

<고급C 프로그래밍 및 실습> 2차 모의고사 문제지

(주의) 2차 인증 날짜 2022.11.21.(월)

※ 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 1-1] (50점) 정수 M과 N을 입력받고, 공백을 포함하는 두 개의 문자열 str1과 str2를 입력받아, str1의 M 번째 단어와 str2의 N 번째 단어를 연결하여 하나의 단어로 만들어 출력하는 프로그램을 작성하시오. 단, 두 단어의 연결은 사전 순서상 빠른 단어 뒤에 순서가 늦은 단어를 연결한다.

- 입력받은 문자열의 최대 길이는 100이고, 각 문자열에 포함된 단어의 최대 개수는 20 이다.
- 첫 번째 단어의 위치는 0번부터 시작한다.
- M과 N의 값이 문자열에 포함된 단어의 수보다 큰 경우는 없다고 가정한다.

▶ strcpy, strcat, strcmp 함수를 각각 최소 1회 이상 사용 (위반 시 각 함수 마다 20% 감점)

▶ 하나의 단어를 만들어 한 번의 출력문으로 출력 (위반 시 100% 감점)

입력 예시 1

2 4 ↳ M=2 (pencil), N=4 (grape)
book desk pencil paper
orange apple banana lemon grape

출력 예시 1

grapepencil ↳ 사전순서가 빠른 grape 먼저

입력 예시 2

1 0 ↳ M=1 (desk), N=0 (orange)
book desk pencil paper
orange apple banana lemon grape

출력 예시 2

deskorange ↳ 사전순서가 빠른 desk 먼저

[문제 1-2] (100점) 정수 N과 M1, M2를 입력받고, N 줄의 공백을 포함하는 문자열을 입력받아, 가장 긴 문자열과 가장 짧은 문자열을 출력하고, 가장 긴 문자열의 M1 번째 단어와 가장 짧은 문자열의 M2 번째 단어를 합성하여 하나의 단어로 만들어 출력하는 프로그램을 작성하시오. 단, 두 단어의 연결은 사전 순서상 빠른 단어 뒤에 순서가 늦은 단어를 연결한다.

- $N \leq 20$ 이고, 입력받은 문자열의 최대 길이는 100이고, 각 문자열에 포함된 단어의 최대 개수는 20이다.
- 첫 번째 단어의 위치는 0번부터 시작한다.
- M1과 M2의 값이 문자열에 포함된 단어의 수보다 큰 경우는 없다고 가정한다.

- 가장 긴 또는 가장 짧은 문자열이 두 개 이상인 경우, 먼저 입력된 문자열을 출력한다.

- ▶ strcpy, strcat, strcmp 함수를 각각 최소 1회 이상 사용 (위반 시 각 함수 마다 20% 감점)
- ▶ 합성된 단어의 출력은 하나의 단어를 만들어 한 번의 출력문으로 출력 (위반 시 100% 감점)

입력 예시 1

```
5 4 2      ↳ N은 줄 수, M1=4 (november), M2=2 (icecream)
book desk pencil paper
orange apple banana lemon grape
red yellow green orange blue white black
candy cake icecream
july august september october november december
```

출력 예시 1

```
july august september october november december   ↳ 가장 긴 문자열
candy cake icecream                                ↳ 가장 짧은 문자열
icecreamnovember
↳ 가장 긴 문자열의 M1번째 단어와 가장 짧은 문자열의 M2번째 단어를 사전 순서로 합성
```

[문제 2] 정수 N을 입력받고, 공백을 포함하지 않는 문자열 한 개를 입력받아 정적 배열 X에 저장한다. 다음, N 줄의 공백을 포함하는 문자열을 입력받아, 동적으로 저장 장소를 할당 받은 후 문자형 포인터 배열 Y에 연결한다. Y의 데이터 타입은 char **이며, 포인터 배열도 동적으로 할당을 받는다.

이 후, Y에 저장된 N 줄의 각 문자열에서 X에 저장된 문자열이 가장 많이 나타나는 문자열을 찾아 출력하는 프로그램을 작성 하시오. 단, 가장 많이 나타나는 문자열이 여러 개인 경우에는 가장 처음에 입력된 문자열을 출력한다.

- 입력되는 모든 문자열의 최대 길이는 100이며, 배열 X에 저장된 문자열의 길이는 Y에 저장된 문자열의 길이 보다 작거나 같다고 가정한다.
- 문자열 AAA에는 문자열 AA가 1개 포함되는 것으로 간주하며, 문자열 AAAA에는 문자열 AA가 2개 포함되는 것으로 간주한다. 즉, 이미 횟수 계산에 사용된 문자는 다음 계산에 포함하지 않는다.
- 배열 X에 저장된 문자열이 Y에 저장된 각 문자열에 한 번도 나타나지 않는 경우에는 NONE을 출력한다.

