Starting the program, I defined a check to see if it is a binary tree. I then defined predicates for the empty binary tree, and the non-empty binary tree. For the non-empty binary tree, I implemented comparisons that would ensure that the elements of the left subtree are less than/equal to the root node, and that the elements of the right subtree are greater than the root node. I then moved on to implement the preorder traversal. To do so, start by visiting the root node, then visiting each node of the left subtree, and then each node of the right subtree. As the tree is traversed, every node that is visited gets appended to a list L. I then moved on to implementing the inorder traversal. To do so, start by visiting each node of the left subtree, then the root node, then each node of the right subtree, and append each node to a list L. I then implemented the postorder traversal by visiting the left subtree, then the right subtree, then the root node, appending each visited node to a list L as the tree is traversed.