<고급C 프로그래밍 및 실습> 1차 모의고사 문제지

2022.10

※ 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.
- 입출력 예시에서 ↦ 이 후는 각 입력과 출력에 대한 설명이다.
- OJ에서 Sample 제출 기능 사용 가능합니다. Sample 제출은 점수에 포함되지 않습니다.

[문제 1] (100점) 정수 N과 M을 입력 받은 후, M개의 정수를 입력받고 (중복 입력 가능), 이 중 중복 입력 횟수가 가장 높은 정수를 출력하는 작업을 N번 반복하는 프로그램을 다음의 함수를 사용하여 작성하시오.

- main() 함수를 제외한 input(), sel_max(), output() 함수에서 **반복문으로 배열을 훑어볼 시, 주소를 이용하여 반복문을 구현한다.** 즉, 포인터가 배열의 각 원소를 순차적으로 가리키도록 하며, 포인터가 가리키는 위치의 원소에 대해 필요한 작업을 수행한다.

1) main( ) 함수

◦ main 함수는 아래와 같으며, **내용 수정이 불가**하다.

```
int main(void){
   int in[100], out[100], *max, i, N, M;

   scanf("%d %d", &N, &M);
   for (i=0; i<N; i++){
      input(in, M);
      max=sel_max(in, M);
      out[i]=*max;
   }
   output(out, N);
   return 0;
}
```

2) input() 함수

◦ 함수 원형: **void input(int *p, int M)**

◦ p는 배열의 시작 주소이며, 정수 M개를 입력받아 p가 가리키는 배열에 저장한다.

3) sel_max() 함수

◦ 인자 : int형 포인터 p와 int형 정수 M

◦ 반환 값 : 선택된 정수가 저장된 배열 원소의 주소

◦ p는 배열의 시작 주소이며, 배열에 저장된 M개의 정수 중 중복 횟수가 가장 높은 정수를 선택하여 해당 정수가 저장된 **배열 원소의 주소를 반환한다.** (정수를 반환하는 것이 아님에 주의 하시오.)

- **최다 중복 정수가 둘 이상인 경우, 가장 처음에 나타나는 정수를 선택 한다.** 예를 들어,

M이 5이고, 입력된 정수가 3, 1, 2, 2, 1인 경우, 1과 2중에 가장 처음에 나타나는 1이 선택 된다. 마찬가지로, 1, 2, 3, 4, 5와 같이 중복된 정수가 없는 경우에는 처음에 나타나는 정수 1이 출력된다.

4) output() 함수

- 함수 원형: **void output(int *p, int N)**
- p는 배열의 시작 주소이며, 배열에 저장된 정수 N개를 출력한다.

▶ main 함수 내용을 수정하는 경우 **(100% 감점, 단, *max만 수정하는 경우 25% 감점)**
▶ 함수의 원형(인자, 반환값)을 지키지 않은 경우 **(input, output, sel_max 각 25% 감점)**
▶ 전역변수를 사용하는 경우 **(100% 감점)**
▶ 함수 내에서 배열 표기 [ ]를 사용하는 경우 **(input, output 각 25% 감점, sel_max 50% 감점)**
▶ 함수 내에서 반복문으로 배열 훑어볼 시, 주소를 이용하여 반복문 구현을 하지 않은 경우 **(input, output 각 10% 감점, sel_max 30% 감점)**

입력 예시 1	출력 예시 1
3 5 ↦ N=3, M=5 1 2 2 1 3 2 4 2 4 3 1 2 3 4 5	□1□2□1

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

[문제 2-1] (50점) 종료조건을 만족할 때까지 (종료조건 -1을 포함하여 최대 100개의) 한 자리 양의 정수를 계속 입력 받아 배열 x에 저장하고, 정수 M을 입력받아, x[M]부터 값이 연속적으로 증가하는 원소들만 모아 하나의 정수를 만들어 출력하는 프로그램을 아래 함수들을 사용하여 작성 하시오. 예를 들어, x[M]부터 입력된 정수가 2, 3, 4, 5, 4, 3, -1인 경우, 값이 증가하는 5까지의 정수들인 2, 3, 4, 5를 모아 만든 정수 2345를 출력한다. 입력 종료조건은 –1의 입력이다.

(1) input 함수

- 함수 원형: **int input(int *p)**
- **배열 x의 시작 주소를 인자**로 받아 종료 조건까지 정수를 입력받아 배열에 저장하고, **배열 원소의 개수를 반환**한다.

(2) sel_next 함수

- 함수 원형: **int *sel_next(int *p, int N, int M)**
- **배열 x의 시작 주소와 배열의 크기 N, 원소 값의 연속적 증가 여부 검사를 시작할 원소의 위치 M을 인자**로 받아, **원소 값의 증가가 끝나는 마지막 원소의 주소를 반환** 한다. 만약, x[M] 다음 원소 값이 x[M]의 값 보다 작거나 같은 경우에는 x[M]의 주소를 반환한다.

(3) number 함수

- 함수 원형: **int number(int *p, int *q)**
- **x[M]의 위치와 sel_next에서 반환된 위치를 인자**로 받아, **두 포인터 사이의 한 자리 정수를 모아, 하나의 정수로 만들어 반환** 한다.

- ▶ 배열의 변수 선언 이외의 배열 표기 [ ]는 사용 금지 (포인터 표기 사용) **(위반 시 100% 감점)**
- ▶ 배열을 반복문으로 훑어볼 시, 주소를 이용하여 포인터를 이동시키며, 반복문 구현 **(위반 시 100% 감점)**
- ▶ 전역변수 사용 금지 **(위반 시 100% 감점)**
- ▶ input, sel_next, number 함수를 사용하지 않거나, 함수의 원형을 지키지 않거나, 함수를 목적에 맞게 사용하지 않은 경우 감점 **(input, number 함수는 각 25%, sel_next 함수는 50% 감점)**
- ▶ 정수를 만들어 출력하지 않고, 배열의 각 원소를 연속으로 출력하는 경우 감점 **(100% 감점)** 즉 number 함수를 제대로 구현하여 main 함수에서 정수를 출력하면 감점 없음.

입력 예시 1

1 2 3 4 5 4 3 -1 1 ↦ M=1	2345 ↦ x[1]인 2부터 연속적으로 증가하는 5까지 모아 2345 출력

입력 예시 2

1 2 3 4 5 4 3 -1 4 ↦ M=4	5 ↦ x[4]인 5 이후 증가하는 수가 없으므로 5 출력

[문제 2-2] (100점) 종료조건을 만족할 때까지 (종료조건 -1을 포함하여 최대 100개의) 한 자리 양의 정수를 입력 받아 배열 x에 저장한 후, 배열의 원소가 연속적으로 증가하는 구간과 연속적으로 감소하는 구간으로 나누어, 연속적으로 증가하는 구간의 원소들을 모아 하나의 정수로 만들고, 연속적으로 감소하는 구간의 원소들을 모아 하나의 정수로 만들어 출력하는 프로그램을 아래 함수들을 사용하여 작성 하시오. 예를 들어, 사용자로부터 입력된 정수가 다음과 같은 경우

1	2	3	4	5	4	3	2	3	4	2	-1

첫 번째 원소인 1부터 값이 증가하는 5까지의 정수를 모아, 12345를 출력하고, 5부터 값이 감소하는 2까지 모아 5432를 출력하고, 다시 값이 증가하는 2, 3, 4를 모아 234를 마지막으로 값이 감소하는 구간인 4와 2를 모아 42를 출력한다. 숫자가 증가하다가 감소하거나, 감소하다가 증가하는 지점의 숫자는 만들어지는 두 정수에 모두 포함된다. 연속해서 동일한 숫자가 입력되는 경우는 없다고 가정하며, 입력 종료조건은 –1의 입력이다. 예시 2와 같이 첫 번째 원소부터 감소하다가 증가할 수도 있다.

(1) input 함수, [문제 2-1]과 동일

- 함수 원형: **int input(int *p)**
- **배열 x의 시작 주소를 인자**로 받아 종료 조건까지 정수를 입력받아 배열에 저장하고, **배열 원소의 개수를 반환**한다.

