**Sum of Right Leaf Nodes:-**

Given a Binary Tree of size **N**, your task is to complete the function **rightLeafSum()**, that should return the sum of all the leaf nodes that are right child of their parent of the given binary tree.  
Example:

Input :

1

/ \

2 3

/ \ \

4 5 8

\ / \

2 6 7

Output :

sum = 2 + 5 + 7 = 14

**Input:**  
The function takes a single arguments as input, the reference pointer to the **root** of the binary tree.  
There are **T** test cases and for each test case the function will be called separately.  
  
**Output:**  
The function should return the sum of all the right leaf nodes of the binary tree.  
  
**Constraints:**  
1<=T<=100  
1<=N<=100  
  
**Example:  
Input:**  
2  
2  
1 2 L 1 3 R  
5  
10 20 L 10 30 R 20 40 L 20 60 R 30 90 L  
**Output:**  
3  
60

**Explanation:  
Test Case 1:**  
Tree formed is as follows:  
      1  
    /    \  
  2      3

Here, there are 2 leaf nodes (2 and 3). Out of these, only 3 is a right child of its parent. Hence, the answer is 3.

**Test Case 2:**  
The tree formed is as follows:  
           10  
         /       \  
      20        30  
    /     \      /  
  40   60  90

Here, there are 3 leaf nodes (40, 60 and 90). Out of these, only 60 is a right child of its parent. Hence, the answer is 60.