고, 관광지의 입장료가 제일 비싼 관광지를 찾아 해당 도시와 관광지 명을 출력하고, 관련 정보를 출력한다. 만일 입장료가 같은 경우가 있으면 그중에 관광지 시작일이 제일 오래된 곳을 출력하여야. (시작일까지 동일하면 그 중 제일 먼저 나온 관광지를 출력한다.)

출력정보는 아래 순서를 따른다.

도시명 관광지명 입장료 관광지시작일 야외/실내구분

입력 예시 1

```
3          ↳ N=3곳 도시
Seoul Busan Yeosu   ↳ 3곳 도시명
3 1 2          ↳ M0, M1, M2 ↳ 3 도시의 관광지 수
Coex  0 I 19970513      ↳ Seoul 첫번째관광지
Duksugung 2000 0 20000101 ↳ Seoul 두번째관광지
Museum 1500 I 19950323   ↳ Seoul 세번째관광지
Haewoondae 0 0 19940830  ↳ Busan 첫번째관광지
Haetul 2500 B 20010305   ↳ Yeosu 첫번째관광지
SeaHouse 5500 0 19851201 ↳ Yeosu 의 두번째관광지
```

출력 예시 1

```
Yeosu SeaHouse 5500 19851201 0
```

- ▶ 구조체 2차원 배열을 위한 구조체 포인터 배열은 쓰지 않은 경우 (100% 감점)
- ▶ 메모리 동적 할당의 성공여부를 체크하지 않거나 사용 후 반환을 하지 않은 경우 (각 경우마다 10%씩 감점)

제출기한 및 방법

- * OJ시스템(<https://ex-oj.sejong.ac.kr/index.php/auth/login>) 내의 **과제5**를 이용하여 제출
- * 제출 마감: **11월 10일(일요일)** 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