In this assignment, you will be implementing two sorting functions and comparing them against the g++ implementation of std::sort.  You will implement InsertionSort and Radix sort.  I have provided a Driver program.

You will submit the following

1. [timer.cppPreview the document](https://canvas.uw.edu/courses/1269504/files/53774976/download?verifier=UvbvI5FnPJ4RF7Oao8eoGc5UKSFwKVM1hKGtfECl&wrap=1)  (provided by me, you must modify this to run your code)
2. [test.sh](https://canvas.uw.edu/courses/1269504/files/53775549/download?verifier=PghZkjBrH9JzgHfRJqc70cDClOZfNtrvsALWPzSg&wrap=1) (provided by me)
3. InsertionSort.h and InsertionSort.cpp -- Your InsertionSort implementation
4. RadixSort.h and RadixSort.cpp -- Your RadixSort implementation
5. Commentary.txt -- A short (1 paragraph) discussion comparing the performance of the 3 sorting algorithms and anything that surprised you about the experiment.  Be sure to include information about what algorithm std::sort uses.
6. Comparison.png -- a graph of the processing time as a function of the size of the vector with the x-axis ranging logarithmically from 100 to 10,000,000 items

To receive any credit, your code must compile and run with the test script above.  You must not have any segmentation faults, the implementation must return (no infinite loops), and the sorting algorithms must sort correctly.  Assignments with debug statements left in them will lose points at the discretion of the grader depending on how much difficulty they add to grading (up to 100 points, if the grader can't determine what is happening in your program, you won't get credit).

Insertion Sort -- 25 points

Radix Sort -- 25 points

Correct Discussion and Comparison output -- 25 points

Correct Driver implementation -- 10 points

Comments and coding style -- 15 points