Program Name- Find the number of sub-arrays having even sum

Project Category- Arrays, Mathematical

Programming Paradigm Used- Cumulative Sum, Modular Operations

Examples-

We have to find the number of subarrays whose sum is an even number.

For example the array- {1, 2, 2, 3, 4, 1} has 9 such possible subarrays-
These are-

1) {1, 2, 2, 3} → Sum = 8
2) {1, 2, 2, 3, 4} → Sum = 12
3) {2} → Sum = 2 (At index 1)
4) {2, 2} → Sum = 4
5) {2, 2, 3, 4, 1} → Sum = 12
6) {2} → Sum = 2 (At index 2)
7) {2, 3, 4, 1} → Sum = 10
8) {3, 4, 1} → Sum = 8
9) {4} → Sum = 4

$O(N^2)$ Time and O(1) Space method [Brute Force]-

We can simply generate all the possible sub-arrays and find whether the sum of all the elements in them is an even or not. If it is even then we will count that sub-array otherwise neglect it.

O(N) Time and O(1) Space Method [Efficient]-

If we do compute the cumulative sum array in **temp[]** of our input array , then we can see that the sub-array starting from **i** and ending at **j**,
has an even sum if **temp[]** if **(temp[j] – temp[i]) % 2 = 0**

Or in other words, **temp[j] = temp[i]** [By considering that modulo operation has been done]

So, instead of building a *cumulative sum array* we will build a *cumulative sum modulo 2 array*, and find how many times 0 and 1 appears in **temp[]** array using handshake formula. [n * (n-1) /2]

Note-

There are two codes of this project-

→ $O(N^2)$ Time and O(1) Space

→ O(N) Time and O(1) Space