CS 172 - Fall 2014
Lab 8
Searching and Sorting

Submit the program Lab8.java on Canvas.

Typical time to completion: 8 to 10 hours

Programs that do not compile will receive a grade of zero.

Comments: Header comments and inline comments are required.

Style: Your program must adhere to the conventions for naming variables, indentation, and spacing, etc.

Lab8.java

Write a Java program named **Lab8.java** that contains the following methods:

1. `public static boolean linearSearch (int [ ] array , int key)`

   Start searching at the beginning of the array. Compare one element at a time with the key. If key is found in the array, returns true. If it was not found, the method returns false.

2.. `public static int linearSearch2 (int [ ] array , int key)`

   Same as method 1 except that if key is found in the array, returns the position of the first occurrence. If not found, returns -1.

3. `public static boolean binarySearch (int [ ] array , int key)`

   Uses the binary search algorithm (see the handout). If key is found in the array, returns true. If it was not found, returns false. The array must be sorted or the binary search method will not work.

4. `public static void bubbleSort (int [] array)`

   Sort the array in non-descending order. Implement the following algorithm:

```
loop for stop = array.length - 2 down to 0
      loop for i = 0 to stop
         if array[i] > array[i+1], swap array[i] with array[i+1]
      end loop
end loop
```

5. `public static void printIntArray (int [ ] array)`

   Print the elements of the array, 10 elements per line with one space in between numbers. The numbers do not have to be aligned in columns.

6. `public static void selectionSort (int [] list)`

   Sort the array in non-descending order. Implement the **selection sort algorithm** using the pseudocode given in the "Chapter 8 – Arrays" handout.

7. `public static void insertionSort (int [] list)`

   Sort the array in non-descending order. Implement the **insertion sort algorithm** using the pseudocode given in the "Chapter 8 – Arrays" handout.

6. Write a main method (in the same class) to test the methods above. Your main method must do the following:

   a) Declare an array of 200 integers
   b) Fill the array with random numbers between 1 and 1000.
   c) Print the array using the printIntArray method.
   d) Use linearSearch to search for 30, 86, and 87.
   e) Use linearSearch2 to locate the first occurrence of 25, 80, and 91.
   f) Use bubble sort to sort the array.
   g) Print the array again.
   h) Use binarySearch to search for 11, 28, 74, and 99.
   i) Declare an 2nd array of 200 integers
   j) Fill the array with random numbers between 1 and 1000.
   k) Print the array.
   l) Sort the array using selectionSort.
   m) Print the array again.
   n) Declare an 2nd array of 200 integers
   o) Fill the array with random numbers between 1 and 1000.
   p) Print the array.
   q) Sort the array using insertionSort.
   r) Print the array again.

Compile and test the program to make sure that it works correctly.