CSF
Hwk08
Up, Up, and Array!

In this lab you will be writing methods involving arrays. Some helper methods have been written for you to use in your own methods and for testing. You may (and should!) use any of your methods or helper methods from lab 7!

1. Write method isMember() that takes an integer array and an integer and returns a boolean. It should return true if the integer is in the array, and false otherwise. The array should not be changed.
Ex: if x = { 10, 3, 4, 1, 9, 10, 3, 6, 9, 2 }, then h8.isMember(x,3) ==> true
h8.isMember(x,7) ==> false

2. Write method subArray() that takes an integer array and two integers (*start* and *end*) and returns an array. It should return an array which is the subArray of the original from the position start to the position end. If start>end, then you should return an empty array ({}). You can initially set the result = {}.
Ex: if x = { 10, 3, 4, 1, 9, 10, 3, 6, 9, 2 }, then h8.subArray(x,3,7) ==> { 1, 9, 10, 3, 6 }
h8.subArray(x,4,4) ==> { 9}, h8.subArray(x,0,1) ==> {10,3}

3. Write method union() that takes two integer arrays and returns an integer array. The returned array should be the "union" of the two arrays, if you think of the arrays as sets. That is, it is all the numbers that are in **either** of the two arrays. Make sure they are sorted when done. The original two arrays should not be affected. Hint) This one is very easy (two lines of code!) if you use two of the methods that you wrote in the last lab as helper methods! The final answer should be sorted.
Ex: if x = { 10, 3, 4 }, and y = { 1, 9, 3, 3, 9, 5, 2 }, then h8.union(x,y) ==> { 1, 2, 3, 4, 5, 9, 10 }

4. Write method intersection() that takes two integer arrays and returns an integer array. The returned array should be the "intersection" of the two arrays, if you think of the arrays as sets. That is, it is all the numbers that are in **both** of the two arrays. Hint: Use your member method as a helper. Make sure they are sorted when done. The original two arrays should not be affected.
Ex: if x = { 10, 3, 4 }, and y = { 1, 9, 3, 3, 9, 5, 2 },
then h8.intersection(x,y) ==> { 1, 2, 3, 4, 5, 9, 10 }

5*. Write method isPalSub() that takes and integer arrays and returns a boolean. It should determine if the array has a palindrome (of length 3 or greater) **anywhere** in the array. That is, is there any sublist greater than length two that is a palindrome? The original arrays should not be affected.
Ex: if x = { 1, 9, 3, 3, 9, 5, 2 }, then h8. isPalSub (x,) ==> true { 9,3,3,9 }
if x = { 1, 9, 3, 3, 4, 5, 2 }, then h8. isPalSub (x,) ==> false