QN Link : <https://www.geeksforgeeks.org/problems/longest-sub-array-with-sum-k0809/1?utm_source=gfg&utm_medium=article&utm_campaign=bottom_sticky_on_article>

Step 1 : Init an empty Hashmap and vars (sum & cnt)

Step 2 : For each index cnt the sum and if sum == k it is the best answer.

Step 3 : Else check whether sum – k is present or not in the map , if so get the length by I – map.get(sum – k)

Step 4 : Get the max length at each iteration and return it.

class Solution{

// Function for finding maximum and value pair

public static int lenOfLongSubarr (int nums[], int n, int k) {

//Complete the function

Map<Integer , Integer> map = new HashMap<>();

int sum = 0;

int max = 0;

for(int i =0 ; i < n ; i++){

sum += nums[i];

if(sum == k) max = Math.max(max , i + 1);

if(map.containsKey(sum - k)){

max = Math.max(max , i - map.get(sum - k));

}

if(!map.containsKey(sum)) map.put(sum , i);

}

return max;

}

}