Java Sequential Search - Practice Problems

All problems in this set should be completed in a project called Storage. Then, create a package called SequentialSearch. Each problem should get its own class and must be named starting with a letter. For example, we could name the second problem in level 1: level 1_2.

Level 1

1. Write a program that creates an array of comic characters {“Spongebob”, “Scooby-Doo”, “Perry the Platypus”, “Spiderman”, “Sully”, “Bugs Bunny”, “Mickey Mouse”}. Then ask the user for their favorite animated character and return the index of its location. If it is not in the list, display -1. Use the Library methods we created in class.
2. Write a program that creates an ArrayList of size 20 with random numbers from 1 to 100 stored in it. Then, ask the user for their favorite number and return that they are a winner if it is in the list and that they are a loser if it is not in the list. Use the library methods we created in class.

Level 2

1. Create a new library called Numbers. Write a method in the Numbers library called gcf. The gcf method should take two integers and return the greatest common factor of the two integers. Remember to test the function from a program to make sure that it works.
    a. In your Numbers library, write a function called lcm. The lcm method should take two integers and return the least common multiple of the two integers. Remember to test the function from a program to make sure that it works.
    b. In your Numbers library, overload a method called mean. The mean method should take a list (one for an array of ints, one for an array of doubles, one for a List of ints and one for a List of doubles) and should return the mean (average) of the list.
2. Create a new library called Points. Write a method in the Points library called slope. The slope method should take four integers (x1, y1, x2, y2) and should return the slope as a decimal. Test the method.
    a. In Points, make a distance method. The distance method should take four integers (x1, y1, x2, y2) and return the distance between the points.
    b. In Points, make a midpoint method. The midpoint should take four integers (x1, y1, x2, y2) and return a string representation of the midpoint of the two points.

Level 3

1. Write a program that generates an array of size 1000. Populate the array with random numbers between 0 and 1,000. Then, generate another random number to search for. Count the number of comparisons needed to find (or not find) the target. Loop the program to run 500 times and find the average number of comparisons needed to locate the target in the array. Ideally, the number should average out to about 500 -showing that the sequential search is linear or of the Big O order O(n).