ASSIGNMENT 1

Implement INSERTION-SORT and MERGE-SORT algorithms from Chapter 2 and measure the performance against the worst-case bounds in the same Chapter.

1. Consider an array of *n* elements:
(a) Initialize the array using a random number generator between 1 and ten times the size of the array. For example, when $n = 10$, each element will have a value between 1 and 100 generated randomly.
(b) Sort the array using the two algorithms and print three values chosen at random positions from approximately the first third, second third and the third third of the sorted array. Your output should say 'Three sorted values chosen randomly" with the values on the next two lines, one for INSERTION-SORT and one for MERGE-SORT. Run this for *n* from 10 to 200 in steps of 10 You will have twenty sets of these values.
(c) Measure the time in microseconds for each run of both algorithms of the *n* element array. You will get twenty runtime values for $n = 10$ through 200. Have these values output in a data file named <YourLastNameFirstInitial_1>.dat.
(d) For every run of each algorithm for size *n* array, find the time per item by dividing the run time by *n*, then averaging it for all runs of the algorithms. Thus, if the algorithm runs 20 times, the *Average_Item-runtime* is the average of all per item run times. This is the time that we consider as $\Theta(1)$.
(e) In part (c), add a column for the theoretical bound for $\Theta(.)$. In the book formulas for bounds, *n* is replaced by *n* x *Average_Item-runtime*.
(f) Now you have the actual run time versus *n* as well as the theoretical bounds.

2. As a result of the above steps, there are five columns in your data files; n, Runtime_INSERTION-SORT Bound_INSERTION-SORT, Runtime_MERGE-SORT, Bound_MERGER SORT.

3. Use gnuplot to plot the data in your data file using lines of difference colors and proper labels. The plot title should be INSERTION-SORT versus MERGE-SORT. There should be a key for all the four graph titles.

4. In a file named <YourLastNameFirstInitial_1.PDF> have the program output (the three randomly selected sorted array values for all runs of both algorithms for *n* ranging from 10 to 200 in steps of 10, and the plot showing all the four graphs.

5. Compress the three files <YourLastNameFirstInitial_1.cpp>, <YourLastNameFirstInitial_1.dat> and <YourLastNameFirstInitial_1.PDF> into one

file <YourLastNameFirstInitial_1.zip> or something such and submit this compressed file as your assignment.