CSCI 36200: Data Structures

Programming Assignment 2

Dimitrije Prosevski

Dr. Snehasis Mukhopadhyay

Due date: March 19, 2019

Contents

Project Description……………………………………………………………………………......2

Comments and Conclusion….………………….…………………………………………............3

# **Project Description**

The goal of the project was to use file stream and stream all the words from the text file into our program. Each word had to go through certain conditionals in order to be recognized as a word. There were 5 important assumptions about the file:

1. End of the file was when file stream encounters character “#”
2. Word must start with letters and must space, punctuation mark or new line.
3. Punctuation marks cannot be part of any word
4. Numbers of lines where the word got repeated are printed in ascending order
5. Each word had to be limited to the first 10 character. If first 10 characters of the words are the same, then the words are the same.

There were two files that had to be run in order to verify correctness of the program:

civilization of science.

science is knowledge.

knowledge is our destiny.

#

---------------------------------------------------------------------------------------------------------------------

set an integer variable x1 to be 2.

set y2 to be 2.

yes, y2 is a variable.

yes, y2 is an integer.

add y2 to x1.

print the result.

convert the result to an integer.

thank you.

#

There are two required data structures for the program to run correctly and efficiently: tree and linked list. Once the code is running, the program goes through each word and stores into lexicographic tree. Where the root node contains the middle-sized word, left node contains smaller and right larger word. The way to determine if the node is smaller or larger is determined by creating an if statement that compares the two strings (words) that returns a Boolean value. Linked list stores the numbers of the line number when each word is encountered where head pointer of the list points to the word itself in tree.

**Comments and Conclusion**

My program includes two headers and two cpp files. One for the binary search tree and one for the linked list. I started off my main by creating an object of the BST called myTree[2]. And I looped the entire program twice where I ran both files “test1.txt” and “test2.txt”. The program goes through the files individually and finds the word count, by braking the files by new lines first and then by braking each line by words. When words are separated each word is stored into temporary “word” variable conditionals are running to check if word counts as a word or is to be ignored (e.i. word starts with a number). The “word” is limited using substr() method, where the limit is 10 characters and then the word is stored into the node of a tree. Conditionals for the punctuation and numbers were done using if statements that check for the ASCII value of the character. If ASCII is matched, then condition passes and program either skips the word or removes the last character depending on the case. Line numbers were tracked in the while loop that loops through the lines. After each line variable “lineCount” is added to the linked list, as a line number, that is connected to the node of the word and “lineCount” get incremented by 1 after each line. Finally print functions are ran to print the tree and the linked list. And the output is streamed into the “output.txt” file.

**Comment:** I would prefer if you could use IDE to run the program (I used code blocks with C++11 compiler. When I try to output over Tesla the output looks slightly different. Thank you.

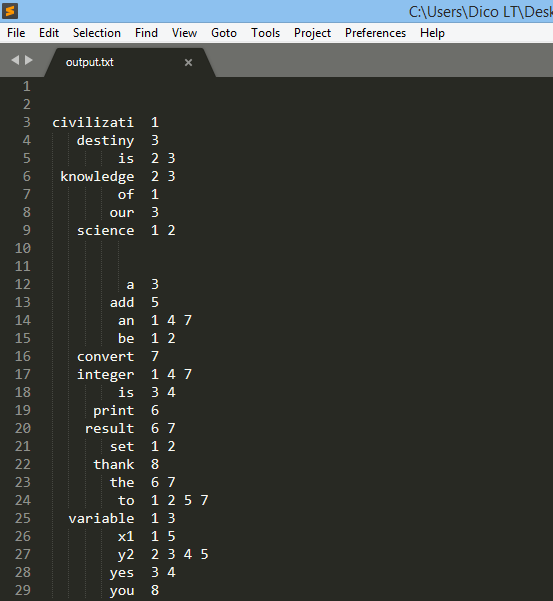
**To run the program:**

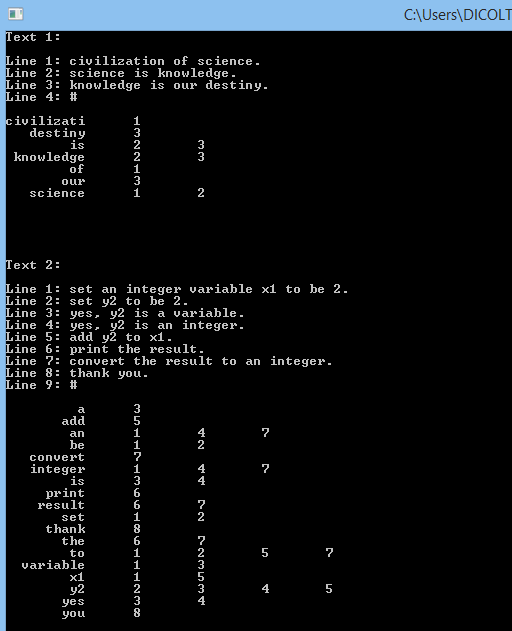
Go to my folder “**dprosev/362Data/Project2\_362**”

run “**make**”

then run “**. /RunAll**”

Output File:



Console output: