COP 3330 Suggested Exercises for Week 10

StringTokenizer, ArrayList, ArrayDeque

1) Rewrite the following exercise from the 1D array section using an ArrayList:

Write a static method that takes in a reference to an ArrayList of Integers and returns the number of inversions in the list. An inversion in a list is an ordered pair (i, j) such that i < j but the element stored at index i is greater than the element stored at index j. Test your method on several arrays:

```
public static int numInversions(ArrayList<Integer> list);
```

2) Write a method that takes in a String which contains spaces and returns an ArrayList of the different tokens in the String. (This is identical to what the .split() method in Python does.)

```
public static ArrayList<String> tokenize(String line);
```

3) Write a shopping simulation with an ArrayDeque of Customer. Create your own Customer class (you can keep it relatively simple, maybe store a name and number of items bought), and process customers by adding them into the grocery line and servicing them at that line. Do this with a menu that allows the user to either add a customer to the back of the line or check out the customer in the front of the line. Use the appropriate methods (addLast, pollFirst, size) as needed. When you add to the back of the line, as the user to enter information for that customer. When a customer is checked out, print out which customer was checked out.

4) Rewrite the class example circuitmath.java to work for integer variables and the addition, subtraction and multiplication operators. Create your own test cases to improve your confidence that your implementation is correct. (Thus, instead of making – a unary operator, change it to be binary and do regular subtraction.)

HashSet

5) Use a HashSet to solve the following Kattis problem:

https://open.kattis.com/problems/cd

Note: to get an accepted status on this problem, you must use a BufferedReader object (and StringTokenizer) instead of a Scanner object. Research the syntax of BufferedReader and what you need to do get the appropriate code to compile. (Hint: One extra include and reporting that main could throw an Exception.)

6) Use a HashSet to solve the following Kattis problem:

https://open.kattis.com/problems/everywhere

7) Read in the number of votes for class president and then read in the names for the votes. Print out all of the unique names that received at least one vote in sorted order. (Hint: Store the names in a set, then copy the names from the set to a list, then sort the list.)

HashMap

8) Take the voting example above and map each unique person receiving a vote to the number of votes they received. Then, print out a summary of the results, sorted in alphabetical order.

9) Write a program that uses a HashMap to track the values of variables in a simple program. Every line of the program must conform to one of these patterns:

variable = value
variable = variable + variable
variable = variable - variable
variable = variable * variable
variable = variable / variable

To make life easy, make the first line of input the number of lines in the program and of course, assume that all variables on the RHS have assigned values already. At the end of the program, print out each variable and its corresponding value.

```
7
apple = 7
orange = 3
banana = apple * orange
kiwi = banana – orange
jackfruit = 8
melon = kiwi / jackfruit
```

10) Imagine storing railways that connect cities. Each city has direct lines to several other cities. We could store this data in a HashMap<String, ArrayList<String> >, where each city is mapped to a list of the cities its connected to directly by a railway. Create your own input format, then store the input in the manner described above and test that you've done this correctly.