

<고급 C프로그래밍 및 실습> 과제 #1 (9장 포인터)

※ 문제에 대한 안내

- 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 기능사용 가능합니다.

※ [문제 1-1]에서 [문제 1-2]까지는 연관된 문제이며, 배점이 각각 50점, 100점이다. [문제 1-1]을 안 풀고 [문제 1-2]만 풀어도 100점을 부여한다. 가장 높은 점수 하나만 반영하며, 합산하지 않는다. [문제 1-1]을 풀고 [문제 1-2]를 풀면 쉽게 문제를 풀 수 있다.

[문제 1-1][레벨 1] (50점) 0 이 입력될 때까지만 정수를 입력받는다. 0은 두 번째부터 입력될 수 있다. 0을 제외한 최대값과 최소값의 차이를 출력하시오. 이 과정을 N번 반복하시오.

1) maxmin() 함수 이용

- 인자: int형 배열 ar
- 배열 표기 [] 사용금지
- 최대값과 최소값의 차이를 반환한다.
- 반환 값: 정수형

2) main 함수에서는

- 사용자로부터 정수 N을 입력 받는다. 아래 과정을 N번 반복한다.
 - 사용자로부터 정수를 입력 받는다. 0이 되면 종료한다. 배열 ar의 크기는 100.
 - maxmin 함수를 호출한다.
 - maxmin 함수에서 전달받은 정수를 화면에 출력한다.

▶ 배열 선언 이후, 배열 표기 [] 는 사용 금지 (포인터 표기 사용) (50% 감점)

▶ 사용자로부터 정수들 입력하여 배열에 저장 시에 포인터 사용 (20% 감점)

▶ 반복문으로 배열 훑어볼 시, 주소를 이용하여 반복문 구현 (강의자료 p. 42 참조) (50% 감점)

(예: for(p = ar ; p < ar+5 ; p++)

여기서 p는 정수형 포인터, ar은 배열의 이름)

▶ 전역변수 사용금지 (100% 감점)

입력 예시 1

```
3          ↳ N
3 7 6 -2 0
13 7 6 1 2 0 -1 14
2 0
```

출력 예시 1

```
9      ↳ 7 - (-2)=9
12     ↳ 13-1=12
0      ↳ 2-2=0
```

입력 예시 2

```
2
13 7 6 6 2 0 -1 5
10 7 6 6 5 3 0 -1 3
```

출력 예시 2

```
11
7
```

[문제 1-2][레벨 2] (100점) 0 이 입력될 때까지만 정수를 입력받는다. 0은 두 번째부터 입력될 수 있다. 0을 제외한 최대값과 최소값의 차이와 가장 가까운 정수를 입력에서 찾아 출력하시오. 이 과정을 N번 반복하시오.

1) maxmin() 함수 이용

- 인자: int형 배열 ar
- 배열 표기 [] 사용금지
- 최대값과 최소값의 차이와 가장 가까운 배열의 원소를 찾아, 그 원소의 주소를 반환한다.
- 가장 가까운 정수가 여러 개이면 배열에서 가장 먼저 나타나는 원소의 주소를 반환한다.
- 반환 값: int형 포인터 즉 주소

2) main 함수에서는

- 사용자로부터 정수 N을 입력 받는다. 아래 과정을 N번 반복한다.
 - 사용자로부터 정수를 입력 받는다. 0이 되면 종료한다. 배열 ar의 크기는 100.
 - maxmin 함수를 호출한다.
 - maxmin 함수에서 전달받은 주소가 가리키는, 배열의 원소를 화면에 출력한다.

