**Question 1: Implement Height balanced binary tree(AVL tree) in the following way(all nodes in the tree should be unique)** **provide an insert function to insert the node in the tree  
Provide a delete function to delete the node from the tree(by value)  
Provide a function to display pre-order,in-order,post-order  
Level,level-order tree traversal  
Provide a function to calculate the sum of all nodes in the left subtree of the root node.**

* The problem is solved by making use of arrays
* Program is written on C with these basic functions.
* (i) **max(int x, int y):** returns maximum of two input integers.
* (ii) **height(struct node\* root)**: this function takes input as the root the tree and returns the height of the tree from that root node.
* (iii) **rotate\_right(struct node\* t1)**: rotates my avl tree in clockwise direction.
* (iv) **rotate\_left(struct node\* t1):** rotates my avl tree in anti-clockwise direction.
* (v) **insert\_node(struct node\* root,int item):** the function that takes the root node of the current tree and some data item as arguments and inserts the data item into my tree while maintianing the AVL tree properties.
* (vi) **delete\_node(struct node\* root, int item):** the function that takes the root node of the current tree and some data item as arguments and deletes the data item from my tree while maintianing the AVL tree properties.
* (vii) **inorder/postorder/preorder(struct node \*root):** These three functions are designed to print the inorder / postorder / preorder traversal of my tree starting from the root node passed as the parameter to the function.
* (viii) **sumTree(struct node \*root):** This function prints my tree starting from the root node passed as the argument to the function.
* (ix) **main()**: main driver function that helps the user make use of all the functionalities provided by the program.
* The input is checked against any exception, by using if and else to check for expected input.  
  If unexpected input is received, The program returns Invalid Input and exits.

**Execution Screenshots:**

     