

Subarray sum equals k in C++

```
#include <iostream>
#include <vector>
#include <unordered_map>
using namespace std;
class SubarraySumEqualsK {
public:
    static int sol(const std::vector<int>& arr, int target)
    {
        int ans = 0;
        std::unordered_map<int, int> map;
        map[0] = 1;
        int sum = 0;

        for (int i = 0; i < arr.size(); i++) {
            sum += arr[i];
            int rsum = sum - target;
            if (map.find(rsum) != map.end()) {
                ans += map[rsum];
            }
            map[sum]++;
        }
        return ans;
    }
};

int main() {
    vector<int> arr = {3, 9, -2, 4, 1, -7, 2, 6, -5, 8, -3, -7, 6, 2, 1};
    int k = 5;
    cout << SubarraySumEqualsK::sol(arr, k) <<
    std::endl;
    return 0;
}
```

Dry Run:

Input:

arr = {3, 9, -2, 4, 1, -7, 2, 6, -5, 8, -3, -7, 6, 2, 1}
k = 5

1. Initialize:

- ans = 0
- map = {0: 1} (We initialize with map[0] = 1 to handle the case where the subarray sum itself equals k).
- sum = 0

2. Iteration 1: i = 0, arr[0] = 3

- sum = 0 + 3 = 3
- rsum = 3 - 5 = -2
- map doesn't have -2, so ans remains 0.
- map[sum]++: map[3] = 1

3. Iteration 2: i = 1, arr[1] = 9

- sum = 3 + 9 = 12
- rsum = 12 - 5 = 7
- map doesn't have 7, so ans remains 0.
- map[sum]++: map[12] = 1

4. Iteration 3: i = 2, arr[2] = -2

- sum = 12 - 2 = 10
- rsum = 10 - 5 = 5
- map doesn't have 5, so ans remains 0.
- map[sum]++: map[10] = 1

5. Iteration 4: i = 3, arr[3] = 4

- sum = 10 + 4 = 14
- rsum = 14 - 5 = 9
- map doesn't have 9, so ans remains 0.
- map[sum]++: map[14] = 1

6. Iteration 5: i = 4, arr[4] = 1

- sum = 14 + 1 = 15
- rsum = 15 - 5 = 10
- map has 10 with count 1, so ans += 1.
- ans = 1
- map[sum]++: map[15] = 1

7. Iteration 6: i = 5, arr[5] = -7

- sum = 15 - 7 = 8
- rsum = 8 - 5 = 3
- map has 3 with count 1, so ans += 1.
- ans = 2
- map[sum]++: map[8] = 1

8. Iteration 7: i = 6, arr[6] = 2

- sum = 8 + 2 = 10
- rsum = 10 - 5 = 5
- map has 5 with count 1, so ans += 1.
- ans = 3

- $\text{map}[\text{sum}]++$: $\text{map}[10] = 2$
- 9. **Iteration 8:** $i = 7$, $\text{arr}[7] = 6$
 - $\text{sum} = 10 + 6 = 16$
 - $\text{rsum} = 16 - 5 = 11$
 - map doesn't have 11, so ans remains 3.
 - $\text{map}[\text{sum}]++$: $\text{map}[16] = 1$
- 10. **Iteration 9:** $i = 8$, $\text{arr}[8] = -5$
 - $\text{sum} = 16 - 5 = 11$
 - $\text{rsum} = 11 - 5 = 6$
 - map doesn't have 6, so ans remains 3.
 - $\text{map}[\text{sum}]++$: $\text{map}[11] = 1$
- 11. **Iteration 10:** $i = 9$, $\text{arr}[9] = 8$
 - $\text{sum} = 11 + 8 = 19$
 - $\text{rsum} = 19 - 5 = 14$
 - map has 14 with count 1, so ans += 1.
 - $\text{ans} = 4$
 - $\text{map}[\text{sum}]++$: $\text{map}[19] = 1$
- 12. **Iteration 11:** $i = 10$, $\text{arr}[10] = -3$
 - $\text{sum} = 19 - 3 = 16$
 - $\text{rsum} = 16 - 5 = 11$
 - map has 11 with count 1, so ans += 1.
 - $\text{ans} = 5$
 - $\text{map}[\text{sum}]++$: $\text{map}[16] = 2$
- 13. **Iteration 12:** $i = 11$, $\text{arr}[11] = -7$
 - $\text{sum} = 16 - 7 = 9$
 - $\text{rsum} = 9 - 5 = 4$
 - map doesn't have 4, so ans remains 5.
 - $\text{map}[\text{sum}]++$: $\text{map}[9] = 1$
- 14. **Iteration 13:** $i = 12$, $\text{arr}[12] = 6$
 - $\text{sum} = 9 + 6 = 15$
 - $\text{rsum} = 15 - 5 = 10$
 - map has 10 with count 2, so ans += 2.
 - $\text{ans} = 7$
 - $\text{map}[\text{sum}]++$: $\text{map}[15] = 2$
- 15. **Iteration 14:** $i = 13$, $\text{arr}[13] = 2$
 - $\text{sum} = 15 + 2 = 17$
 - $\text{rsum} = 17 - 5 = 12$
 - map doesn't have 12, so ans remains 7.
 - $\text{map}[\text{sum}]++$: $\text{map}[17] = 1$
- 16. **Iteration 15:** $i = 14$, $\text{arr}[14] = 1$
 - $\text{sum} = 17 + 1 = 18$
 - $\text{rsum} = 18 - 5 = 13$
 - map doesn't have 13, so ans remains 7.
 - $\text{map}[\text{sum}]++$: $\text{map}[18] = 1$

Final Answer:

After processing all the elements, the number of subarrays whose sum equals 5 is 7.

	Output:7
Output:- 7	