Lab 7 (40 points total)
The purpose of this lab is to continue learning the BlueJ Integrated Development Environment (IDE) and provide hands on practice for Chapter 7 concepts. There are several learning objectives to this assignment

- Writing Class and Method Definitions with a focus on Arrays.
- Introduction to ArrayLists
- Using debugging techniques
- Incorporating documentation and style into your code

Lab 7 Part1 (20 points)
Use the following UML to create a class called ArrayMethods. ArrayMethods will need to use the java.util.Arrays class (requires import). **NOTE-Arrays is a static class, meaning you do not need to create an object. For example you can just do Arrays.copyOf()**

ArrayMethods
- a: int [ ] (initialized to {7, 8, 8, 3, 4, 9, 8, 7})
Lab 7 PART 1 - Methods + count (): int (1pt) + sum (): int (1pt) + average (): double (1pt) +findMax (): int (3pts) +findIndexOfMax (): int (3pts) +findLast (int key): int (4pts) +findAll(int key): int [ ] (4pts) +getArray(): int[ ] (1pt) +copyArray (): int [ ] *(Note: Use Arrays.copyOf())* (2pts) +print(int [ ] a): void **//provided below**
Lab 7 PART 2 - Methods +sortArray (int [ ] inArray): void (3pts) +reverseArray(int[ ] inArray): int[ ] (2pts) +genFiveByFiveRows():void //generates int[5][5] and then prints int[ ][ ] (2pts) +genFiveByFiveCols():void //generates char[5][5] and then prints char[ ][ ] (3pts)

1) count () returns the number of values that are in the array
   (Note: Use an enhanced for (for each) loop with a statement in this method to compute the number of values) vs .length //HINT – Just use a simple accumulator
2) sum () returns the sum of the values that are in the array
3) average () returns the average of the array as a double. **HINT-What does (double)sum()/count() do?**
4) findMax () returns the value of the largest integer in the array. **HINT: set max=a[0] and then run through a[ ], if a value is bigger, update max. You will do similar for findIndexOfMax(), also see pg 428.**
5) findIndexOfMax () returns the index of the value of the largest integer in the array
6) findLast (int key) returns the index of the last value of the array based on the index parameter that is passed by value or -1 if the value 'key' is not found. **You will want to start at the right most index, for example, a.length-1 and work to the left (index[0])**

7) findAll(int key) creates and returns a new array containing the index(es) of every occurrence of a target value. Return an empty array of length 0 that contains nothing if the target value does not occur. Use the following array below as a print method to print the resultant findAll [ ]

   Hints: use 2 loops for this. The first counts how many times the target occurs. Next create a new array, to hold this many indexes. The second loop puts the indexes into the new array. Make sure after you put the value in the new array that you move to the next element.

8) getArray() is a simple getter for int[ ] a instance var.
9) copyArray() creates a copy of the a[ ] using the Arrays.copyOf(). HINT: copyOf requires two arguments - http://docs.oracle.com/javase/7/docs/api/java/util/Arrays.html

   NOTE: It was necessary to import java.util.Arrays into ArraysMethodsDemo to do this.

Other items of interest in ArrayMethods class

a) Use Arrays.copyOf() to create copyArray().
b) Arrays.sort() is the Java API method for sorting an array
c) Arrays.equals () is the Java API method for determining if two arrays are equal. Returns `true` if the two specified arrays of ints are *equal* to one another. Two arrays are considered equal if both arrays contain the same number of elements, and all corresponding pairs of elements in the two arrays are equal. In other words, two arrays are equal if they contain the same elements in the same order.

```
/**
 prints an int array, nicely formatted
 @param  a[ ]   int array to print
*/
public void print(int a[ ])
{
    System.out.print("{");
    int i;
    // print elements before the last, separated by commas
    for (i = 0; i < a.length - 1; ++i)
        System.out.print(a[i] + ", ");
    // print last element.  Careful here to handle length 0
    if (a.length > 0)
        System.out.print(a[i]);
    System.out.println("}");
}
```

Lab 7Pt2 (20pts)

Lab7Pt2 – (Continuation of Lab7 Pt1) (10pts)

1) Use sortArray() to sort int[ ] copyOfA into ascending order. One of the easiest implementations is to create an outer for loop, start with index[0] and compare to an inner for loop that starts with index[1] through index[length-1], if the value index[innerLoop] is less than value index[0], set a temp var to the value of index[innerLoop], set

index[innerLoop] to value index[outerLoop], then set index[outerLoop] to temp var value. Then do the same with index[1] compared to index[2] through index [length-1] and swap as necessary. This repeats until the outer loop index gets to length-1. **DO NOT USE Arrays.sort()** *See pg 450+.*