(2) sel_next 함수, [문제 2-1]과 다름

- 함수 원형: **int *sel_next(int *p)**
- **배열의 한 원소의 주소 p를 인자**로 받아, p가 가리키는 원소부터 원소 값의 **증가 또는 감소**가 끝나는 **마지막 원소의 주소를 반환** 한다. 위 예제의 경우, x[4]의 주소가 인자로 전달되면, x[7]의 주소가 반환된다.

(3) number 함수, [문제 2-1]과 동일

- 함수 원형: **int number(int *p, int *q)**
- **배열의 두 원소의 주소 p과 q를 인자**로 받아, **두 포인터 사이의 한 자리 정수를 모아 하나의 정수로 만들어 반환** 한다.

- ▶ 배열의 변수 선언 이외의 배열 표기 [ ]는 사용 금지 (포인터 표기 사용) **(위반 시 100% 감점)**
- ▶ 배열을 반복문으로 훑어볼 시, 주소를 이용하여 포인터를 이동시키며, 반복문 구현 **(위반 시 100% 감점)**
- ▶ 전역변수 사용 금지 **(위반 시 100% 감점)**
- ▶ input, sel_next, number 함수를 사용하지 않거나, 함수의 원형을 지키지 않거나, 함수를 목적에 맞게 사용하지 않은 경우 감점 **(input, number 함수는 각 25%, sel_next 함수는 50% 감점)**
- ▶ 정수를 만들어 출력하지 않고, 배열의 각 원소를 연속으로 출력하는 경우 감점 **(100% 감점)** 즉 number 함수를 제대로 구현하여 main 함수에서 정수를 출력하면 감점 없음.

입력 예시 1

```
1 2 3 4 5 4 3 2 3 4 2 -1
```

출력 예시 1

```
12345
5432
234
42
```

입력 예시 2

```
8 6 4 2 1 0 3 5 7 1 4 -1
```

출력 예시 2

```
864210
357     ↦ 0은 두 번째 정수에 포함이 되지만, 0357이 아니라 357 출력
71
14
```

[문제 3] (100점) 최대 8자리 양의 정수를 문자열로 입력 받은 후, 홀수가 연속되는 경우, 두 홀수 사이에 '+'를, 짝수가 연속되는 경우에는 두 짝수 사이에 '*'를 추가한 문자열을 출력하는 프로그램을 작성하시오.

1) "문자열 길이 구하기 표준 문자열 함수"를 두 번 사용하시오. 이 외의 표준 문자열 함수 사용에 대한 제약사항 없음
2) StringAdd() 함수를 다음과 같이 정의하여 사용하시오:
   - 인자: ① char arr[ ] ② '+' 혹은 '*'가 저장된, char형 변수 ch,
     ③ int형 변수 index
   - 해당 배열의 해당 인덱스가 나타내는 위치에 '+' 혹은 '*' 를 삽입한다.
   - 반환 값: 없음
3) Convertor() 함수를 다음과 같이 정의하여 사용하시오:
   - 인자: char형 변수 x
   - 숫자 문자를 숫자로 변환한다.
   - 반환값: 숫자 문자에 해당하는 정수

4) main( ) 함수를 다음과 같이 작성하시오.
   - 문자열을 사용자로부터 입력 받아 char형 배열 str[ ]에 저장한다. 이때 %c 사용금지
   ① 홀수가 연속되는 경우에 StringAdd() 함수를 호출하여, 두 홀수 사이에 '+'를 추가한다.
   ② 짝수가 연속되는 경우에도 ①과 비슷한 방법으로 처리한다.
   - 위의 작업을 반복한다.
   - 문자열을 화면으로 출력한다. 이때 %c 사용금지

▶ 함수를 정의하여 사용하는 문제이므로 전역변수 사용금지 **(위반 시 100% 감점)**
▶ 문자열을 사용자로부터 입력 받을 때 %c 사용금지 **(위반 시 50% 감점)**
▶ 문자열을 화면으로 출력할 때 %c 사용금지 **(위반 시 50% 감점)**
▶ 문자열 길이 구하기 표준 문자열 함수를 두 번 사용하지 않으면 **(40% 감점)**
▶ 함수 사용하지 않으면 **(각각 30% 감점)**

입력 예시 1

132456

출력 예시 1

1+32*456

입력 예시 2

33332222

출력 예시 2

3+3+3+32*2*2*2

※ [문제 4-1]에서 [문제 4-2]까지 연관된 문제이며, 배점이 각각 50, 100점이다. [문제 4-1]을 안 풀고 [문제 4-2]만 풀어도 100점을 부여한다. 가장 높은 배점 하나만 반영한다. 합산하지 않는다.

