CSCD 211
Exam 2 Study Guide

Linked Lists -- Given the linked list code below, answer the questions that follow.

You'll be given some standard linked list code (singly-linked, no dummy head node,) and be asked to trace through the code to determine the output. Suggestion for this part: **Draw pictures.**

```
public class LinkedList
{
      private class Node
      {
            private Node next;
            private Comparable data;

            public Node(Comparable data)
            {
                  this(data, null);
            }

            public Node(Comparable data, Node next)
            {
                  this.data = data;
                  this.next = next;
            }
            public Comparable getData(){ return this.data; }
            public Node getNext(){return this.next;}

      }//end nested Node class

      private Node head;
      private int numItems;

      public boolean isEmpty()
      {
            return this.head == null;
      }

      public void deleteAll()
      {
            this.head = null;
            this.numItems = 0;
      }


      private static void mystery(Node curr)
      {
            while (curr != null)
            {
                  System.out.println(curr.data)
            }
            curr = curr.getNext();
      }
```

1. Describe completely what the mystery method does.

2. Write four of the following methods for the LinkedList class: **add, delete**, **sort**, **toString**, **addOrdered**.

Recursion Section

3. Show the output and the total number of calls that will be made to the method **two** based on an initial call in main() of: **two(7)** . (Hint: Draw a recursion tree…)

```
void two(int n)
{
      if (n > 2)
      {
            two(n-2);
            two(n-3);
            two(n-2);
      }
      System.out.println(n);
}
```

4. Shown below is the iterative version of binary search. Write the recursive version! Note that the initial call to the recursive version would look like this:

```
int index = binarySearch(array, target, 0, array.length-1);

public static int binarySearch(Comparable [] array, Comparable target)
{
     int first=0, last=array.length-1, mid;

     while (first <= last)
     {
           mid = (first + last) / 2;
           if (array[mid].compareTo(target) < 0)
                first = mid + 1;
           else if (array[mid].compareTo(target) > 0)
               last = mid - 1;
           else
                return mid;
     }//end while
     return -1;//not found
}//end binarySearch
```