No of subarrays with odd sum in C++

```
#include <iostream>
using namespace std;

int nos(int arr[], int n) {
    long long ans = 0;
    int even = 0;
    int odd = 0;
    int sum = 0;

    for (int i = 0; i < n; i++) {
        sum += arr[i];
        if (sum % 2 == 0) {
            ans += odd;
            even++;
        } else {
            ans += 1 + even;
            odd++;
        }
    }
    return ans % 1000000007;
}

int main() {
    int arr[] = {1, 2, 3, 4, 5, 6, 7};
    int n = sizeof(arr) / sizeof(arr[0]);

    cout << nos(arr, n) << endl;

    return 0;
}
```

Input:

arr = {1, 2, 3, 4, 5, 6, 7}

🔍 Key Variables Tracked:

- sum → cumulative sum from start to current index
- even → count of prefix sums that are even so far
- odd → count of prefix sums that are odd so far
- ans → count of subarrays with odd sum

🧮 Dry Run Table:

i	arr[i]	sum	sum%2	Action	ans	even	odd
0	1	1	1 (odd)	Add 1 + even (0) → ans += 1	1	0	1
1	2	3	1 (odd)	Add 1 + even (0) → ans += 1	2	0	2
2	3	6	0 (even)	Add odd (2) → ans += 2	4	1	2
3	4	10	0 (even)	Add odd (2) → ans += 2	6	2	2
4	5	15	1 (odd)	Add 1 + even (2) → ans += 3	9	2	3
5	6	21	1 (odd)	Add 1 + even (2) → ans += 3	12	2	4
6	7	28	0 (even)	Add odd (4) → ans += 4	16	3	4

✔ Final Output:

16