**Data Structures**

**Assignment # 5**

**Due Date: Friday, March 27, 2020**

**Section A:** Refer Text Book; **“Data Structures and Algorithms using JAVA 4th Edition”** and do the following,

**Chapter# 3:**

**C-3.7**

Describe a good algorithm for concatenating two singly linked lists *L* and *M*, with header sentinels, into a single list *L* ′ that contains all the nodes of *L* followed by all the nodes of *M*.

**C-3.8**

Give a fast algorithm for concatenating two doubly linked lists *L* and *M*, with header and trailer sentinel nodes, into a single list *L* ′.

**C-3.12**

Describe a recursive method for converting a string of digits into the integer it represents. For example, "13531" represents the integer 13,531.

**C-3.15**

Write a short recursive Java method that finds the minimum and maximum values in an array of int values without using any loops.

**C-3.16**

Describe a recursive algorithm that will check if an array *A* of integers contains an integer *A*[*i*] that is the sum of two integers that appear earlier in *A*, that is, such that *A*[*i*] = *A*[*j*] +*A*[*k*] for *j*,*k* > *i*.

**C-3.17**

Write a short recursive Java method that will rearrange an array of ***int*** values so that all the even values appear before all the odd values.

**P-3.1**

Write a Java program for a matrix class that can add and multiply arbitrary two-dimensional arrays of integers.

**Chapter # 5:**

**R-5.1**

Suppose an initially empty stack *S* has performed a total of 25 push operations, 12 top operations, and 10 pop operations, 3 of which generated ***StackEmptyExceptions***, which were caught and ignored. What is the current size of *S*?

**R-5.7**

Suppose an initially-empty queue *Q* has performed a total of 32 enqueue operations, 10 front operations, and 15 dequeue operations, 5 of which generated ***QueueEmptyExceptions***, which were caught and ignored. What is the current size of *Q?*

**C-5.4**

Describe how to implement the stack ADT using two queues. What is the running time of the push() and pop() methods in this case?

**P-5.1**

Implement the stack ADT with a doubly linked list.

**P-5.4**

Implement the queue ADT using an array.

**P-5.5**

Implement the entire queue ADT using a singly linked list