**AIM:** Write a program to perform the following operations:

a) Insert an element into a AVL tree

b) Delete an element from a AVL tree

c) Search for a key element in a AVL tree

**ALGORITHM:**

 **Insertion**

* Perform standard BST insertion.
* Update height of the current node.
* Calculate the **balance factor** (difference between left and right subtree heights).
* If the node becomes unbalanced, perform appropriate rotation:
  + **Left-Left (LL) Rotation**
  + **Right-Right (RR) Rotation**
  + **Left-Right (LR) Rotation**
  + **Right-Left (RL) Rotation**

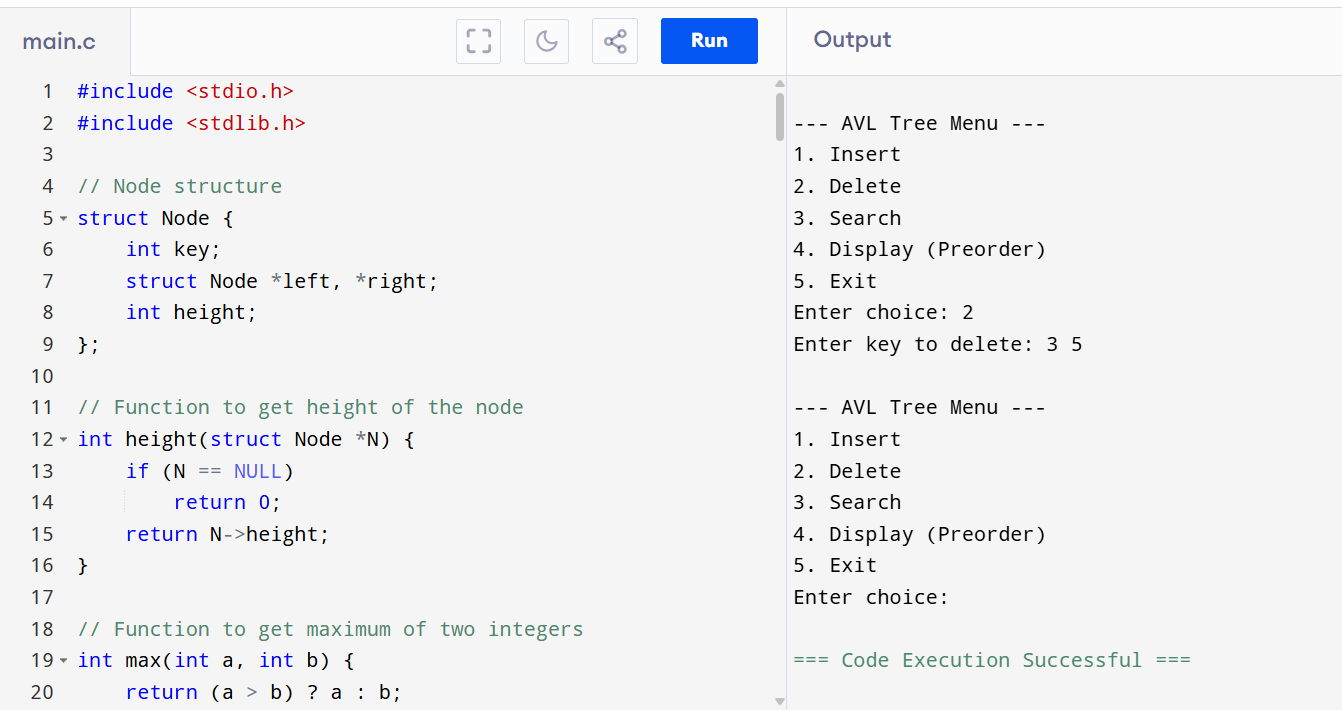
 **Deletion**

* Perform standard BST deletion.
* Update height and balance factor.
* If unbalanced, perform the same rotations as in insertion.

 **Search**

* Traverse like in a BST:
  + If key < node → search left.
  + If key > node → search right.
  + If key == node → found.

**CODE:**

****

**OUTPUT:**

--- AVL Tree Menu ---

1. Insert

2. Delete

3. Search

4. Display (Preorder)

5. Exit

Enter choice: 2

Enter key to delete: 3 5

--- AVL Tree Menu ---

1. Insert

2. Delete

3. Search

4. Display (Preorder)

5. Exit

Enter choice: