A. Objectives

1. Implement a sorting technique for Strings (10 points)
2. Implement a recursive binary search for Strings (10 points)
3. Use methods to do separate and repetitive work. The main method should not have more than 20 lines of code (10 points)
4. Implement the project correctly. (60 points)

10 points will be awarded for use of meaningful identifiers, consistent indentation, explanatory comments in your code and properly formatted output.

B. Description

We would like to implement a word correction program, similar to those available in email and cellphone clients. The program starts and the user can type a word and press enter; if the word has some meaning then the program reports “Correct”, else will show a word that could have been intended but resulted in something from mistyping; if the entered word does not resemble a valid word it reports “No suggestions”. For example, if the user typed “calss” or “clsas”, then the program should predict the correct word “class”; if the user typed “clssss”, then the program reports “No suggestions”.

Construct the program using the following guidelines:

1. Load a dictionary of accepted words into a data structure when the program starts (this can be done either by reading the words from a file, or have the user type in a list of valid words at the beginning of the program)
2. Create a separate method that takes input from the user in the form of Strings to simulate the word checking utility until the user enters “DONE”
3. Create a method that is called for each of the words entered, checks if the word is valid and reports “Correct”, “No suggestions” or suggests a valid word
4. To implement the word checking, sort the list of valid words using your own sorting method, and then search for the entered word on the sorted list using binary search
5. When searching through the list, check if entered word matches with or is an anagram of a valid word. If there is an exact match with a valid word, then report “Correct”; if the words compared are anagrams, then suggest the valid word, else continue searching. If there are no more words to compare with, then the program reports “No suggestions”.

C. Constraints

1. Using data structures from Java Collections is not allowed; implement your own data structures. If students find code somewhere else that they want to include in the program, they need to cite the reference as well.

2. The project is due by Wednesday, November 25, 2015, 11:59 PM PDT, using Blackboard.

3. This is not a group project. Copying code from others or using an unfair means is strictly not allowed and plagiarism charges will be imposed on students who do not follow this.