CSCI 201 – Computer Science 1

Lab assignment 7: Verifying sortedness of a sequence
To be completed by Tuesday March 12.

Objectives: *Design and implement a function for validating sortedness when reading data from an input stream into an array. Learn the concept of designing loops with multiple exit conditions.*

IMPORTANT. *For the tests, we have the following possibilities: file ends first, array fills first, data in file is not sorted. All these cases should be tested.*

Sometimes there is a need to check if the data in a file is sorted. In this lab we will write a function `readSortedArray`. When we call this function, an input filestream and an array of float values are specified. The function uses a loop that behaves in the following way:

- As the items are read, the function compares adjacent items to verify that they are in non-decreasing order.
- As long as items are in order, they are placed in successive locations in the array.
- If an adjacent pair of items is not in order, the function abandons the process and returns -1.
- If the end of the file or the end of the array is reached, the function returns the number of items that were read.

Question 1. Consider the function `readSortedArray`. List the parameters that will specify the array and the input filestream. Write the header for the function `readSortedArray`.

Question 2. Design a loop in psuedocode to read the data into the array. Note that the first item will be read outside the loop.

Question 3. Implement your function in C++, using your design from Question 2.

Question 4. Write a driver to test your function. The main program will declare two arrays: one of 20 float values, and another of 30 values, and open the input file and an output file. It then calls the function to store the items from the file into the first array. Then it displays the result: if the file was not sorted, a message is printed; otherwise it calls the `display` function from the previous lab, to print the values that were stored in the array. Next, the program closes the input file, clears the input stream, opens the file again and calls the function to store the items from the file into the second array and again displays the result. Test it on multiple files, including the empty file, a file with less than 20 items, a file with exactly 20 items, a file with more than 30 items and an unsorted file.

Upload answers to Questions 1 and 2 and the script for Question 4.