EGRE 246 Advanced Engineering Programming
Homework #1 – Pointers

This homework must be your own (individual) work as defined in the course syllabus and discussed in class.

1) In this assignment, you will write a C program that reads in a list of integers from an input file, sorts it from smallest to largest, and writes out to an output file.

   You are provided with a *hw1.c* file that includes a `main()` function. The *hw1.c* file also includes the function prototypes for 3 functions that you must write. The function prototypes for those functions are shown below:

```
int read_array(FILE *readfile, int *array, int max_size);
void sort_array(int *array, int size);
void write_array(FILE *writefile, int *array, int size);
```

   You must write the code for these three functions in a file named *hw1_functions.c*. The functions must be written such that they work with the unchanged `main()` function and prototypes provided in the *hw1.c* file. ***In addition, your functions must be written to use only pointer arithmetic to handle the array operations – there cannot be ANY square brackets (i.e., '[' or ']') ANYWHERE in your hw1_functions.c file.***

   The `read_array()` function must read the number from the file and place them in the array in the order that they are read. The third argument, `max_size` specifies the maximum number of elements in the array. If the input file contains more than `max_size` integers, your `read_array()` function should print out an error message to the user and then either not put any more elements in the array and return to the `main()` function, or exit the program – your choice. However, it should not crash if there are more than `max_size` integers in the input file. The `read_array()` function must return the actual number of elements placed into the array.

   The `sort_array()` function should sort the elements in the array in order from smallest to largest. The `size` argument specifies the actual number of elements in the array.

   The `write_array()` function should write the elements from the array into the output file in the order in which they are stored in the array. Again, the `size` argument specifies the number of elements in the array.

You are also provided with a sample input file, *hw1_input.txt*. The contents of that file look like this:

```
100
23
-56
984
0
-3
409
-384
91
-63
```

When you run your completed program using the *hw1_input.txt* file, it must generate an output file called *hw1_output.txt* that looks like this:

```
-384
-63
-56
-3
0
23
91
100
409
984
```

For this assignment, you may turn in only one file – your *hw1_functions.c* file. Your functions will be tested with an unchanged version of the *hw1.c* and *hw1_input.txt* files. If your functions do not compile or run correctly with the unchanged versions, your submission will be graded appropriately. If your functions contain any square brackets (‘[‘ or ‘]’), your submission will be graded with a maximum grade of 50%.

Remember the class policy on late submissions – no late submissions are allowed unless prior arrangement is made with the instructor.