[문제 4-1] (50점) 문자열에 포함된 부분 문자열을 다른 문자열로 치환하는 프로그램을 문자열 함수를 이용하여 다음과 같이 작성 하시오.

(1) 공백을 포함하는 문자열 하나를 입력받고, 공백을 포함하지 않는 두 개의 문자열을 추가로 입력받는다. 첫 번째 문자열을 문자열A, 두 번째와 세 번째 문자열을 각각 문자열B와 문자열C라 한다. 문자열A는 영문 대소문자, 공백, 그리고 구두점(.)들로 이루어져 있고, 문자열B와 문자열C는 영문 대소문자로만 이루어져 있다.

(2) 문자열A에 포함된 모든 문자열B를 문자열C로 치환하고, 치환이 완료된 문자열을 문자열 출력(%s)으로 출력한다. 문자열A에 포함된 문자열B를 찾을 때, 대소문자를 구분한다. 즉, Abc와 abc는 같은 문자열이 아니다.

(3) 문자열A의 최대길이는 100이고, 문자열B와 문자열C의 최대길이는 10이다. 문자열B와 문자열C의 길이는 같지 않을 수 있으며, 두 문자열 모두 문자열A 보다 짧다. 치환이 완료된 최종 문자열의 최대길이는 250이다.

(4) 치환은 왼쪽 문자에서부터 순서대로 수행한다. (출력 예시 2 참조)
즉 AAA에는 AA가 1 개 있는 것으로 한다. AAAA에는 AA가 2개 있다. 즉 이미 한번 치환에 이용된 문자는 다음 치환에 사용하지 않는다.

(5) 사용 가능한 문자열 함수: strcpy, strcat, strlen, strcmp, strncmp

(힌트) strncmp 함수 원형
int strncmp(char *lhs, char *rhs, int size)
(예) strncmp(st1, st2, N) 두 문자열에서 각 단어의 앞부분 N개 문자만 비교한다.

- 프로그램 작성 시 다음의 조건들이 지켜져야 한다.

▶ strcpy, strcat, strlen, strcmp, strncmp 이외의 다른 문자열 함수 사용 시 **(50% 감점)**

▶ 문자열을 치환하여 하나의 문자열을 만들어 문자열 출력(%s)하지 않고, 문자 출력(%c)을 하거나, 문자열 출력(%s)을 반복해서 하는 경우 **(100%감점)**

입력 예시 1

```
I have a pen. I have an applepen. PPAP.
pen       ↦ 문자열 B
mouse     ↦ 문자열 C
```

출력 예시 1

```
I have a mouse. I have an applemouse. PPAP.
```

입력 예시 2

```
Sequence GTCGTCG  ↦ 문자열 A
GTCG              ↦ 문자열 B
CAGC              ↦ 문자열 C
```

출력 예시 2

```
Sequence CAGCTCG  ↦ 왼쪽부터 순서대로 바뀜
                  ↦ 두 번째 GTCG는 바뀌지 않음
```

[문제 4-2] (100점) 문자열에 포함된 단어를 다른 단어로 치환하는 프로그램을 문자열 함수를 이용하여 다음과 같이 작성 하시오.

(1) 공백을 포함하는 두 개의 문자열을 입력받는다. 첫 번째 문자열을 문자열A, 두 번째 문자열을 문자열B라 한다. **두 문자열 모두 영문 대소문자와 공백으로만 이루어져 있다**.

(2) 문자열A와 문자열B를 공백을 기준으로 단어로 나눈다. (공백과 공백 사이의 문자열을 하나의 단어라 한다. 단어와 단어 사이에는 하나의 공백이 존재한다.)

(3) 문자열A에 포함된 단어가 문자열A에 2회 이상 중복되어 나타나면 첫 번째 단어를 제외한 나머지 단어는 문자열B에 포함된 단어로 치환한다. 치환이 완료된 문자열을 문자열 출력(%s)으로 출력한다.

