CSC 228-60 Data Structures and Algorithms, Spring 2018
Instructor: Dr. Natarajan Meghanathan

<u>Quiz 3 (Take Home)</u>: Linked List vs. Dynamic Array Implementation of the List ADT: Impact of Insertion Index, Data Size and List Size: *Memory and Run Time Analysis*

Submission: Submit a hardcopy/printout of the report in class): Due: Feb. 16th @ 12 PM

In this quiz, you will analyze the impact of the data size (more precisely, the size of an object) and the list size (number of objects stored in the list) on the performance of the Linked List and Dynamic Array implementations of the List ADT.

You are given the code comprising of the following classes:
- HOA (stands for Home Owners Association) that contains an array of house addresses (each of type string; a string object is of size 24 bytes in C++). The array size is currently 1.
- Node class that goes with the Linked List implementation
- LinkedList class that goes with the Linked List implementation
- ArrayBasedList class that goes with the Dynamic Array-based implementation

The main function contains the code to insert a sequence of HOA objects to a Linked List and a Dynamic Array based List. The for loop for the insertions is written in such a way that one can control where could insertion occur in a current List. The version provided inserts at the end of the List (because of the statement insertIndex = i and the call to the insertAtIndex function uses the insertIndex as one of the arguments). The timers have been appropriately setup to measure the run time for the insertions.

The number of trials is 5 for all the experimental runs.

You have to measure the average run time per insertion on a three-dimensional setup as follows:
(1) insertionIndex = 0 (i.e., insertion occurs at the beginning of the current List)
= i (i.e., insertion occurs at the end of the current List)
= i/2 (i.e., insertion occurs at the middle of the current List)
(2) Data size: Array size for the string of house addresses in the HOA class = 1, 4, 16. Each string object takes 24 bytes. Hence, the size of the data (HOA object) will be correspondingly 24, 96 and 384 bytes.
(3) List size: The list size is varied with values of 1000, 5000 and 10000.

Report:
(1) Show three tables (one table for each of the three different values of the insertionIndex, as shown above) showing the average run time results (in microseconds) obtained for Data Size (24, 96 and 384 bytes) vs. List Size (1000, 5000 and 10000).
(2) Show execution screenshots for Data Size = 24 and 384 bytes; List Size = 1000 and 10000.
(3) Similar to the analysis done in class, compute the memory usage (in bytes) for a Linked List and Dynamic Array for the following combinations of values: Data Size (24 and 384 bytes) and List Size (1000 and 10000 bytes).