In order to form a balanced search tree from an array, we need to take the following steps:

1. Get the median value of the array

2. Insert that initial value as the “root”, or parent node of the tree

3. Use recursion to divide the array in half and insert the values to the left of the median (the values lower than the median) on the left subtree of the root

4. Use recursion to do the same on the right side of the median, creating the right subtree

This method ensures that the tree is a binary tree, whose requirement is for each node to have a maximum of 2 children. Because the array is already sorted, the tree is a binary search tree because each node on the left side of the root are less than their ancestor nodes and each node on the right side of the tree are greater than their ancestor nodes.

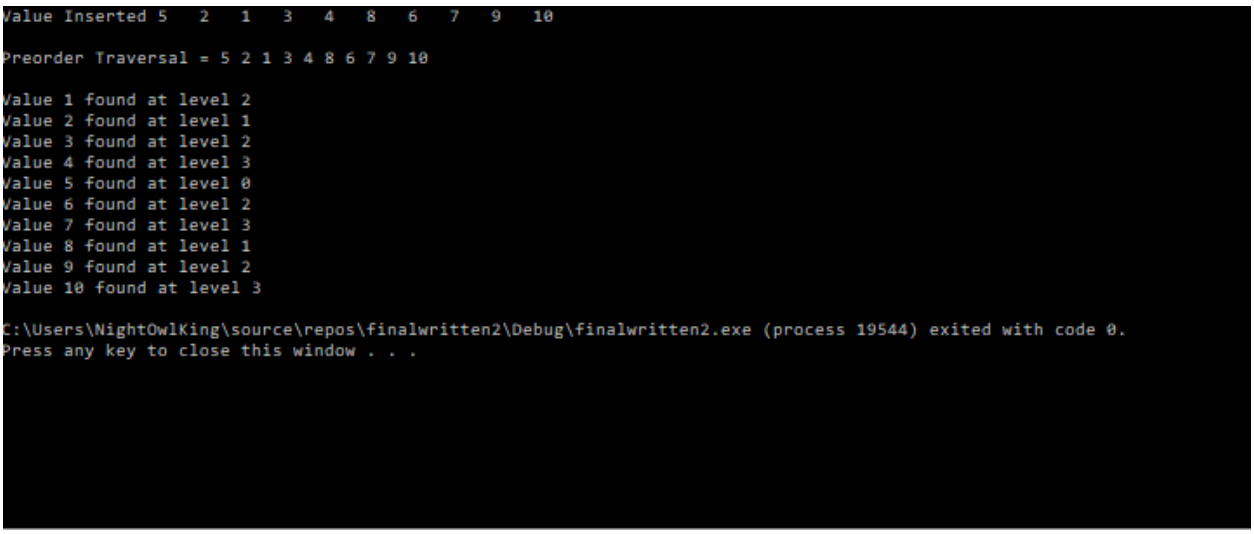
The findLevel value works by the following steps:

1. Look at the root value to see if the search value is identical.

2. If the value is not found, go down a level on the left subtree, and keep doing that recursively until you find the value.

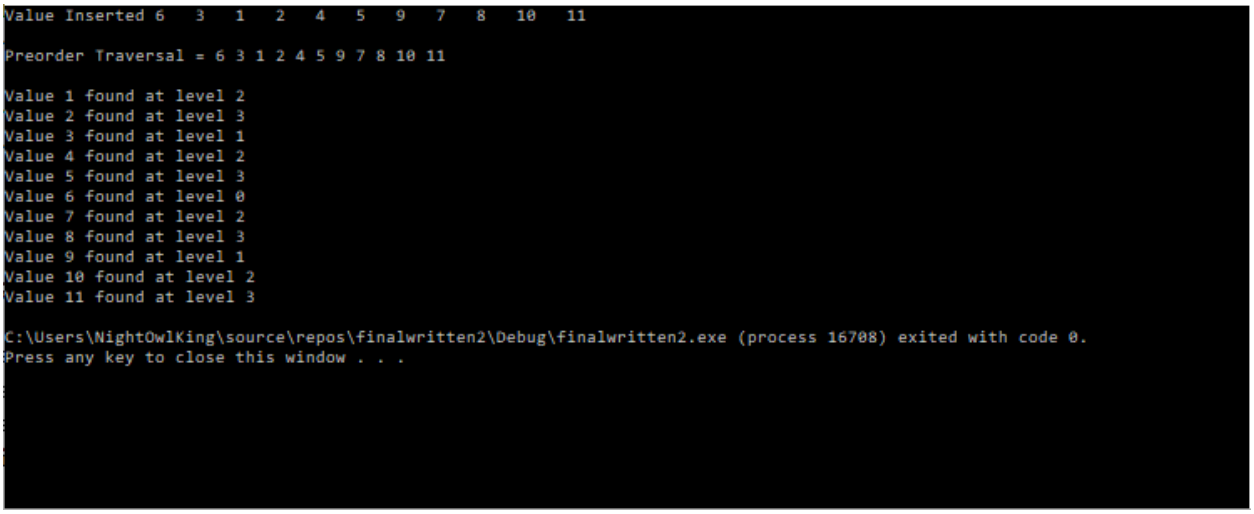
3. If the value is not found on the left subtree, go down the right subtree until you find the value.

4. If the value is not found, return an error

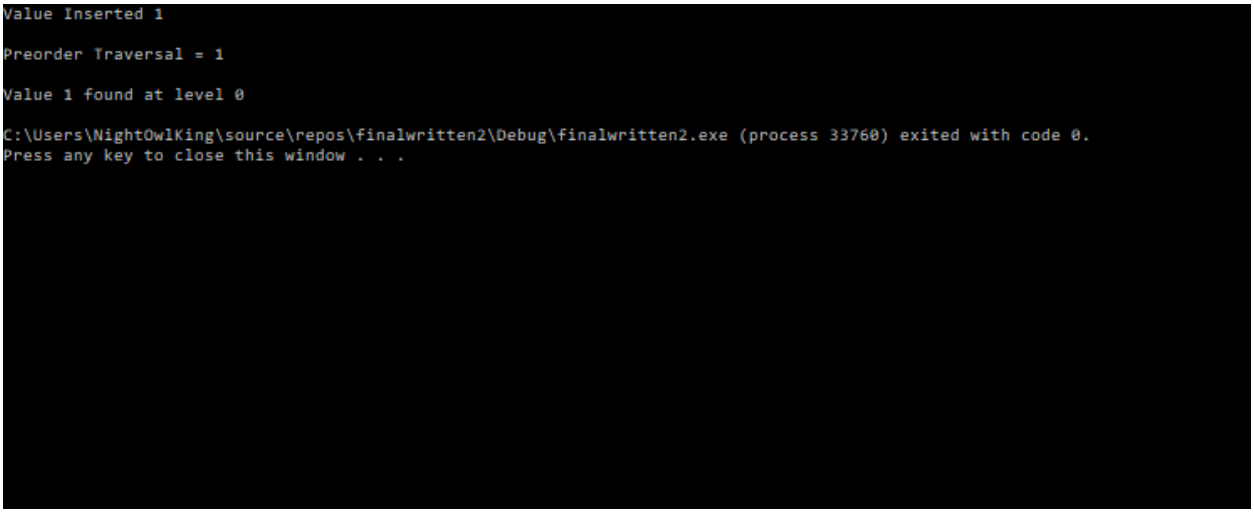


This is an example of the program with the input array of { 1,2,3,4,5,6,7,8,9,10 }. The order of the values inserted and the preorder traversal are shown in order to confirm that the values were inserted correctly. As shown in the program, the values are identical, meaning that the program worked correctly. Furthermore, the FindLevel function iterates through the array, displaying the levels the

values are in. The left and right subtrees are no more than 1 level from each other, meaning that the tree is a balanced binary search tree



This is an example of the program with the input array of { 1,2,3,4,5,6,7,8,9,10,11 }. As with the first sample array, the order of input matches the preorder traversal. Furthermore, the levels of the left and right subtrees are no more than 1 from each other, making it a balanced binary search tree.



This is an example of the program with the input array of { 1 }. This tree has only a single value. Nevertheless, both the convertarraytoBST() function and FindLevel functions work on it.