Count all possible groups of size 2 or 3 that have sum as multiple of 3

Given an unsorted integer (positive values only) array of size 'n', we can form a group of two or three, the group should be such that the sum of all elements in that group is a multiple of 3. Count all possible number of groups that can be generated in this way.

```
Input: arr[] = {3, 6, 7, 2, 9}
Output: 8
// Groups are {3,6}, {3,9}, {9,6}, {7,2}, {3,6,9},
//           {3,7,2}, {7,2,6}, {7,2,9}


Input: arr[] = {2, 1, 3, 4}
Output: 4
// Groups are {2,1}, {2,4}, {2,1,3}, {2,4,3}
```

We strongly recommend to minimize the browser and try this yourself first.

The idea is to see remainder of every element when divided by 3. A set of elements can form a group only if sun of their remainders is multiple of 3. Since the task is to enumerate groups, we count all elements with different remainders.

```
1. Hash all elements in a count array based on remainder, i.e,
   for all elements a[i], do c[a[i]%3]++;
2. Now c[0] contains the number of elements which when divided
   by 3 leave remainder 0 and similarly c[1] for remainder 1
   and c[2] for 2.
3. Now for group of 2, we have 2 possibilities
   a. 2 elements of remainder 0 group. Such possibilities are
      c[0]*(c[0]-1)/2
   b. 1 element of remainder 1 and 1 from remainder 2 group
      Such groups are c[1]*c[2].
4. Now for group of 3,we have 4 possibilities
   a. 3 elements from remainder group 0.
      No. of such groups are c[0]C3
   b. 3 elements from remainder group 1.
      No. of such groups are c[1]C3
   c. 3 elements from remainder group 2.
```

```
      No. of such groups are c[2]C3
   d. 1 element from each of 3 groups.
      No. of such groups are c[0]*c[1]*c[2].
5. Add all the groups in steps 3 and 4 to obtain the result.
```

```
#include<stdio.h>

// Returns count of all possible groups that can be formed from elements
// of a[].
int findgroups(int arr[], int n)
{
    // Create an array C[3] to store counts of elements with remainder
    // 0, 1 and 2.  c[i] would store count of elements with remainder i
    int c[3] = {0}, i;

    int res = 0; // To store the result

    // Count elements with remainder 0, 1 and 2
    for (i=0; i<n; i++)
        c[arr[i]%3]++;

    // Case 3.a: Count groups of size 2 from 0 remainder elements
    res += ((c[0]*(c[0]-1))>>1);

    // Case 3.b: Count groups of size 2 with one element with 1
    // remainder and other with 2 remainder
    res += c[1] * c[2];

    // Case 4.a: Count groups of size 3 with all 0 remainder elements
    res += (c[0] * (c[0]-1) * (c[0]-2))/6;

    // Case 4.b: Count groups of size 3 with all 1 remainder elements
    res += (c[1] * (c[1]-1) * (c[1]-2))/6;

    // Case 4.c: Count groups of size 3 with all 2 remainder elements
    res += ((c[2]*(c[2]-1)*(c[2]-2))/6);

    // Case 4.c: Count groups of size 3 with different remainders
    res += c[0]*c[1]*c[2];

    // Return total count stored in res
    return res;
}

// Driver program to test above functions
int main()
{
    int arr[] = {3, 6, 7, 2, 9};
    int n = sizeof(arr)/sizeof(arr[0]);
    printf("Required number of groups are %d\n", findgroups(arr,n));
```

```
    return 0;
}
```

Output:

```
Required number of groups are 8
```

Time Complexity: O(n)
Auxiliary Space: O(1)

This article is contributed by Amit Jain. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

18 Comments Category: Arrays

Related Posts:

- Longest Span with same Sum in two Binary arrays
- Count Inversions of size three in a give array
- Find the subarray with least average
- Count triplets with sum smaller than a given value
- Find zeroes to be flipped so that number of consecutive 1's is maximized
- Reorder an array according to given indexes
- Find maximum value of Sum(i*arr[i]) with only rotations on given array allowed
- Find maximum average subarray of k length

(Login to Rate and Mark)

2.6 Average Difficulty : **2.6/5.0**
Based on **3** vote(s)

Add to TODO List
Mark as DONE

Writing code in comment? Please use code.geeksforgeeks.org, generate link and share the link here.