ASSIGNMENT 1A

Assignment 1A tests your knowledge of Recursion (Chapter 18).

1. Design a driver class called ***YourNameAssignment1A*** (replace ***YourName*** with your actual name) with the following exact/precise methods (exact names, spelling, caps, parameters, returned values, functionality) in this order:

Method	Description
YourName*Method** (replace ***YourName with your actual name)	A recursive method that finds the number of occurrences of a specific letter in a word (any case - counting both lower and upper case of it). The method should receive the character for the *Letter* and the string with the *Word* as parameters and return the number of occurrences of the *Letter* in *Word*. For example, *YourName*Method('a', "cat") returns 1, *YourName*Method('a', "at") returns 1, *YourName*Method('a', "t") returns 0, *YourName*Method('a', "Azalea") returns 3, *YourName*Method('E', "bee") returns 2, *YourName*Method ('a', "CALAMATA") returns 4, *YourName*Method('a', "dog") returns 0.

YourName*Table** (replace ***YourName with your actual name)	A method that computes the number of occurrences of each `Letter` in the alphabet (from 'A' to 'Z') in a `Word` received as parameter, using the *YourName*Method above. The method should output "WORD=`Word`:" where `Word` is the parameter, an empty line and then the numbers in a table format (use `printf` to format your table in columns) with first column of 8 characters showing the letter and the second column of 6 characters showing the corresponding number of occurrences computed from the *YourName*Method `(Letter, Word)` call. The method should output an empty line after the table.
	For example, for the `Word` is `"cat"`, the output should be:
	`WORD=cat:`
	`Letter Number`
	`A 1`
	`B 0`
	`C 1`
	`D 0`
	…
	`T 1`
	…
	`Z 0`

main	The main method should call the ***YourName*Table** method 8 times for the words "*cat*", "*dog*", "STUDENT", "Dallas", "College", "*JAVA*", "*Programming*", and *"Supercalifragilisticexpialidocious"*[1]. The method should also output at the end your full name on a separate line.

2. Implement the class in NetBeans IDE and JAVA: Create a JAVA project called ***YourNameAssignment1A*** (same name as the driver class) and add your code to ***YourNameAssignmen1A*** project/driver class. Your program should **well-documented/commented** (have documenting comments for every line of code). You should not add other classes or methods.

3. Create the screenshot document for your code and output: Create a Microsoft Word document from the **YourName-Assignment1A.docx** template and call it ***YourName-Assignment1A*.docx** (replace ***YourName*** with your actual name) Add to the document your **complete screenshots** of the NetBeans editor window showing the **complete JAVA source code** for *YourNameAssignment1A* class and **complete screenshots** the **complete output**. If the entire class JAVA source code or the output does not fit in one screenshot or the screenshots cannot be easily read, create multiple screenshots and add multiple screenshots to the screenshot document. Please keep the screenshots in order (look at the line numbers). If your output is longer than a line and does not fit on one screen, Wrap Text in your output panel. See the instructions on how to use the template and how to create screenshots. Do not paste text or anything else instead of the screenshots.

4. Submit your work: Submit ***YourName-Assignment1A.docx*** on eCampus under the **Assignment 1A.** Do not archive the files (e.g. no RAR, ZIP, etc.) or submit other types/formats of files (e.g. no CLASS, PDF, JAVA, etc.). Please take time after submitting to review the file and check it is correct and resubmit it if there are any errors.

[1] Since the ***YourName*Table** outputs an empty line after each table, there should be an empty line between the 8 outputs.