**Name :**

**Hafsa Waseem**

**Roll no:**

**SU92-BSSEM-S24-014**

**Subject :**

**DSA (Lab)**

**Section :**

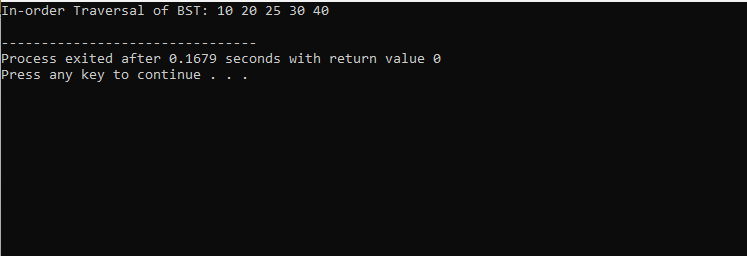
**3A**

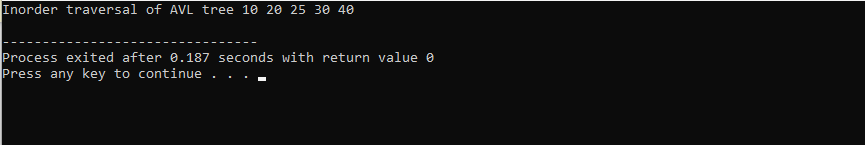
**Submitted to:**

**Sir Rasikh**

**Task no 12:**

**1.insert and traverse for BST**

**2.insert and traverse for AVL** ****

****

**Explain :**

The code shows how to create and work with a binary search tree (BST) and an AVL tree in C++. In a BST, values are inserted so that smaller ones go left and larger ones go right, keeping the tree sorted. In-order traversal prints the values in order. The AVL Tree is a self-balancing version of BST. After each insertion, it checks if the tree is unbalanced and fixes it using rotations. This keeps the AVL tree more balanced and efficient than a regular BST.

**Top of Form**

**Bottom of Form**