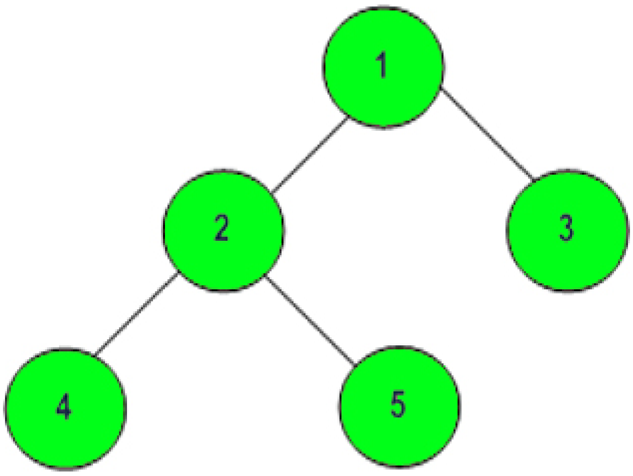
Kth Ancestor in a Tree

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.



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.

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

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

```python
def kthAncestor(root,k, node):
    #code here
    res = {}
    helper(root,res,None)
    parent = None
    for key in res:
        if key.data == node:
            node = key
            break
    while node:
        node = res[node]
        k = k-1
        if k==0 and node!=None:
            return node.data
    return node if node else -1





def helper(root,res,parent):
    if root is None:
        return
    res[root] = parent
    helper(root.left,res,root)
    helper(root.right,res,root)
```