

<고급C 프로그래밍 및 실습> 2차 모의고사 문제지

2022.11.21

※ 문제지의 무단 배포 및 사용을 원칙적으로 금지합니다.

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

※ 문제에 대한 안내

- 문제지는 총 9페이지이고, 총 5문제 100점 만점이고, 문제의 순서는 난이도와 관계없다.

- 출력 예시에서 □는 각 줄의 맨 앞과 맨 뒤에 출력되는 공백을 의미한다.

- 입출력 예시에서 ↳ 이 후는 각 입력과 출력에 대한 설명이다.

- Sample Submit 가능합니다.

[문제 4-1] (50점) 공연 예약 정보를 출력하는 프로그램을 함수와 동적 할당을 이용하여 다음의 조건들을 지켜 작성하시오.

(1) 구조체 struct reserve를 다음과 같이 선언하여 사용한다.

```
struct reserve{
    int id;           // 고객 id
    char grade[4];    // 구매한 티켓의 등급
    int price;        // 구매한 티켓의 가격
};
```

(2) 정수 N을 입력받고, 고객의 예매 정보를 저장할 크기 N의 struct reserve 타입의 배열을 동적으로 할당 받는다.

(3) N개의 예매 정보(고객 id와 구매한 티켓의 등급)를 입력받아, 티켓 가격과 함께 구조체 배열에 저장한다. 한 공연에는 4개 등급의 티켓이 있고, 각 티켓의 등급과 가격은 아래 표와 같다.

티켓 등급	가격
VIP	140,000원
R	110,000원
S	85,000원
A	55,000원

(4) 예매 정보 입력을 위해 input 함수를 사용한다.

○ 함수 원형: void input(struct reserve *p, int N)

○ 구조체 배열의 시작 주소와 배열의 크기를 인자로 받아, 입력받은 고객 id, 티켓 등급과 티켓 가격을 저장한다.

○ input 함수에서는 배열 표기 []를 사용하지 않고, 포인터를 움직이며 반복문을 구현한다.

(5) 예매 정보를 출력할 고객의 id를 입력받아, 해당 고객이 구매한 티켓의 등급과 티켓 가격을 입력순서대로 출력하고, 티켓 구매에 사용된 총 금액을 출력한다. 티켓 구매에 사용된 총 금액을 계산하기 위해 cal_sum 함수를 사용한다.

○ 함수 원형: int cal_sum(struct reserve *p, int N, int id)

○ 구조체 배열의 시작 주소와 배열의 크기, 그리고 고객 id를 인자로 받아, 해당 고객이 구매

한 티켓의 총 구매 금액을 계산하여 반환한다.

- cal_sum 함수에서는 배열 표기 []를 사용하지 않고, 포인터를 움직이며 반복문을 구현한다.

(6) 단, main 함수에서의 주소 표현 방식은 제한 사항 없다.

- ▶ 구조체 배열을 동적으로 할당받지 않으면 (100% 감점)
- ▶ 동적 할당 후 오류 확인을 하지 않으면 (10% 감점)
- ▶ 동적 할당받은 메모리를 해제하지 않으면 (10% 감점)
- ▶ 함수를 사용하지 않거나 함수원형을 지키지 않으면 (함수마다 30% 감점)
- ▶ input과 cal_sum 함수에서 구조체 배열을 반복문으로 훑어볼 시, 포인터를 이동시키며, 반복문 구현 (p[i]/p+i 표기 금지) (위반 시 함수마다 30% 감점)

입력 예시 1

출력 예시 1

10	↳ N=10명	A 55000
123 A		A 55000
210 VIP		R 110000
123 A		A 55000
123 R		R 110000
210 S		385000
210 VIP		
123 A		↳ 123번 고객이 구매한 티켓의 종류와 티켓 한 장의
523 A		↳ 가격을 입력순서대로 출력하고, 티켓 구매 금액의
123 R		↳ 총 합 출력
523 VIP		
123	↳ id=123, 123번 고객의 정보 출력	

[문제 4-2] (100점) 공연 예약 정보를 출력하는 프로그램을 함수와 동적 할당을 이용하여 다음의 조건들을 지켜 작성하시오.

(1) 구조체 struct reserve와 struct tot_price를 다음과 같이 선언하여 사용한다.

<pre>struct reserve{ int id; // 고객 id char grade[4]; // 구매한 티켓의 등급 int price; // 구매한 티켓의 가격 };</pre>	<pre>struct tot_price{ int id; // 고객 id int total_price; // 티켓 구매에 사용한 총 금액 };</pre>
--	---

(2) 정수 N을 입력받고, 고객의 예매 정보를 저장할 크기 N의 struct reserve 타입의 배열을 동적으로 할당 받는다.

