**Aaron Williams**

**Assignment 3 (Homework Unit 4)**

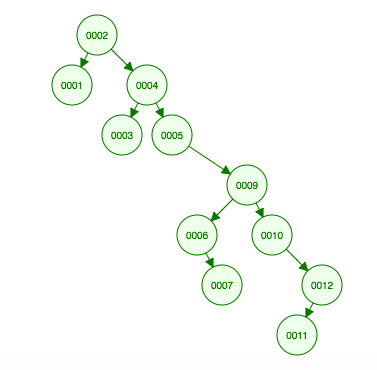
**10 September 2017**

**Assignment Description**

The purpose of this assignment is to begin with source code for a binary search tree (tree.cpp) and complete functions for inserting a node, deleting a node, and searching for a node while displaying relevant outputs as functions were built out.

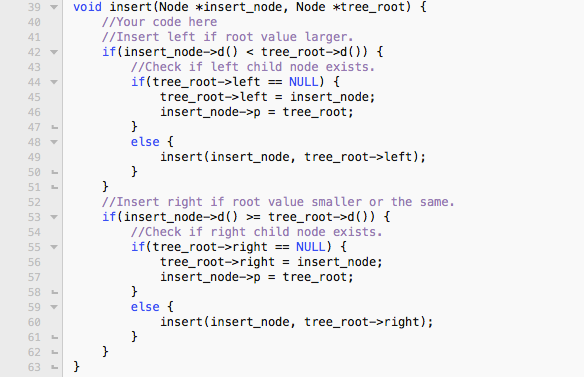
**Logic and Outputs**

1) The first task was to implement the insert function and then display the results of inserting 2, 1, 4, 5, 9, 3, 6, 7, 10, 12, and 11 into the empty binary search tree. In theory, the binary tree should appear as the following:



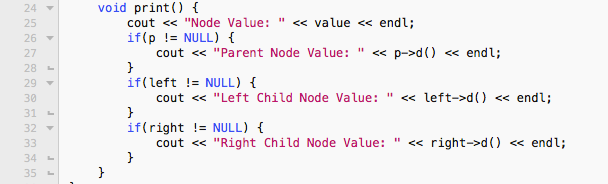
The insert method function was created to check if the node to be inserted was less than or greater than the associated root/parent node. Following the principles of a binary tree, if the node value was less than the root/parent node, it was inserted to an appropriate position on the left. If the node value was more than or equal to the root/parent node, it was inserted to an appropriate position on the right.

Insert Function:

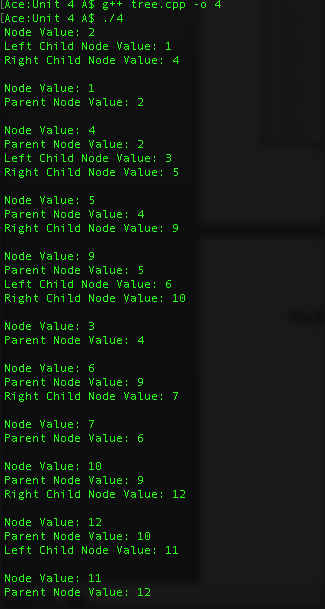


The nodes were created in the main function and inserted after their creation. Once the insertions were completed, all nodes were printed. In order to display the properties of each node, the print function was edited and implemented.

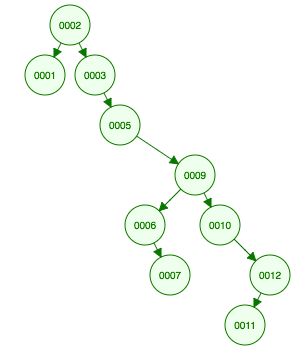
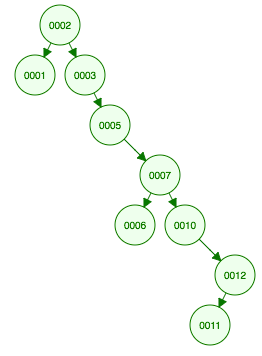
Updated Print Function:



Current Main Output:

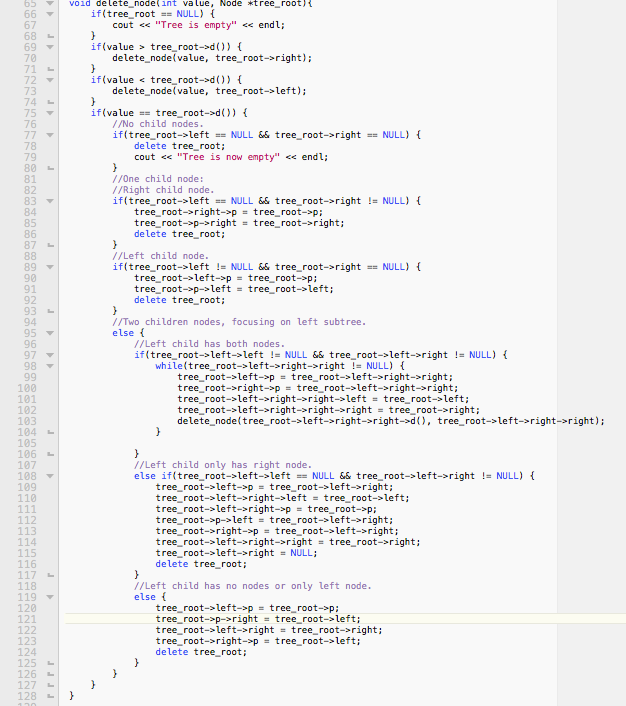


2) The next task was to implement the delete\_node function and show the results of deleting 4 then 9 from the binary search tree used in part 1. In theory, the binary tree should appear as the following (left picture shows after deletion of 4 and right picture shows after deletion of 9):

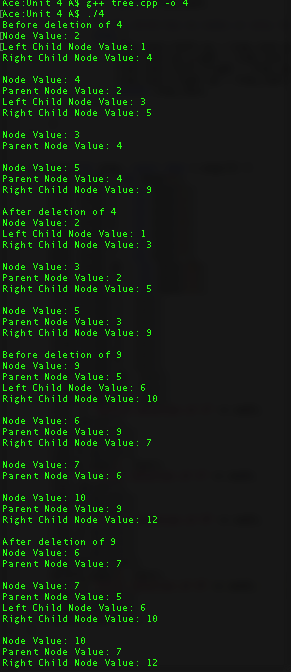


The delete\_node function was created to check the value parameter and determine if it was smaller, larger or equal to the tree\_root node parameter value. The function was set up recursively to call itself until the desired node to be deleted was found. In order to determine how to replace the deleted nodes, conditional statements were used to determine the structure of the nodes left subtree. Once the structure was determined for the subtree, relevant pointers were adjusted and the node was deleted.

Delete\_node Function:

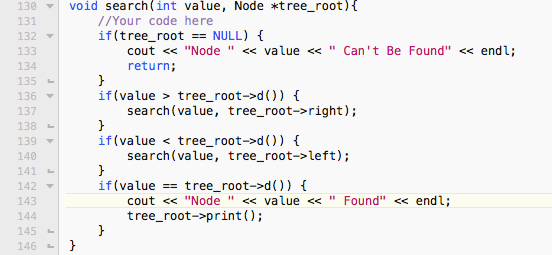


Current Main Output:



3) The final task was to implement the search function and display the results of searching for 12 and then 4 from the binary search tree resulting from part 2. In theory, the node with value 12 should be found, but 4 should not. The search function determines if the value is larger than the tree\_root node, and calls itself recursively until the node with the input value is found or can’t be found.

Search Function:



Current Main Output:

