CSE1322 Assignment 2

Background:

Word guessing games are a very common childhood activity. Typically you are told how many letters are in the word, and then you guess letters. Letter by letter, you either see more of the word, or you run up against a deadline. Examples of these games are apple tree, hangman, and even the TV show Wheel of Fortune.

You have decided to go try out for Wheel of Fortune, but you want to put your new found coding skills to the test first. You are going to write a helper program, which will help you guess the correct words.

This program will ask you how many letters are in the word, then it'll ask you to give it some letters that you know, and it'll show you all the possible words of that length with those letters.

There are two different ways that you might want to use it. For example you might want to find all words which have the letters o and p in them in any order. "Top" being an example, "Pot" being another. But you might also want to be able to find words with a particular pattern in them, for example, all words with "pp" in them (such as happy).

In order to simplify this, we are going to use a dictionary of the 900 most common words in English. When we get to the FileIO Module in a few weeks, you could modify this to load in a full dictionary which would have 100,000+ words. Then it would be even more powerful, but even with the 900 most common words, it'll help a lot.

Your tasks:

1) In the zip file with these instructions, you also were given an English.java and an English.cs file. If you are a java student you only need English.java, if you are a C# student you only need English.cs.

2) Import the appropriate file into your project (either copy/paste, or import the file). You do not need to change anything in this class, so don't modify any code in there.

3) In your main class, write the following methods:
    a) wordContainsLetter. It should take in a word (String) and a letter (char) and it should return a boolean which is true if the word contains that letter, or false if not.

       There are many ways to do this, but the simplest is to use a foreach loop that iterates over the String as if it's an array of characters. Both C# and Java have a string method called toCharArray which converts a string to an array of characters. It may be helpful.
    b) wordContainsString. It should take in two strings, and determine if one string is in the other. For example if given "happy" and "pp" it should return true. If given "happy" and "ab" it should return false. It should only return true if the entire second string is in the first string. Both languages have methods for doing this. Check out the docs for String objects here and look for a useful method:

       Java:
       https://docs.oracle.com/javase/7/docs/api/java/lang/String.html

       C#:
       https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/strings

    c) guessWordWithLetters. It will take in an English object, a length and a string of letters. For example, you might be passed the English Object, 3 and "at". You'll then look for all words that are 3 letters long, that have an 'a' and a 't' in them and print them out (e.g. tax, art and eat). It will return void.

       What you will want to do here, is iterate over every word in the English object's words list/arraylist. As you look at each word, you'll decide if it's the right length, then you'll check each of the letters that were passed in to see if they are in the word. If it's the right length and all the letters are in there, print it out, otherwise move onto the next word.

    d) guessWordWithPattern. It will take in an English object, a length, and a pattern (string). For example you might be passed the English Object, 3 and ct. This will only print words which have c followed by a t in them. Again this method returns void.

4) Finally, you'll write a driver in your main class. It should do the following:
    a) Prompt the user to enter the length of the word they are looking for "How many letters are in the word?"
    b) Read in the response.
    c) Prompt the user with a menu asking if they want to search for letters, or by pattern. It should read:

        Do you want me to look for letters or a pattern?
        1. Letters
        2. Pattern
    d) Read in the user's response. If they choose 1, ask them for the letters "What letters are in the word?". Now call guessWordWithLetters, passing the appropriate arguments. If they choose 2, ask for the pattern "What pattern is in the word?". Now call guessWordWithPattern, passing the appropriate arguments.

        In either case print out "It might be any of these…"

Sample Output:

User input in red

How many letters are in the word?
3
Do you want me to look for letters or a pattern?
1. Letters
2. Pattern
1
What letters are in the word?
at
It might be any of these...
act
art
eat
tax

Second Run:

How many letters are in the word?
5
Do you want me to look for letters or a pattern?
1. Letters
2. Pattern

2
What pattern is in the word?
pp
It might be any of these...
apply
happy

Third Run:
How many letters are in the word?
3
Do you want me to look for letters or a pattern?
1. Letters
2. Pattern

1
What letters are in the word?
a
It might be any of these...
act
age
ago
air
all
and
any
arm
art
ask
bad
bar
can
car
day
eat

far
law
lay
man
may
pay
say
sea
tax
war
way

Submitting your answer:

Please follow the posted submission guidelines here:
https://ccse.kennesaw.edu/fye/submissionguidelines.php

Ensure you submit before the deadline listed on the lab schedule for CSE1322L here:
https://ccse.kennesaw.edu/fye/courseschedules.php

Rubric:

- Successfully imported the English.java or English.cs file into the project. [10 points]
- Method wordContainsLetter [15 points total]
    - Takes in a word and a letter [4 points]
    - Returns a bool/boolean [4 points]
    - Searches through the word and successfully returns true if the letter is in there. [7 points]
- Method wordContainsString [7 points total]
    - Takes in a word and a string [4 points]
    - Successfully returns true or false based on whether the string is in the word [3 points]
- Method guessWordWithLetters [22 points]
    - Takes in English object, length and string of letters [6 points]
    - Goes through every word in English object identifying ones which are the correct length [5 points]
    - Breaks up the letters string and searches for each one in the word [6 points]
    - Keeps track of whether each letter is in the word and only prints words that are the correct length, and also have all the letters [5 points]
- Method guessWordWithPattern [16 points]

    - Takes in English object, length and pattern [6 points]
    - Looks at each word in English object identifying ones which are correct length [5 points]
    - If the word has the pattern, print it, otherwise don't [5 points]
- Main Method [30 points]
    - Correctly instantiates an English object. [5 points]
    - Correctly asks the user for the number of letters, reads response [5 points]
    - Correctly presents the menu of choices to the user, reads their response [5 points]
    - Correctly asks the user for a pattern or letters, reads them into a string [5 points]
    - Calls the guessWordWithLetters when #1 is chosen [5 points]
    - Calls the guessWordWithPattern when #2 is chosen [5 points]