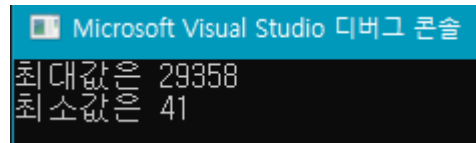


## <실습 과제 #7>

1. 크기가 10인 1차원 배열에 난수를 저장한 후에, 최대값과 최소값을 출력하는 프로그램을 작성하라.

※ rand( )함수 사용, srand(time(NULL)) 이용



```
Microsoft Visual Studio 디버그 콘솔
최대값은 29358
최소값은 41
```

2. 아래 실행결과를 참고 하여, 2개의 정수 배열 a, b를 전달받아 서로 같은 배열인지를 검사하는 함수 array\_equal(int a[ ], int b[ ], int size)를 작성하고 검사하라. 이 함수는 a[0]과 b[0], a[1]과 b[1], ..., a[size-1]과 b[size-1]이 각각 같은지를 검사한다. 만약 전체 요소가 같다면 1을 반환하고 그렇지 않으면 0을 반환한다.

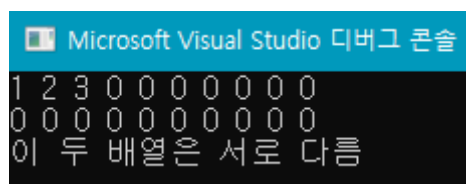
※ 배열의 크기는 N\_DATA라는 기호상수를 정의하고 10으로 지정한다.

※ 배열의 구성요소를 출력해주는 void array\_print(int a[ ], int size)를 작성하고, 배열의 구성 요소를 출력할 때는 항상 array\_print함수를 사용한다.

※ main 함수 내에서 A 배열은 {1,2,3} 으로 초기화 한다.

※ main 함수 내에서 B 배열은 모두 0으로 초기화 한다.

※ main 함수 내에서 array\_equal 함수의 반환값을 이용하여 판단 후 마지막 문장이 출력되도록 한다.

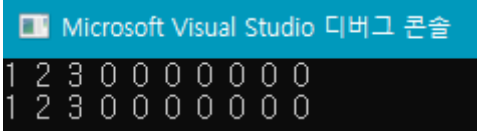


```
Microsoft Visual Studio 디버그 콘솔
1 2 3 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
이 두 배열은 서로 다름
```

3. 앞선 2번에서 작성한 코드를 바탕으로, 이번에는 array\_equal 함수 대신, 정수 배열 A를 다른 정수 배열 B에 복사하는 함수 void array\_copy(int \*a, int\*b, int size)를 작성하고 테스트하라.

```
int main(void)
{
    int A[N_DATA] = { 1, 2, 3 };
    int B[N_DATA] = { 0 };

    array_print(A, N_DATA);
    array_copy(A, B, N_DATA);
    array_print(B, N_DATA);
    return 0;
}
```



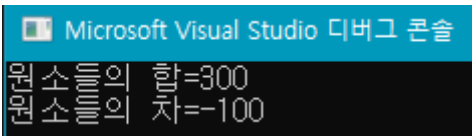
Microsoft Visual Studio 디버그 콘솔

```
1 2 3 0 0 0 0 0 0
1 2 3 0 0 0 0 0 0
```

4. 2개의 정수의 합과 차를 동시에 반환하는 함수를 작성하고 테스트 하라. 포인터 매개 변수를 사용한다.

※ void get\_sum\_diff(int x, int y, int \*p\_sum, int \*p\_diff) 를 작성하시오.

※ 계산에 쓰일 값은 각각 200, 100을 사용한다.



Microsoft Visual Studio 디버그 콘솔

```
원소들의 합=300
원소들의 차=-100
```

5. 2개의 정렬된 정수 배열 A[]와 B[]가 있다고 가정하자. 이 2개의 배열을 합쳐서 하나의 정렬된 배열 C[]로 만드는 함수 merge를 작성하고 아래 실행결과를 참고하여 테스트 하시오.

※ void merge(int \*A, int \*B, int \*C, int size) 를 작성하시오.

※ 배열 A와 B는 똑같은 크기로 정의되어 있다고 가정한다.

※ 배열 C에는 충분한 공간이 확보되어 있다고 가정한다.

※ 두 배열을 합치는 알고리즘은 다음과 같다.

1. 먼저 A[0]와 B[0]를 비교한다. 만약 A[0]가 B[0]보다 작으면 A[0]를 C[0]에 복사한다.

2. 다음, A[1]과 B[0]를 비교한다. 만약 B[0]가 A[1]보다 작으면 이번에는 B[0]를 C[1]에 복사한다.

3. 같은 방식으로 남아있는 요소들을 모두 비교하여 C 배열에 복사한다.

4. 만약 A배열 또는 B배열 중 어느 하나가 모든 요소를 복사하게 되어 남아있는 요소가 없을 경우, 더 이상 비교 알고리즘 없이 나머지 배열에 남아있는 요소들을 모두 C 배열에 복사한다.

```
int main(void)
{
    int i;

    int A[] = { 2, 5, 7, 8 };
    int B[] = { 1, 3, 4, 6 };
    int C[8];

    array_print("A", A, N_DATA);
    array_print("B", B, N_DATA);

    merge(A, B, C, 4);

    array_print("C", C, 2 * N_DATA);

    printf("\n");
    return 0;
}
```

```
void merge (int *A, int *B, int *C, int size)
```

```
{
```

```
    배열 인덱스 참조를 위한 카운터 변수 선언
```

```
    for( a 배열, b 배열 둘 다 요소가 남아있는 동안 )
```

```
    {
```

```
        각 요소의 대소 비교 후 C 배열에 저장
```

```
    }
```

```
    // 두 배열 중 한 배열이라도 먼저 비게 된다면,
```

```
    for(배열 요소 끝까지)
```

```
    {
```

```
        남아있는 요소를 C 배열에 저장
```

```
    }
```

```
}
```

```
Microsoft Visual Studio 디버그 콘솔
A[] = 2 5 7 8
B[] = 1 3 4 6
C[] = 1 2 3 4 5 6 7 8
```