Qn Link : <https://www.geeksforgeeks.org/smallest-subarray-with-sum-k-from-an-array/>

public class Main {

static int smallestSubarraySumK(int arr[], int n, int K)

{

HashMap<Integer,Integer> mp = new HashMap<>();

int currPrefixSum = 0;

int result = Integer.MAX\_VALUE;

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

currPrefixSum += arr[i];

if(currPrefixSum == K){

int currLen = i + 1;

result = Math.min(result, currLen);

}

int requirePrefixSum = currPrefixSum - K;

if(mp.containsKey(requirePrefixSum)){

int foundIdx = mp.get(requirePrefixSum);

int currIdx = i;

result = Math.min(result, currIdx

- foundIdx);

}

mp.put(currPrefixSum, i);

}

if(result >= Integer.MAX\_VALUE) return -1;

// return the result

return result;

}

}