

## <고급C 프로그래밍 및 실습> 중간고사 (1차 인증시험) 문제지

2020.10.22 (목)

학번	학과	이름

### ※ 문제에 대한 안내

- 문제지는 총 10페이지이고, 총 5문제 500점 만점이고, 문제의 순서는 난이도와 관계없다.
- 특별한 언급이 없으면 문제의 조건에 맞지 않는 입력은 입력되지 않는다고 가정하라.
- 입출력 예시에서  $\mapsto$  이 후는 각 입력과 출력에 대한 설명이다.
- 한 문제가 여러 소문제로 분리된 경우, **가장 높은 점수 하나만** 최종 점수로 간주한다.
- OJ에서 Sample Submit 기능사용가능하며, 점수에는 포함되지 않는다.

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

**[ 문제 1-1 ] (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 인자로 받아 종료 조건까지 정수를 입력받아 배열에 저장하고, 배열 원소의 개수를 반환한다.

#### (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 인자로 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 인자로 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 <span style="float: right;">↦ M=1</span>

출력 예시 1

2345
<span style="float: right;">↦ x[1]인 2부터 연속적으로 증가하는 5까지 모아 2345 출력</span>

입력 예시 2

1 2 3 4 5 4 3 -1
4 <span style="float: right;">↦ M=4</span>

출력 예시 2

5
<span style="float: right;">↦ x[4]인 5 이후 증가하는 수가 없으므로 5 출력</span>

**[ 문제 1-2 ] (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 1-1 ]과 동일

- 함수 원형: `int input(int *p)`
- 배열 x의 시작 주소를 인자로 받아 종료 조건까지 정수를 입력받아 배열에 저장하고, 배열 원소의 개수를 반환한다.

(2) sel\_next 함수, [ 문제 1-1 ]과 다름

- 함수 원형: `int *sel_next(int *p)`
- 배열의 한 원소의 주소 p를 인자로 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 1-1 ]과 동일

- 함수 원형: `int number(int *p, int *q)`

- 배열의 두 원소의 주소 p와 q를 인자로 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 2-1]에서 [문제 2-2]까지 연관된 문제이며, 배점이 각각 50, 100점이다. [문제 2-1]을 안 풀고 [문제 2-2]만 풀어도 100점을 부여한다. 가장 높은 배점 하나만 반영한다. 합산하지 않는다.

**[ 문제 2-1 ] (50점)** 사용자로부터 여러 단어가 포함된 공백을 포함하는 하나의 문장을 하나의 문자열로 입력 받고 (입력 시 %c 사용금지), 문자의 개수가 가장 긴 단어를 인쇄하는 (%c 사용금지) 프로그램을 작성하시오.

- 문자열은 알파벳 소문자로만 구성된다고 가정한다.
- 문자의 개수가 동일한 단어는 입력되지 않는다고 가정한다.
- 입력되는 문자열의 길이는 최대 100 이다.
- 입력되는 문자열에서, 단어와 단어 사이에 공백문자가 한 개 있다.
- 입력받은 문자열에서, 첫 문자가 공백인 경우는 없다고 가정한다.

- ▶ 문자열 입력 시 gets 사용 (위반 시 100% 감점)
- ▶ 문자열을 화면으로 출력할 때 %c 사용금지 (위반 시 50% 감점)
- ▶ 강의자료에 제시된 표준 문자열 함수 모두 사용 가능  
단, strtok 와 같이 수업 시간에 다루지 않은 표준 문자열 함수 사용금지 (위반 시 50% 감점)

입력 예시 1

why so serious

출력 예시 1

serious

입력 예시 2

icecream makes everything better

출력 예시 2

everything

**[ 문제 2-2 ] (100점)** 사용자로부터 여러 단어가 포함된 **공백을 포함하는** 여러 개의 문장을 여러 개의 문자열로 입력 받고 (입력 시 %c 사용금지), 입력된 문자열들 중 가장 짧은 문자열에 포함된 가장 짧은 단어와 가장 긴 문자열에 포함된 가장 긴 단어를 각각 인쇄하는 (%c 사용금지) 프로그램을 작성하시오.

- 첫 입력은 입력될 문자열의 개수이다.
- 각 문자열은 알파벳 소문자로만 구성된다고 가정한다.
- 하나의 문자열 내에는 문자의 개수가 동일한 단어는 존재하지 않는다고 가정한다.
- 길이가 동일한 문자열은 입력되지 않는다고 가정한다.
- 입력되는 문자열의 개수는 최소 2 이고 최대 10 이며, 각 문자열의 길이는 최대 100 이다.
- 입력되는 각 문자열에서, 단어와 단어 사이에 공백문자가 한 개 있다.
- 입력받은 문자열에서, 첫 문자가 공백인 경우는 없다고 가정한다.

▶ 문자열 입력 시 gets 사용 (위반 시 100% 감점)

▶ 문자열을 화면으로 출력할 때 %c 사용금지 (위반 시 50% 감점)

▶ 강의자료에 제시된 표준 문자열 함수 모두 사용 가능

단, strtok 와 같이 수업 시간에 다루지 않은 표준 문자열 함수 사용금지 (위반 시 50% 감점)

입력 예시 1

2  
why so serious  
icecream makes everything better

출력 예시 1

so  
everything

입력 예시 2

3  
you make me smile  
c language midterm exam  
i see you

출력 예시 2

i  
language

예시 2에서 3번째 문자열이 가장 짧은 문자열이다. 3번째 문자열에서 가장 짧은 단어는 "i" 이다.  
예시 2에서 2번째 문자열이 가장 긴 문자열이다. 2번째 문자열에서 가장 긴 단어는 "language" 이다.

※ [문제 3-1]에서 [문제 3-2]까지 연관된 문제이며, 배점이 각각 50, 100점이다. [문제 3-1]을 안 풀고 [문제 3-2]만 풀어도 100점을 부여한다. 가장 높은 배점 하나만 반영한다. 합산하지 않는다.

**[ 문제 3-1 ]** (50점) 첫 번째 줄에서, 영문 대소문자와 공백들로만 이루어진 하나의 문자열을 입력받는다. 두 번째 줄에서 영문 대소문자로만 이루어진 문자열 하나를 입력받는다. 첫 번째 입력받은 문자열은 공백을 기준으로 단어가 구분되며, 두 번째 문자열은 하나의 단어를 의미한다.

첫 번째 입력받은 문자열에 있는 (공백으로 구분되는) 단어들 중에서, 두 번째 입력받은 단어와 **같지 않는 단어만** 차례대로 모두 출력하시오. **힌트:** 그러므로 하나의 문자열을 만들어 출력하지 말고, 여러 번의 %s를 이용하여 출력이 가능하다.

- 1) 첫 번째 입력받는 문자열의 전체 길이는 (널 문자 포함하여) 150을 넘지 않는다.
- 2) 첫 번째 입력받는 문자열을 구성하는 총 단어 개수는 15개 이하이다.
- 3) 두 번째 입력받는 문자열의 길이는 (널 문자 포함하여) 10을 넘지 않으며, 항상 첫 번째 입력받은 문자열의 길이보다는 짧다.
- 4) 대소문자는 구분한다.