- ▶ 배열 선언 이후, 배열 표기 [] 는 사용 금지 (포인터 표기 사용) (50% 감점)
- ▶ 사용자로부터 정수들 입력하여 배열에 저장 시에 포인터 사용 (20% 감점)
- ▶ 반복문으로 배열 훑어볼 시, 주소를 이용하여 반복문 구현 (강의자료 p. 42 참조) (50% 감점)
(예: for(p = ar ; p < ar+5 ; p++)
여기서 p는 정수형 포인터, ar은 배열의 이름)
- ▶ 전역변수 사용금지 (100% 감점)

입력 예시 1

```
3           ↳ N
3 7 6 2 4 0
13 7 6 1 2 0 -1 14
2 0
```

출력 예시 1

```
6       ↳ 7-2=5
13      ↳ 13-1=12
2       ↳ 2-2=0
```

입력 예시 2

```
2
13 7 6 6 2 0 -1 5
10 7 6 7 5 3 0 -1 3
```

출력 예시 2

```
13
7
```

※ [문제 2-1]에서 [문제 2-2]까지는 연관된 문제이며, 배점이 각각 50점, 100점이다. [문제 2-1]을 안 풀고 [문제 2-2]만 풀어도 100점을 부여한다. 가장 높은 점수 하나만 반영하며, 합산하지 않는다. [문제 2-1]을 풀고 [문제 2-2]를 풀면 쉽게 문제를 풀 수 있다.

[문제 2-1, 2-2] 로또 번호와 관련된 문제이다. 로또 번호는 1~45까지의 숫자 6개로 구성되며, 6개의 숫자들에 중복된 숫자는 포함되지 않는다.

[문제 2-1][레벨 1] (50점) 로또 번호를 입력하면 이의 유효성을 확인하는 프로그램을 작성하려고 한다. 1~45까지의 숫자 6개를 입력 받은 후, 입력 결과에 중복되는 숫자가 존재하는 경우, 해당 숫자와 그 숫자의 중복 횟수를 알려주는 프로그램을 작성하시오. 입력되는 숫자는 모두 1~45 사이라고 가정하며, 숫자가 출력되는 순서는 입력된 순서와 동일하다.

- ▶ 배열 선언 이후, 배열 표기 [] 는 사용 금지 (포인터 표기 사용) (50% 감점)
 - ▶ 사용자로부터 정수들 입력하여 배열에 저장 시에 포인터 사용 (20% 감점)
 - ▶ 반복문으로 배열 훑어볼 시, 주소를 이용하여 반복문 구현 (50% 감점)
- (예: `for(p = ar ; p < ar+5 ; p++)` 여기서 p는 정수형 포인터, ar은 배열의 이름)

입력 예시 1

출력 예시 1

1 2 3 3 4 5

3 2 ↳ 3이 2번 입력됨

입력 예시 2

출력 예시 2

20 15 20 15 20 1

20 3 ↳ 20가 3번 입력됨
15 2 ↳ 15가 2번 입력됨
↳ 20이 먼저 입력되었으므로 15보다 먼저 출력됨

입력 예시 3

출력 예시 3

10 5 8 2 38 45

0 ↳ 중복된 숫자가 없는 경우 0을 출력함

[문제 2-2][레벨 2] (100점) 로또 번호를 입력하면 이의 유효성을 확인하고, 유효하지 않은 경우 유효한 번호로 변경해주는 프로그램을 작성하려고 한다.

1) main 함수

- 6개의 숫자를 입력 받는다.
- correct 함수를 호출한다.
- 중복된 숫자가 존재하는지 여부를 출력한다. (중복 없음: 0, 중복 있음 1)
- 보정된 숫자 6개를 출력한다.

2) correct 함수

- 함수 원형: `int correct(int *arr)`
- 인자(arr): 숫자가 저장되어 있는 배열의 이름
- 중복된 숫자가 존재하는지 알아내고, 중복된 숫자가 존재하는 경우, newnum 함수를 호출하여 중복된 숫자를 중복되지 않는 숫자로 대체한다.
단, 중복된 숫자들 중 먼저 나오는 숫자를 대체한다.
예) 1 3 3 3 4 5 가 입력된 경우, 3 이 중복되므로 이들 중 맨 앞의 3 이 첫 번째로 2로 대체 되고, 그 다음으로 두 번째 3 이 6으로 대체 됨. 최종 결과는 1 2 6 3 4 5 가 됨.
즉, 아래와 같은 순서로 변경 됨. 1 3 3 3 4 5 --> 1 2 3 3 4 5 --> 1 2 6 3 4 5
- 반환 값 : 배열에 중복된 숫자가 존재하는 경우 0, 존재하는 경우 1을 반환한다.

3) newnum 함수

- 함수 원형: `int upper(int *arr)`
- 인자(arr): 숫자가 저장되어 있는 배열의 이름
- 1~45 사이의 숫자들 중 배열에 존재하지 않으면서 가장 작은 숫자를 찾아준다.
- 반환 값 : 찾아낸 가장 작은 숫자를 반환한다.

- ▶ 배열 선언 이후, 배열 표기 [] 는 사용 금지 (포인터 표기 사용) (50% 감점)
- ▶ 사용자로부터 정수들 입력하여 배열에 저장 시에 포인터 사용 (20% 감점)
- ▶ 배열에 저장 된 정수들 화면에 출력 시에 포인터 사용 (20% 감점)
- ▶ 반복문으로 배열 훑어볼 시, 주소를 이용하여 반복문 구현 (50% 감점)
(예: `for(p = ar ; p < ar+5 ; p++)` 여기서 p는 정수형 포인터, ar은 배열의 이름)
- ▶ 전역변수 사용금지 (100% 감점)

입력 예시 1

출력 예시 1

1 2 3 3 4 5	1 ↳ 중복되는 숫자(3)가 존재하므로 1을 출력함 1 2 6 3 4 5 ↳ 중복되는 숫자 중 먼저 등장하는 3을 배열에 존재하지 않으면서 1~45 사이에서 가장 작은 숫자인 6으로 대체함
-------------	--

입력 예시 2

출력 예시 2

20 15 20 15 20 1	1 ↳ 중복되는 숫자(15, 20)가 존재하므로 1을 출력함 2 3 4 15 20 1 ↳ 중복되는 숫자들 중 먼저 등장하는 20, 15, 20 을 배열에 존재하지 않으면서 1~45 사이에서 가장 작은 숫자인 2, 3, 4 로 대체함
------------------	--

입력 예시 3

출력 예시 3

10 5 8 2 38 45	0 \mapsto 중복된 숫자가 없는 경우 0을 출력함 10 5 8 2 38 45 \mapsto 중복된 숫자가 없는 경우 입력된 숫자를 그대로 출력
----------------	--

[문제 3][레벨 1.5] (100점) 정수 N 을 입력받는다. 알파벳과 숫자로만 구성된 N 개 문자를 입력 받는다. 입력 받은 N 개의 문자들에 대해서 대문자와 소문자를 출력하고, 숫자 문자의 경우 숫자 문자를 연결하여 만든 정수에 2를 곱하여 출력하시오.

- 모든 함수에서 (즉 input 함수와 main 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.
- 배열 선언을 제외하고 배열 기호 []를 사용하지 않는다.
- 입력되는 N 개 문자 안에 대문자 최소 1개, 소문자 최소 1개, 숫자 문자 최소 1개를 포함한다.
- 여기서 $5 \leq N \leq 20$

1) input 함수

- 함수 원형: **void input(char *p, int N)**
- p 는 배열의 시작 주소이며, 문자 n 개를 사용자로부터 $\%c$ 를 이용하여 입력받아 p 가 가리키는 배열에 저장한다.
- 반환 값 : 없음

2) upper 함수

- 함수 원형: **void upper(char *p, int N)**
- 입력받은 포인터 변수가 가리키고 있는 곳에 저장된 문자 중에서 대문자만 화면으로 출력한다.
- 반환 값 : 없음

3) lower 함수

- 함수 원형: **void lower(char *p, int N)**
- 입력받은 포인터 변수가 가리키고 있는 곳에 저장된 문자 중에서 소문자만 화면으로 출력한다.
- 반환 값 : 없음

4) double 함수

- 함수 원형: **int DOUBLE(char *p, int N)**
- 입력받은 포인터 변수가 가리키고 있는 곳에 저장된 숫자 문자를 차례로 연결하여 만든 정수에 2를 곱하여 반환한다. 이 정수는 main 함수에서 화면으로 출력한다.
- 반환 값 : 정수

- ▶ 함수의 원형(인자, 반환값)을 지키지 않거나 목적에 맞게 함수를 사용하지 않거나 위에서 설명된 기능을 정확하게 구현하지 않은 경우 (input, upper, lower, DOUBLE 각 20% 감점)
- ▶ 배열 선언 이외에 배열 표기 []를 사용하는 경우 (main, input, upper, lower, DOUBLE 각 20% 감점)
- ▶ 반복문으로 배열 훑어볼 시, 주소를 이용하여 반복문 구현을 하지 않은 경우 (main, input, upper, lower, DOUBLE 각 20% 감점)
- ▶ 전역변수를 사용하는 경우 (100% 감점)

입력 예시 1

출력 예시 1

12 FIV12grandE5	↪ N = 12	FIVE grand 250	↪ 125 × 2 = 250
--------------------	----------	----------------------	-----------------

입력 예시 2

출력 예시 2

8 offC5U5T	CUT off 110
---------------	-------------------

[문제 4][레벨 2] (100점) 두 개의 배열에 정수를 입력 받고 이를 적절히 정렬하는 프로그램을 작성한다.

- 두 개의 배열 a와 b에 각각 정수를 n개와 m개 입력 받는다.
- 각 배열에 숫자를 입력 받을 때는 0이 들어오기 전까지만 입력 받으며, n과 m은 최소 1이고, 최대 10이다.
- 입력되는 정수의 개수 n, m은 입력으로 주어지지 않으므로 계산해야 된다.
- 배열 a에는 배열 a와 b에 저장된 정수들 중 가장 큰 n개를 내림차순으로 저장한다.
- 배열 b에는 배열 a와 b에 저장된 정수들 중 가장 작은 m개를 오름차순으로 저장한다.
- * 단, 해당 프로그램이 적용될 시스템의 메모리 제한으로 인해 배열 a와 b 이외의 다른 배열은 절대로 사용할 수 없다고 가정한다.

- ▶ 배열 선언 이후, 배열 표기 [] 는 사용 금지 (포인터 표기 사용) (50% 감점)
 - ▶ 반복문으로 배열 훑어볼 시, 주소를 이용하여 반복문 구현 (50% 감점)
- [예: for(p = ar ; p < ar+5 ; p++) : p는 정수형 포인터, ar은 정수형 배열의 이름]

입력 예시 1

출력 예시 1

10 50 70 0 20 100 0	100 70 50 10 20
------------------------	--------------------

입력 예시 2

```
-10 20 40 -5 10 0
-100 200 -50 -15 -5 0
```

출력 예시 2

```
200 40 20 10 -5
-100 -50 -15 -10 -5
```

입력 예시 3

```
100 20 90 30 20 100 200 40 50 10 0
5 55 85 105 205 5 15 55 25 45 0
```

출력 예시 3

```
205 200 105 100 100 90 85 55 55 50
5 5 10 15 20 20 25 30 40 45
```

제출기한 및 방법

- * OJ시스템(<https://ex-oj.sejong.ac.kr/index.php/auth/login>) 내의 과제1을 이용하여 제출
- * 제출 마감: **9월 29일(일요일)** 밤 11시 59분까지 제출
- * 과제 점수는 위 마감일 전에 제출된 가장 마지막 코드를 기준으로 부여
(마감일 이후에 제출되는 코드는 채점에서 제외됨)
- * OJ시스템의 점수는 참고로만 사용
 - 문제의 조건을 만족시켜서 코딩했는지 조사 후 최종 점수 부여
 - 코드 2줄당 1줄의 비율로 반드시 주석을 적을 것(주석이 없을 경우 감점)
 - 보고서 등 기타 제출물 없음