**Name:**

**Advanced Programming in C++**

**Lab Exercise 5/13/2020**

**Application of Trees**

In this exercise you will explore several aspects of trees. For questions 4 – 6, use the main.cpp file found in the following location:

https://www.github.com/nmessa

1. Define the following terms:
   1. root
   2. child
   3. leaf
   4. parent
   5. depth
2. What is the smallest number of levels required to store 100,000 nodes in a binary tree?
3. What is the smallest and largest possible number of leaves in a binary tree containing exactly six non-leaf nodes?
4. Using the IntBinaryTree class that you worked on previously, write a function

bool IntBinaryTree::isLeaf(TreeNode \*tn)

{

if (tn->left == NULL && tn -> right == NULL)

return true;

else

return false;

}

that returns true if node is a leaf. This function should be called by the private displayInOrder, displayPreOrder, and displayPostOrder functions.

1. Using the IntBinaryTree class that you worked on previously, write a function:

int IntBinaryTree::sumTree(TreeNode \* tn)

{

if (tn != NULL)

return tn->value + sumTree(tn->left) + sumTree(tn->right);

else

return 0;

}

that returns the sum of the values contained in the nodes.

1. Using the IntBinaryTree class that you worked on previously, write a function:

int IntBinaryTree::treeCount(TreeNode \*tn)

{

if (tn != NULL)

return 1 + treeCount(tn->left) + treeCount(tn->right);

else

return 0;

}

that returns the number of nodes found in the tree.

When you have completed these functions, run main to make sure it works turn a screenshot of your program output.