Pre-lab Worksheet

CS163 Lab Session #4 – Other Linked Lists

Each group member should collaborate on this worksheet. All online students should participate!
The goal of these worksheets is to help prepare you for the next step in programming!

Goals:

- Experience implementing circular and doubly linked lists from Topic #5
- Experience critical thinking.
- Build test plans and experience unit testing as member functions are developed

_____*1.* **Circular Linked lists.** *Assume that a circular linked list has a rear pointer – which points to the last item in the list. That node, in turn, points to the first node in the list*

a. When working with linked lists, the first thing to consider is the special cases. What are the special cases for finding a matching item in a **circular linked list**?

- There is only one item in the list
- The item does not exist

b. For a linear linked list, what is the stopping condition when we want to traverse through all of the data (eg., for display) current == NULL

c. How would this change for a circular linked list? (Handle all special cases)

This would change because you would have to test if the first item repeated itself

_____2. Create the algorithm for removing the last item in a **circular linked list**?

Using the rear pointer, traverse until the next node is the node to be deleted, then have a temporary pointer point to it, point around it, delete the node, then have tail point to the new tail pointer

_____3. Create the algorithm for inserting into a **doubly linked list**?

Traverse to before the desired insertion location, create a to_add node, copy the data into the node then insert.
Insertion consists of connecting the node using the following order: connect the next pointer of to_add to the node that is after the current node, then connect the previous node of the to_add to the current node, then connect the previous node that is now after the to_add pointer to the to_add pointer, then change the next pointer of the current node to the to_add node