**September 10, 2015 – Programming Exercises – Arrays**

1. **arrayEqual\_lastname.c (20pts.)** Have two arrays A and B, each of **10 integers**. Write a **function** that tests if every element of array A is equal to its corresponding element in array B. In other words, the function must check if A[0] is equal to B[0], A[1] is equal to B[1], and so forth. The function is **to return 1 if all elements are equal and 0 if at least one element is not equal**.

**int isEqual(int A[], int B[], int size);**

**Note:** **Initialize the arrays in main()** then call the function. **Always have make a constant value (#define) for the size of the array**, so that you can change its value during testing.

1. **arrayReverse\_lastname.c (20pts.)** Write a function that reverses the elements of an array so that the last element becomes the first, the second from the last becomes the second, and so forth. The function is to reverse the element in place –that is, **without using another array.** (It is permissible to use a variable to hold an element temporarily.) Then test your program twice, once with an even number of elements in the array and another with an odd number of elements in the array.

**void reverseArray(int A[], int size);**

**Note:** **Initialize the arrays in main()** then call the function. **Always have make a constant value (#define) for the size of the array**, so that you can change its value during testing. **You may use the function, displayAll(), from the previous exercise (array\_lastname.c) to display the elements of the array.**

**ASSIGNMENT: SHORT BONDPAPER, HANDWRITTEN, DUE September 15, 2015**

1. Would you know how a gymnastics competition is scored? Well, for each performance, each judge give a score in the range 1 to 10. Then, the highest and lowest scores are canceled, and the rest are averaged. The average is the score of the performance.

For example, suppose there were 7 judges, and the performance gets the following scores:

**9.20 9.00 8.75 9.40 9.50 8.90 9.60**

The average would be 9.2 (that’s (9.2+9.0+9.4+9.5+8.9)/5)

Write a program that reads 7 scores as input and outputs the average. The program should use an array of floats. Note that you need to determine the highest and the lowest score and exclude them in the computation.

**Break apart the program into the following functions:**

1. **inputScores()** – accepts an array and its size and will get input from the user the scores.
2. **displayHigh()** - accepts an array and its size and will display and returns the Highest score
3. **displayLow()** - accepts an array and its size and will display and returns the Lowest score
4. **displayAverage()** - accepts an array and its size and will display the average score but should exclude the highest and lowest score. This function will call displayHigh() and displayLow()
5. **main()** – will call inputScores() and displayAverage()

Include the function prototypes in your program.

Example: **void inputScores(float scores[], int size);**

Below is a sample output of this exercise.

**Enter 7 scores: 9.20 9.00 8.75 9.40 9.50 8.90 9.60**

**High score: 9.60**

**Low score: 8.75**

**Average score: 9.2**