CSF
Hwk07
Hip, Hip, Array!

In this lab you will be writing a number of methods involving arrays. Some helper methods have been written for you to use in your own methods and for testing.

1. Write method isSorted() that takes an integer array and returns a boolean. It should return true if the array is sorted, and false otherwise. The array should not be changed.

2. Write method replace() that takes an integer array and two integers (*s* and *r*) and returns void. It should change the array so that all occurrences of *s* (search) are replaced by the integer *r* (replace).
Ex: if x = { 10, 3, 4, 1, 9, 10, 3, 6, 9, 2 }, then after the call: h7.replace(x,3,1),
x will be { 10, 1, 4, 1, 9, 10, 1, 6, 9, 2 }.

3. Write method median() that takes an integer array and returns an integer. It should return the median value in the array. You may assume that the length of the array is odd. The original array should not be changed.

4. Write method concat() that takes two integer arrays and returns an integer array. The returned array should be the "concatenation" of the two arrays. The original arrays should not be affected.
Ex: if x = { 10, 3, 4 }, and y = { 1, 9, 10, 3, 6, 9, 2 },
then h7.concat(x,y) ==> { 10, 3, 4, 1, 9, 10, 3, 6, 9, 2 }

5. Write method rotateR() that takes an integer array and an integer *n* and returns an integer array. The returned array should be the same as the original, except that all values are "shifted" to the right by *n* positions. The original array should not be affected.
Ex: if x = { 10, 3, 4, 1, 9, 10, 3, 6, 9, 2 }, then h7.rotateR(x,3) ==> { 6, 9, 2, 10, 3, 4, 1, 9, 10, 3 }
Hint: This one isn't too bad if you use the % operator to help you out.

6. Write method mode() that takes an integer array and returns an integer. It should return the mode of the array (the value that appears the greatest number of times). If there is a tie, return the smallest mode The original array should not be changed. This one will take some good thought!

7. Write method unique() that takes an integer array and returns an integer array. The returned array should contain only the unique integers in the original array. That is, it returns an array without the duplicates. The original array should not be affected. Note that the returned array will in general be shorter than the array passed in. Hint: Use sort() (on a copy!), determine how big the new array will be and create it, and then load the unique values into the new array. The result will be the unique numbers sorted. This isn't easy.
Ex: if x = { 10, 3, 4, 1, 9, 10, 3, 6, 9, 2 }, then h7.unique(x) ==> { 1, 2, 3, 4, 6, 9, 10}

8.* Write method mySort() that takes an integer array and returns void. It should sort the array.
Of course, you can't use the sort built into Java!