# Sum of the Longest Bloodline of a Tree (Sum of nodes on the longest path from root to leaf node)

Given a binary tree of size **N.** Your task is to complete the function **sumOfLongRootToLeafPath(),** that find the sum of all nodes on the longest path from root to leaf node.
If two or more paths compete for the longest path, then the path having maximum sum of nodes is being considered.

Examples:

Input : Binary tree:
4
/ \
2 5
/ \ / \
7 1 2 3
/
6
Output : 13

```
    **4**
   /  \
  **2**    5
 / \ / \
7  **1** 2   3
  /
 **6**
```

The highlighted nodes **(4, 2, 1, 6)** above are
part of the longest root to leaf path having
sum = (4 + 2 + 1 + 6) = 13

```
def sumOfLongRootToLeafPath(root):
    #:param root: root of the given tree.
    #:RETURN: SUM

    #code here
    ref = [0]
    length = [0]
    helper(root,ref,length,0,0)
    return ref[0]

def helper(root,ref,length,temp,tempLen):
    if root is None:
        return
    temp = temp+root.data
    if root.left is root.right:
        if tempLen>=length[0]:
            length[0] = max(length[0],tempLen)
            ref[0] = max(temp,ref[0])
    helper(root.left,ref,length,temp,tempLen+1)
    helper(root.right,ref,length,temp,tempLen+1)
```