COMP500 / ENSE501: Week 11 – Exercise:

EXERCISE NAME: *Bubble Sort*

The following program source code is incomplete:

```c
#include <stdio.h>

typedef int(*compare_func)(int, int);

void swap(int* p1, int* p2);
int ascending(const int a, const int b);
int descending(const int a, const int b);
void print_array(int* the_array, const int size);
void bubble(int* the_array, const int size, compare_func func);

int main(void)
{
    int data[] = { 2, 16, 4, 81, 1, 43, 12, 23, 50 };

    printf("Unsorted:         ");
    print_array(data, 9);

    // TODO: 2) CALL bubble WITH data SORT ascending:

    printf("Ascending Order:  ");
    print_array(data, 9);

    // TODO: 3) CALL bubble WITH data SORT descending:

    printf("Descending Order: ");
    print_array(data, 9);

    return 0;
}

void bubble(int* the_array, const int size, compare_func func)
{
    for (int pass = 1; pass <= size; ++pass)
    {
        for (int count = 0; count <= size - 2; ++count)
        {
            // TODO: 1) Insert code here...
        }
    }
}
```

Define the **swap** function. This function must swap the values pointed to by the **p1** and **p2** reference parameters. This function must have no side effects.

Define the **ascending** function. This function must return is **b** less than **a**. This function must have no side effects.

Define the **`descending`** function. This function must return is **b** greater than **a**. This function must have no side effects.

Define the **`print_array`** function such that it outputs **`the_array`** input parameter in the following style:

```
{ 2, 16, 4, 81, 1, 43, 12, 23, 50 }
```

At **`// TODO: 1)`** implement the following pseudo code inside the nested loop:

```
SET result AS func(the_array[count], the_array[count + 1])

IF result
    CALL swap WITH &the_array[count], &the_array[count + 1])
ENDIF
```

At **`// TODO: 2)`** call the **`bubble`** function passing in the **`data`** array, the **`data`** array's dimension and address of the **`ascending`** function.

At **`// TODO: 3)`** call the **`bubble`** function passing in the **`data`** array, the **`data`** array's dimension and address of the **`descending`** function.

Once complete, your program should output the following:

```
Unsorted:          { 2, 16, 4, 81, 1, 43, 12, 23, 50 }
Ascending Order:   { 1, 2, 4, 12, 16, 23, 43, 50, 81 }
Descending Order: { 81, 50, 43, 23, 16, 12, 4, 2, 1 }
```

Ensure the program output is exactly as described, and that the whitespace of your source code is well formatted.