```python
"""
CS241 Homework 08
Written by Chad Macbeth
"""

class TreeNode:
    """
    This class holds a basic binary tree node
    For this assignment, you should not need to change the class.
    """
    def __init__(self, data=""):
        self.data = data
        self.left = None
        self.right = None

def construct_tree():
    """
    This function constructs a simple balanced binary search tree.
    Normally, we would add functions to our tree to be able to automatically insert
    the values into the correct places, but for this assignment, this
    function will simply construct it manually.
    :return:
    """

    # You should not need to change anything here
    root = TreeNode("50")

    root.left = TreeNode("26")

    root.left.left = TreeNode("12")
    root.left.left.left = TreeNode("8")
    root.left.left.right = TreeNode("16")

    root.left.right = TreeNode("43")
    root.left.right.left = TreeNode("34")
    root.left.right.right = TreeNode("46")

    root.right = TreeNode("83")

    root.right.left = TreeNode("59")
    root.right.left.left = TreeNode("56")
    root.right.left.right = TreeNode("72")

    root.right.right = TreeNode("93")
    root.right.right.left = TreeNode("91")
    root.right.right.right = TreeNode("99")


    return root

def print_tree(node):
    """
    This functions should use RECURSION to print out all the nodes of the tree IN ORDER.
    :param node:
    :return:
    """

    # TODO: Put your code here
    # The stopping case is when node is None.  Otherwise, we traverse
    # to the left, then print the current, then traverse to the right
    if node != None:
        print_tree(node.left)
        print(node.data)
        print_tree(node.right)


```

```
def main():
    """
    Call functions to construct a tree and print it.
    :return:
    """

    # You should not change anything here.
    root = construct_tree()
    print_tree(root)

if __name__ == "__main__":
    main()

```