Longest sub-array having sum k

Given an array **arr[]** of size **n** containing integers. The problem is to find the length of the longest sub-array having sum equal to the given value **k**.

Examples:

```
Input : arr[] = { 10, 5, 2, 7, 1, 9 },
            k = 15
Output : 4
The sub-array is{5, 2, 7, 1}.

Input : arr[] = {-5, 8, -14, 2, 4, 12},
            k = -5
Output : 5
```

Efficient Approach:

Following are the steps:

1. Initialize **sum** = 0 and **maxLen** = 0.
2. Create a hash table having **(sum, index)** tuples.
3. For i = 0 to n-1, perform the following steps:
    a. Accumulate arr[i] to **sum**.
    b. If sum == k, update **maxLen** = i+1.
    c. Check whether **sum** is present in the hash table or not. If not present, then add it to the hash table as **(sum, i)** pair.
    d. Check if **(sum-k)** is present in the hash table or not. If present, then obtain index of **(sum-k)** from the hash table as **index**. Now check if maxLen < (i-index), then update **maxLen** = (i-index).
4. Return **maxLen**.

Time Complexity: O(n).

Auxiliary Space: O(n).