I didn't have enough time to create a visual representation of the tree to show that the code creates a balanced tree. To compensate, I'll explain how it performs the creation while remaining balanced. Simply put, the inputted values are sorted once all values have been collected. Once sorted, the first node within the tree is the medium value of the array SIZE. This means that the values on the left and right are ensured to be the same height/depth on each side. The values are then put into the tree with two different loops, left of the medium element and right of the medium element (Load left values, load right values).

Once the tree has been created, then the program goes through a loop find only the left children of each node, once the last node is found and left child of that is NIL, then the node's value is displayed.