CS 1302A – HW 7 (Spring 2016)
Total: 100 pts
Due: Wednesday, April 6 11:59 pm

This homework has 2 problems which deal with recursion (Ch 18)

Eclipse Reminder

- You will create a Java Project in Eclipse with the name: *hw7_FLastname*
- You will have two packages: *prob1, prob2*

Problem 1

You will write a number of recursive methods as described below. Make sure your method names, parameters, and returns types exactly match the descriptions as your code in *main* will be replaced for testing. Do the following:

a. Create a *prob1* package and a class named *RecursionTester*. Define the following static methods in *RecursionTester*

b. Consider this function: $power(x,n) = x^0 + x^2 + x^4 + \cdots + x^m$, where m is the largest even power that is less than or equal to n. For example: $power(3,5) = 3^0 + 3^2 + 3^4$. Note, this would give the same result as $power(3,4)$. Write a recursive static method to evaluate this function for given m and n.

c. Write a recursive static method named *reverseString* that accepts an integer and returns the integer as a string in reverse order (without using any String or Character methods). For example, 4295 would be returned as "5924".

d. Write a recursive static method named *countOccurrences* that accepts a string, *str* and a character, *key.* The method returns the number of occurrences of *key* in *str*. For example: countOccurrences("enzeme entene orf", 'e') returns 6. You are required to write an efficient method (*i.e.* you must use a helper method).

e. Write a recursive static method named *makePalindrome* that accepts a string and returns a palindrome of that string. For example, makePalindrome("abc") returns "abccba". You may use the substring function (*i.e.* you don't have to use a helper).

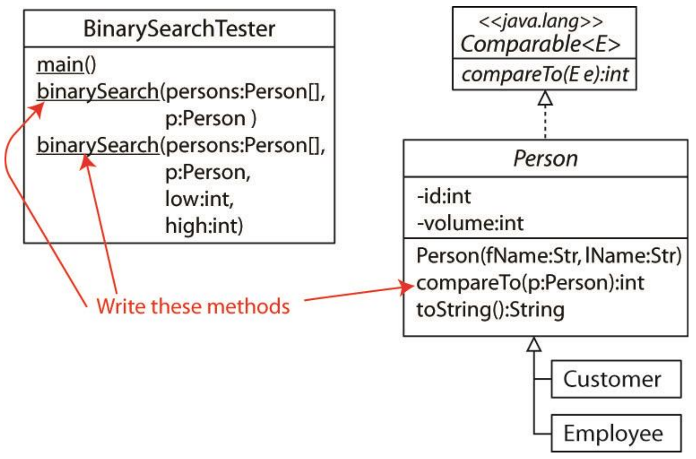
   Copy and paste the following test code into the main for testing:

```
System.out.println("Test 1a: power(): " + power(2, 3));
System.out.println("Test 1b: power(): " + power(3, 4));
System.out.println("Test 1c: power(): " + power(3, 5));
System.out.println("Test 2a: reverseString(): " + reverseString(6));
System.out.println("Test 2b: reverseString(): " + reverseString(29));
System.out.println("Test 2c: reverseString(): " + reverseString(1234567));
System.out.println("Test 3a: countOccurrences(): " + countOccurrences("z", 'z'));
System.out.println("Test 3b: countOccurrences(): " + countOccurrences("a", 'z'));
System.out.println("Test 3c: countOccurrences(): " + countOccurrences("zaz", 'z'));
System.out.println("Test 3d: countOccurrences(): " + countOccurrences("azazzaaza ab ", 'a'));
System.out.println("Test 4a: makePalindrome(): " + makePalindrome("a"));
System.out.println("Test 4b: makePalindrome(): " + makePalindrome("ab"));
System.out.println("Test 4c: makePalindrome(): " + makePalindrome("abc"));
```

Problem 2

Do the following:

a. Create a *prob2* package.
b. Download the file, *hw7_prob2.zip*, unzip, and drag the file, *BinarySearchTester* into the *prob2* package.



The code in *BinarySearchTester* contains the classes shown on the right. You will write the three methods indicated (method signatures are provided in the code). The code works in this way:

- *binarySearch* implements the binary search algorithm to search for a *Person* object in an array of *Person* objects.
- *compareTo* compares Person objects based on their last name, and then their first name. For instance, "Zack Black" would be first, followed by "Archie Dozier", and then followed by "Bob Dozier"

The *main* that is provided tests the code.

Submission

Make sure you zip the ENTIRE folder *hw7_FLastname*!! Submit your zip file to Blazeview by the due date. The name of your file should be: *hw7_FLastname.zip*.

Grading

I will RANDOMLY select one problem for grading. For example, if problem 1 is selected, I will only go to the package/folder *prob1* for grading. So make sure all your Java codes are in the correct packages/folders.

Additional Requirements:

1) No late submission will be accepted.
2) Please exactly follow the naming rules described above. You will be deducted 10 points for incorrect naming.
3) Write comments at the beginning of your Java source file(s) to indicate your name, student ID, "CS 1302-A Homework 7", and the due date.
4) Make sure that your programs are well-documented, readable, and user friendly. Make sure you document your programs by adding comments to your statements/blocks so that I can understand your code. You will

get 0 to 10 points based on the helpfulness of your comments. You will be deducted 10 points for no comments.

5) It is your responsibility to make sure your programs can compile and run correctly. Please be aware that programs that do not compile may receive ZERO credit.
6) When grades are returned to you via BlazeView, you have 7 days to meet with the instructor to discuss on grade changes. After 7 days, the grades are written in stone and can't be changed after that point.