Midterm
CS 163: Data Structures

Name: _____
;

Email Address: _____

1. (25 points) Short Answer: Limit your answers to 1-2 sentences.

Think about what we have talked about using linear linked lists versus arrays. Answer these questions:

a) List one advantage of using a linear linked list:

b) List one advantage of using an array

c) What memory issues exist with linear linked lists:

d) What memory issues exist with statically allocated arrays:

e) What memory issues exist with dynamically allocated arrays:

f) Create a node structure for a linear linked list of names; where, a name is stored in a dynamically allocated array of characters.

g) How would (f) change if the node was for a doubly linked list?

2. (25 points) Efficiency Discussions: Limit your answers to 1-2 sentences.

We have talked a lot this term about efficiency as we have discussed different data structures and Abstract Data Types…keep this in mind as you answer the following questions

ADT choices: Absolute Ordered Lists, Relative Ordered Lists, Stacks, Queues

Data Structure Choices: Linear Linked List, Circular Linked List, Doubly Linked List, Array

a) When have we seen the worst RUN TIME efficiency this term…

ADT: _____ Data Structure: _____

ADT choices: Absolute Ordered Lists, Relative Ordered Lists, Stacks, Queues

Data Structure Choices: Linear Linked List, Circular Linked List, Doubly Linked List, Array

b) We have also talked a lot this term about MEMORY efficiency. Tell me about the worst memory issues we have encountered so far this term…

ADT: _____ Data Structure: _____

ADT choices: Absolute Ordered Lists, Relative Ordered Lists, Stacks, Queues

Data Structure Choices: Linear Linked List, Circular Linked List, Doubly Linked List, Array

c) If we were to implement an absolute ordered list (one WITH holes) which data structure would be best for run-time efficiency:

Data Structure: _____

d) If we were to implement an stack which data structure would be best for memory efficiency:

Data Structure:_____

3. (25 points) Stacks and Queues – implement code in C++

a) Write the implementation of the **dequeue** operation when implemented using a **linear linked list**

int dequeue () { //data is a struct of one dynamically allocated array of chars

b) Write the implementation of the **push** operation
when implemented using a **linear linked list**

int push (char name []) { //the argument is the name to be pushed onto the stack

3. (25 points) C++ Code

a. Write a function that will copy every node in a linear linked list (keep it simple – just a list of integers) to a new linear linked list; assume that the new list has a head pointer that is NULL to begin with. Make no other assumptions about the list that you are copying from.

b. Assume you have a linear linked list. Let's assume it is a list of integers. Write the code to remove the last node from this list (only given a head pointer as an argument):

c. How would (b) change if there was a “tail” pointer in addition to a head pointer?

d. Show the code to add a node at the beginning of a DOUBLY linked list (given a head pointer) of integers.