**Counts Zeros Xor Pairs**

Given an array A[] of size N. Find the number of pairs (i, j) such that  
AiXOR Aj = 0, and 1 ≤ i < j ≤ N.

**Example 1:**

**Input :** arr[ ] = {1, 3, 4, 1, 4}

**Output :** 2

**Example 2:**

**Input :** arr[ ] = {2, 2, 2}

**Output :** 3

**Step 1 :** We know that , the Xor of the same number is zero .

**Step 2 :** So , take the count of each number in the array and store it in a hash map.

**Step 3 :** It is not possible to make xor , when the occurrence is zero so we calculate the cnt only when occuremce is > 1.

**Step 4 :** By using (n \* (n – 1) ) / 2

class Complete{

public static long calculate (int arr[], int n) {

//Complete the function

Map<Integer , Integer> map = new HashMap<>();

int xor = 0;

int cnt = 0;

for(int i = 0 ; i < n ; i++){

map.put(arr[i] , map.getOrDefault(arr[i] , 0) + 1);

}

for(int i : map.keySet()){

int val = map.get(i);

cnt += (val \* (val - 1)) / 2;

}

return cnt;

}}