

# Zero Sum Subarrays

---

You are given an array `arr[]` of size `n`. Find the total count of sub-arrays having their sum equal to 0.

## Example 1:

**Input:** `n = 6`

`arr[] = {0,0,5,5,0,0}`

**Output:** 6 **Explanation:** The 6 subarrays are [0], [0], [0], [0], [0,0], and [0,0].

## Example 2:

**Input:** `n = 10`

`arr[] = {6,-1,-3,4,-2,2,4,6,-12,-7}`

**Output:** 4 **Explanation:** The 4 subarrays are [-1 -3 4] [-2 2], [2 4 6 -12], and [-1 -3 4 -2 2]

## Your Task:

You don't need to read input or print anything. Complete the function **findSubarray()** that takes the array `arr` and its size `n` as input parameters and returns the total number of sub-arrays with 0 sum.

**Expected Time Complexity :**  $O(n)$

**Expected Auxilliary Space :**  $O(n)$

```
import collections
class Solution:
    #Function to count subarrays with sum equal to 0.
    def findSubArrays(self,arr,n):

        #return: count of sub-arrays having their sum equal to 0
        sumFreq = collections.defaultdict(int)
        sumFreq[0] = 1
        count = 0
        sum = 0
        for i in range(n):
            sum = sum+arr[i]
            if sum in sumFreq:
                count = count+sumFreq[sum]
                sumFreq[sum] = sumFreq[sum]+1
            else:
```

```
        sumFreq[sum] = 1  
    return count
```