CSSSKL142 – Lab8

Arrays

University of Washington, Bothell

Create a class called **Lab8** and in it implement all of the methods below. Also, write a main method with test calls to all of these methods. Don't forget to turn in your file to Canvas before the end of the lab today.

1. Write a function called **median**, that takes as parameter a full, sorted array of doubles and returns the median of the list. For a sorted list of odd length, the median is the middle value. For a sorted list of even length, the median is the average of the two middle values. Make an example function call in your main.
2. Write a function called **isSorted** that takes an array of doubles as a parameter and returns true if the list is in sorted (non-decreasing) order and returns false otherwise. Make an example function call in your main.
3. Write a function called **findCommon** that takes three arrays of positive integers as parameters. The first two array parameters are filled with ints. Fill the third array parameter with all the values that are uniquely in common from the first two arrays and the rest of the array with zeros. For example:

   (a) **a1[]** contains: `3 8 5 6 5 8 9 2`

   (b) **a2[]** contains: `5 15 4 6 7 3 9 11 9 3 12 13 14 9 5 3 13`

   (c) **common[]** should contain: `3 5 6 9 0 0 0 0`

4. Write a function called **rotateRight** that takes an array of integers as an argument and rotates values in the array one to the right (i.e., one forward in position), shifting the value at the end of the array to the front. For example, if the array called **list** stores [3, 8, 19, 7] before the function is called, it should store [7, 3, 8, 19] after the function is called. Another call on **rotateRight** would leave the list as [19, 7, 3, 8]. Another call would leave the list as [8, 19, 7, 3].
5. Write a function **count** that takes an array of integers and a target value as parameters and returns the number of occurrences of the target value in the array. For example, if an array called **list** stores the sequence of values [3, 5, 2, 1, 92, 38, 3, 14, 5, 73] then the following call **int n = count(list, 3);** would return 2 because there are 2 occurrences of the value 3 in the list.
6. Write a function called **stretch** that takes an array of integers as an argument, and returns a new array **twice** as large as the original that 'replaces' every integer from the original list with a pair of integers, each half the original, and then returns it. If a number in the original list is odd, then the first number in the new pair should be one higher than the second so that the sum equals the original number. For example, suppose a variable called **list** stores the sequence of values [18, 7, 4, 24, 11]. The number 18 is stretched into the pair (9, 9), the number 7 is stretched into (4, 3), the number 4 is stretched into (2, 2), the number 24 is stretched into (12, 12) and the number 11 is stretched into (6, 5). Thus, the call:

   ```
   stretch(list);
   ```

   should replace list with the following list which is twice the length of the original:

   ```
   [9, 9, 4, 3, 2, 2, 12, 12, 6, 5]
   ```