

[8장 실습1] 함수의 인자전달 방식

Q1. 2차원 평면 상의 좌표를 다루는 함수들을 정의하고, 다음의 프로그램을 완성하십시오.

▶ **display_point()** 함수

- x축과 y축 좌표값을 전달받아 좌표 형태로 출력하는 함수(실행 예 참고)

▶ **compute_distance()** 함수

- 두 점의 좌표를 전달받아 두 점 사이의 거리를 계산해서 반환해주는 함수(출력문 없음)
- 두 점사이의 거리 $= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

▶ **move_point()** 함수

- x축과 y축 좌표값, 그리고 이동량을 전달받아 x축과 y축 좌표값을 각각 이동량만큼 변경하는 함수

```
#include <stdio.h>
```

```
#include <math.h>
```

// 각 함수의 선언

```
int main()
```

```
{
```

```
    int x1, x2, y1, y2, move;
    double distance;
```

```
    printf("p1 좌표 입력(x축, y축 좌표값 입력): ");
    scanf("%d %d", &x1, &y1);
```

```
    printf("p2 좌표 입력(x축, y축 좌표값 입력): ");
    scanf("%d %d", &x2, &y2);
```

```
    printf("p1 = ");
    display_point(x1, y1);
```

```
    printf("p2 = ");
    display_point(x2, y2);
```

// compute_distance()를 호출하여 두 점 사이의 거리를 계산한 후 출력(실행 예 참고)

```
    printf("좌표 이동량 입력: ");
    scanf("%d", &move);
```

//move_point()를 호출하여 move만큼 p2에 해당하는 좌표값 변경

```
    printf("p2을 x축과 y축으로 각각 %d 만큼 이동 = ", move);
    display_point(x2, y2);
```

```
    return 0;
```

```
}
```

// 각 함수의 정의

```
C:\WINDOWS\system32\cmd.exe
p1 좌표 입력(x축, y축 좌표값 입력): 1 1
p2 좌표 입력(x축, y축 좌표값 입력): 3 7
p1 = (1, 1)
p2 = (3, 7)
두점 사이의 거리 = 6.32456

좌표 이동량 입력: -1
p2을 x축과 y축으로 각각 -1 만큼 이동 = (2, 6)
계속하려면 아무 키나 누르십시오 . . .
```

```
C:\WINDOWS\system32\cmd.exe
p1 좌표 입력(x축, y축 좌표값 입력): -2 4
p2 좌표 입력(x축, y축 좌표값 입력): 3 -5
p1 = (-2, 4)
p2 = (3, -5)
두점 사이의 거리 = 10.2956

좌표 이동량 입력: 3
p2을 x축과 y축으로 각각 3 만큼 이동 = (6, -2)
계속하려면 아무 키나 누르십시오 . . .
```

[8장 실습2]: 정수형 배열을 다루는 함수

※ 다음의 요구사항에 맞게 배열을 다루는 각 함수를 정의하여 프로그램을 완성하시오((단, 제시된 함수의 원형 변경 불가).

-
- ▶ `int getArray(int p[]);` //포인터의 산술 연산 사용
 - 사용자에게 몇 개의 정수를 입력할지 입력받고, 입력받은 개수만큼 정수를 입력받아 배열에 저장한 후, 실제 입력된 정수 데이터의 개수를 반환하는 함수
 - ▶ `void printArray(const int p[], int size);` //포인터의 색인화 사용
 - 배열의 주소와 실제 크기를 전달받아 화면에 출력하는 함수
 - ▶ `double computeAverage(const int p[], int size);` //포인터의 증가연산 사용
 - 배열의 주소와 실제 크기를 전달받아 평균을 계산하여 반환하는 함수
 - ▶ `void copySquareArray(int p1[], const int p2[], int size);`
 - 두 배열의 주소와 실제 크기를 전달받아 첫 번째 배열의 각 원소값을 제곱하여 두 번째 배열의 대응되는 원소에 저장하는 함수
 - ▶ `void swapArray(int p1[], int p2[], int size);` //포인터의 산술 연산 사용
 - 배열의 주소와 실제 크기를 전달받아, 두 배열의 내용을 서로 교환하는 함수
 - 두 배열의 대응되는 원소의 값을 교환할 때, 두 정수의 내용을 교환하는 `swap()` 함수를 사용한다.
-

```
#include <stdio.h>
#define MAX 50
```

```
int getArray(int p[]);           //포인터의 산술 연산 사용
void printArray(const int p[], int size); //포인터의 색인화 사용
double computeAverage(const int p[], int size); //포인터의 증가연산 사용
void copySquareArray(int p1[], const int p2[], int size);
void swapArray(int p1[], int p2[], int size); //포인터의 산술연산 사용
int main()
{
    int arr1[MAX], arr2[MAX];
    double average;
    int no; // 배열의 실제 크기

    no = getArray(arr1);

    printf("\narr1 = ");
    printArray(arr1, no);

    average = computeAverage(arr1, no);
    printf("arr1의 평균: %g \n", average);

    copySquareArray(arr2, arr1, no);
    printf("\n==== squireArray() 함수 호출후 =====\n");
    printf("arr2 = ");
    printArray(arr2, no);

    swapArray(arr1, arr2, no);
    printf("\n==== swapArray() 함수 호출후 =====\n");
    printf("arr1 = ");
    printArray(arr1, no);
    printf("arr2 = ");
    printArray(arr2, no);

    return 0;
}
```

```
C:\WINDOWS\system32\cmd.exe
몇개의 정수를 입력하시겠습니까? 5
5개의 정수 입력: 8 10 5 11 7

arr1 = {8, 10, 5, 11, 7}
arr1의 평균: 8.2

==== squireArray() 함수 호출후 =====
arr2 = {64, 100, 25, 121, 49}

==== swapArray() 함수 호출후 =====
arr1 = {64, 100, 25, 121, 49}
arr2 = {8, 10, 5, 11, 7}
계속하려면 아무 키나 누르십시오 . . .
```