1. Write a static method called `aveArray` that is passed an integer array as an input. It must first determine the average of the values in the array. Then it determines how many values in the array are greater than the average, and returns a new array containing only those values.

For example, if passed the following array:

   \{13, 10, 15, 11, 12\}

then the array to be returned would contain: \{13, 15\} (since the average is 12.2)
2. What is printed when the following program is run?

```java
public class Example {
    public static void main (String [] args) {
        String word = "lurcvkty";
        char[] array = new char[word.length()];

        char achar = ' * ';
        for (int j = 0; j < array.length; j++) {
            array [j] = word.charAt(j);
        }

        System.out.println("before:");
        for (int j = 0; j < array.length; j++) {
            System.out.print(array[j]);
        }
        System.out.println();
        aMethod (array, achar);

        System.out.println("after:");
        for (int j = 0; j < array.length; j++) {
            System.out.print(array[j]);
        }
        System.out.println();
    }

    public static void aMethod (char[] arr, char ch) {
        int place = arr.length - 2;
        for (int i = arr.length - 2; i > 1; i -= 2){
            arr[i] = ch;
        }
    }
}
```

Output:
3. Show how the contents of the following array are sorted using each of the following:
   insertion, selection, and bubble sort. How many comparisons are required by each of the
   sorts?

   Array = {12, 14, 7, 6, 9}

Compare MergeSort to SelectionSort. Include the comparison of running times for best/worst
case of each algorithm running on an array of n elements as well as the comparison of the
memory requirements.

Which sorting algorithm among those studied in the course would terminate first when sorting an
array in which only a few elements are out of order. Justify your answer.
4. Given the `guess` method written below, what is the value of `answer`, where `answer = guess(2, 3)`? Use a stack to show your work.

```java
public static int guess(int a, int b) {
    if (b == 1)
        return a
    else
        return a + guess(a, b - 1)
}
```
5. Objects, Classes, and Arrays

The Coins class has the following instance variables:

```java
public class Coins {
    private String type; // penny, nickel, dime, or quarter. Lowercase!
    private int value;   // in cents - 5 for nickel, 10 for dime, etc.
    private int count;   // number of coins of that type
}
```

a. Write an instance method for the Coins class called `setValue()` that sets the `value` instance variable of the Coins object that invokes it to 1 for a penny, 5 for a nickel, 10 for a dime, or 25 for a quarter, as identified by the `type` instance variable of the invoking object. You may assume that the value of the `type` instance variable is always stored as a lowercase value.

b. Write a 2-arg constructor for the Coins class that sets the `type` instance variable to the lowercase equivalent of the first argument, the `count` instance variable to the second argument, and uses the `setValue()` instance method from part a to set the `value` instance variable.

c. Write a `getValue()` accessor method for the Coins class that returns the total value, in change, of the Coins object passed to it. For example, if a Coins object with a `type` instance variable of “nickel” and a `count` instance variable of 3 invoked this method, then a value of 15 should be returned.
5. Objects, Classes, and Arrays (continued)

d. Complete the `println` code in the following `UseCoins` class so that the value of `coin1` is printed using the `getValue()` accessor method from part c.

```java
public class UseCoins {
    public static void main(String[] args) {
        coin1 = new Coins("nickel", 3);  
        System.out.println("Total value = " + ____________________);
    }
}
```

e. The Wallet class defines an array of `Coin` objects. Add a static method to this class called `totalValue()` that is passed an array of `Coin` objects (as shown) and returns the total value, in change, of all of the objects in that array.

```java
public class Wallet {
    public static void main(String[] args) {
        Coins[] coins = new Coins[4];  
        coins[0] = new Coins("nickel", 14);  
        coins[1] = new Coins("penny", 12);  
        coins[2] = new Coins("quarter", 10);  
        coins[3] = new Coins("dime", 4);  
        System.out.println("Total value = " + totalValue(coins));  
    }
    // add code here
}
```
6. Define a new class to represent course information in the following way:
each course object is characterized by a name, maximum enrollment and is assigned a classroom
type, that is either “computerized” or “regular”. Your class must define a separate instance
variable to represent each of the characteristics mentioned above.

You do not need to define any methods, but you have to define a constructor. The constructor
should assign the values to the instance variables representing course name and maximum
enrollment to the values passed in through the parameters, and should assign the value of
instance variable representing the classroom type according to the following rule: all classes with
a name that starts with “CS” or “IT” must be assigned a computerized classroom, all others – a
regular classroom.

(Hint: substring(startPos, endPos) is an instance method of class String that extracts from the
calling string the substring starting at startPos and ending at endPos-1)

How should the class you defined be modified to be used for representing a node in a linked list
of course objects? Do not merely describe the modifications in English, but provide the Java
statements that have to be added to the class definition.
7. Given the following class definition:

```java
public class ListNode {
    private String data;
    private ListNode link;

    public ListNode( ){
        link = null;
        data = null;
    }
    public ListNode(String newData, ListNode linkValue){
        data = newData;
        link = linkValue;
    }
    public void setData(String newData){
        data = newData;
    }
    public String getData( ){
        return data;
    }
    public void setLink(ListNode newLink){
        link = newLink;
    }
    public ListNode getLink( ){
        return link;
    }
}
```

Assume the figure below depicts the state of memory achieved after a set of statements that are not presented here is executed, and that `head` is of type `ListNode`, as defined above.

What would be the effect of the following code on the above list? Describe it in words and also modify the above diagram to show any changes.

```java
ListNode first = head;
ListNode next = head.getLink();
next = next.getLink();
first.setLink(next)
```