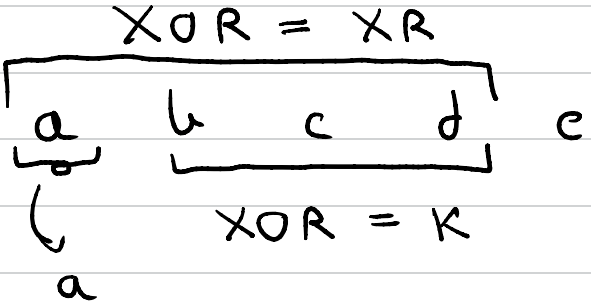
NUMBER OF SUBARRAYS WITH XOR K

✡ In this problem we are supposed to return the number of subarrays where all the elements XOR'd return K.

Brute force solution like in all subarray questions is to calculate all subarrays find their XOR's and increment the count if it matches. Time complexity is nearly $O(N^3)$.

Better solution is to carry on the XOR from previous iteration to remove the last loop. This improves our time complexity of $O(N^2)$.

Optimal solution is to use the prefix Sum logic but instead making it work for XOR. How we do that is



$$a \wedge K = XR$$
$$a \wedge K \wedge K = XR \wedge K$$
$$a = XR \wedge K$$

We use two pointers, which will track the start and end of our subarray

Pseudocode :

```
numberOfSubarraysWithXORK(arr, N, K) {
    map mpp
    mpp[0] = 1
    counter = 1
    x = 0
    for(i = 0 → N) {
        x = x ^ arr[i]
        req = x ^ K
        if(req in mpp) {
            counter += mpp[req]
        }
        mpp[x] += 1
    }
    return counter
}
```