**UCS 2312 Data Structures Lab**

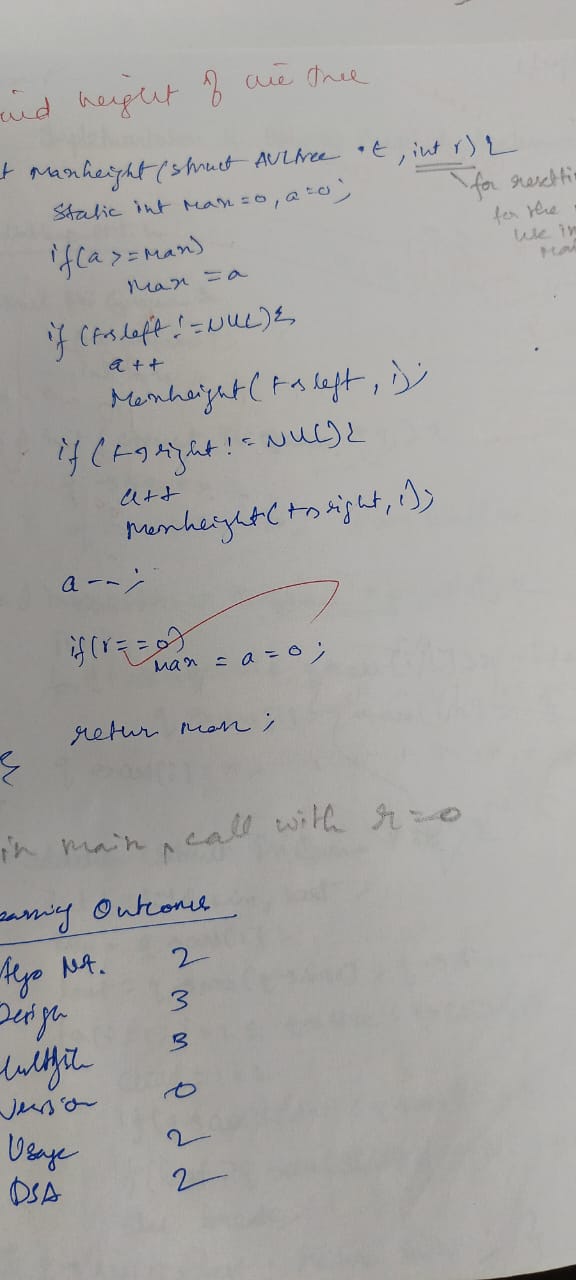
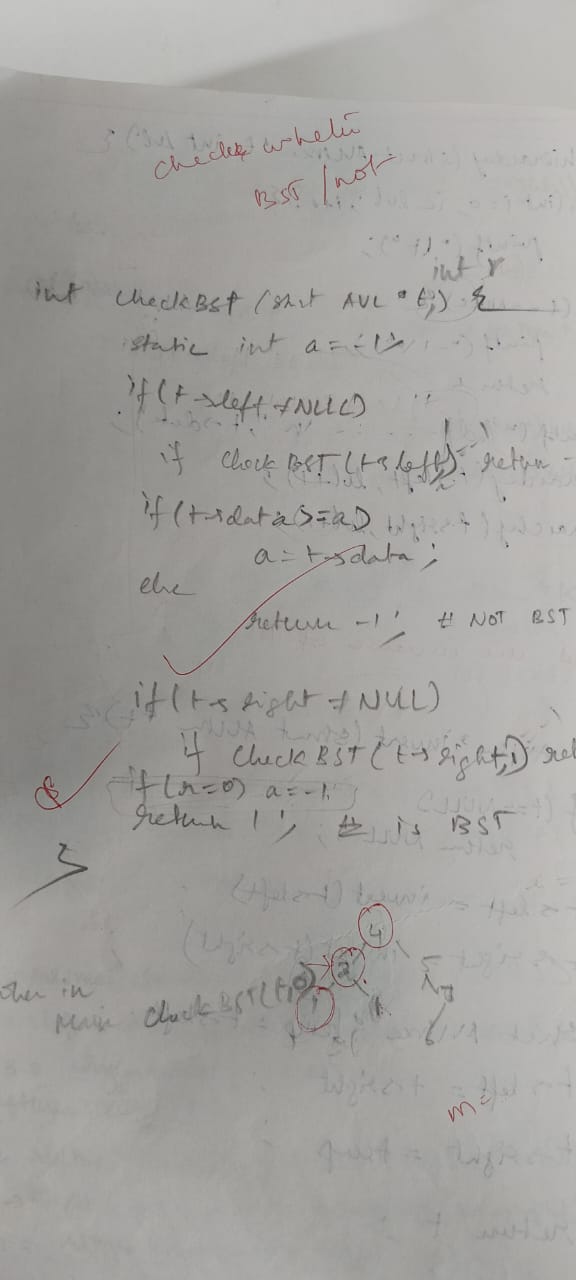
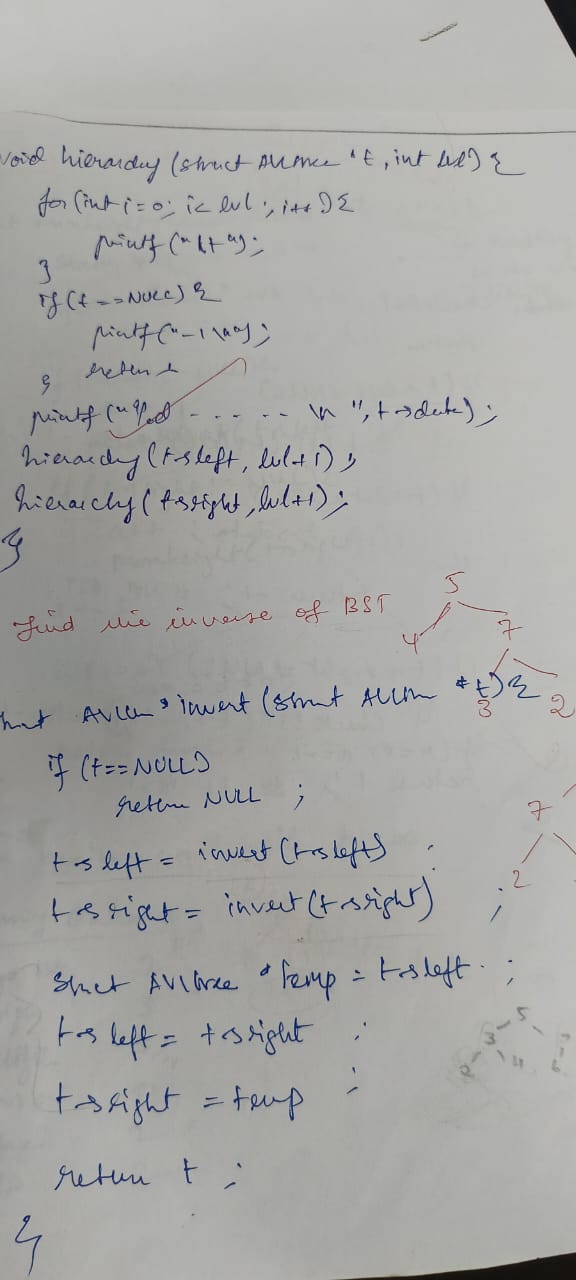
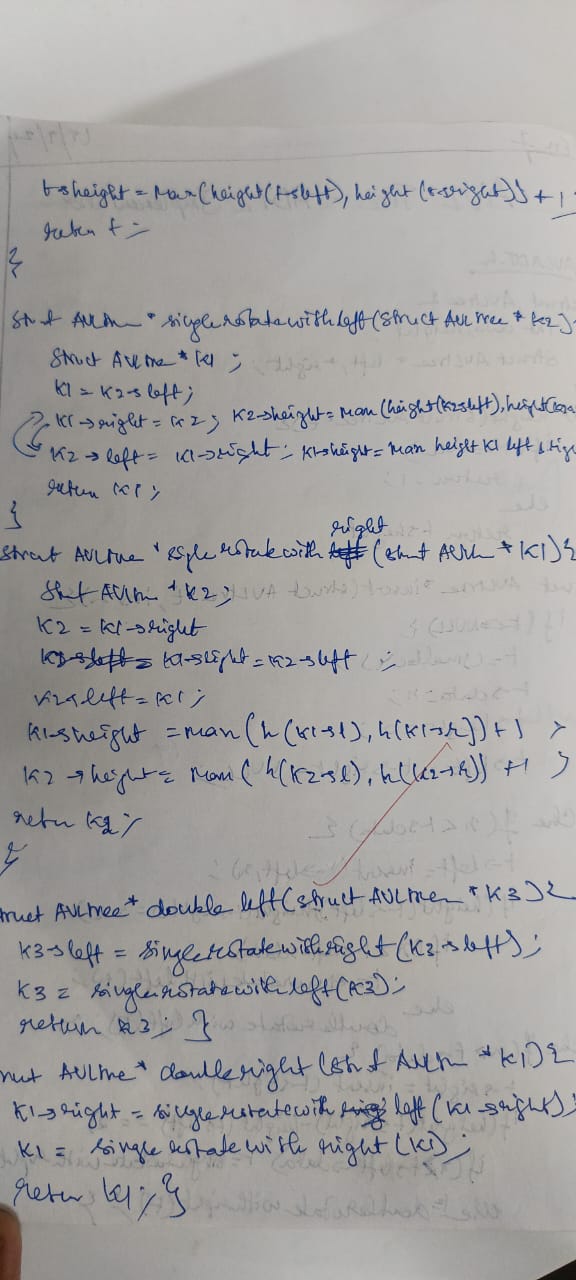
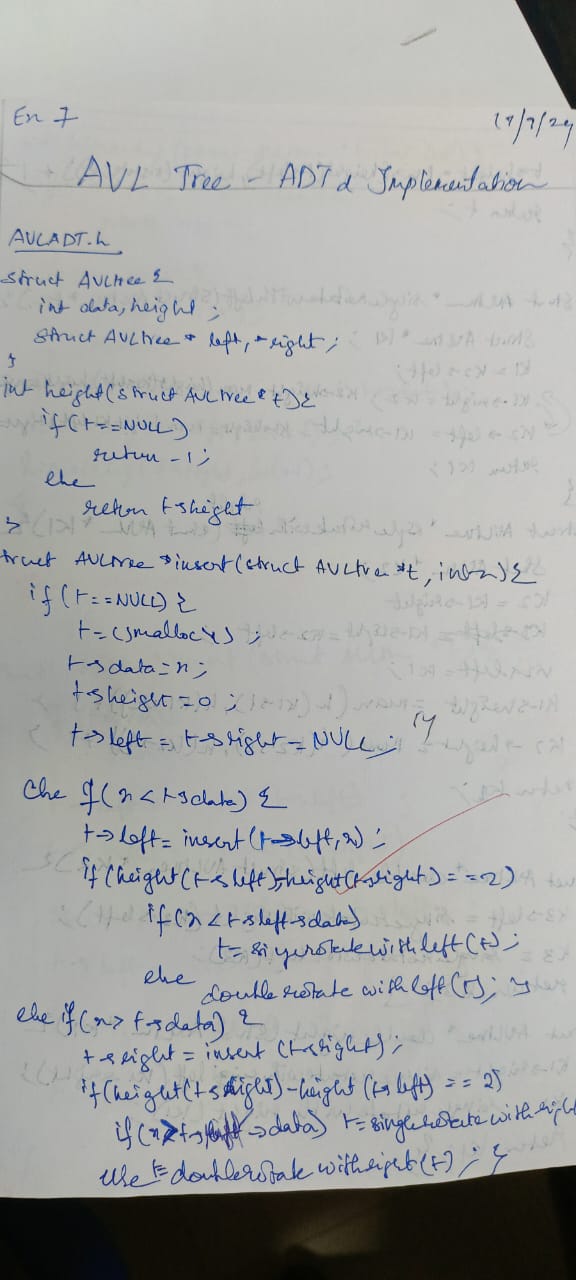
**Assignment 7: Implementation of AVL Tree**

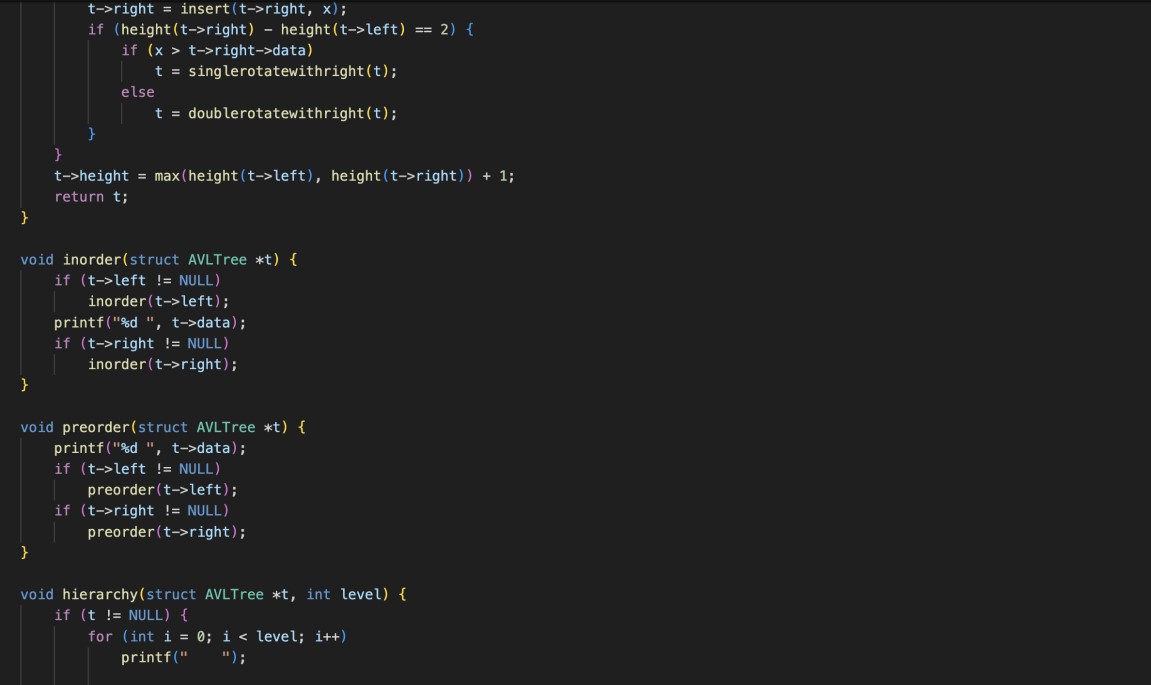
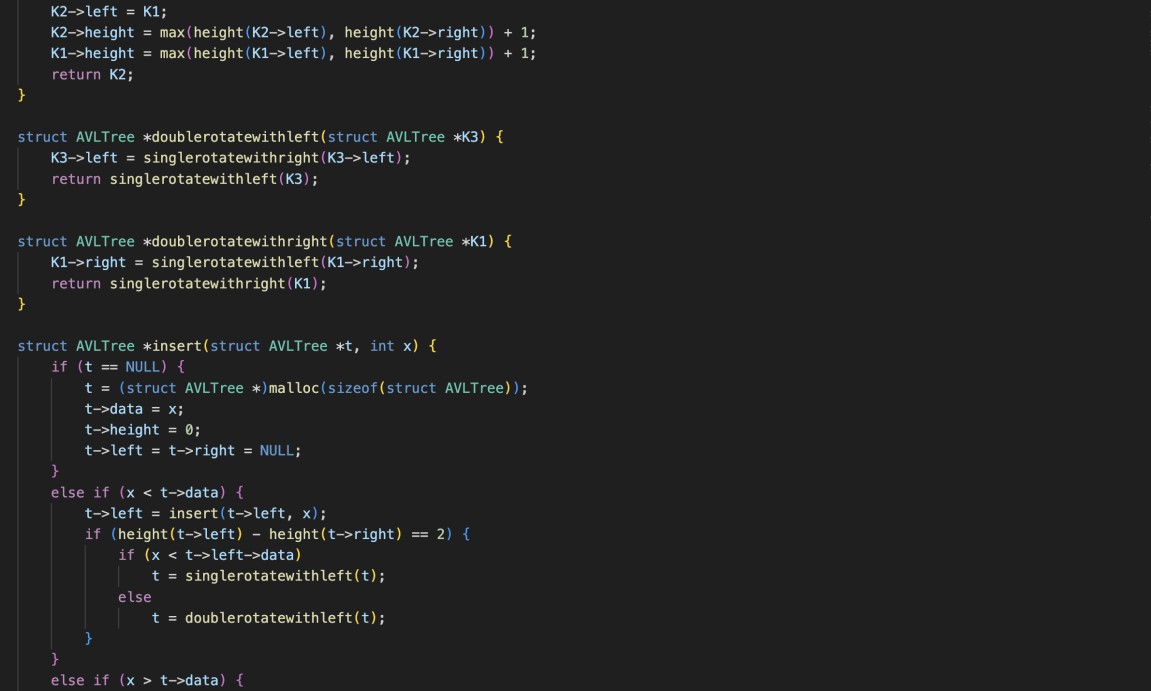
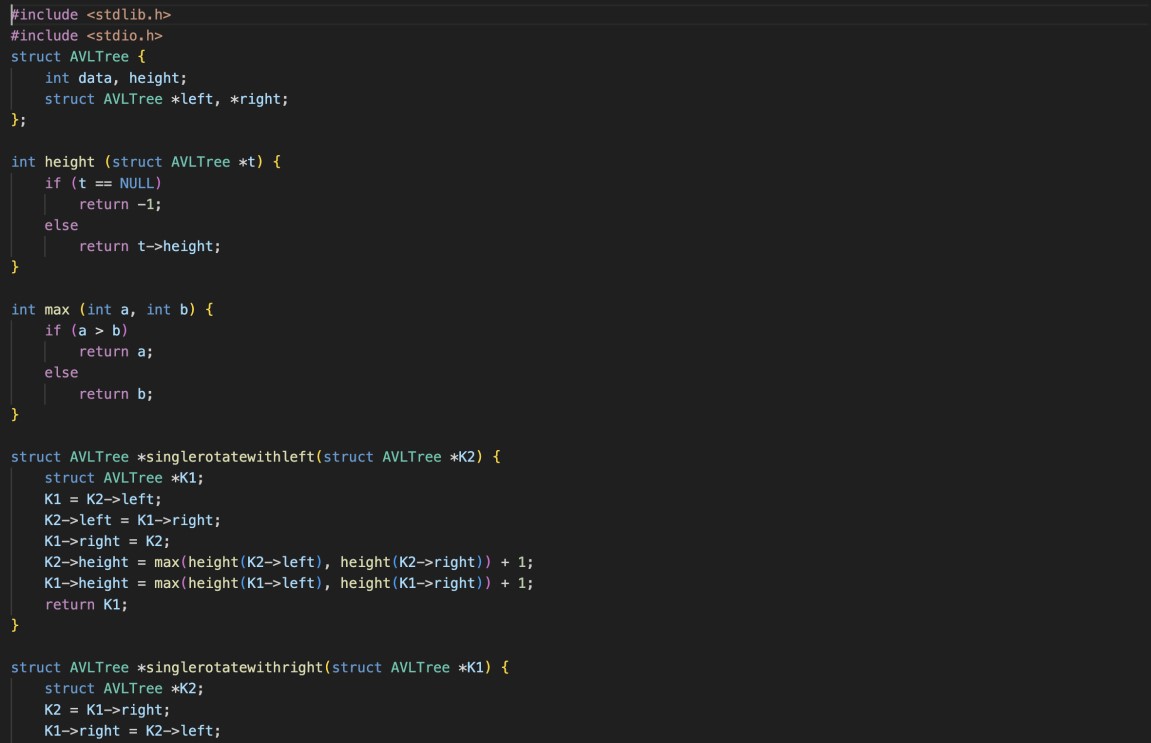
Design an ADT for the AVL Tree data structure with the following functions. Each node consists of a character data, address of left, right and parent nodes [CO1, K3]

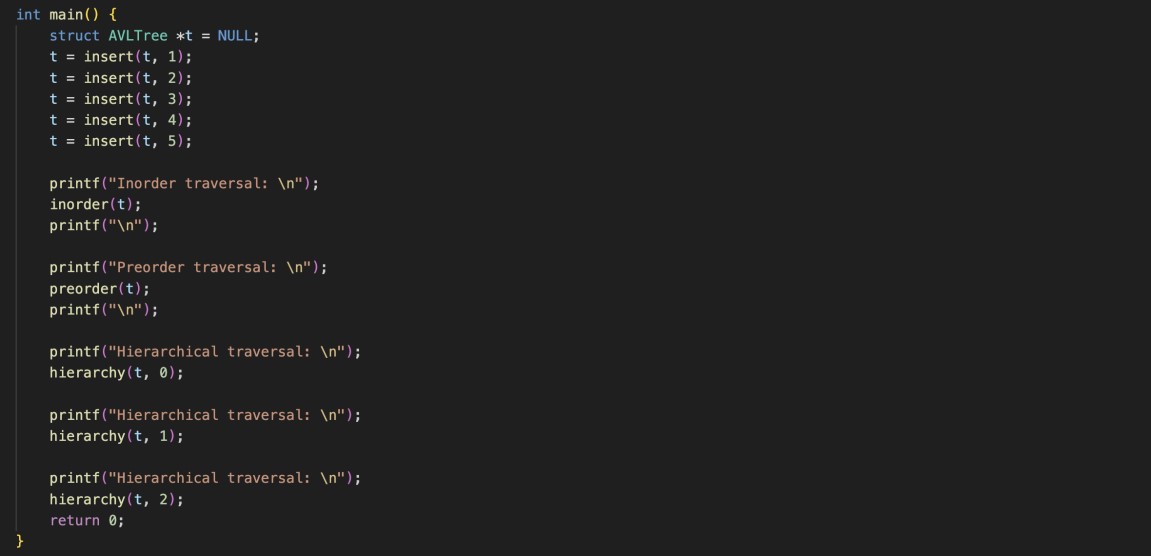
1. insertAVL(t, data) – insert data into BST
2. hierarchical(t) – display the tree in hierarchical fashion
3. findParent(t, key) – will return the parent of the given data
4. findLeaf(t) – returns the number of leaf nodes
5. findDepth(t,x) – returns the depth of the node x

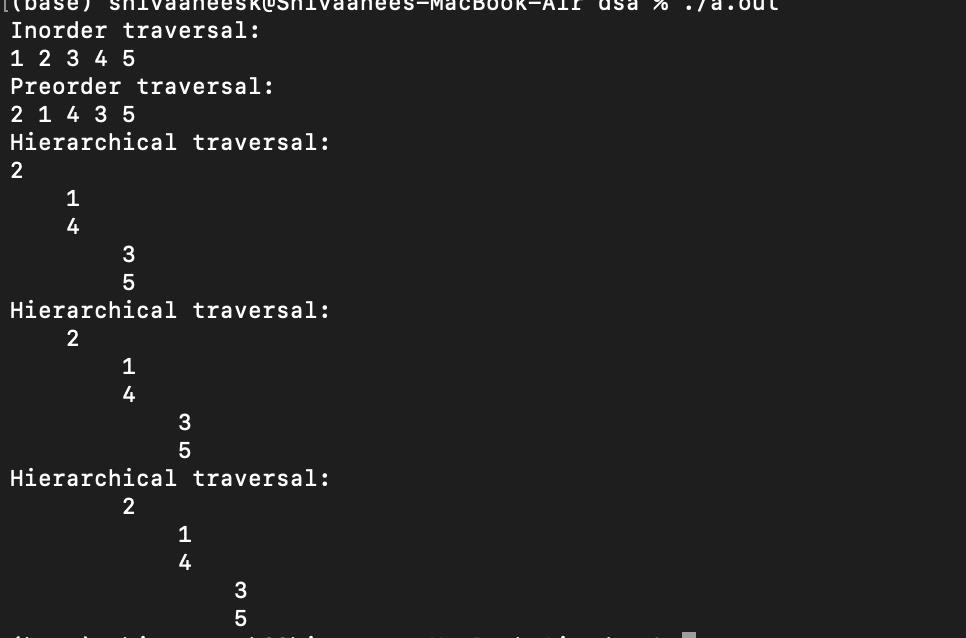
Demonstrate the AVL ADT with the insertion of the following character data one at a time.

**H, I, J, B, A, E, C, F, D, G, K, L**









Technical Outcomes

|  |  |  |
| --- | --- | --- |
| Design | 2 |  |
| Understanding of DS | 2 | Needs improvement |
| Use of DS | 3 |  |
| Debugging | 2 |  |

Best Practices

|  |  |  |
| --- | --- | --- |
| Design before coding | 2 | Needs improvement |
| Usage of Algo | 2 |  |
| Multifile | 3 |  |
| Versioning | 0 | didnt |