- ▶ 강의노트에 있는 문자열 처리 함수들 중 2가지 이상 반드시 사용 (**위반 시 각 25% 감점**)
- ▶ 입력받는 문자열을 저장하는 2개 배열에서, 필요한 문자열의 길이보다 배열 크기가 작은 경우에 **10% 감점**
- ▶ 출력 시, %c 혹은 %d 사용 금지 (**위반 시 100% 감점**)

입력 예시 1

출력 예시 1

I have a <u>pen</u> I have an applepen PPAP <u>pen</u>	I have a I have an applepen PPAP
---	----------------------------------

입력 예시 2

출력 예시 2

Sequence GTCGTCG GTCG	Sequence GTCGTCG    ↳    입력받은 문자열 그대로 출력
--------------------------	--

**[ 문제 3-2 ]** (100점) 영문 대소문자와 숫자문자, 그리고 공백들로만 이루어진 하나의 문자열을 입력받는다 (문자열 A라 하자). 이후, 숫자문자로만 이루어진 문자열 하나를 입력받는다 (문자열 B라 하자). 마지막으로, 영문 대소문자로만 이루어진 문자열 하나를 더 입력받는다 (문자열 C라 하자). 문자열 A는 공백을 기준으로 단어가 구분되며, 공백은 연속으로 여러 번 나올 수 있다. 또한 각 단어는 숫자문자로만 혹은 영문자로만 구성된다. 문자열 B와 문자열 C는 각각 하나의 단어를 의미한다. 문자열 A에서 문자열 B와 일치하는 단어를 찾아 문자열 C로 치환하여 출력하는 프로그램을 작성하시오.

- 1) 문자열 A의 전체 길이는 (널 문자 포함하여) 150을 넘지 않는다.
- 2) 문자열 A를 구성하는 총 단어 개수는 15개를 넘지 않는다.
- 3) 문자열 B와 문자열 C는 (널 문자 포함하여) 10을 넘지 않으며, 항상 문자열 A의 길이보다는 짧다.
- 4) 대소문자는 구분한다.

- 5) 치환 완료된 문자열의 길이는 (널 문자 포함하여) 300자를 넘지 않는다.  
 6) 최종 문자열 출력 시, 여러 개의 공백은 한 개의 공백으로 만들어 출력하며, 하나의 문자열로 출력한다. (하나의 %s, 혹은 한 번의 puts로 출력)

- ▶ 강의노트에 있는 문자열 처리 함수들 중 3가지 이상 반드시 사용 (2가지 함수만 사용 시 25% 감점, 1가지 이하의 함수 사용 시 50% 감점)  
 ▶ 입력받는 문자열을 저장하는 3개 배열에서, 필요한 문자열의 길이보다 배열 크기가 작은 경우에 10% 감점  
 ▶ 출력 시 %s 한 번 사용하거나, puts 함수 한 번 사용하시오. (위반 시 100% 감점)

입력 예시 1

```
I have a 111 I have an applepen PPAP
111
mouse
```

출력 예시 1

```
I have a mouse I have an applepen PPAP
```

↳ □는 공백 한 개를 의미한다.

입력 예시 2

```
www sejong 23 kr
23
co
```

↳ 문자열 A  
↳ 문자열 B  
↳ 문자열 C

출력 예시 2

```
www sejong co kr
```

↳ □는 공백 한 개를 의미한다.

※ [문제 4-1]에서 [문제 4-3]까지 연관된 문제이며, 배점이 각각 30, 60, 100점이다. [문제 4-3]만 풀어도 100점을 부여한다. 가장 높은 배점 하나만 반영한다. 합산하지 않는다.

**[ 문제 4-1 ] (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.  $N \leq 100$ 이고,  $M \leq N$ 이다.
- 다음은 N명 지원자의 이름, 필기시험 점수, 면접시험 점수가 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

출력 예시 1

7 3      ↳ N=7, M=3	Ccccc 91.6
Aaaaa 75 85	Ggggg 86.0
Bbbbb 86 77	Fffff 84.4
Ccccc 94 82	
Ddddd 78 70	
Eeeee 65 85	
Fffff 83 90	
Ggggg 90 70	

[ 문제 4-2 ] (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 4-1 ]과 같을 때, M명 합격자의 이름과 최종 점수를 출력하는 프로그램을 작성하시오. 부동소수 자료형은 double을 사용하시오.

- 출력순서는 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	Ccccc 91.6	
Aaaaa 75 85	Ggggg 84.8	
Bbbbb 86 80	Bbbbb 84.8	↳ 3명의 동점자 Bbbbb, Ffffff, Ggggg중 필기시험 점수가 높은 Ggggg, Bbbbb 순으로 선발되고, Fffff는 탈락
Ccccc 94 82		
Ddddd 78 70		
Eeeee 65 85		
Fffff 85 84		
Ggggg 90 64		

[ 문제 4-3 ] (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 4-1 ]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 낮은 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

<pre> 7 3      ↳ N=7, M=3 Aaaaa 1995-10-07 75 85 Bbbbbb 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                 </pre>	<pre> Ccccc 91.6 Bbbbb 84.8 Fffff 84.8                 </pre> <div style="margin-left: 20px;">                         ↳ 필기와 면접 시험 점수가 같은                          3명의 동점자 Bbbbb, Ffffff,                          Ggggg중 생일이 낮은                          Bbbbb, Fffff 순으로 선발되고,                          Ggggg는 탈락                     </div>
--	--

※ [문제 5-1]에서 [문제 5-2]까지 연관된 문제이며, 배점이 각각 50, 100점이다. [문제 5-1]을 안 풀고 [문제 5-2]만 풀어도 100점을 부여한다. 가장 높은 배점 하나만 반영한다. 합산하지 않는다.

**[ 문제 5-1 ] (50점)** 양의 정수 N과 N개의 단어를 입력받아 각 단어에서 모음이 나온 회수를 구조체 변수의 멤버 변수에 저장한다. 모음 개수가 가장 많은 단어부터 모음 개수가 가장 적은 단어 까지 차례대로 출력하는 프로그램을 작성하시오. 아래의 구조체 형을 사용하여 구조체 배열을 선언하여 사용하고 모든 단어에 대하여 모음의 개수를 구하는 count\_word 함수를 사용한다.

- 구조체 형 : struct word { char letters[21]; int count; };
- 1 <= N <= 100
- 사용자로부터 입력받는 단어는 공백(빈칸) 없이 영문자 최대 20자 이다.
- 함수 원형 : void count\_word(struct word \*word, int N);
  - 인자 : 단어를 가리키는 구조체 배열의 시작 주소와 단어의 개수 N
  - 반환값 없음.
  - 구조체 배열내의 count값들을 구하여 채운다. (모음개수)
  - 함수 내부에서는 구조체 배열을 반복문으로 순회할 경우 주소를 이용하여 반복할 것



(구조체 포인터 사용)

- 함수 내부에서 letters 멤버 변수에 대한 작업 (반복문, 멤버변수 체크 등) 에서는 배열 기호를 사용할 수 있다. (**letters[i] 표현은 사용 가능**)

- 메인함수에서 정수 N과 N개의 단어를 입력받고 count\_word함수를 호출하여 모음의 개수를 구한 후, 단어의 모음개수 순으로 정렬하여 출력한다. 만일 모음의 개수가 같은 경우에는 먼저 나온 단어를 출력한다. **메인함수에서는 배열기호 사용가능.**
- 모음은 A, E ,I, O, U, a, e, i, o, u 이다.

- ▶ 함수 내에서 구조체 포인터 사용하지 않거나, 구조체 배열을 반복문으로 순회할 경우 주소를 이용하여 포인터를 이동시키며, 반복문 구현을 하지 않은 경우 (100%감점)
- ▶ 함수 내부에서 화면 출력을 하거나, 전역변수 사용 시 (100%감점)
- ▶ 모음의 개수를 구조체 배열 안에 저장하지 않으면 (100%감점)
- ▶ 지정된 함수 원형을 지키지 않으면 (100% 감점)
- ▶ 필요한 단어의 길이보다 적게 배열크기를 잡은 경우 (10%감점)
- ▶ 정렬에서 구조체 원소교환 시 멤버 변수값을 각각 이동시킨 경우 (30%감점)