(3) N개의 예매 정보(고객 id와 구매한 티켓의 등급)를 입력받아, 티켓 가격과 함께 구조체 배열에 저장한다. 한 공연에는 4개 등급의 티켓이 있고, 각 티켓의 등급과 가격은 [문제 4-1]의 표와 같다.

(4) 예매 정보 입력을 위해 input 함수를 사용한다.

- 함수 원형: void input(struct reserve *p, int N)
- 구조체 배열의 시작 주소와 배열의 크기를 인자로 받아, 입력받은 고객 id, 티켓 등급과 티켓 가격을 저장한다.

- input 함수에서는 배열 표기 []를 사용하지 않고, 포인터를 움직이며 반복문을 구현한다.
- (5) 예매를 한 고객의 수를 계산한다. 동일 고객이 여러 장의 티켓을 구매하는 경우, N개의 예매 정보에는 동일 고객의 id가 여러 번 나타난다. 중복된 id를 제외하여, 고객의 수를 계산한다. 고객 수 계산에 cal_num 함수를 사용한다.
 - 함수 원형: **int cal_num(struct reserve *p, int N)**
 - 구조체 배열의 시작 주소와 배열의 크기를 인자로 받아, 고객의 수를 계산하여 반환한다.
 - cal_num 함수에서는 배열 표기 []를 사용하지 않고, 포인터를 움직이며 반복문을 구현한다.
- (6) **고객 수 크기의 struct tot_price 타입 배열을 동적으로 할당 받는다.**
- (7) struct tot_price 타입의 구조체 배열에 고객 id와 해당 고객이 티켓 구매에 사용한 총 금액을 계산하여 저장한다. **동일 고객의 id가 반복 저장되어서는 안 된다.** 각 고객의 총 티켓 구매 금액 계산을 위해 cal_sum 함수를 호출하여 사용한다.
 - 함수 원형: **void cal_sum(struct reserve *p, int N, struct tot_price *q)**
 - 예매 정보가 저장된 구조체 배열의 시작 주소와 배열의 크기, 그리고, 고객의 총 티켓 구매 금액을 저장할 구조체 배열의 시작 주소를 인자로 받아, 고객 id를 저장하고, 해당 고객이 티켓 구매에 사용한 총 금액을 계산한다. **고객 id 저장 순서는 해당 고객 id가 예매 정보에 처음 나타나는 순이다.** (입출력예시 1 참고)
 - cal_sum 함수에서는 배열 표기 []를 사용하지 않고, 포인터를 움직이며 반복문을 구현한다.
- (8) struct tot_price 타입의 구조체 배열에 저장된 고객 id와 티켓 구매 총 금액을 main 함수에서 출력한다. **main 함수에서의 주소 표현 방식은 제한 사항 없다.**

- ▶ struct reserve와 struct tot_price 타입의 구조체 배열을 동적으로 할당받지 않으면 (100% 감점)
- ▶ 동적 할당받을 struct tot_price 타입의 구조체 배열의 크기를 계산하지 않으면 (50% 감점)
- ▶ 동적 할당 후 오류 확인을 하지 않으면 (각 5% 씩 감점)
- ▶ 동적 할당받은 메모리를 해제하지 않으면 (각 5% 씩 감점)
- ▶ 함수를 사용하지 않거나 함수원형을 지키지 않으면 (함수마다 20% 감점)
- ▶ input, cal_num, cal_sum 함수에서 구조체 배열을 반복문으로 훑어볼 시, 포인터를 이동시키며, 반복문 구현 (p[i]/p+i 표기 금지) (위반 시 함수마다 20% 감점)

입력 예시 1

10 ↳ N=10명
 123 A
 210 VIP
 123 A
 123 R
 210 S
 210 VIP
 123 A
 523 A
 123 R
 523 VIP

출력 예시 1

123 385000
 210 365000
 523 195000
 ↳ 세 명 고객의 id와 티켓 구매에 사용한 총 금액 출력

[문제 5-1] (50점) N 줄의 공백이 포함된 문자열을 입력받아, 각 문자열에 포함된 단어 수 순서대로 문자열을 정렬하는 프로그램을 동적 할당을 이용하여 다음의 조건들을 지켜 작성하시오.

(1) 구조체 struct string을 다음과 같이 선언하여 사용한다.

```
struct string{
    char *str;    // 문자열을 가리키는 포인터, 문자 배열 (char str[])로 선언한 경우 50% 감점
    int words;    // str이 가리키는 문자열에 포함된 단어 수
};
```

(2) 정수 N을 입력받고, 크기 N의 struct string 타입의 구조체 배열을 동적으로 할당 받는다.

(3) N 줄의 문자열을 입력받아, 문자열 크기에 맞는 문자 배열을 동적으로 할당 받아 str에 연결한다. 입력되는 문자열의 최대 크기는 100이다.

