CSc 227 — Program Design and Development
Spring 2014 (McCann)

http://www.cs.arizona.edu/classes/cs227/spring14/

Section Activity #8: Adding `lastIndexOf (E item)` to `CS227LinkedList`

Your Names: _____

Directions: In groups of two (or three, if need be), complete the following activity. This section activity will be graded; all students in the group will receive the same score. Make sure that the names of all group members are on the page you submit to your section leader. Section Meeting 13 (2014/04/16-17)

Background: We've examined the idea of using linked lists of node objects to represent lists of data items. Your SL has just finished talking about using a "tail" reference to keep track of the last node in the list.

One of Java's `List` interface routines is `int lastIndexOf (E item)`, which returns the (0-based) position in the list of the last occurence of `item`. For example, if the list contains the letters M, T, W, R, F, S and S, `lastIndexOf()` would report that the position of the last 'S' is 6.

Task: Write an implementation of the `lastIndexOf()` method as an addition to our `CS227LinkedList` class (from `T09n01.java`). If the given item isn't a member of the list, return the value -1. Assume that you have access to (and must maintain, if necessary) both head and tail list references, as well as an occupancy variable. You may use any methods of our `CS227ListInferface` interface that you wish to use; both it and the `Node` class from `T09n01.java` are given on the back of this page.

```
public int lastIndexOf (E item) {









}
```

CS227ListInterface:

```
interface CS227ListInterface<E>
{
    public     int   append (E item);
    public     int  prepend (E item);
    public     int   insert (int location, E item);
    public       E   delete (int location);
    public boolean  isEmpty ();
    public boolean   isFull ();
    public     int     size ();
    public     int capacity ();
    public  String toString ();
}
```

The Node Class:

```
class Node<E>
{
    private        E  data;
    private  Node<E>  next;

    public Node ()
    {
        next = null;
        data = null;
    }

    public Node (E object)
    {
        data = object;
        next = null;
    }

    public       E  getData ()             { return data; }
    public Node<E>  getNext ()             { return next; }

    public    void  setData (E object)     { data = object; }
    public    void  setNext (Node<E> node) { next = node; }
}
```