```
'''
        Time complexity: O(N)
        Space complexity: O(H)

        where N is the number of nodes in the input tree
        and H is the height of the input tree.
'''

from sys import stdin, setrecursionlimit
import queue

setrecursionlimit(10 ** 6)

class BinaryTreeNode:
    def __init__(self, data):
        self.data = data
        self.left = None
        self.right = None


def getSum(root):

    if root is None :
        return 0


    leftSum  = getSum(root.left)
    rightSum = getSum(root.right)

    return (leftSum + rightSum + root.data)




#Taking level-order input using fast I/O method
def takeInput():
    levelOrder = list(map(int, stdin.readline().strip().split(" ")))
    start = 0

    length = len(levelOrder)

    root = BinaryTreeNode(levelOrder[start])
    start += 1

    q = queue.Queue()
    q.put(root)

    while not q.empty():
        currentNode = q.get()

        leftChild = levelOrder[start]
        start += 1

        if leftChild != -1:
            leftNode = BinaryTreeNode(leftChild)
            currentNode.left =leftNode
            q.put(leftNode)

        rightChild = levelOrder[start]
        start += 1

        if rightChild != -1:
            rightNode = BinaryTreeNode(rightChild)
            currentNode.right =rightNode
            q.put(rightNode)

    return root

# Main
root = takeInput()
print(getSum(root))
```