Question :

You are given a binary Tree. Find the number of nodes with one child

Tags:

Tree, Binary Tree, math

Input Description:

Number of nodes in a binary tree.
Nodes as an array

Output Description :

Number of nodes with one child in a binary tree

Solution :

```
class Node:
    def __init__( self , data ) :
        self.data = data
        self.left = None
        self.right = None
class BinaryTree:
    def __init__(self):
        self.root = None
def createBinaryTree(lst, n, node, i):
    if i < n:
        if lst[i] == -1:
            return node
        temp = Node(lst[i])
        node = temp
        node.left = createBinaryTree(lst, n, node.left, (2*i)+1)
        node.right = createBinaryTree(lst, n, node.right, (2*i)+2)
    return node

def countChild(root, count):
    if root == None:
        return
    if (root.left is None and root.right is not None) or (root.left is not None and root.right is None):
        count.append(count.pop() + 1)

    countChild(root.left, count)
    countChild(root.right, count)
    return count
```

```
n = int(input())
lst = [int(x) for x in input().split()[:n]]
tree = BinaryTree()
tree.root = createBinaryTree(lst, n, tree.root, 0)
count = [0]
count = countChild(tree.root, count)

print(count[0])
```

Test Cases :

Test Case 1 :
Input :
7
2 3 4 5 -1 4 -1
Output :
2

Test Case 2:
Input :
7
2 3 -1 5 6 7 8
Output :
1

Test Case 3:
Input :
17
6 2 9 2 1 7 9 4 8 11 12 16 -1 12 -1 -1 -1
Output :
2

Test Case 4:
Input :
12
2 3 6 4 8 11 12 -1 3 -1 16 -1
Output :
2

Test Case 5:
Input :
6
5 3 5 8 4 2

Output :
1