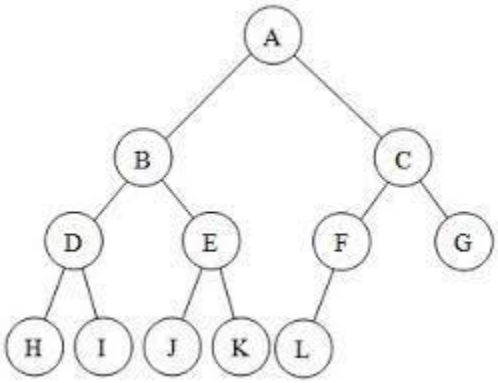
Given a Complete Binary Tree, you have to tell the number of nodes present in it.

Constraints:

1 <= Number of Nodes ('N') <= 500000.



You will **only** be given the **pointer to the root node and nothing else**!
Answer of the above tree will be : 12.
Each node is of the form:

```
struct Node {
  struct Node *left, *right;
  int val
}
```

H = 4

H=3

3^(h-1) when height is starting from 0-> n-1

3^(h-1-1) when height is starting from 1-> n

```
Import math
Def height(root) :
        root==null:
                Return 0
        Hleft = height(root.left)
        Hright = height(root.right)

        Return math.max(Hleft, Hright) + 1
}
```

```
Def get_elemets_sum_at_a_height(root, current_level, at_level, sum):
        If root==null:
                Return 0
        If current_level == at_level:
                Sum += root.data
        Get_elemets_sum_at_a_height(root.left, current_level, at+1_level, sum)
        get_elemets_sum_at_a_height(root.right, current_level+1, at_level, sum)


If __name__ == '__main__':
        heightOfthetree = height(root)
        Total_elements_till_h-1 = 2^(heightOfthetree-1-1)
        Sum = 0
        get_elemets_sum_at_a_height(root, 0, heightOfthetree-1, sum)
        print("Total Sum: ", Total_elements_till_h-1 + Sum)




Def deciderNode(root):

        If root==null:
                Return None

        If root.left & !root.right:
                Return root
```

Given ***N*** distinct elements of an array, compute the sum of **(min+max)** across all the subarrays of the array . Mathematically, he asks him to calculate,
$\sum_{i=1}^{N}\sum_{j=i}^{N}(min(A_i, A_{i+1}, \ldots, A_j) + max(A_i, A_{i+1}, \ldots, A_j))$

Input Format

*First line of the input contains an integer **N**, the size of the array.*
*Next line contains **N** space separated integers of the given array **A**.*

Output Format

Output the required answer in a separate line.

Constraints

- $1 \leq N \leq 10^5$
- $1 \leq A_i \leq 10^6$

Eg :-
All the elements are distinct
N = 3
A = [1, 2, 3]
Expected answer = {1} + {2} + {3} + {1, 2} + {2, 3} + {1, 2, 3}
= (1 + 1) + (2 + 2) + (3 + 3) + (1 + 2) + (2 + 3) + (1 + 3) = **24**
So, **24** is the required answer.

[1,2,3]

```
Len_arr = len(arr)
For i in range(1, Len_arr-1):
        A = 0
        B = A + i
```

[41,3,6,7,8,91]

6

{41} + {3} + {6} + {7} + {8} + {91}

1. Sum = 2* (each uique_element)

{41, 3} {3, 6} {6, 7} {7,8} {8, 91}
{41, 3, 6} {3, 6, 7} { 6, 7, 8} { 7, 8, 91}
{41, 3, 6, 7} {3, 6, 7, 8} { 6, 7, 8, 91}
{41, 3, 6, 7, 8} {3, 6, 7, 8, 91}

{41, 3, 6, 7, 8, 91}

3 = 9 + 2 [lesser than 5 elements out of 6 elements]
6 = 4 + 2 [lesser than 4 elements out of 6 elements]
7 = 4 + 2 [lesser than 3 elements out of 6 elements]

41 = 4 + 2 [lesser than 1 elements out of 6 elements]
8 = 4 + 2 [lesser than 2 elements out of 6 elements]
91 = 4 + 2 [lesser than 0 elements out of 6 elements]