

## 8. 형식지정 입출력함수 - scanf() / printf()

형식지정은 % 뒤에 여러 가지 '변환문자'를 사용하여 지정한다.

즉, % 뒤에 있는 변환문자에 따라 해당하는 인수의 입출력 기능이 달라진다.

변환기호	기능
%c	인수를 단일문자로 변환시킨다.
%d	인수를 부호 있는 10진수로 변환시킨다.
%u	인수를 부호 없는 10진수로 변환시킨다.
%o	인수를 8진수로 변환시킨다.
%x	인수를 16진수로 변환시킨다.
%s	인수를 포인터형으로 변환시킨다(문자열 입출력).
%f	인수를 실수형으로 변환시킨다.

```
#include <stdio.h>
void main()
{
    int a;
    char b[20];
    scanf("%d", &a);    //정수 입력
    scanf("%s", b);    //문자열 입력
    printf("%5d\n", a); //정수 출력
    printf("%10s\n", b); //문자열 출력
}
```

### ◆ 함수 scanf()

[원형] int scanf(const char \*format, .....);

- 표준입력장치로부터 지정된 형식에 맞게 자료를 읽어 들인다.
- scanf()는 변수의 주소를 인수로 사용한다.
- 해서, 일반변수 앞에는 &를 붙이고, 배열명이나 포인터에는 &를 붙이지 않는다.

예 : int a;

char b[5];

scanf("%d", &a); → 정수형 변수 a에 자료를 입력받는다.(변수 앞에 &를 붙여야 한다)

scanf("%s", b); → 배열 b에 자료를 입력받는다.(변수 앞에 &를 붙이지 않는다)

기출문제 분석

1. 다음은 숫자를 처리하는 C 프로그램이다. 프로그램에서 ㉠과 ㉡에 들어갈 내용과 3 2 1 4를 입력하였을 때의 출력결과를 바르게 짝지은 것은? (단, 다음 프로그램에 문법적 오류는 없다고 가정한다) [2019년 우정 9급]

```

#include <stdio.h>
#include <stdlib.h>
void a (int n, int *num) {
    for (int i = 0; i < n; i++)
        scanf("%d", &(num[i]));
}
void c(int *a, int *b) {
    int t;
    t = *a; *a = *b; *b = t;
}
void b(int n, int *lt) {
    int a, b;
    for (a = 0; a < n-1; a++)
        for (b = a + 1; b < n; b++)
            if(lt[a]>lt[b]) c( ㉠, ㉡ );
}
void main() {
    int n;
    int *num;
    printf("How many numbers?");
    scanf("%d", &n);
    num = (int *)malloc(sizeof(int) * n);
    a(n, num);
    b(n, num);
    for (int i = 0; i < n; i++) printf("%d ", num[i]);
}

```

㉠	㉡	출력 결과
① lt+a	lt+b	1 2 3 4
② lt+a	lt+b	1 2 4
③ lt[a]	lt[b]	4 3 2 1
④ lt[a]	lt[b]	4 2 1

☞ 자판으로 입력한 값을 정렬하여 출력하는 프로그램

- 프로그램을 문법적으로 오류가 발생되지 않도록 완성하면 다음과 같다.

```

#include <stdio.h>
#include <stdlib.h>
void a(int n, int *num) {           //a(3, num)
    int i;
    for(i = 0; i < n; i++)         //자판으로 입력한 나머지 값을 차례로 입력 받음
        scanf("%d", &(num[i]));  //num[3] = {2, 1, 4}처럼 배열이 초기화 된다.
}
void c(int *a, int *b) {           //교환(포인터 a, b가 가리키는 값을 서로 교환)
    int t;
    t = *a; *a = *b; *b = t;
}
void b(int n, int *lt) {           //b(3, num)이므로, lt[3] = {2, 1, 4}와 같다.
    int a, b;
    for(a = 0; a < n-1; a++)      //선택정렬(오름차순)
        for(b = a + 1; b < n; b++)
            if(lt[a]>lt[b]) c(lt+a, lt+b); //호출 : 첫 번째 호출은 c(lt+a, lt+b) = c(2, 1)과 같다.
}
void main() {                     //먼저, 프로그램 개요를 설명하면
    int i, n;                     //자판으로 3 2 1 4를 입력하면
    int *num;                     //첫번째 3은 정렬할 자료수를 의미하고
    printf("How many numbers?"); //나머지 2 1 4는 정렬대상 자료를 의미
    scanf("%d", &n);              //3 2 1 4를 입력하면, n = 3이 된다.
    num = (int *)malloc(sizeof(int) * n); //배열크기는 3, int num[3]:과 같음
    a(n, num);                   //호출 : a(3, num)
    b(n, num);                   //호출 : b(3, num)
    for(i=0; i<n; i++) printf("%d ", num[i]); //정렬 결과 출력, num[3] = {1, 2, 4}
}

```

// 함수 scanf("%d", &n);에서 입력되는 원리

- scanf("%d", &n);에서 자판으로 3 2 1 4를 입력하면, n = 3이 된다.(띄어쓰기를 해야 함)  
↓ 나머지 2 1 4는 다음에 나오는 scanf("%d", &(num[i]));에서 입력처리 된다.
- scanf("%d", &(num[i])); //num[3] = {2, 1, 4}처럼 배열이 초기화 된다.

2. 다음 C 프로그램을 실행하면서 사용자가 1, 2, 3, 4를 차례대로 입력했을 때, 출력 결과는?  
[2022년 지방 9급]

```
-----  
#include <stdio.h>  
int main()  
{  
    int ary[4];  
    int sum = 0;  
    int i;  
    for (i = 0; i < 4; i++)  
    {  
        printf("%d번 째 값을 입력하시오 : ", i + 1);  
        scanf("%d", &ary[i]);  
    }  
    for (i = 3; i > 0; i--)  
        sum += ary[i];  
    printf("%d \n", sum);  
    return 0;  
}  
-----
```

- ① 3                      ② 6  
③ 9                      ④ 10

♣ C 프로그램

// 1, 2, 3, 4를 차례대로 입력하면, 배열은 다음과 같은 값을 가진다.

	ary[0]	ary[1]	ary[2]	ary[3]
ary[4]	?	?	?	?
	↓	↓	↓	↓
	1	2	3	4

```
for (i = 3; i > 0; i--)  
    sum += ary[i];            // sum = 4 + 3 + 2 = 9  
printf("%d \n", sum);        // 출력 : 9
```

정답 : ③

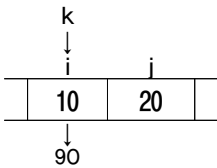
3. 다음의 C 프로그램을 실행한 결과로 옳은 것은? (단, 아래의 scanf() 함수의 입력으로 90을 타이핑했다고 가정) [2015년 서울 9급]

```

int main( )
{
    int i = 10;
    int j = 20;
    int *k = &i;
    scanf("%d", k);
    printf("%d, %d, %d\n", i, j, *k);
    return 0;
}
    
```

- ① 10, 20, 10            ② 10, 20, 90
- ③ 90, 20, 10           ④ 90, 20, 90

♣ 메모리 구조



• 포인터 k가 i를 가리킨다.

```

scanf("%d", k);           //k가 가리키는 곳에 90을 입력한다.
printf("%d, %d, %d\n", i, j, *k); //90, 20, 90이 출력된다.
    
```

정답 : ④