

과제 #1

```
1 #include <stdio.h>
2
3 int Max(int* pArr, int size); // Max함수 선언
4 int Min(int* pArr, int size); // Min함수 선언
5 void Sorting(int* pArr, int size); // Sorting함수 선언
6
7 int main()
8 {
9     int ab[] = { 20, 34, 12, 24, 54, 91, 9, 40, 81, 10 }; // ab[]를 주어진 배열로 설정
10    int* pArr = ab;
11    int y;
12    printf("배열의 최댓값: %d\n", Max(ab, 10));
13    printf("배열의 최솟값: %d\n", Min(ab, 10));
14    printf("정렬 전 배열: ");
15    for (y = 0; y < 10; y++) {
16        printf("%d ", ab[y]);
17    }
18    printf("\n");
19    printf("내림차순 정렬 후 배열: ");
20    Sorting(ab, 10);
21 }
22
23
24 int Max(int* pArr, int size)
25 {
26     int i; // for문에 사용하기 위한 정수형 변수 i 선언
27     int MAX = pArr[0]; // 배열의 맨 처음 수를 MAX에 대입
28     for(i = 0; i < size; i++) {
29         if (pArr[i] > MAX)
30             MAX = pArr[i]; // i를 size까지의 범위로 함으로써 배열 내 모든 원소를 비교할 수 있게 만들고, 하나씩 비교 후 제일 큰 숫자를 MAX에 대입
31     }
32     return MAX; // MAX값을 리턴한다
33 }
34
35 int Min(int* pArr, int size)
36 {
37     int i; // for문에 사용하기 위한 정수형 변수 i를 선언
38     int MIN = pArr[0]; // 배열의 맨 처음 수를 MIN에 대입
```

```
35 int Min(int* pArr, int size)
36 {
37     int i; // for문에 사용하기 위한 정수형 변수 i를 선언
38     int MIN = pArr[0]; // 배열의 맨 처음 수를 MIN에 대입
39     for (i = 1; i < size; i++) { // size 까지를 범위로 모든 원소를 비교하는 반복을 시행 함.
40         if (pArr[i] < MIN) // 만약 배열[i]의 값이 MIN보다 작다면?
41             MIN = pArr[i]; // 그 값을 MIN에 대입한다
42     }
43     return MIN;
44 }
45
46 void Sorting(int *pArr, int size)
47 {
48     int i, j; // 정수형 변수 i, j 선언
49     int temp; // 값 임시 저장용 정수형 변수 temp 선언
50
51     for (i = 0; i < size - 1; i++) { // 아래 설명
52         for (j = 0; j < size - i - 1; j++) { // 아래의 4개
53             if (pArr[j] < pArr[j + 1]) { // ab[x] < ab[x+1] 즉 배열의 다음 원소가 더 크다면 아래를 실행한다
54                 temp = pArr[j]; // ab[x] 값을 temp에 임시저장한다
55                 pArr[j] = pArr[j + 1]; // ab[x+1] 값을 ab[x]에 저장한다 (내림차순이라 크게 앞으로 가야 해서)
56                 pArr[j + 1] = temp; // ab[x+1] 값을 temp에 넣는다. 이렇게 하면 1차적으로 큰 값이 앞으로 간다. 하지만 전부 정렬되지는 않기에 i를 사용한 for문을 한번 더 만든다
57             }
58         }
59     }
60
61     for (i = 0; i < size; i++) {
62         printf("%d ", pArr[i]); // 배열 출력
63     }
64 }
```

Microsoft Visual Studio 디버그 콘솔

```
배열의 최댓값: 91
배열의 최솟값: 9
정렬 전 배열: 20 34 12 24 54 91 9 40 81 10
내림차순 정렬 후 배열: 91 81 54 40 34 24 20 12 10 9
```

코드

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

```
int Max(int* pArr, int size); // Max함수 선언
```

```
int Min(int* pArr, int size); // Min함수 선언
```

```
void Sorting(int* pArr, int size); // Sorting함수 선언
```

```
int main()
```

```
{  
  
    int ab[] = { 20, 34, 12, 24, 54, 91, 9, 40, 81, 10 }; // ab[]를 주어진 배열로 설정  
  
    int* pArr = ab;  
  
    int y;  
  
    printf("배열의 최대값: %d\n", Max(ab, 10));  
  
    printf("배열의 최소값: %d\n", Min(ab, 10));  
  
    printf("정렬 전 배열: ");  
  
    for (y = 0; y < 10; y++) {  
        printf("%d ", ab[y]);  
    }  
  
    printf("\n");  
  
    printf("내림차순 정렬 후 배열: ");  
  
    Sorting(ab, 10);  
  
}
```

```
int Max(int* pArr, int size)
```

```
{
```

```

int i; // for문에 사용하기 위한 정수형 변수 i 선언

int MAX = pArr[0]; // 배열의 맨 처음 수를 MAX에 대입

for(i = 0; i < size; i++) {

    if (pArr[i] > MAX)

        MAX = pArr[i]; // i를 size까지의 범위로 함으로써 배열 내 모든 원소를 비교할 수 있게 만들고, 하나
        씩 비교 후 제일 큰 숫자를 MAX에 대입

}

return MAX; // MAX값을 리턴한다
}

```

```

int Min(int* pArr, int size)

{

    int i; // for문에 사용하기 위한 정수형 변수 i를 선언

    int MIN = pArr[0]; // 배열의 맨 처음 수를 MIN에 대입

    for (i = 1; i < size; i++) { // size 까지를 범위로 모든 원소를 비교하는 반복을 시행 함.

        if (pArr[i] < MIN) // 만약 배열[i]의 값이 MIN보다 작다면?

            MIN = pArr[i]; // 그 값을 MIN에 대입한다

    }

    return MIN;

}

```

```

void Sorting(int *pArr, int size)

{

    int i, j; // 정수형 변수 i, j 선언

    int temp; // 값 임시 저장용 정수형 변수 temp 선언

    for (i = 0; i < size - 1; i++) { // 아래 설명

        for (j = 0; j < size - i - 1; j++) { // 아래의 4개

            if (pArr[j] < pArr[j + 1]) { // ab[x] < ab[x+1] 즉 배열의 다음 원소가 더 크다면 아래를 실행한다

```

temp = pArr[j]; // ab[x] 값을 temp에 임시저장한다

pArr[j] = pArr[j + 1]; // ab[x+1] 값을 ab[x]에 저장한다 (내림차순이라 크게 앞으로 가야 해서)

pArr[j + 1] = temp; // ab[x+1]값을 temp에 넣는다. 이렇게 하면 1차적으로 큰 값이 앞으로 간다.

하지만 전부 정렬되지는 않기에 i를 사용한 for문을 한번 더 만든다

}

}

}

for (i = 0; i < size; i++) {

printf("%d ", pArr[i]); //배열 출력

}

}
