**CS203 HW10: Palindromes**

**Fall 2012**

**Due Date:** Wednesday, April 25 at 11 am

**Introduction:** This assignment will give you a chance to practice reading text from a file, writing text to files, and implementing a recursive method to determine if a word is a palindrome.

This assignment should be completed individually.

**Specification:**

1. (1 point) Complete the isPalindrome method. It should return true if a string is a palindrome (reads the same forward and backward) and false, otherwise. The pseudocode for this method is given in the starter file. This method should be written **recursively**.

Complete 2 – 7 in the main method:

2. (2 points) Open the file words.txt for reading and read the file line-by-line. Each word is on its own line in the file. For each word in the file, trim it, make it lowercase, and check to see if it is a palindrome. If it is a palindrome, write it on its own line to the file palindromes.html. The file palindromes.html should be an html file and have as the format:

<html>

<body>

<h1>List of Palindromes</h1>

Word1 <br />

Word2 <br />

Word3 <br />

…etc

</body>

</html>

Therefore, you should write the first three lines to palindromes.html first. Then, write each palindrome to the file. After you write the word, write <br /> to the file to get a newline in the html file. Then, write the last two lines to palindromes.html.

3. (.5 points) Print to the screen a message such as, “Finished finding palindromes and words starting with “a” and ending with “z”.” Print to the **screen** the total number of words in the words.txt file (update a counter as your read each word from the file).

4. (.5 points) Print to the **screen** the total number of palindromes in words.txt (update a counter as you find them).

5. (2 points) Choose some other property to implement regarding the words in the file. For example, you could find all words that start with a certain letter, find words longer than a certain number of characters, find words containing a certain substring, etc. Create another text file youChooseName.html to place the words that satisfy your property. This should also be an html file with formatting as in part (2). You should include this part in the code for part (2), so you read the file words.txt just once. Do something other than starts with ‘a’ and ends with ‘z’, as shown in Tammy’s solution.

6. (.5 points) Print to the screen the number of words that satisfy your chosen property.

7. (.5 points) **Close** all three files when finished and catches any necessary exceptions. If an exception occurs, the program should print an informative message about what error occurred and exit (System.exit(0);).

You should be able to open the files your program produces in a web browser to view the contents.

**Additional Enrichment:**

1. Add more file processing techniques (like you implemented for #4 above).

2. Reverse the characters of all the words in dictionary.txt and print the reversals to a file.

3. Find other text files containing data that interests you and write a program to process the text.

4. Sort words in the file (shortest to longest). You will likely want to use a Vector to store the words and just sort a subset of the words to save time.

**Logistics:**

1. Download the starter code from Moodle.

2. When you are finished, upload your BlueJ project, the text files you used/created, and your summary report as a single zip file named username\_HW10.zip.

**Grading Guidelines:**

Your program will be graded on a scale of 0 to 7 in two categories:

 Code Quality: design, whitespace, indentation, comments

 Code Operation: functionality of code with respect to specification

Your summary report will be graded on a scale of 0 to 6 based on:

 Correct use of vocabulary

 Completeness

 Clarity

 Organization

 Use of headings/proper formatting

**Report Guidelines and Format:**

***Introduction and System Use:***Describe the contents of the html files produced by your program.

***System Description:***

1. Describe your isPalindrome method. Give a call sequence diagram for the following:

 isPalindrome(“racecar”)

2. Include a short description of how your try/catch block(s) work and what happens if an exception is thrown.

***Testing and Evaluation:***

If your program meets all specification, write “Meets specification” here. If not, describe what is missing and/or not working as specified.

1. Copy/paste the results of your terminal window here.

2. What is the longest palindrome in words.txt that starts with the letter “d”?

3. Modify your code so that it instead opens the file words2.txt (do not include words2.txt as a file in the bluej folder). What does you program do? What does it print?

4. Evaluate the quality of your code. Is it easy to follow? Good whitespace? Proper indentation? Proper names?

***Conclusion:***

1. Since this is the final assignment of the semester, describe how you will apply what you learned in CS203 to your major / future career.

2. How long did you spend on this assignment?

3. By typing your name here, you are acknowledging that the code and report you are submitting are your own.

**Appendix:** Copy and paste code here (use Courier font so characters line up correctly)