**Explanation:**

## BST:

A **BST** is a tree data structure where:

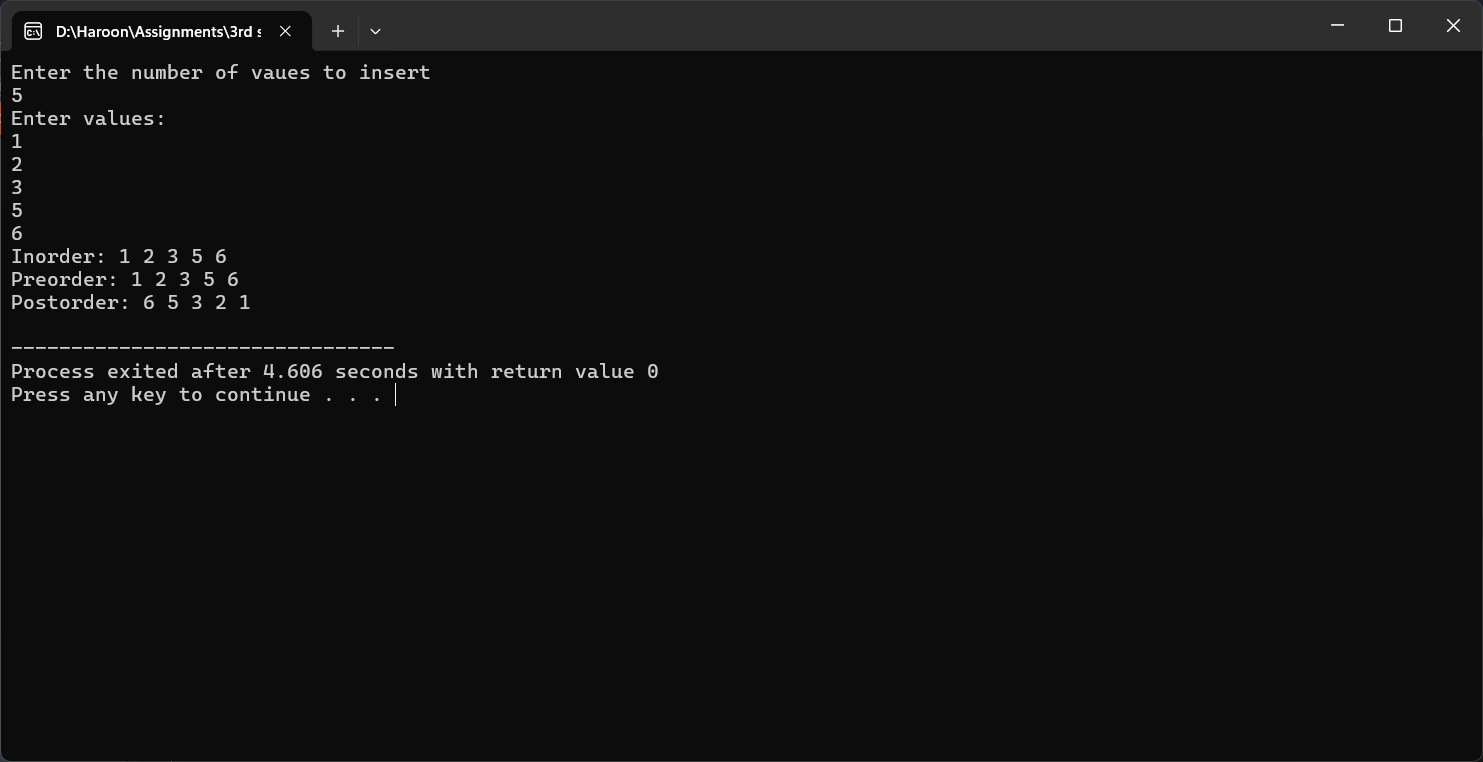
Each node has a value.

Left child contains values **less than** the node.

Right child contains values **greater than** the node.

This makes searching, inserting, and traversing fast — ideally in O(log⁡n)O(\log n)O(logn) time.

**OutPut:**



**AVL:**

An **AVL Tree** is a self-balancing Binary Search Tree (BST), where the difference in height (balance factor) between the left and right subtree of any node is at most 1. This keeps the tree balanced and ensures faster insertions, deletions, and lookups (in O(log⁡n)O(\log n)O(logn) time).

**Output:**