- ▶ 표준 문자열 함수 중 strlen과 strcpy를 각 1회 이상 사용하지 않으면 (30% 감점)
- ▶ 배열 Y의 저장장소를 동적으로 할당 받아 입력받은 N개의 문자열들을 저장하지 않으면 (50% 감점)
- ▶ N개의 문자열들을 저장할 때, 각 문자열 저장에 필요한 공간보다 큰 공간을 할당 받으면 (20% 감점)
- ▶ 동적으로 할당 받은 메모리 해제를 하지 않으면 (20% 감점)
- ▶ 동적 할당 시 정상적으로 메모리 할당이 되었는지를 체크하는 부분을 한 번 이상 사용하지 않으면 (10% 감점)

입력 예시 1

```
3 abc
abcabc abcabc   ↳ abc 4회
ABCabc ABCabc   ↳ abc 2회
abc abc abc abcd ↳ abc 4회
```

출력 예시 1

```
abcabc abcabc   ↳ 반복 횟수가 같은 문자열 중
                  ↳ 먼저 입력된 문자열 출력
```

입력 예시 2

```
3 aaa
aaaaaa aaa aaaa   ↳ aaa 4회
aaa aaa aaa       ↳ aaa 3회
aaaaa aaaaa aaaaa ↳ aaa 3회
```

출력 예시 2

```
aaaaaa aaa aaaa
```

입력 예시 3

```
3 abc
ab cab c abd
ABCab cABC
a bca bca bcd
```

출력 예시 3

```
NONE
```

[문제 3] 정수 M과 N을 입력받고, M행 N열의 2차원 배열을 동적으로 할당받아 0으로 초기화한다. 다음, 세 개의 정수 x, y, z를 입력받고, 배열의 x행 y열에 z를 저장하는 작업을 0, 0, 0 이 입력 될 때까지 반복한다. x값은 0부터 M-1 사이의 값이며, y값은 0부터 N-1사이의 값이 입력되며, 같은 원소에 중복된 값 입력은 없다. 즉, 1, 2, 3 입력 후, 다시 1, 2, 5가 입력되는 경우는 없다고 가정한다. 이렇게 만들어진 2차원 배열을 출력하고, 배열의 각 행 또는 각 열에 연속해서 나타나는 0 의 최대 길이를 계산하여 출력한다.

	0	1	2	3	4
0	5	15	0	25	0
1	0	0	35	0	0
2	0	45	0	55	0

- 위 배열에서 연속으로 0이 나타나는 최대 횟수는 행 방향으로 2이고, 열 방향으로 3이므로, 3이 출력 된다.

- 다음의 두 함수를 사용하여 프로그램을 작성 한다.

1) check_row함수

○ 함수 원형: **int check_row(int **p, int M, int N)**

○ 인자: 2차원 배열의 주소 p, 2차원 배열의 행 수 M, 2차원 배열의 열 수 N

○ p가 가리키는 M X N의 2차원 배열의 각 행을 검사하여 0이 연속해서 나타나는 횟수를 계산한다.

○ 반환값 : 각 행에 연속된 0의 최대 수

2) check_col함수

○ 함수 원형: **int check_col(int **p, int M, int N)**

○ 인자: 2차원 배열의 주소 p, 2차원 배열의 행 수 M, 2차원 배열의 열 수 N

- p가 가리키는 M X N의 2차원 배열의 각 열을 검사하여 0이 연속해서 나타나는 횟수를 계산한다.
- 반환값 : 각 열에 연속된 0의 최대 수

- ▶ 동적으로 할당 받은 MXN의 2차원 배열을 사용하지 않는 경우 (100% 감점)
- ▶ 함수의 원형(인자, 반환값)을 지키지 않거나 함수를 사용하지 않은 경우 (check_row, check_col 각 25% 감점)
- ▶ 동적으로 할당 받은 메모리 해제를 하지 않으면 (20% 감점)
- ▶ 전역변수를 사용하는 경우 (100% 감점)

입력 예시 1

```
3 5
0 0 5
0 1 15
0 3 25
1 2 35
2 1 45
2 3 55
0 0 0
```

출력 예시 1

```
□5□15□0□25□0
□0□0□35□0□0
□0□45□0□55□0
3          ↳ 연속된 0의 최대 길이 (열방향)
```

입력 예시 2

```
4 4
0 1 1
0 3 3
1 0 1
1 2 3
2 2 0   ↳ 0이 입력되는 경우 있음
2 3 6
3 1 2
0 0 0
```

출력 예시 2

```
□0□1□0□3
□1□0□3□0
□0□0□0□6
□0□2□0□0
3          ↳ 연속된 0의 최대 길이 (행방향)
```

[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 구매한 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

```

10      ↳ N=10명
123 A
210 VIP
123 A
123 R
210 S
210 VIP
123 A
523 A
123 R
523 VIP
123     ↳ id=123, 123번 고객의 정보 출력
    
```

출력 예시 1

```

A 55000
A 55000
R 110000
A 55000
R 110000
385000

↳ 123번 고객이 구매한 티켓의 종류와 티켓 한 장의
↳ 가격을 입력순서대로 출력하고, 티켓 구매 금액의
↳ 총 합 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