$\begin{array}{c} \textbf{CSCE 221 Cover Page} \\ \textbf{Homework Assignment} \ \#2 \end{array}$

 ${\bf Last\ Name}$

First Name

UIN

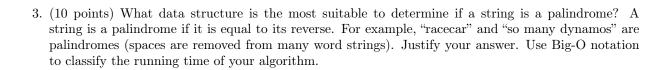
User Name	E-mail address				
current homework. If you fai	the table below including web il to cite sources you can get a ce website: http://aggiehono	a lower number of points			
Type of sources					
People					
Web pages (provide URL)					
Printed material					
Other Sources					
I certify that I have listed all the sources that I used to develop the solutions/codes in the submitted work. On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work.					
Your Name		Date			

Homework 2

due October 18 at 11:59 pm.

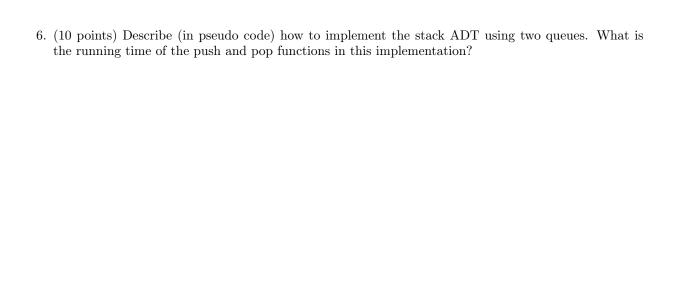
l.	(20)	points) Linked list questions.
		Write a recursive function in C++ that counts the number of nodes in a singly linked list.
	(b)	Write a recurrence relation that represents the running time for your algorithm.
	(c)	Solve this relation and provide the classification of the algorithm using the Big-O asymptotic notation.

2.	(20 points) Write a recursive function that finds the maximum value in an array of int values without using any loops.
	(a) Write a recurrence relation that represents running time of your algorithm.
	(b) Solve this relation and classify the algorithm using the Big-O asymptotic notation.



4. (10 points) Solve C-5.2 on p. 224

(20 points) What is the amortized cost of the stack push operation when the additional stack-arr memory is allocated by each of these two strategies? Do calculations to support your answer.
(a) Doubling strategy – double the size of the stack-array memory if more memory is needed.
(b) Incremental strategy – increase the size of the stack-array by a positive constant c if more memoris needed.



7. (10 points) Solve C-5.8 on p. 224

8. (20	points) Consider the quick sort algorithm.
(a)	Provide an example of the inputs and the values of the pivot point for the best, worst and average cases for the quick sort.
(b)	Write a recursive relation for running time function and its solution for each case.

(a)	Write a recurrence relation for running time function for the merge sort.
(b)	Use two methods to solve the recurrence relation.
(c)	What is the best, worst and average running time of the merge sort algorithm? Justify your
(C)	answer.

9. (15 points) Consider the merge sort algorithm.