

〈고급C 프로그래밍 및 실습〉 2차 모의고사 문제지

2017.11.16

학과	학번	이름

※ 문제에 대한 안내

- 문제지는 총 6페이지이고, 총 5문제 100점 만점이고, 문제의 순서는 난이도와 관계없다.
- 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.
- **Sample Submit** 가능합니다.

[문제 1] 정수 M과 N을 입력 받은 후, M개의 정수를 입력받아 배열 X에, N개의 정수를 입력받아 배열 Y에 저장 한다. 다음, 정수 K를 입력받아, X와 Y에 저장된 정수를 K개씩 번갈아가며 출력 하는 프로그램을 다음의 함수를 사용하여 작성 하시오. 단, M과 N은 100보다 작은 K의 배수이며, X에 저장된 정수부터 K개씩 출력한다.

- input과 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. 단, main 함수에서의 주소 표현 방식은 제한 사항 없음.

1) input 함수

- 함수 원형: **void input(int *p, int N)**
- p는 배열의 시작 주소이며, 정수 N개를 입력받아 p가 가리키는 배열에 저장한다.

2) output 함수

- 함수 원형: **void output(int *p, int K)**
- p는 배열의 한 원소의 주소이며, 이 원소부터 시작하여 배열에 저장된 정수 K개를 출력한다.

- ▶ 함수의 원형(인자, 반환값)을 지키지 않거나 목적에 맞게 함수를 사용하지 않은 경우 (**input, output 각 50% 감점**)
- ▶ 배열 선언 이외에 배열 표기 []를 사용하는 경우 (**main, input, output 각 25% 감점**)
- ▶ 반복문으로 배열 훑어볼 시, 주소를 이용하여 반복문 구현을 하지 않은 경우 (**input, output 각 25% 감점**)
- ▶ 전역변수를 사용하는 경우 (**100% 감점**)

입력 예시 1

```
6 3      ↪ M=6, N=3
1 2 3 4 5 6
7 8 9
3        ↪ K=3
```

출력 예시 1

```
□1□2□3□7□8□9□4□5□6
```

입력 예시 2

```
4 8      ↪ M=4, N=8
1 2 3 4
5 6 7 8 9 10 11 12
2        ↪ K=2
```

출력 예시 2

```
□1□2□5□6□3□4□7□8□9□10□11□12
```

↪ k개씩 번갈아 출력 후 남은 정수 연속 출력

[문제 2] 세 개의 복소수를 입력받고, 절대 값이 가장 큰 복소수와 절대 값이 가장 작은 복소수의 합과 차를 구하여, 아래 출력예시와 같이 $\pm bi$ 형태로 출력하는 프로그램을 다음의 함수를 이용하여 작성 하시오. **단, a와 b는 소수점 첫째자리까지 출력한다.**

- 복소수를 표현하는 complex 구조체를 정의하여 사용하시오.

- 실수부를 나타내는 실수형 변수, 허수부를 나타내는 실수형 변수, 절대값을 나타내는 실수형 변수

힌트1) 복소수 $a+bi$ 의 절대값 계산은 $a^2 + b^2$ 의 공식으로 계산되며, 절대 값 비교 시에는 $a^2 + b^2$ 으로도 비교 가능하다. ($\sqrt{a^2 + b^2}$ 계산 시, `#include<math.h>` 선언 후 `sqrt()` 함수 사용)

힌트2) double형 실수 입력은 `scanf("%lf", ...)`를, float형 실수 입력은 `scanf("%f", ...)`를 사용한다.

힌트3) 실수 출력 시 `printf("%+.1f", ...)`를 사용하면, 양수인 경우에는 +, 음수인 경우에는 - 기호가 숫자 앞에 표시된다. 0인 경우에는 +0.0으로 표시된다.

1) add함수

- 인자: 두 개의 struct complex형 변수
- 두 복소수의 덧셈 결과를 반환한다.
- 반환값: struct complex형 변수 값

2) sub함수

- 인자: 두 개의 struct complex형 변수
- 첫 번째 복소수에서 두 번째 복소수를 뺀 결과를 반환한다.
- 반환값: struct complex형 변수 값

3) abs_val함수

- 함수 원형: `void abs_val(struct complex *p1, struct complex *p2, struct complex *p3)`
- p1, p2, p3이 가리키는 구조체 변수의 절대 값을 계산하고, p1이 가리키는 변수에 절대 값이 가장 큰 복소수를, p2가 가리키는 변수에 절대 값이 가장 작은 복소수를 저장한다.
- 반환값 : 없음.

