Dan Bonnett
Lab 4 Additional Questions

1) To find the second to last node in a singly kinked list, follow this algorithm:
   A) pointer1 and pointer2 both start out as pointing to the head node.

   B) pointer2 gets advanced by n-1 nodes and pointer1 still points to the head.

   C) Advance pointer1 and pointer2 by one node

   D) When pointer2 equals null we know we have reached the tail and pointer1 now points to the nth to the last node. The answer can be returned as the node that pointer1 is pointing to.

2) Concatenate singly linked lists L and M

   A) If L is empty then return M.

   B) If M is empty then return L.

   C) Store the head node of L in a pointer variable, we'll use v.

   D) Move v to the tail

   E) Store the first node of M in the next field of the node pointed to by V.

   F) Return L

   G) Declare this new list as a new linked list L'.

3) How to determine whether circularly linked lists J and K have the same sequence of elements.

   To do this you must iterate through K to see if list K has the same set of values as J. Start with element 1 in J, and search for that element in K. If K contains that element, then move onto searching for element 2 of J in list K.