CIS 212 Assignment 5: 100 points

The goal of this assignment is to provide experience with reading data from a file, writing data into a file, sorting data using Selection Sort and Merge Sort, and evaluating the efficiency of the two sorting algorithms.

1. [10] Download the following file into the same directory as your Java program: https://classes.cs.uoregon.edu/17F/cis212/examples/phonebook_test.txt

Write your Java program to read all the data from the above file into an ArrayList of phonebook entries, where each entry consists of separate fields for the name and phone number (e.g., a class with private String variables would be considered good). The file is a plain text file, where every line has the following format: "PhoneNumber LastName, FirstName".

2. [20] Implement a method that takes an ArrayList of phonebook entries and a String as inputs, and output all the phonebook entries which contain this input String in either the first name or the last name into another plain text file named "Output.txt" in the same directory as your Java program. Every line of this file should follow the same format as above. For example, if the input String is "new", then the following should be the content of your "Output.txt":

```
754270500 Eh, Ainew
592030468 Fannew, Ouf
880526690 Wounoot, Onew
752360287 Bussok, Neinnew
865147504 Eipysh, Wainnew
213798053 Anouh, Annew
682280707 Cheifoh, Ainew
607750213 Zeivam, Onew
317288872 Faunt, Dinnew
```

If the input String is not contained by any name, then your "Output.txt" is an empty file.

3. [20] Implement a method that takes an ArrayList of phonebook entries as an argument and returns a sorted copy of the list using Selection Sort to sort alphabetically by phonebook last name. Your implementation must not modify the input list. You should implement your own sorting code here, and not simply use the Collections sort method. Cite any source used in the comments of your code.

4. [20] Implement a method that takes an ArrayList of phonebook entries as an argument and returns a sorted copy of the list using Merge Sort to sort alphabetically by phonebook last name. Your implementation must not modify the input list. Again, you should implement your own sorting code here, and not simply use the Collections sort method. Cite any source used in the comments of your code.

5. [10] Implement a method that takes an ArrayList of phonebook entries as an argument and returns true if the input list is sorted alphabetically by phonebook last name (false otherwise). Use this method to test your sorting implementations.

6. [10] Implement a method that invokes your own Selection Sort and Merge Sort codes, and report the elapsed *sorting* time in seconds, excluding the time of reading data from the file. Your output should look something like (but may not necessarily be):

```
Selection Sort: 27.094

Merge Sort: 0.433
```

7. [10] Write code that is clear and efficient. Specifically, your code should be indented with respect to code blocks, avoid unnecessarily repetitive code, avoid code that is commented out or otherwise unused, use descriptive and consistent class/method/variable names, etc.

Please zip only your Java source file(s), i.e., .java file(s), into a zipped file, rename that file as <Your Full Name>Assignment5.zip, e.g., BillGatesAssignment5.zip, and then upload that file to Canvas. Do not put the Java source file(s) in a folder and zip that folder; instead, please directly zip all the Java source files into a single zipped file. You are free to use whatever tools/IDEs you prefer to complete your assignment.