Hints: You will want to use two for loops for this, one nested in the other and will need to use a temporary value to swap the value being evaluated with the value at an index if the value at the index is less than the value being evaluated. See pg 450+

2) reverseArray(int[ ] inArray) uses the inputted array to reverse the order and returns either the parameter array or another array produced in the method. **DO NOT USE Collections or ArrayList methods. You must create your own** 😊
There are a few ways to do this. Here are a couple of hints.
    a. Start at index[0] and index[lenth-1] and swap values, you will need a temp var to do this. After each swap move the left (start) index one to the right and the right (end) index one to the left, until the left (start) index > the right (end) index and return the param int[ ] –or-
    b. Create a new array the same size as a[ ]. Copy the value index[length-1] to the new array index [0] and move left to right from the original int [ ] and right to left in the new int [ ]. This is similar to when we reversed the words in Lab4. Then return the assign the new array to the param var so that the array passed in now points to the memory location of the new int [ ].

3) genFiveByFiveRows() generates and prints out a 5x5 **int [ ][ ]**. The columns are A->E and the rows are 1->5. For this method, row 1 will have all ones (1's), row 2 will have all two's (2's), row 3 all three's(3's), row 4 all four's (4's), and row 5 all five's (5's). Since a 2D array, you will need to have 2 for loops. The outer for loop will be for [row] and the inner for loop for [col]. *See pg 463+.*
    a. Tips – Print the top row header (this can be hard coded), For the left row number to show the row, add 1 to the row index to get the correct number. Since you want the rows the same number across, put the value of the row+1 in for the [row][col] value.

4) genFiveByFiveCols() generates and prints out a 5x5 **char[ ][ ]**. The columns are A->E and the rows are 1->5. For this method, col 1 will have all A's, col 2 will have all B's, col 3 all C's, col 4 all D's, col 5 all E's. *See pg 463+.*
    a. Tips – Print the top row header this can be hard coded), For the left row number to show the row, add 1 to the row index to get the correct number. Since you want the rows to be the same as the column header you will need to vary the [row][col] value based on the col for loop index value.
    b. Remember a char is really just an int. Don't forget that you can cast (char). What is the value of (char)('Z'-1)? Hope you guessed 'Y'.

Lab7 Pt2 – Problem 2 (10pts)

Use the following UML Diagram to create TV_Actor class (5pts). **See pg 472+ and pptx slides 69+.**

TV_Actor
- name: String - show: String
+ TV_Actor() //Set name to "Unknown Name", show to "Unknown Show" + TV_Actor(inName:String, inShow:String)

+ getters/setters for name and show **//Will be a total of 4**
+ toString(): String //changes default .toString() – **Provided below**

Update driver class named TV_ActorDemo_Starter (5pts):

1) Import `java.util.ArrayList`
2) Create an ArrayList TV_Actor collection that is called `list`
3) Add three TV_Actors to the list. For each actor, you can do this in one command or create a TV_Actor object with a TV_Actor temp var and pass in the objects to the list.
    a. Bart, The Simpsons
    b. Maggie, The Simpsons
    c. Lisa, The Simpsons
4) Check out the printArrayList() that accepts an ArrayList TV_Actor collection and does not return anything (void)
    a. The method prints out "Printing ArrayList<TV_Actor> list"
    b. Then, prints the current contents of the list. The enhanced for (for each) loop makes this super compact!
5) Call/Invoke printArrayList passing in `list` as an arg
6) Remove the entry in `list` at index 1
7) Invoke printArrayList passing in `list` as an arg
8) Replace the entry at `list` index 1 with an TV_Actor with name Butthead who was in show B&B
9) Invoke printArrayList passing in `list` as an arg
10) Add an entry in `list` at index 1, a TV_Actor with name Beavis who was in show B&B
11) Invoke printArrayList passing in `list` as an arg
12) Remove the entry at `list` index 0
13) Invoke printArrayList passing in `list` as an arg

```
/**
* Code to print out a TV_Actor object
* put in TV_Actor class file, this overrides the default .toString()
* @return formatted TV_Actor output
*/
public String toString(){
    String output = "Name: " + name + ", Show: " + show;
    return output;
}
```

Submitting your work

For all labs you will need to provide a copy of all .java files. **No need to provide .class files. I cannot read these.** ***NOTE – For Replit, please update Main.java to another name such as TempProb.java, ProChall3.java, etc.*** In addition to your .java files, you will need to provide output files of your console. The name of the output file should match the class name and have the .txt extension such as TempProbOut.txt, ProChall3Output.txt. For GUIs such as JOptionPane, you will instead need to create screenshots. For Windows users, Snipping Tool is a great way to do this. Chromebook - Shift+Ctrl+Show Windows. Mac OS users, you can see how to take screenshots using the following url - https://support.apple.com/en-us/HT201361.