

7주차 실습

Topic

- 입, 출력에 대한 복습
- Pointer
- Array

scanf가 헛갈릴 땐

```
1  #include <stdio.h>
2
3  int main(void) {
4      int a;
5      long long b;
6      char c[100];
7      // 32bit integer
8      scanf("%d",&a);
9      // 64bit integer
10     scanf("%lld",&b);
11     // string
12     scanf("%s",c);
13     // EOF
14     while(scanf("%d",&a)!=EOF);
15     return 0;
16 }
```

- read *formatted* data from stdin : format과, 변수 목록을 지정해줌
- 공백문자(space, newline, tab)를 자동으로 무시한다
 - ex) scanf ("%d%d", &a, &b) ; 만 해도 두 개 숫자 입력
- format
 - %d : int형 데이터
 - %lld : long long형 데이터
 - %f : float형 데이터
 - %lf : double형 데이터
 - %s : char[]형 데이터
 - and more..

} &를 붙여준다
(printf에는 해당없음!)
- return value : 입력이 끝난 경우 EOF를 리턴
- 참고 : <http://www.cplusplus.com/reference/cstdio/scanf/?kw=scanf>

Example (숫자 값 입력 받기)

Input Value

1 2 3 4 5 1 2 5 6 17 8

문제의 조건에서 **n 값이 주어졌을 때** ($n < 1000$) [n : 입력 받을 숫자의 개수]

```
int num[1000]; /* 원래는 다른 방법을 통해 n개 만큼 설정(동
적할당)을 해 주어야 하나, 아직 배우지 않은 범위이므로 max
값인 1000만큼 할당을 해줍니다. */
for (i = 0; i < n; i++)
{
    scanf("%d", &num[i]);
}
```

이렇게 될 경우 띄어쓰기 공백으로 숫자들을 구분하면서 num[0] 부터 num[10]까지 순서대로 1 2 3 4 5 1 2 5 6 17 8 숫자가 들어가게 됩니다.

Example (숫자 값 입력 받기)

Input Value

11
1 2 3 4 5 1 2 5 6 17 8

입력에서 **n 값이 주어졌을 때** ($n < 1000$) [n : 입력 받을 숫자의 개수]

```
int num[1000]; /* 원래는 다른 방법을 통해 n개 만큼 설정(동
적할당)을 해 주어야 하나, 아직 배우지 않은 범위이므로 max
값인 1000만큼 할당을 해줍니다. */
scanf("%d", &n); //n 값을 우선 읽습니다.
for (i = 0; i < n; i++)
{
    scanf("%d", &num[i]);
}
```

Example (숫자 값 입력 받기)

Input Value

1 2 3 4 5 1 2 5 6 17 8

문제의 조건에서 **n 값이 주어지지 않았을 때** ($n < 1000$) [n : 입력 받을 숫자의 개수]

```
int num[1000]; /* 원래는 다른 방법을 통해 n개 만큼 설정(동
적할당)을 해 주어야 하나, 아직 배우지 않은 범위이므로 max
값인 1000만큼 할당을 해줍니다. */
int i = 0;
while(scanf("%d", &num[i]) != EOF)
{
    i++;
}
```

scanf의 return value는 읽힌 변수의 개수를 return 합니다. 그래서 위의 경우 정상적인 상황이라면 scanf("%d", &num[i])은 1을 return 하게 됩니다. 하지만 더 이상 읽을 value가 없고 **input의 끝에 도달하게 되면 EOF 값을 리턴**하게 되고 while문은 종료됩니다. While 문이 종료되고 난 후 i 값은 입력 받은 숫자의 개수가 됩니다.

Example (문자 입력 받기)

Input Value

abcde efghi

문제의 조건에서 **n 값이 주어지지 않았을 때** ($n < 1000$) [n : 문자열의 길이]

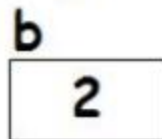
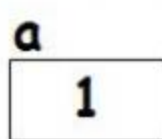
```
char num[1000]; /* 원래는 다른 방법을 통해 n개 만큼 설정  
(동적할당)을 해 주어야 하나, 아직 배우지 않은 범위이므로  
max 값인 1000만큼 할당을 해줍니다. */  
int i = 0;  
while(scanf("%c", &num[i]) != EOF)  
{  
    i++;  
}
```

%c 단위로 입력을 받을 경우, 줄 바꿈도 하나의 문자(\n)로 처리가 되므로 그 부분을 잘 고려하시기 바랍니다.

