SENG 513 - Assignment 2

Due date: February 11, 2017 at 23:59pm

Weight: 10% of the final grade.

Note: This is an **individual** assignment. Group work is **NOT** allowed.

Objectives:

The purpose of this assignment is to practice JavaScript programming.

Description:

Implement a function `getStats(txt)` which will compute information from text provided to it in the string parameter "txt". The function `getStats()` must return an object with a number of attributes. These attributes will correspond to the different types of information that your code will compute from the given text. Some of the attribute descriptions below include computations done on '**words**'. For this assignment we will define "**word**" to be a sequence of alphanumeric characters, converted to lowercase letters. For example the text "Hello, World!7x7" contains 3 words: "hello", "world" and "7x7".

Here is the list of required attributes and their descriptions:

`nChars: integer / worth 5 points`

Will contain the total number of characters in the text, including all white spaces.

`nWords: integer / worth 10 points`

Will contain the total number of words in the text. For example, "Hello, World 1!" contains three words: "Hello", "World" and "1".

`nLines: integer / worth 5 points`

Will contain the number of lines in the text. The only time this will be '0' is when the text is empty.

`nNonEmptyLines: integer / worth 10 points`

Will contain the number of lines in the text containing at least one visible character. We will define visible character as any character other than whitespace.

`maxLineLength: integer / worth 5 points`

Will contain the length of the longest line. Line length will be computed by counting the number of characters in the line, including any trailing white spaces, except newline '\n'.

`averageWordLength: float / worth 10 points`

Will contain the average word length in the text. Example: text "Hello, World 1!" would have average word length equal to (5+5+1)/3 = 3.666666.

`palindromes: array of strings / worth 15 points`

Will contain a list of unique palindromes in the text. Palindrome is a word with length > 2, which reads the same forward and backwards. Example: "Kayak, mom, MOM, XXx and 10z01 zz" contains 4

unique palindromes: [“kayak”, “mom”, “xxx”, “10x01”]. Palindromes should be reported in the same order they appear in the text.

`longestWords: array of strings / worth 20 points`

Will contain the 10 longest words in the text. In case of ties, the secondary sorting criteria should be alphabetical sorting. Example: “0, XXX, YYYY, AAAA, BBB” will yield a list: [“yyyy”, “aaaa”, ”bbbb”, “xxx”, ”0”].

`mostFrequentWords: array of strings / worth 20 points`

Will contain the 10 most frequent words in the text, concatenated with their respective frequencies. Use alphabetic sorting to to resolve frequency ties. The results will include the corresponding frequencies appended to the actual words surrounded by brackets. Example: the text “The,the,THE,and,AND,and,it,IT” should yield a list [“and(3)”, “the(3)”, “it(2)”].

Requirements:

A skeleton web page has been created for you to use. The skeleton page has a text field where you can enter arbitrary text, or even copy/paste prepared text into it. It has a button called “Compute Text”. This button will invoke a function getStats(), which you will have to write. The results of calling the function will be the automatically displayed below. The skeleton code can be found on github:

https://github.com/uofc-seng513w17/W17-assignment2

You **have to** use this skeleton for your assignment, and you have to put your entire implementation into a file called “`code.js`”. You cannot modify any other part of the skeleton. Your TAs may decide to automate the grading process, so it is important that your entire implementation is contained in the `code.js` file.

Additional requirements:

- Your assignment must be publicly viewable. That means you must host it on a publicly available web server. We discussed some possible hosting choices in the class.
- You may **not** use any external JavaScript libraries for this assignment.
- You may use any built-in methods & functions provided by your browser.
- Your implementation must work at least in Google Chrome **and also** in Mozilla Firefox browsers.

Submission:

Submit a ZIP or TAR archive of your assignment, containing 2 files only:

- readme.txt or readme.pdf, containing your name, ID and URL of your site;
- code.js, containing your JavaScript.

Warning: your code.js file will be compared to all other submissions from the entire class, using an automatic plagiarism detection system.

General information about all assignments:

1. **Due time** is listed at the top of the assignment**.** Late assignments or components of assignments will not be accepted for marking without your instructor’s approval for an extension beforehand. What you have submitted in D2L as of the due date is what will be marked.
2. **Extensions** may be granted for reasonable cases, but only by the course instructor, and only with the

receipt of the appropriate documentation (e.g., a doctor's note). Typical examples of reasonable cases for an extension include: illness or a death in the family. Cases where extensions will not be granted include situations that are typical of student life, such as having multiple due dates, work commitments, etc. Forgetting to hand in your assignment on time is not a valid reason for getting an extension.

3. After you submit your work to D2L, make sure that you check the content of your submission. It's your responsibility to do this, so make sure that your submit your assignment with enough time before it is due so that you can double-check your upload, and possibly re-upload the assignment.
4. All assignments should include contact information, including full name, student ID and tutorial section, at the very top of each file submitted.
5. Assignments must reflect **individual** work. Group work is **not allowed** for assignments, nor can you copy the work of others. For further information on plagiarism, cheating and other academic misconduct, check the information at this link: http://www.ucalgary.ca/pubs/calendar/current/k-5.html.
6. You can and should submit many times before the due date. D2L will simply overwrite previous submissions with newer ones. It's better to submit incomplete work for a chance of getting partial marks, than not to submit anything.
7. Only one file can be submitted per assignment. If you need to submit multiple files, you can put them into a single container. The container types supported will be ZIP and TAR. No other formats will be accepted.
8. Assignments will be marked by your TAs. If you have questions about assignment marking, contact your TA first. If you still have question after you have talked to your TA then you can contact your instructor.