CSCI 3302 Pair Programming Assignment 02 (100 Points)

Due: Sept 22, 8:00 AM

GitHub Link: Pair Program 02

Objectives:

- Demonstrate understanding of recursion.
- Demonstrate how to implement operations on arrays.

Assignment Assistance:

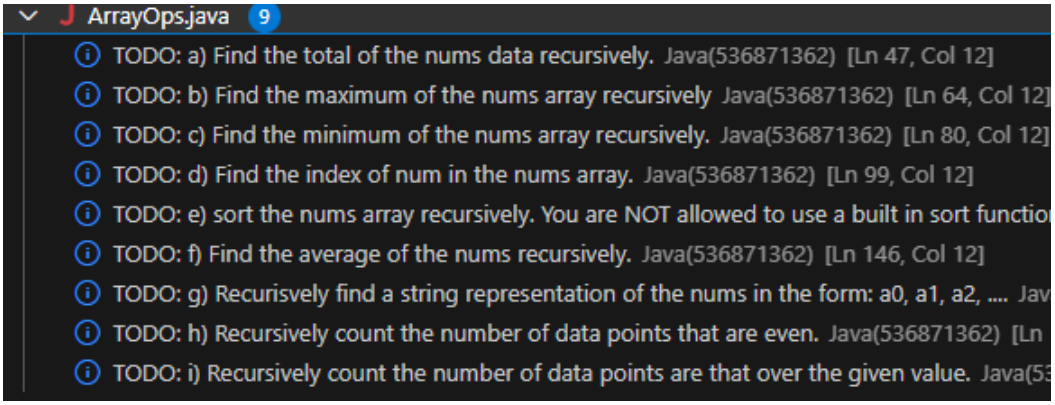
- This homework assignment is due before the date and time specified above.
- You and your partner must work on this TOGETHER. One person is the DRIVER, the other is the NAVIGATOR. If you are unfamiliar with paired programming, please google and get acquainted. Here is a quick video with an overview: https://youtu.be/q7d_JtyCq1A
- This assignment is restricted to you and your partner. You may not receive help from any other person except the instructor or the AARC (help from the AARC must be well documented!).
- Any resource used (other than Dr. Becnel or the course text) must be documented in the code (as comments) detailing the source and describing exactly what was learned and how that information was used. Submissions will be severely penalized if copied in part or in whole from any source.
- If you need help, visit your instructor during his posted office hours. If your schedule cannot accommodate any of these times, then email your instructor to schedule a different time.

Problem Description:

1. After getting access to the repository, acclimate yourself to the files
   a. `ArrayOps` – a class that performs basic operations on an array of integers
      - `nums` – an array of integers on which operations are performed
   b. `TestArrayOps` – a simple test program for the ArrayOps class. **You should comment and uncomment blocks of code as you go to test methods one at a time**.

2. You should start by understanding the code you were given. In particular, notice a few things:
   a. The sample size is assumed to be a positive integer (see constructor).
   b. Each method in `ArrayOps` that is public has a private helper method. **These private methods should use a recursive solution. This is where you will implement your solution.**
   c. The test file has a few options that you can use for test data. Simply comment/uncomment out your choice.

3. In VS Code, if you look under Problems, you will see a lot of TODOs.



   a. `total` – sum up all elements of the array and return the result.

   b. `max` – find the maximum value in the array and return the result

   c. `min` – find the minimum value in the array and return the result

   d. `indexOf` – finds the location (index) of a value in the array; returns -1 if the value is not in the array.

   e. `sorts` – sorts the array of nums. You cannot use a built-in Java function. You should recursively sort the elements. There are some hints in the comments.

   f. `average` – returns the average value of all numbers in the array

   g. `toString` – returns a string listing all numbers in the array in the form $[a_1, a_2, \ldots, a_{n-1}]$.

   h. `numEven` – returns an integer representing a count of the total number of elements in the array

   i. `findNumOver` – returns an integer corresponding to a count of array elements that are greater than the given value.

4. Add appropriate comments to all your methods. You may consider using `/**` for the Java doc comment. You should also add an appropriate file heading to the file which includes both your name and your partner's name.

5. Your submission should compile with no compiler errors or warnings.

6. You may write any `private` helper methods if needed.

HINTS:

- While testing your code, change the data or create new data files to ensure you fully test your methods.
- To start, comment out the majority of the `TestArrayOps.java` file. As you implement methods, uncomment the corresponding portions of the file to test. Note: This file is not meant to be a complete test.

SUBMISSION:

- Review the Evaluation below to ensure you have met all the requirements.
- Only one person needs to turn in/commit the assignment. However, both members of the team are expected to understand the solutions and be able to answer questions about the solution.
- Commit all files to GitHub. Upload a backup copy to D2L. Your submitted files should NOT contain a main method or testing code. You may include testing files in your repository; however, these will not be considered when grading. If you wish to include non-working code for insight into your thought process, make sure to contain it within comment blocks and ensure that submission successfully compiles.

EVALUATION

a)	Project is late or not submitted at all.	-100
b)	Project does not run/compile.	-50
c)	Project compiles with warnings.	-30
d)	Project does not correctly implement the interface.	-30
e)	Calculations/output are for requirements incorrect.	-10 each
f)	Code is not well organized or properly indented.	-5
g)	Code is inadequately commented for readability.	-5
h)	Code does not contain the student's name, course section, and date of submission.	-5
i)	Code is not submitted to the Github	-15