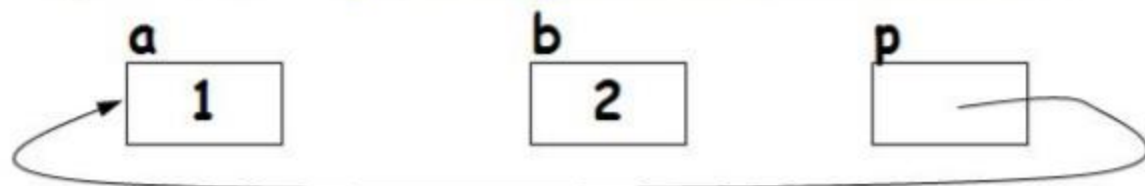
다음 실습 시간에는 문자열을 한번에 입력 받을 수 있는 방법 ("%s") 에 대해서 알아볼 것입니다.

Pointers

```
int a = 1, b = 2, *p;
```



```
p = &a; "p is assigned the address of a"
```



```
b = *p; "b is assigned the value pointed to by p"
```

```
b = *p;  $\Leftrightarrow$  b = a;
```


HW 1

Description

Implement a 'swap()' function.

'swap()' function takes two integer pointer variable as parameters, and swaps values of two variables.

Function Prototype

```
void swap(int *a, int *b);
```

Test Code

```
void swap(int *a, int *b){  
    // you should implement this function  
}  
  
// lines below this must be commented out when submitting your code.  
int main(){  
    int a,b;  
    scanf("%d %d", &a, &b);  
    swap(&a, &b);  
    printf("%d %d\n", a, b);  
}
```

HW 1

Example

Input	Output
1 2	2 1
10 10	10 10

HW 2

Description

You are given a number of student N and N scores.

You should answer Q questions - What is n th ranker's score?

The rank of the student A is defined as "the number of students who scored better than A " + 1.

Input

The 1st line of input contains the number of students N ($1 \leq N \leq 1000$),

next line contains the score of N students ($0 < \text{score} < \text{INT_MAX}$). There is no same score.

The 3rd line contains the number of question Q ($1 \leq Q \leq N$)

Last line contains the rank of Q questions, separated by a space.

Output

For each question, output a single integer score.

HW 2

Example

Input	Output
5 10 8 6 5 2 2 3 4	6 5
7 18 4 9 13 2 10 6 4 3 1 5 6	10 18 6 4

HW 3

Description

Implement `array_cmp()` function.

`array_cmp()` function takes length of array, array A, and array B.

It returns 1 if decimal representation of A is smaller than B. Returns -1 if A is greater than B. Returns 0 if A and B are same. Decimal representation of array A means the number we can get by concatenating every elements of A.

Function Prototype

`int array_cmp(int N, int* A, int* B)`

N is length of two array. It is guaranteed that length of A and B are same.

Constraints

$N \leq 10000$

$0 \leq A[i], B[i] \leq 9$

HW 3

Example

Parameters	Return value
N = 5 A = {1,2,3,4,5} B = {3,3,3,3,3}	1
N = 5 A = {5,4,3,2,1} B = {3,3,3,3,3}	-1

HW 3

Example

Parameters	Return value
N = 5 A = {1,2,3,4,5} B = {1,2,3,4,5}	0
N = 2 A = {3,4} B = {3,3}	-1

HW 4

Description

For given two array A, B, find out how much times that the array B is in the array A and where it is.

Input

The 1st line of input contains the size of array A. $N(1 \leq N \leq 1000)$,
next line contains the elements of A. $(0 < \text{element} < \text{INT_MAX})$.

The 3rd line contains the size of array B. $M(1 \leq M \leq N)$

Last line contains the elements of B. $(0 < \text{element} < \text{INT_MAX})$.

Output

1st line : number of appearance of array B in A

2st line : locations of appearances

HW 4

Example

Parameters	Return value
5 3 1 2 1 2 2 1 2	2 1 3
6 0 1 0 1 0 1 1 0	3 0 2 4

1, 3 번 문제 풀 때 주의할 사항

- 함수의 prototype은 주어져 있고, 내부 구현만 완성하면 되는 문제. (Implement a function)
- 채점 서버에서 미리 만들어 둔 main 함수와 합쳐서 컴파일 후 실행한다.
- 각자 main함수를 작성해 테스트 하는 것은 좋지만, 제출할 때는 서버와 충돌하지 않도록 자신의 main함수를 주석 처리한 뒤 제출한다.

본인 컴퓨터에서 코딩 및 테스트 할 때

```
2 void swap(int *a, int *b){
3     // you should implement this function
4 }
5
6 // lines below this must be commented out when
7 int main(){
8     int a,b;
9     scanf("%d %d", &a, &b);
10
11     swap(&a, &b);
12
13     printf("%d %d\n", a, b);
14 }
```

채점 서버에 제출할 때

```
1
2 void swap(int *a, int *b){
3     // you should implement this function
4 }
5
```

or

```
2 void swap(int *a, int *b){
3     // you should implement this function
4 }
5
6 // lines below this must be commented out when
7 /*
8 int main(){
9     int a,b;
10    scanf("%d %d", &a, &b);
11
12    swap(&a, &b);
13
14    printf("%d %d\n", a, b);
15 }
16 */
```