**[Transform to Sum Tree](https://practice.geeksforgeeks.org/problems/d7e0ce338b11f0be36877d9c35cc8dfad6636957/1)**

Given a Binary Tree of size N , where each node can have positive or negative values. Convert this to a tree where value of each node will be the sum of the values of all the nodes in left and right sub trees of the original tree. The values of leaf nodes are changed to 0.  
**Note:**You have to modify the given tree in-place.

**Example 1:**

**Input:**

10

/ \

-2 6

/ \ / \

8 -4 7 5

**Output:**

20

/ \

4 12

/ \ / \

0 0 0 0

**Explanation:**

(4-2+12+6)

/ \

(8-4) (7+5)

/ \ / \

0 0 0 0

**Example 2:**

**Input:**

10

/ \

-2 6

/ \ / \

8 -4 7 5

/ \ / \ / \ / \

  2 -2 3 -5 9 -8 2 8

**Output:**

29

/ \

2 23

/ \ / \

0 -2 1 10

  / \ / \ / \ / \

0 0 0 0 0 0 0 0

**Explanation:**

(2+6+8-4+7+5+2-2+3-5+9-8+2+8)

/ \

(8-4+2-2+3-5) (7+5+9-8+2+8)

/ \ / \

(2-2) (3-5) (9-8) (2+8)

/ \ / \ / \ / \

0 0 0 0 0 0 0 0

**Your Task:**  
You dont need to read input or print anything. Complete the function**toSumTree()** which takes root node as input parameter and modifies the given tree in-place.

**Note:** If you click on Compile and Test the output will be the in-order traversal of the modified tree.

**Expected Time Complexity:**O(N)  
**Expected Auxiliary Space:**O(height of tree)

**Constraints:**  
1 ≤ N ≤ 104