- ▶ 구조체를 사용하지 않은 경우 (100% 감점)
- ▶ 함수의 원형(인자, 반환값)을 지키지 않거나 함수를 사용하지 않은 경우 (add, sub, abs_val 각 30% 감점)
- ▶ 전역변수를 사용하는 경우 (100% 감점)

입력 예시 1

2.3 4.5	↪ 첫 번째 복소수 (=2.3+4.5i)
3.4 5.0	↪ 두 번째 복소수 (=3.4+5i)
-1.4 7.2	↪ 세 번째 복소수 (=-1.4+7.2i)

출력 예시 1

0.9+11.7i	↪ 복소수의 합
-3.7+2.7i	↪ 복소수의 차

입력 예시 2

2.3 4.5	↪ 첫 번째 복소수 (=2.3+4.5i)
3.4 5.0	↪ 두 번째 복소수 (=3.4+5i)
1.4 -7.2	↪ 세 번째 복소수 (=1.4-7.2i)

출력 예시 2

3.7-2.7i	↪ 복소수의 합
-0.9-11.7i	↪ 복소수의 차

[문제 3] 정수 N을 입력받고, 공백을 포함하지 않는 문자열 한 개를 입력받아 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.

- Y를 구조체에 대한 포인터 변수로 선언하고, 크기 N의 구조체 배열을 할당 받아도 됨. 이 경우, 구조체 배열의 각 원소는 문자열 포인터 변수를 멤버로 포함하여야 함.
- 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)
- ▶ 문자열 입출력 시에 %s를 사용하지 않고, %c 혹은 %d를 사용하면 (50% 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 4] 컴퓨터공학과 1학년 학생들의 고급C언어 인증 시험 결과를 확인하고자 한다. 사용자로부터 학생 수 N을 입력받고 이후 **학번, 1차 인증 점수, 2차 인증 점수**를 입력할 수 있는 구조체 N개를 동적으로 할당받아 학생들의 정보를 입력받는다. 인증 통과 규칙은 다음과 같다.

- 조건 1: 1차 인증 점수 40%, 2차 인증 점수 60%의 합산으로 전체 학생의 60%를 인증통과자로 선발 한다. 예를 들어, N이 7인 경우 7명의 60% 인 4명을 선발 한다.
- 조건 2: 이후 탈락된 학생 중, 2차 인증 점수 우수자를 선발하되 전체 인증 통과자가 총 학생 수의 80%가 되도록 선발 한다. 예를 들어, N이 7인 경우 총 선발 인원이 7명의 80%인 5명이 되게 한다. 즉, 조건 1에 의한 선발 이후, 1명을 추가 선발 한다.
- 단 조건 1, 2에서 동점자가 커트라인에 있을 경우 모두 통과시킨다.
- 조건 1에 의해 선발된 학생이 동점자로 인해 전체 학생의 80%이상인 경우에는 조건 2에 의한 2차 선발은 없다.

위의 조건으로 인증 통과한 학생들을 선발한 후 학번을 오름차순으로 정렬하여 출력하시오. 관련된 코딩 규칙은 다음과 같다.

- 적어도 5개의 학생 점수를 입력(N은 5이상)
- 학번은 모두 5자리의 정수
- 각 인증 점수는 0점 ~ 100점 사이의 값

힌트1) 구조체 내부에 두 인증 점수의 총점을 저장하는 실수형 멤버 변수를 추가해서 사용

힌트2) 통과자 저장을 위한 별도의 구조체 N개를 동적할당 받아서 사용

- ▶ 동적 할당을 사용하지 않으면 (100% 감점)
- ▶ 동적으로 할당 받은 메모리 해제를 하지 않으면 (10% 감점)
- ▶ typedef 사용하지 않으면 (20% 감점)

▶ 시간 관계상, 동적 할당 시 NULL 반환값 체크하지 않으면 (10% 감점)

입력 예시 1

```
10
12345 15 86  ↳ 57.6 ... 7위
12346 52 68  ↳ 61.6 ... 5위
12347 55 90  ↳ 76 ... 2위
12348 32 88  ↳ 65.6 ... 4위
12349 22 12  ↳ 16 ... 9위
12350 65 9   ↳ 31.4 ... 8위
12351 1 100  ↳ 60.4 ... 6위
12352 2 10   ↳ 6.8 ... 10위
12353 50 90  ↳ 74 ... 3위
12354 90 100 ↳ 96 ... 1위
```

출력 예시 1

```
12345
12346
12347
12348
12349 ↳ 총점은 9위지만(60%탈락) 2차점수가 8위보다 높음
12351
12353
12354
```

입력 예시 2

```
6
12345 80 100
12121 80 100
12358 80 100
12364 80 100
13548 80 100
13154 80 100
```

출력 예시 2

```
12121
12345
12358
12364
13154
13548 ↳ 모두 동점이므로 모두 통과
```

입력 예시 3

```
7
11325 78 12  ↳ 38.4 ... 7위
11215 45 65  ↳ 57 ... 4위
13125 55 50  ↳ 52 ... 6위
11111 12 87  ↳ 57 ... 4위
12233 1 100  ↳ 60.4 ... 3위
13158 70 90  ↳ 82 ... 1위
14589 67 78  ↳ 73.6 ... 2위
```

출력 예시 3

```
11111
11215
12233
13158
14589
```

[문제 5] 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)
```