(4) 치환 작업은 문자열A의 첫 단어부터 순차적으로 진행되며, 문자열A의 각 단어에 대해 앞부분에 동일한 단어가 나온 적이 있는지 확인하고 치환한다. 치환될 단어는 문자열B의 첫 번째 단어부터 순차적으로 선택된다. 단, 문자열B의 모든 단어는 문자열A에 포함되어있지 않다고 가정하고, 문자열B의 단어 수는 문자열A의 단어 수보다 많다고 가정한다.

(5) 문자열A에 포함된 동일 단어는 대소문자를 구분한다. 즉, Abc와 abc는 같은 단어가 아니다.

(6) 문자열A와 문자열B의 최대길이는 100이고, 치환이 완료된 최종 문자열의 최대길이는 250이다.

(7) 사용 가능한 문자열 함수: strcpy, strcat, strlen, strcmp, strncmp

- 프로그램 작성 시 다음의 조건들이 지켜져야 한다.

▶ strcpy, strcat, strlen, strcmp, strncmp의 문자열 함수 중 3 종류 이상 사용 **(미사용 시 각 25% 감점)**

▶ strcpy, strcat, strlen, strcmp, strncmp 이외의 다른 문자열 함수 사용 시 **(50% 감점)**

▶ 문자열을 치환하여 하나의 문자열을 만들어 문자열 출력(%s)하지 않고, 문자 출력(%c)을 하거나, 문자열 출력(%s)을 반복해서 하는 경우 **(100%감점)**

입력 예시 1

red orange red yellow green red blue purple yellow ↦ 문자열 A
white black gray pink brown blush crimson garnet vermilion indigo ↦ 문자열 B

출력 예시 1

red orange white yellow green black blue purple gray
↦ 문자열A의 중복 단어 red/red/yellow는 문자열B의 첫 번째, 두 번째, 세 번째 단어인 white/black/gray로 치환

※ [문제 5-1]에서 [문제 5-3]까지 연관된 문제이며, 배점이 각각 30, 60, 100점이다. [문제 5-3]만 풀어도 100점을 부여한다. 가장 높은 배점 하나만 반영한다. 합산하지 않는다.

[문제 5-1] (30점) K 전자는 필기시험 점수의 80%와 면접시험 점수의 20%를 합산한 총점이 높은 순으로, N명의 지원자 중 M명의 신입직원을 선발 한다. 단, **총점이 같은 동점자는 없다고 가정한다.** 각 지원자의 이름과 필기시험, 면접시험 점수가 아래 예시와 같이 주어질 때 M명 합격자의 이름과 최종 점수를 출력하는 프로그램을 작성하시오.

- 먼저, 지원자 수 N과 합격자 수 M이 입력된다. N≦100이고, M≦N이다.
- 다음은 N명 지원자의 이름, 필기시험 점수, 면접시험 접수가 순서대로 입력된다.
- 이름은 공백을 포함하지 않는 최대 10자의 문자열 이며, 필기시험과 면접시험 점수는 100점 만점이다.
- 합격자의 이름, 필기시험 점수의 80%와 면접시험 점수의 20%가 반영된 총점을 소수점 첫째 자리까지 출력한다. 부동소수 자료형은 double을 사용하시오.
- **출력순서**는 선발순서(총점이 높은 합격자부터 낮은 합격자 순) 이다.
- main() 함수에서 지원자를 정렬하는 과정에 아래의 swap() 함수를 호출하여 사용한다.

(1) swap() 함수

- 함수 원형: **void swap(struct person *p, struct person *q)**
- p와 q는 구조체 배열의 두 원소의 주소이며, 두 원소의 내용을 바꾸는 작업을 한다.

