Instructions

Write one method per task (all of these methods should go in the same class). **If a name for a method is specified, make sure it has that name!!!** Otherwise, it will break the testing suite.

Before writing a method, it is recommended to FIRST write an algorithm or make a flowchart depicting the high-level operations being performed. **For methods with arguments, make sure to test out your methods on a variety of inputs to ensure they are working correctly!** Once you are satisfied that your code works correctly, make sure you've followed all the style guidelines.

Tasks: Arrays and Methods

1. Write a method 'public static void printIntArray(int[] theArray)' that takes in an array of integers theArray, then prints out each of the integers in the array, one per line (use a loop). (Nothing is returned by this method.)
2. Write a method 'public static int sumArray(int[] theArray)' that takes in an array of integers, then returns a single integer that is the **sum** of all of the integers in the array (hint: you can reuse most of the code from the NumberSummer example from the loops unit).
3. Write a method 'public static int averageArray(int[] theArray)' that takes in an array of integers, then returns a single integer that is the **average** of all of the integers in the array (hint: this will be very similar to the previous method; you could even call the previous method to streamline the process if you want. Think about how to compute an average given a sum.).
4. Write a method 'public static int maxArray(int[] theArray)' that takes in an array of integers, then returns a single integer that is the **maximum** of all of the integers in the array (hint: you can reuse most of the code from FindArrayMax.java).
5. Write a method 'public static int minArray(int[] theArray)' that takes in an array of integers, then returns a single integer that is the **minimum** of all of the integers in the array (hint: very similar to maxArray).

Style

These style guidelines should be followed at all times:

- Source code file names and class names should start with an uppercase (capital) letter.
- Source code file names and class names should be relevant to the function of the code.
- Variable names should start with a lowercase letter (unless they are program-wide static final constants, in which case they should be written in all caps).
- Variable names should reflect their purpose, for readability.
- Every time you open a block of code with a {, all of the code inside the block (i.e., everything up until the corresponding }) should be indented. The amount to indent your code by is up to personal preference (I like 2-4 spaces), but you should keep it consistent throughout a program.
- Use line breaks (newlines, as in what you get when you hit the 'Enter'/'Return' key on your keyboard) appropriately to make your code more readable.
- Write a block comment (that's the one that starts with /* and ends with */) at the start of your program containing a brief description of what the program does, along with your name or hacker name.
- Write comments throughout your code whenever you do anything complicated or particularly interesting. A good rule of thumb is to comment anything you don't think you'd be able to immediately understand, if you looked at your code 2 months from now.