60) 1442. Count Triplets That Can Form Two Arrays of Equal XOR Given an array of integers arr.
We want to select three indices i, j and k where (0 <= i < j <= k < arr.length).
Let's define a and b as follows:

- a = arr[i] ^ arr[i + 1] ^ ... ^ arr[j - 1]
- b = arr[j] ^ arr[j + 1] ^ ... ^ arr[k]

Note that ^ denotes the bitwise-xor operation.
Return the number of triplets (i, j and k) Where a == b.

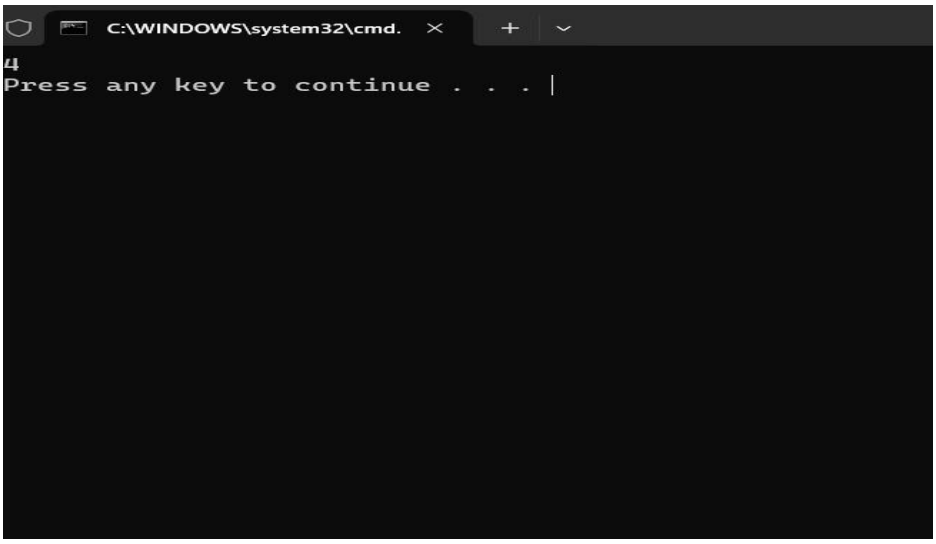
Example 1:
Input: arr = [2,3,1,6,7]
Output: 4
Explanation: The triplets are (0,1,2), (0,2,2), (2,3,4) and (2,4,4)

CODE:

```
def countTriplets(arr):
    n = len(arr)     prefix_xor = [0] * (n + 1)     count = 0          for i in
range(n):         prefix_xor[i + 1] = prefix_xor[i] ^ arr[i]
        for i in range(n):         for j in range(i + 1, n + 1):              if
prefix_xor[i] == prefix_xor[j]:                count += j - i - 1
return count

arr = [2, 3, 1, 6, 7] print(countTriplets(arr))
```

OUTPUT:



TIME COMPLEXITY : **O(n2)**