**Print all nodes that don't have sibling:-**

Given a Binary Tree of size N, find all the nodes which don't have any sibling. Root node can not have a sibling.

**Example 1:**

**Input :**

37

/

20

/

113

**Output:** 20 113

**Explanation:** 20 and 113 dont have any siblings.

**Example 2:**

**Input :**

1

/ \

2 3

**Output:** -1

**Explanation:** Every node has a sibling.

**Your Task:**  
You dont need to read input or print anything. Complete the function**noSibling()**which takes the root of the tree as input parameter and returns a list of integers containing all the nodes that don't have a sibling in sorted order. If all nodes have a sibling, then the returning list should contain only one element -1.

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

**Constraints:**  
1 ≤ N ≤ 10^4  
All nodes have distinct values.