

### 3. 포인터

#### 1) 포인터를 이용한 매개변수 전달

두 정수를 입력받고 두 수의 값을 서로 바꾼 뒤 각각 출력하시오. 단, 포인터를 사용하여 s 두정수의 값을 서로 바꾸는 wap() 함수를 작성하시오.

[제약 사항]

반드시 포인터를 사용하여 swap() 함수를 작성

[입력]

정수형 2 개

[출력]

서로 값이 바뀐 정수 2 개

해답/결과:

```
#include <stdio.h>
void swap(int* a, int* b) {
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}
int main(void) {
    int a,b;
    scanf("%d %d", &a, &b);

    swap(&a, &b);
    printf("%d %d", a,b);
}
```

3

5

5 3

## 2) 포인터 연산

먼저 크기가 5 인 정수형 1 차원 배열을 선언한 뒤 [1,2,3,4,5]로 값을 초기화한다. 그 후 해당 배열의 첫 번째 원소부터 마지막 원소까지 배열 포인터의 덧셈 연산을 사용하여 모든 원소를 한 칸씩 띄워 출력하는 코드를 작성하시오. 단, 배열의 인덱스 접근법인 []를 사용해선 안됩니다.

### [제약 사항]

배열의 인덱스 접근법인 []를 사용하지 않고 배열 포인터만을 사용해야 한다.

### [입력]

없음

### [출력]

배열의 모든 원소

해답/결과:

```
#include <stdio.h>
int main(void) {
    int arr[5] = {1,2,3,4,5};

    for(int i=0; i<5; i++) {
        printf("%d ", *(arr+i));
    }
}
```

1 2 3 4 5

### 3) 포인터 반환

총 5 개의 정수를 입력받아 정수형 1 차원 배열에 저장하고 이 중 가장 작은 원소가 있는 곳의 주소값을 리턴하는 `find()` 함수를 작성하고 `main()` 함수에서 포인터 변수를 활용해 해당 주소값을 넘겨받아 출력하시오. 단, 배열의 인덱스를 넘겨선 안되고 반드시 주소값을 넘겨야한다.

[제약 사항]

`find()` 함수에서 배열의 인덱스를 넘겨선 안되고 반드시 주소값을 리턴

[입력]

정수형 5 개

[출력]

배열의 원소 중 최솟값

해답/결과:

```
#include <stdio.h>
int* find(int arr[]) {
    int min = 0;
    for(int i=1; i<5; i++) {
        if(arr[min] > arr[i])
            min = i;
    }
    return &arr[min];
}
```

```
int main(void) {
    int arr[5];
    int num=0;
    int *p;
    for(int i=0; i<5; i++) {
        scanf("%d", &num);

        arr[i] = num;
    }
    p = find(arr);
    printf("%d", *p);
}
```

4 2 3 5 1