Computer Science 1 - Program 5
Individual Project: Database Queries
Assigned: 3/31/04 (Wednesday)
Due: 4/16/04 (Friday) at 11:55pm (WebCT time)

Note: This version of the assignment is ONLY applicable for students who do not have a partner for the project. In particular, each lab section can have at most one student working alone. The TA's have a list of which students these are.

Binary Tree

The functionality you have to provide is as follows:

1) Inserting a node into an existing tree. (Note that if a "word" is already stored in the binary tree, then rather than a new node being created, the integer counter for the existing node should be incremented.) Also, note that even though a string passed into the insert function may be in mixed case, you should store the string in the tree in **all lower case.**

2) Deleting a node from an existing tree. (Note that if a "word" has a count of more than 1, then this should not delete a physical node in the tree. Rather it should simply decrement the count of the appropriate node it the tree. If the count of a "word" is 1, then the node should be deleted.)

3) Searching for a node in an existing tree. For this function, an integer should be returned, representing the number of times the word being searched for appeared in the summary. (Clearly if no corresponding node is found in the tree, 0 should be returned, otherwise, the appropriate count should be returned.)

Testing your program

Since you do not have another student working with you, you need to write a reasonable "testing" program for your code. Your testing program should allow the user to create multiple trees, and allow them to insert, delete and search for words from the tree. (Thus, each time the user inserts a value, you need to FIRST ask them which tree they want to insert it into.) When searching, your program should reply with ***the number of times*** the word being searched for is in the tree. You may design this portion of the program as you wish, but make sure that you leave ample directions for the TA in a separate file called "readme.txt" or "readme.doc" so that they can properly test and grade your program.

All other information for the assignment (such as what to include when you turn it in) can be found on the actual program writeup.