**CS3330 LAB 1**

**Objective:**

* Using the String Object
* Practice creating more methods
* Use Arrays
* Use FileIO
* Java Documentation

**Purpose:**

Design and implement a class that reads a list of names from a file, counts the number of prefixes, suffixes, and substrings given a user inputted string. Use the sample code below to help create your project and class. The main method code is given to use.

**Downloads (use babbage to download the lab file):**

wget babbage.cs.missouri.edu/~mremtf/cs3330/downloads/lab1.zip

**You will need to change the Java Project to match what we want. If you fail to do so, that will result in 5 points off your lab score.**

**Expected project name:**

**<yourpawprint>.cs3330.lab1**

**Methods to implement:**

**public static String[] initArrayOfStrings(String filePath, int num)**

Allocates an array of Strings size of num, and reads the contents from a file located at the filePath. **Note: Don’t allow for buffer overflow of your array that will result in a zero. Don’t assume the file is the size of num.**

**public static void displayArrayOfStrings(String[] arrayOfStrings)**

Displays the contents of the passed array of strings. If the array is empty will not display contents.

**public static void displaySortedArrayOfStrings(String[] arrayOfStrings)**

Displays the contents of the passed array of strings in ascending order. Use can use the Arrays.sort(Object[] objArray) method or write your own sort.

**public static int oldSchoolSubstringOccurrences(String[] arrayOfStrings, String substring)**

Counts the number string in the array that contains the substring using the C way traversing an array. Counts the number string in the array that starts with the prefix except **charAt(int index) and length().**

**public static int oldSchoolPrefixOccurrences(String[] arrayOfStrings, String prefix)**

Counts the number string in the array that starts with the prefix using the C way traversing an array. **Cannot use Java built-in String functions except** **charAt(int index) and length().**

**public static int countSuffixOccurrences(String[] arrayOfStrings, String suffix)**

Counts the number string in the array that end with the suffix.

**public static int countPrefixOccurrences(String[] arrayOfStrings, String prefix)**

Counts the number string in the array that start with the prefix.

**public static int countSubstringOccurrences(String[] arrayOfStrings, String substring)**

Counts the number string in the array that contains the substring.

**Program Output**

**Suffix and Substring Test**

Names stored in Array of Strings

matt

johnathan

erik

sue

jonah

dan

george

dorothy

fred

north

Names stored in Array of Strings

dan

dorothy

erik

fred

george

johnathan

jonah

matt

north

sue

Enter a string to do comparison with: **an**

Manual substring search, found: 2

Manual prefix search, found: 0

Number of suffix found:2

**Prefix Test**

Names stored in Array of Strings

matt

johnathan

erik

sue

jonah

dan

george

dorothy

fred

north

Names stored in Array of Strings

dan

dorothy

erik

fred

george

johnathan

jonah

matt

north

sue

Enter a string to do comparison with: **jo**

Manual substring search, found: 2

Manual prefix search, found: 2

Number of suffix found:0

**GRADE GUIDE**

**30 possible points + 5 possible bonus points**

**If your program does not compile, produce any input/output (I/O) because most of the source code is commented out then your lab will receive a grade of zero points. If you lab has any runtime errors, the lab will receive zero points. If you don’t have your name on all your class files, you will receive a zero points.**

**Grading Rubric**

**5 points for Java Document comments**

**5 points for initArrayOfStrings and displayArrayOfStrings**

**5 points for oldSchoolSubstringOccurrences**

**5 points for oldSchoolPrefixOccurrences**

**5 points for countSuffixOccurrences and displaySortedArrayOfStrings**

**5 points for countPrefixOccurrences and countSubstringOccurrences**

**Bonus:**

Change all the string arrays to an ArrayList<String> Collection and update all affected logic and code. Output is the same as the lab output.