CS 141 – Introduction to Programming and Problem Solving, HW4

Note: This assignment is worth 7pts out of your course total.

If you've ever driven on California streets, you've probably noticed that California license plates always consist of a number, three letters, and then another three numbers.
For example: 5NPT839 4KDD232 7FTW142

On a long highway drive in California, one fun way to pass the time is to play the license plate game, in which you look at the license plate of the nearest car, and then try to find a word that uses the letters in that license plate in the order in which they appear.

For example, for the license plate 5NPT839, you might use the word INPUT.
For 4KDD232, you could use KIDDING.
For 7FTW142, you could use AFTERGLOW.

For a word to match a license plate, it has to contain all the letters in the license plate in the order in which they occur. This means, for instance, that FTW doesn't match WATERFALL because the letters are in the wrong order. Similarly, NNN doesn't match ENTIRE, since only one of the N's was matched. Certain letter combinations are a lot harder to find words for than others. The letter combination YRW only matches a handful of words (like EYEBROW) and some combinations don't match anything at all. There are no English words that match QQQ, for example. Your job in this part of the assignment is to write a program that prompts the user for a three-letter string, and then prints all English words that match that pattern. Your program should sit in a loop prompting the user to enter a three-letter string (which can be upper-case or lower-case), re-prompting as necessary until the user enters a string consisting of three letters. Then, your program should list all English words that match that string and the game restarts by prompting again for 3 letter word. Your program should exit if the user type "exit". We've provided you a file (wordsEn.txt) containing all words in a dictionary, which you'll need for this assignment. There are many ways you can implement this program. Although we haven't talked about efficiency this quarter, file reading is slower than most other program operations. For full credit, your program should only read in the dictionary file once. Other than that, don't worry about efficiency.

Submission: To be submitted as **FirstnameLastname-141-A4.zip** file on Blackboard (under Assignments). Include all the .java files used in your program in the zip file.

In addition, each .java file must contain the following information at the top:

//Your name
//CS141
//Assignment 4
//Date

Due: Wed. May 16, 4.00 PM