MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG213 Data Structures – Programming Assignment 3

Date handed out: **Thursday 11 August 2016**
Date submission due: **Saturday 20 August 2016**

"Word Count!"

This assignment aims to help you practice binary tree abstract data types, specifically BST (Binary Search Tree) and AVL trees. You will write a program that processes a text file and counts the number of occurrences of words in that word file. You will implement this both with BST and AVL trees. The purpose of this assignment is to produce a file that lists these words in a file in an alphabetical order and also compares the performance of search with a BST and an AVL tree implementation.

Programming Requirements
Your program will work as a command line application. Your program needs to accept the name of the input file. A sample run would be as follows:

```
WordCount example.txt myoutput.txt
```

In this case, your program is called "WordCount" and it will process the text file called "example.txt". Once the processing is completed, the output will be written to another file called "myoutput.txt". For this assignment, you need to use agrc/argv for command line arguments.

You need to write two versions of this program: one that uses a BST and one that uses an AVL tree. Your program needs to open and process this file by building up a BST or AVL tree of words and counts as it progresses. Once all the processing is completed then write out the words in the tree in alphabetical order, one word per line, along with the number of occurrences of that word. Your program should ignore the case of the words, so that "apple" and "Apple" are considered the same. However, words that are actually spelled differently, such as "apple" and "apples" are considered to be different words. Your program needs to also ignore punctuations.
A sample output might look like the following:

25	The
12	quick
5	brown
6	fox
7	jumps
8	over
13	lazy
15	dog

You can find a text file attached to this program called "metuncccng.txt", you can use that file as a sample text file.

As part of this assignment, you also need to submit a **short report** that compares the search performances of BST and AVL implementations. In this short report, you need to show how your implementations perform when you search the word in far left child of your tree (first word in the alphabetical order), you search the word in far right child of your tree (last word in the alphabetical order), and eight other words that you randomly choose in your tree. In your report, you also need to state the worst-case search performance of BST and AVL tree, and also its generalization with Big-O notation.

Grading

Your program will be graded as follows:

Grading Point	**Mark (20 pts)**
Processing command line arguments (i.e., file names)	5 pts
Reading the text file word by word	10 pts
Creating a structure for each binary tree node that will store a word and also the count of that word	5 pts
Creating a structure for each AVL node that will store a word and also the list count of that word	5 pts
Creating the binary search tree for each word (need to have all the necessary functionality of the BST tree – insert, search, etc)	25 pts
Creating the AVL tree for each word (need to have all the necessary functionality of the AVL tree – insert, search, etc)	25 pts
Writing the text file	10 pts
Report (including search performance times for BST and AVL tree)	15 pts