(4) 문자열에 포함된 단어의 수를 세어 words에 저장한다. 공백과 공백 사이가 하나의 단어이다.

(5) 문자열에 포함된 단어 수가 가장 많은 문자열부터 가장 적은 문자열까지 내림차순으로 정렬하고, 문자열과 단어 수를 출력한다. 단어 수가 같은 경우에는 입력순으로 출력한다.

- ▶ 구조체 배열을 동적 할당받지 않으면 (50% 감점)
- ▶ 문자열을 저장할 문자 배열을 동적 할당받지 않으면 (50% 감점)
- ▶ 동적 할당받은 문자 배열의 크기가 문자열 저장에 필요한 공간보다 크거나 작으면 (10% 감점)
- ▶ 동적 할당 후 오류 확인을 하지 않으면 (구조체 배열 5%, 문자 배열 5% 감점)
- ▶ 동적 할당받은 메모리를 해제하지 않으면 (구조체 배열 5%, 문자 배열 5% 감점)

입력 예시 1

```
5
why so serious
icecream makes everything better
you make me smile
c language midterm and final
see you
```

출력 예시 1

```
c language midterm and final 5
icecream makes everything better 4
you make me smile 4
why so serious 3
see you 2
```

[문제 5-2] (100점) N 줄의 공백이 포함된 문자열을 입력받아, 각 문자열에 포함된 단어 수 순서대로 문자열을 정렬하고, 각 문자열에 가장 많이 나타나는 단어를 출력하는 프로그램을 동적 할당을 이용하여 다음의 조건들을 지켜 작성하시오.

(1) 구조체 struct string을 다음과 같이 선언하여 사용한다.

```
struct string{
    char *str;    // 문자열을 가리키는 포인터, 문자 배열 (char str[])로 선언한 경우 50% 감점
    int words;    // str이 가리키는 문자열에 포함된 단어 수
};
```

(2) 정수 N을 입력받고, 크기 N의 struct string 타입의 구조체 배열을 동적으로 할당 받는다.

(3) N 줄의 문자열을 입력받아, 문자열 크기에 맞는 문자 배열을 동적으로 할당 받아 str에 연결한다. 입력되는 문자열의 최대 크기는 100이다.

- (4) 문자열에 포함된 단어의 수를 세어 words에 저장한다. 공백과 공백 사이가 하나의 단어이다.
- (5) 문자열에 포함된 단어 수가 가장 많은 문자열부터 가장 적은 문자열까지 내림차순으로 정렬한다. 단어 수가 같은 경우에는 입력순으로 정렬한다.
- (6) 각 문자열을 단어로 나누어 동적으로 할당받은 2차원 배열에 저장한다. 먼저, 단어 수와 같은 크기의 문자 포인터 배열을 동적으로 할당받고, 문자열을 단어로 나누어 각 단어가 저장될 문자 배열을 동적으로 할당받아 포인터 배열에 연결한다.
- (7) 각 문자열 마다 저장된 단어 중 중복 횟수가 가장 많은 단어를 찾아, 해당 문자열, 문자열의 단어 수, 중복 횟수가 가장 많은 단어, 해당 단어의 중복 횟수를 출력한다. 중복 횟수가 가장 많은 단어가 둘 이상인 경우, 사전 순서상 빠른 단어를 출력한다.
- (8) 각 문자열에 대한 출력을 마친 후, 단어 저장을 위해 동적으로 할당받은 2차원 배열은 해제 (free)한다.

- ▶ 구조체 배열을 동적 할당받지 않으면 (50% 감점)
- ▶ 문자열을 저장할 문자 배열을 동적 할당받지 않으면 (50% 감점)
- ▶ 단어들을 저장할 2차원 문자 배열을 동적 할당받지 않으면 (50% 감점)
- ▶ 동적 할당받은 문자 배열의 크기가 문자열 저장에 필요한 공간보다 크거나 작으면 (10% 감점)
- ▶ 동적 할당 후 오류 확인을 하지 않으면 (동적 할당 시 마다 5%씩 감점)
- ▶ 동적 할당받은 메모리를 해제하지 않으면 (해제가 필요한 시점 마다 5%씩 감점)

입력 예시 1

```
5
why why why so so so serious
icecream makes everything better and better
you make me smile
c language exam and exam and exam and exam
see you later later later later
```

출력 예시 1

```
c language exam and exam and exam and exam 9 exam 4
why why why so so so serious 7 so 3
icecream makes everything better and better 6 better 2
see you later later later later 6 later 4
you make me smile 4 make 1
```

- ↳ 문자열, 단어수, 빈도수 높은 단어, 빈도수 출력
- ↳ 빈도수 높은 단어가 둘 이상인 경우, 사전 순서상 빠른 단어 출력