The Assignment:

You will implement and test a revised sequence class that uses a linked list to store the items.

Purposes:

Ensure that you can write a small class that uses the linked list toolkit to create and manipulate a linked list.

Before Starting:

Read all of Chapter 5, with particular attention to Sections 5.3 and 5.4.

Files that you must write:

1. sequence3.h
2. Download sequence3.h
3. : The header file for the new sequence class. Actually, you don't have to write much of this file. Just start with the provided version and add your name and other information at the top. If some of your member functions are implemented as inline functions, then you may put those implementations in this file too. By the way, you might want to compare this header file with your first sequence header file sequence1.h and second sequence header file sequence2.h. The linked list version no longer has a CAPACITY constant nor a DEFAULT_CAPACITY constant because the items are stored on a linked list instead of an array.
4. sequence3.cxx:The implementation file for the new sequence class. You will write all of this file, which will have the implementations of all the sequence's member functions.

Other files that you may find helpful:

1. sequence_test3.cxx
2. Download sequence_test3.cxx
3. :This is the same interactive test program that you used with the earlier sequences. If you want to use it with the new sequence, then copy it to your

project and open it with your editor. The namespace for this assignment is: main_savitch_5.
4. sequence_exam3.cxx
5. Download sequence_exam3.cxx
6. :This is a non-interactive test program that will be used to grade the correctness of your new sequence class.
7. node1.h
8. Download node1.h
9. and node1.cxx
10. Download node1.cxx
11. : Copy these files to your project. They contain the node class ad helper functions. You may use these files without changing them.

The Sequence Class Using a Linked List

Discussion of the Assignment

Your sequence class for this assignment will differ from the your previous sequence in the following ways:

- The sequence's items are now stored on a linked list. The head pointer of the linked list is a private member variable of the sequence class. I suggest that you also have a tail pointer as an additional private member variable of the sequence class. The reason for the tail pointer is explained in Section 5.4 of the class text.
- Because you are dynamically allocation memory within your sequence class, you will need to define a copy constructor, an assignment operator, and a destructor.

Start by declaring the new sequence's private member variables in sequence3.h. You might try declaring these variables yourself, and then compare your solution with the suggestion in Section 5.4.

Once again, do your work in small pieces. For example, my first version of the sequence had only a constructor, start, insert, advance, and current. My other member functions started out as stubs.

Do not use size(), many_nodes or is_item() in your code, other than for assert().

Use the interactive test program and the debugger to track down errors in your implementation. If you have an error, *do not start making changes until you have* *identified the cause of the error.* If you come to me for help, we will always ask you to do the following:

1. Show us the invariant that describes how your private member variables implement the sequence class.
2. Use the debugger to show us the problem!