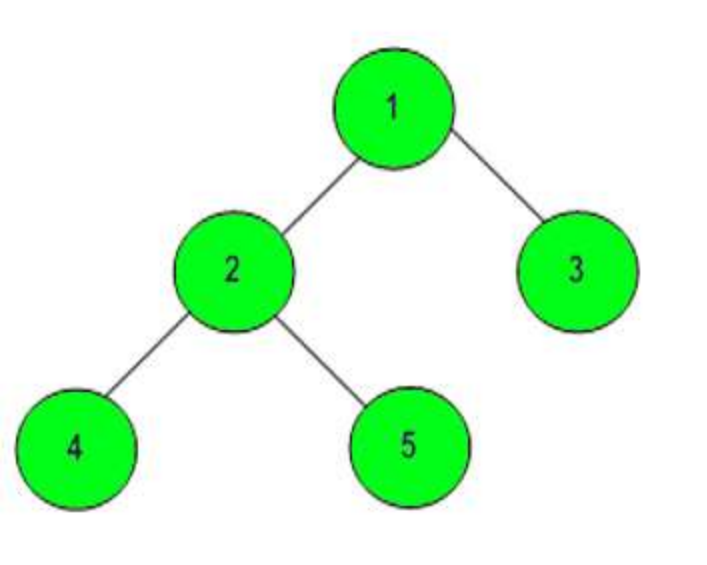
Kth Ancestor in a Tree

Given a binary tree of size **N**, a **node,** and a positive integer **k**., Your task is to complete the function **kthAncestor()**, the function should return the **kth** ancestor of the given node in the binary tree. If there does not exist any such ancestor then return -1.

Note: It is guaranteed that the **node** exists in the tree.

Example 1:



Input:

```
K = 2 Node = 4
```

Output: 1

Explanation:

```
Since, K is 2 and node is 4, so we
first need to locate the node and
look k times its ancestors.
Here in this Case node 4 has 1 as his
2nd Ancestor aka the Root of the tree.
```

Example 2:

```
Input:
k=1
node=3
      1
    /   \
   2     3

Output:
1
Explanation:
K=1 and node=3 ,Kth ancestor of node 3 is 1.
```

Your Task:

You are asked to complete the function **kthAncestor()** which accepts **root** of the tree, **k** and **node** as input parameters, and returns the **kth ancestor** of Node which contains node as its value.

Expected Time Complexity: O(N)
Expected Auxiliary Space: O(N)

Constraints:

$1<=N<=10^5$
1<= K <= 100
1 <= Node.data <= N