PFII Lab 12: Binary Trees

In this lab I learned about the creation of Binary Search Trees using nodes and manipulated them for inserting an element as well as printing the sorted tree. Something very interesting that I observed in this lab was the similarities between Binary Tree links and Doubly Linked List links as BST uses **left** and **right** while DLL uses **prev** and **next**. Another observation of mine that related to memory management using BST is the fact that you cannot traverse upward on the BST, rather only being able to go down in the left or right direction.

When testing in the main, I had to **initialize the node for the BST as NULL** rather than just declaring it when there are 2 or more trees in a single main(). The focal concept of this lab was Binary Search Tree organization on how the data is sorted from left to right traveling down the tree from the root to the leaves. This lab gave practice with previous concepts like the arrow operator (->) was done, but the applications and content were the new Binary Trees.

Program Output:

9 10 11 13 15 20

0x1edcc40

4 6 11 12 13 14 19 21 68 101 102 369

0