**Sum of Binary Tree:-**

Given a Binary Tree of size **N**, your task is to complete the function **sumBt()**, that should return the sum of all the nodes of the given binary tree.  
  
**Input:**

First line of input contains the number of test cases T. For each test case, there will be two lines:

1. First line of each test case will be an integer N denoting the number of parent child relationships.
2. Second line of each test case will print the **level order traversal** of the tree in the form of **N space separated triplets**. The description of triplets is as follows:
   1. Each triplet will contain **three space-separated** elements of the form (int, int char).
   2. The first integer element will be the value of **parent.**
   3. The second integer will be the value of corresponding **left** or **right child**. In case the child is null, this value will be **-1**.
   4. The third element of triplet which is a character can take any of the three values **‘L’**, **‘R’** or **‘N’**. L denotes that the children is a left child, R denotes that the children is a Right Child and N denotes that the child is NULL.

Please **note** that the relationships are printed only for internal nodes and not for leaf nodes.  
  
**Output:**  
The function should return the sum of all the nodes of the binary tree.  
  
**User Task:**  
As it is a functional problem. So, you don't need to worry about the input you just have to complete the function **sumBT()**that takes a **node** as a parameter and returns the sum of all the nodes.  
  
**Constraints:**  
1<=T<=100  
1<=N<=100  
  
**Example:  
Input:**  
2  
2  
1 2 L 1 -1 N  
6  
1 2 L 1 3 R 2 -1 N 2 -1 N 3 3 L 3 -1 N  
**Output:**  
3  
9