Implementation of AVL tree:-

Creating a node of type structure that has two integer values and two node pointers of type structure. Creating a node of type structure pointer and initializing its values, finally returning the node pointer.

Following are the functions used in the program:-

1. max(): This function is used to find the maximum of 2 integers. It returns the maximum value.
2. height(): This function is used to get the height of the AVL tree. It recursively calls the left and right child nodes to get the maximum height.
3. rotate\_right(): This function rotates the AVL tree in the right direction to balance the height of the tree.
4. rotate\_left(): This function rotates the AVL tree in the left direction to balance the height of the tree.
5. balance(): This function returns the balance factor of the node for which this function call is made.
6. insert\_node(): This function inserts a new node into the AVL tree. If the new node violated the balancing condition then it will handle the cases and rotate the tree as required to balance the height.
7. min\_node(): This function finds the node with the minimum value and returns the node.
8. delete\_node(): This function deletes a node from the AVL tree by the value provided by the user if the node is present else simply returns. After deletion operation, it again balances the height of the AVL tree.
9. preorder(): Function to print the preorder traversal of the AVL tree.
10. inorder(): Function to print the inorder traversal of the AVL tree.
11. postorder(): Function to print the postorder traversal of the AVL tree.
12. printLevel(): Function to print the data item present in the level for which it is called.
13. height\_l(): Function to get the height of the AVL tree for level order traversal.
14. levelorder(): Function to print the levelorder traversal of the AVL tree.
15. sumOfLeftSubtree(): Function to find the sum of all nodes in the left subtree of the root node.
16. main(): The program execution starts from the main() function itself. As per the choice of the user can do the various operations and traversals for the AVL tree.

OUTPUT:





