**K distance from root:-**

Given a Binary Tree of size **N** and an integer **K**. Print all nodes that are at distance k from root (root is considered at distance 0 from itself). Nodes should be printed from left to right. If k is more that height of tree, nothing should be printed.

For example, if below is given tree and k is 2. Output should be 4 5 6.

          1  
       /     \  
     2        3  
   /         /   \  
  4        5    6   
     \  
      8

**Example 1:**

**Input:**

K = 0

  1

  / \

  3 2

**Output:** 1

**Example 2:**

**Input:**

K = 3

  3

  /

  2

  \

  1

  / \

  5 3

**Output:** 5 3

**Your Task:**  
You don't have to take input. Complete the function **Kdistance()**that accepts **root** nodeand **k**as parameter and **return**the **value**of the **nodes**that are at a distance k from the root. (The values are returned as vector<int> in **cpp,**as ArrayList<Integer> in **java**and list in python)  
**Expected Time Complexity:**O(N).  
**Expected Auxiliary Space:**O(Height of the Tree).

**Constraints:**  
1 <= N <= 104  
0 <= K <= 30