# COP 3502C
# Final term Assignment 2
# Total points: 13

**Introduction:** For this assignment you have to write a c program that will utilize the Binary Search Tree (BST).

**What should you submit?**
Write all the code in a single file and upload the .c file.

Please include the following lines in the beginning of your code. By writing this line you also agree that you have written the code:

/* COP 3502C Final term Assignment 2
This program is written by: Your Full Name */

**Compliance with Rules:** UCF Golden rules apply towards this assignment and submission. Assignment rules mentioned in the syllabus, are also applied in this submission.

Do not post or share part or full solution of your assignment anywhere!

**Additional notes:** The TAs and course instructor can call any students to explain their code for grading.

## Problem

In this assignment, you have to read a text file (in.txt) that contains a set of words. The first line of the file contains 3 numbers **(N, S, D).** These numbers represent the sequence of input words in your file.

**N:** represents the number of words to read to build binary search tree. You have to write a recursive insert code to create and insert these words into the binary search tree. After inserting all the items, you should show the words in Pre order, In order, and Post order. So, you need to create three functions for this purpose.

**S:** represents the number of the words to search from the tree. These S words are placed after the first N words in the input file.You need to implement a search function that will be able to search these words in your BST.

Additionally, the search words will also be used to look in the binary search tree and count number of words in the tree that come before the search word, alphabetically. You will need to create a recursive function with the following prototype ***CountBefore(treeNode* root, char searchKey[]),*** where treeNode is the node structure of your tree.

**D:** represents the number of words to be deleted from the BST. This list of words are placed after N+S words in the input file. Write a recursive delete function for your task. After deleting all the items, also show the tree in only in-order traversal.

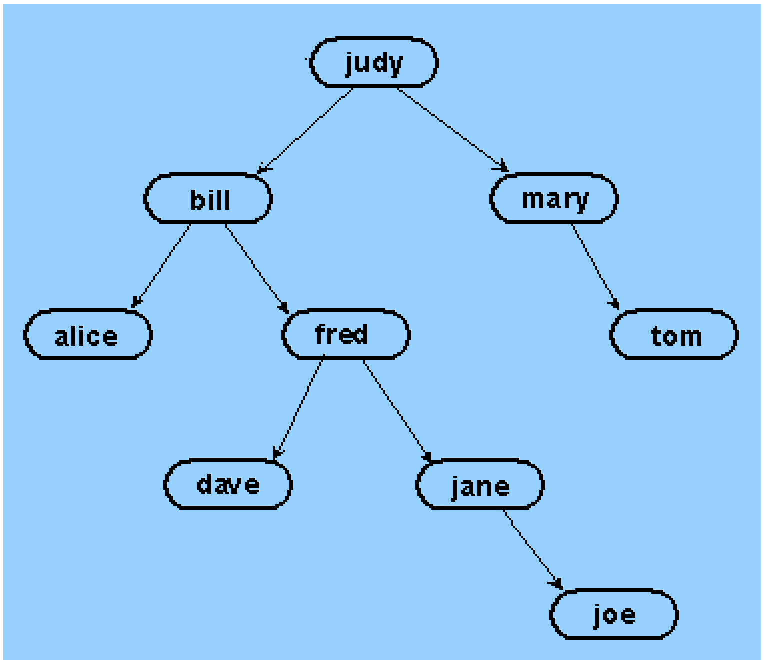
**Other required functions:** In addition to the general functions for the above tasks, your code should also contain the following functions.

**<u>totalCharacters(*treeNode* root*):</u>** This function takes the root of the tree in parameter and returns the total number of characters in the whole Binary Search Tree. You need to call this function after creating the tree and also at the end of deleting the requested nodes from the tree.

**<u>isBalanced(*treeNode* root*):</u>** We call the tree as imbalanced if the differences between the height of left subtree and height of right subtree is more than one. This function takes the root of the tree in parameter and returns 1 if the tree is balanced, otherwise it returns 0. In this process, you might consider implementing a height function to get the height of a subtree. The isBalanced function should be called after creating the tree and also at the end of deleting the requested nodes from the tree.

## Example:

The figure below shows an example of BST with strings.



All the output must be written to the out.txt file according the sample output shown bellow. You may output in the console, however, we will mainly see the out.txt file for our given input file.Do not hardcode your code only for the example provided below. If we see you have hard-coded any output, you will get zero for your full assignment submission.

# Example

**ample Input file in.txt:**

```
9 3 4
judy
mary
bill
alice
tom
fred
jane
joe
dave
jane
jones
judy
alice
mary
judy
fred
```

**Sample Output file out.txt:**

```
Pre Order: judy bill alice fred dave jane joe mary tom
In Order: alice bill dave fred jane joe judy mary tom
Post Order: alice dave joe jane fred bill tom mary judy
Total characters: 35
Height left= 3, height right = 1. The tree is imbalanced!
Search Phase:
jane: Found, Items before: 4
jones: Not found, Items before: 0
judy: Found, Items before: 6
Delete Phase:
alice: deleted
mary: deleted
judy: deleted
fred: deleted
In Order: bill dave jane joe tom
```

**Requirement:**

You must use file IO, Binary Search Tree and relevant traversal for your solution. All the information must be extracted from the tree.

Your program must have to compile in EUSTIS to get credit.

## Rubric:

1) Implementing insertion: 3 (-0.33 for each test case. 9 test cases)
2) In order, postorder, and pre-order: 1 (-0. 5 for in-order, 0.25 for post, 0.25 for pre-order)
3) Implementing total char function and displaying results: 1 point
4) Implementing isbalance function and displaying the heights of the left and right sub-trees and balance status: 1 point
5) Implementing search: 1.5(-0.5 each test case. 3 test cases)
6) Implementing countbefore function: 1.5 (-0.50 each test case 3 test cases)
7) Implementing Delete: 4 (1 for each test case. 4 test cases)

**See hint bellow for comparing strings.**

## Hints: use the strcmp() function of string.h to compare strings. Use strlen() function to get the length of a string.