**Largest subarray with 0 sum**

Submissions: [11772](https://practice.geeksforgeeks.org/problem_submissions.php?pid=700254)  Accuracy:

46.94%

   Difficulty: [Easy](https://practice.geeksforgeeks.org/Easy/1/0/)   Marks: 2

Show Topic Tags   

[Amazon](https://practice.geeksforgeeks.org/company/Amazon/)[MakeMyTrip](https://practice.geeksforgeeks.org/company/MakeMyTrip/)

Given an array having both positive an negative integers . Your task is to complete the function**maxLen** which returns the length of maximum subarray with 0 sum . The function takes two arguments an array**A** and**n** where n is the size of the array A .   
  
**Input:**  
The first line of input contains an element T denoting the No of test cases. Then T test cases follow. Each test case consist of 2 lines. The first line of each test case contains a number denoting the size of the array A. Then in the next line are space separated values of the array A .  
  
**Output:**  
For each test case output will be the length of the largest subarray which has sum 0 .  
  
**Constraints:**  
1<=T<=100  
1<=N<=100  
-1000<=A[]<=1000  
  
**Example:  
Input**  
1  
8  
15  -2  2  -8  1  7  10 23  
  
**Output**  
5  
  
**Explanation**  
In the above test case the  largest subarray with sum 0 will be -2  2  -8  1  7  
  
**Note:**The **Input/Ouput** format and **Example** given are used for system's internal purpose, and should be used by a user for **Expected Output** only. As it is a function problem, hence a user should not read any input from stdin/console. The task is to complete the function specified, and not to write the full code.

\*\* For More Input/Output Examples Use ['Expected Output'](https://practice.geeksforgeeks.org/problems/largest-subarray-with-0-sum/1#ExpectOP) option \*\*

Contributor: Amit Khandelwal, Harshit Sidhwa  
[Author: Shubham Joshi 1](https://auth.geeksforgeeks.org/user/Shubham%20Joshi%201/practice/)

<https://practice.geeksforgeeks.org/problems/largest-subarray-with-0-sum/1>

static int MaxSubArraySum(int[] arr)

{

HashMap<Integer, Integer> dic =

new HashMap();

int max\_len = -1;

int sum = 0;

for(int i =0; i<arr.length; i++)

{

sum += arr[i];

if (arr[i] == 0 && max\_len == 0)

max\_len = 1;

if (sum == 0)

max\_len = i + 1;

if(!dic.containsKey(sum))

{

//dic[sum] = i;

dic.put(sum, i);

}

else

{

//max\_len = Math.Max(max\_len, i - dic[sum] );

//dic[sum] = i;

max\_len = Math.max(max\_len, i - dic.get(sum));

}

}

return max\_len;

}