Test 4 Example Questions:

1. Quiz 5
2. Closed lab 18 questions, and
3. The two programs developed for Closed lab 19 and closed lab 20
4. Write C++ client program or function using unsorted or sorted list class (linked list implementation) as discussed in class.

5. All of the operations below are to be done **in the main/client program only**. You should not assume that a linked list version of the list class has been implemented. Show C++ code to:
    a. Read a list of integer numbers from a data file and add the numbers into a **linked list** of integer type.
    b. delete the first node in the list
    c. delete the k^{th} node in the list (position k could be in the middle, or at the end of the list)
    d. delete ALL the nodes in the list
    e. insert a new node with value 10 as the first node in the list
    f. insert a new node with value 5 as the last node in the list
    g. display the values in all the nodes
    h. find the sum of the values in all the nodes
    i. free memory of all the nodes in the list
    j. create a deep copy of the existing list

6. All of the operations below are to be done assuming a **sorted linked list class** has been implemented.
    a. Show code in the **client program** to:
        i. Display all the values in the list one by one;
        ii. add a new, user-supplied, value into the list;
        iii. look to see if a user specified value is in the list or not;
        iv. delete a user specified value from the list.
    b. Show how the **insert method** is implemented as it appears in the implementation file, list.cpp. Make sure your code handles situations where a new value to be added may be the added as the first node in the list, or as a node in the middle of the list or at the end of the list.
    c. Show how the delete method is implemented as it appears in the implementation file, list.cpp. Your code should work for any of these possible situations:
        i. What if the node with value 8 is the first node in the list?
        ii. What if the node is in the middle of the list?
        iii. What if the node is at the end of the list?
        iv. What if the list was empty?
        v. What if the value to be deleted is not in the list?
    d. Show the copy constructor as it is shown in the implementation file
    e. Show the destructor as it is shown in the implementation file