▶ 구조체 배열을 사용하지 않은 경우 **(100% 감점)**
▶ swap() 함수를 사용하지 않거나 함수의 원형을 지키지 않은 경우 **(20% 감점)**
▶ swap() 함수 내에서 구조체 대입 연산을 사용하지 않은 경우 **(10% 감점)**

입력 예시 1

```
7 3          ↦ N=7, M=3
Aaaaa 75 85
Bbbbb 86 77
Ccccc 94 82
Ddddd 78 70
Eeeee 65 85
Fffff 83 90
Ggggg 90 70
```

출력 예시 1

```
Ccccc 91.6
Ggggg 86.0
Fffff 84.4
```

[문제 5-2] (60점) K 전자는 필기시험 점수의 80%와 면접시험 점수의 20%를 합산한 총점이 높은 순으로, N명의 지원자 중 M명의 신입직원을 선발 한다. 단, **총점이 같은 동점자가 있으면, 필기시험 점수가 더 높은 사람을 선발한다. 필기시험 점수와 면접시험 점수가 모두 같은 동점자는 없다고 가정한다.** 다른 조건은 [문제 5-1]과 같을 때, M명 합격자의 이름과 최종 점수를 출력하는 프로그램을 작성하시오. 부동소수 자료형은 double을 사용하시오.

- **출력순서**는 선발순서이다. 즉, 총점이 높은 합격자부터 낮은 합격자 순, 총점이 같은 경우에는 필기시험 점수가 높은 합격자부터 낮은 합격자 순이다.

▶ 구조체 배열을 사용하지 않은 경우 **(100% 감점)**
▶ swap() 함수를 사용하지 않거나 함수의 원형을 지키지 않은 경우 **(20% 감점)**
▶ swap() 함수 내에서 구조체 대입 연산을 사용하지 않은 경우 **(10% 감점)**

입력 예시 1	출력 예시 1
7 3 ↦ N=7, M=3 Aaaaa 75 85 Bbbbb 86 80 Ccccc 94 82 Ddddd 78 70 Eeeee 65 85 Fffff 85 84 Ggggg 90 64	Ccccc 91.6 Ggggg 84.8 Bbbbb 84.8 ↦ 3명의 동점자 Bbbbb, Ffffff, Ggggg중 필기시험 점수가 높은 Ggggg, Bbbbb 순으로 선발되고, Fffff는 탈락

[문제 5-3] (100점) K전자는 필기시험 점수의 80%와 면접시험 점수의 20%를 합산한 총점이 높은 순으로, N명의 지원자 중 M명의 신입직원을 선발한다. 단, **총점이 같은 동점자가 있으면, 필기시험 점수가 더 높은 사람을 선발하고, 필기시험 점수와 면접시험 점수도 모두 같은 경우에는 생년월일이 늦은 사람을 선발한다. 모든 시험 점수와 생년월일까지 같은 동점자는 없다고 가정한다.**

N명 지원자의 정보는 이름, 생년월일, 필기시험 점수, 면접시험 점수 순서대로 입력되며, 생년월일은 입력과 같이 생년 4자리, 생월 2자리, 생일 2자리가 '-' 문자로 구분되어 입력된다. 다른 조건은 [문제 5-1]과 같을 때, M명 합격자의 이름과 최종 점수를 출력하는 프로그램을 작성하시오.

- **출력순서**는 선발순서이다.
  1) 총점이 높은 합격자부터 낮은 합격자 순
  2) 총점이 같은 경우에는 필기시험 점수가 높은 합격자부터 낮은 합격자 순
  3) 모두 같은 경우에는 생년월일이 가장 늦은 합격자부터 빠른 합격자 순
- 부동소수 자료형은 double을 사용하시오.

▶ 구조체 배열을 사용하지 않은 경우 **(100% 감점)**
▶ swap() 함수를 사용하지 않거나 함수의 원형을 지키지 않은 경우 **(20% 감점)**
▶ swap() 함수 내에서 구조체 대입 연산을 사용하지 않은 경우 **(10% 감점)**

입력 예시 1	출력 예시 1
7 3 ↦ N=7, M=3 Aaaaa 1995-10-07 75 85 Bbbbb 1991-11-15 90 64 Ccccc 1994-12-08 94 82 Ddddd 1993-07-17 78 70 Eeeee 1995-10-07 65 85 Fffff 1991-11-05 90 64 Ggggg 1990-08-21 90 64	Ccccc 91.6 Bbbbb 84.8 Fffff 84.8 ↦ 필기와 면접 시험 점수가 같은 3명의 동점자 Bbbbb, Ffffff, Ggggg중 생일이 늦은 Bbbbb, Fffff 순으로 선발되고, Ggggg는 탈락