PRACTICE PROBLEMS: LINEAR DATA STRUCTURES

A simple rule I follow: Honor the coder and their code. The constraints they endured are not ours to know. Make it better if you can.. —Woody Zulill

Course: CS 5002
Fall 2018
Due: No due date

OBJECTIVES

After you complete this assignment, you will be comfortable with:

- The notion of a data structures
- The need for different data structures
- Linked lists
- Doubly linked lists
- Stacks
- Queues

RELEVANT READING

- Introduction to Data Structures
- Basic Data Structures in Python
- Linked List
- Doubly Linked List
- Stack
- Queue

EXERCISES

Question 1

Write a function `count(int number)` that counts the number of times a given integer `number` occurs in a list.

(a) What is the run time of the given function, expressed as function as the size of the list, n.

Question 2

Write a function *getNth(int n)* that takes an integer index `n`, and returns the data value stored in the node at the `n`-th

Points: _____ out of 20

position. Your function should follow 0-based indexing, so for a list 42, 13, 666 `getNth()` with index 1 should return 13.

Question 3

Write a function `insertNth(int n, Node node)` which inserts a new node at index `n` within a list. The caller can specify any index in the range [0..length], but should not be allow to specify an index outside of that range.

Question 4

Write a function `sortList()` which, given a list, rearranges its nodes so that they are sorted in increasing order.

Question 5

Given two lists, write a function `mergeLists(list1, list2)` that merges their nodes to make one list, taking nodes alternately between two lists. So, given two lists with elements 1, 5, 10 and 2, 4, 6, functions `merge Lists(list1, list2)` should return 1, 2, 5, 4, 10 ,6. If one list is shorted than the other, once you run out of the elements in one list, take all of the remaining elements from the other list.

(a) What is the run time of the given function, expressed as function as the size of the lists, n_1 and n_2.

Question	Points	Score
1	10	
2	10	
3	15	
4	15	
5	20	
Total:	70	