8th Jan 2023

First:-

Absolute difference divisible by K Easy

Given an array of integers of size **n** and an integer **k**, find all the pairs in the array whose absolute difference is divisible by k.

Example 1:

```
Input:
n = 3
arr[] = {3, 7, 11}
k = 4
Output:
3
Explanation:
(11-3) = 8 is divisible by 4
(11-7) = 4 is divisible by 4
(7-3) = 4 is divisible by 4
```

Example 2:

```
Input:
n = 4
arr[] = {1, 2, 3, 4}
k = 2
Output :
2
Explanation:
Valid pairs are (1,3), and (2,4).
```

Your Task:
You don't need to read input or print anything. Your task is to complete the function **countPairs()** which takes integers n, array arr[ ], integer k as input parameters and returns the number of pairs whose absolute difference is divisible by k.
Note: The answer may be large so use 64-bit integer.

Expected Time Complexity: O(n + k)
Expected Auxiliary Space: O(k)

Constraints:
$2 \le n \le 10^5$
$1 \le k, arr[i] \le 10^5$

CODE SECTION:-

```
long long countPairs(int n, int arr[], int k) {
      // code here
      unordered_map<int,int>m;
      long long ans=0;
      for(int i=0;i<n;i++){
         int rem=arr[i]%k;
         ans+=m[rem];
         m[rem]++;
         // cout<<ans<<" ";
      }
      return ans;
   }
```

Second:-

Largest subarray with 0 sum :: Easy

Given an array having both positive and negative integers. The task is to compute the length of the largest subarray with sum 0.

Example 1:

```
Input:
N = 8
A[] = {15,-2,2,-8,1,7,10,23}
Output: 5
Explanation: The largest subarray with
sum 0 will be -2 2 -8 1 7.
```

Your Task:
You just have to complete the function **maxLen()** which takes two arguments an array **A** and **n,** where n is the size of the array A and returns the length of the largest subarray with 0 sum.

Expected Time Complexity: O(N).
Expected Auxiliary Space: O(N).

Constraints:
1 <= N <= 10^5
-1000 <= A[i] <= 1000, for each valid i

Code section:-

```
long long countPairs(int n, int arr[], int k)
    {
        // code here

        unordered_map<int, int> m;
        long long ans = 0;

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

            int rem = arr[i] % k;
            ans += m[rem];
            m[rem]++;
            // cout<<ans<<" ";
        }

        return ans;
    }
```