Count Of Zeros Sum Subarray in C++

```
#include <iostream>
#include <unordered_map>
#include <vector>

using namespace std;

int sol(const vector<int>& arr) {
    int count = 0;
    unordered_map<int, int> map;
    int sum = 0;
    map[0] = 1;

    for (int i = 0; i < arr.size(); ++i) {
        sum += arr[i];

        if (map.find(sum) != map.end()) {
            count += map[sum];
            map[sum]++;
        } else {
            map[sum] = 1;
        }
    }

    return count;
}

int main() {
    vector<int> arr = {2, 8, -3, -5, 2, -4, 6, 1, 2, 1,
-3, 4};
    int result = sol(arr);
    cout << result << endl;
    return 0;
}
```

Dry Run:

Initial Values:

- count = 0
- map = {0: 1}
- sum = 0

Iteration Breakdown:

i	arr[i]	sum (cumulative sum)	map[sum]	count (after update)	map (updated)
0	2	2	map[2] = 0	0	{0: 1, 2: 1}
1	8	10	map[10] = 0	0	{0: 1, 2: 1, 10: 1}
2	-3	7	map[7] = 0	0	{0: 1, 2: 1, 10: 1, 7: 1}
3	-5	2	map[2] = 1	1	{0: 1, 2: 2, 10: 1, 7: 1}
4	2	4	map[4] = 0	1	{0: 1, 2: 2, 10: 1, 7: 1, 4: 1}
5	-4	0	map[0] = 1	2	{0: 2, 2: 2, 10: 1, 7: 1, 4: 1}
6	6	6	map[6] = 0	2	{0: 2, 2: 2, 10: 1, 7: 1, 4: 1, 6: 1}
7	1	7	map[7] = 1	3	{0: 2, 2: 2, 10: 1, 7: 2, 4: 1, 6: 1}
8	2	9	map[9] = 0	3	{0: 2, 2: 2, 10: 1, 7: 2, 4: 1, 6: 1, 9: 1}
9	1	10	map[10] = 1	4	{0: 2, 2: 2, 10: 2, 7: 2, 4: 1, 6: 1, 9: 1}
10	-3	7	map[7] = 2	6	{0: 2, 2: 2, 10: 2, 7: 3, 4: 1, 6: 1, 9: 1}
11	4	11	map[11] = 0	6	{0: 2, 2: 2, 10: 2, 7: 3, 4: 1, 6: 1, 9: 1, 11: 1}

Final Values:

- **count = 6**
- **map = {0: 2, 2: 2, 10: 2, 7: 3, 4: 1, 6: 1, 9: 1, 11: 1}**

	Output: The total number of subarrays with sum equal to 0 is **6**. **Final Output:** 6
Output: 6	