입력 예시 1

```
5
TimeISEnough
MyNameIsKim
HelloFRIEND
CATCHMEIFYOUCAN
NOPAINNOGAIN
```

출력 예시 1

```
6 TimeISEnough
6 CATCHMEIFYOUCAN
6 NOPAINNOGAIN
4 MyNameIsKim
4 HelloFRIEND
```

입력 예시 2

```
6
preview
Released
The
Second
EDITION
POPular
```

출력 예시 2

```
4 Released
4 EDITION
3 preview
3 POPular
2 Second
1 The
```

**[ 문제 5-2 ] (100점)** 양의 정수 N과 N개의 단어를 입력받아, 이 중 단어 3개를 이용하여 끝말잇기를 할 때 만들어지는 가장 긴 문장을 찾아 그 문장의 길이와 문장을 출력하는 프로그램을 작성하여라. 단, 단어 3개의 모음의 개수가 홀수, 짝수, 홀수 개인 단어순으로 끝말잇기가 되어야 한다. 대문자는 모두 소문자로 변환하고, 화면출력 할 때 소문자로만 출력한다. 끝말잇기는 단어의 마지막 문자로 시작하는 단어를 찾아 연결하는 방식으로 진행되고 단어와 단어 사이에는 공백을 1개

두어 연결한다. 단어는 한번만 사용이 될 수 있다. [ 문제 5-1 ]과 동일하게 아래의 구조체 형을 사용하여 구조체 배열을 만들고, [ 문제 5-1 ]의 count\_word함수를 약간 확장하여 사용한다.

- 3개 단어가 위 방식대로 연결이 되지 않을 경우 **NONE**을 출력한다. 중복된 단어는 없다. 만일 같은 길이의 문장이 여러 개 만들어 지면, 그 중 문장의 첫 단어가 제일 나중에 입력된 문장을 출력한다. (그 외 같은 길이의 문장은 만들어 지지 않는다고 가정한다.)

- 구조체 형 : struct word { char letters[21]; int count; };
- 1 <= N <= 100
- 사용자로부터 입력받는 단어는 공백(빈칸) 없이 영문자 최대 20자 이다.
- 함수 원형 : void count\_word(struct word \*word, int N);
  - 인자 : 단어를 가리키는 구조체 배열의 시작 주소와 단어의 개수 N
  - 반환값 없음.
  - 대문자는 소문자로 변경한다. 구조체 배열 내의 각 원소의 count 멤버 변수에 모음개수를 저장한다.
  - 함수 내부에서는 배열기호 사용금지 및 모든 배열을 반복문으로 순회할 경우 주소를 이용하여 반복할 것 (구조체 포인터와, 문자형 포인터 사용)  
(letters[i] 표현도 사용 가능하지 않음)

- ▶ 함수 내부에서 화면 출력을 하거나, 전역변수 사용 시 (100%감점)
- ▶ 모음의 개수를 구조체 배열 안에 저장하지 않으면 (100%감점)
- ▶ 지정된 함수 원형을 지키지 않으면 (100% 감점)
- ▶ 필요한 단어의 길이보다 적게 배열크기를 잡은 경우 (10%감점)
- ▶ 함수 내에서 구조체 포인터 사용하지 않거나, 구조체 배열을 반복문으로 순회할 경우 주소를 이용하여 포인터를 이동시키며, 반복문 구현을 하지 않은 경우 (100%감점)
- ▶ 함수 내에서 문자형 포인터 사용하는 대신에 [ ] 기호를 사용하면 (20%감점)

입력 예시 1

```
8
Toe
Rat
TAKE
Eight
Table
Tiger
Rare
Enrgy
```

출력 예시 1

```
15 rat table enrgy
    ↳ "rat table enrgy"
      "rat toe enrgy"
      "rat take enrgy"
      중에 제일 긴 것의 길이와 문장
```

입력 예시 2

```
3
TestIsMyWay
Hello
World
```

출력 예시 2

```
NONE
```