

[Feedback](#)

[Home](#) [Product](#) [Pricing](#) [About](#) [Blog](#) [PR](#) [Contact](#)

Welcome, John Smith [Dashboard](#) | [Log out](#)



[Dashboard](#) [Create tests](#) [Tasks & Stats](#)

[Settings](#)

You are here: [Dashboard](#) > **Test: U3C7KH-YPF**

[extend](#)

[create similar test](#)

**Medium** [change](#)  
**U3C7KH-YPF**

Score:

# 75

of 100

score: 75 of 100

[1. equi](#)

Find an index in an array such that its prefix sum equals its suffix sum

### Task description

Equilibrium index of a sequence is an index such that the sum of elements at lower indexes is equal to the sum of elements at higher indexes. For example, in a sequence A:

A[0]=-7 A[1]=1 A[2]=5 A[3]=2 A[4]=-4 A[5]=3 A[6]=0  
3 is an equilibrium index, because:

A[0]+A[1]+A[2]=A[4]+A[5]+A[6]  
6 is also an equilibrium index, because:

A[0]+A[1]+A[2]+A[3]+A[4]+A[5]=0  
(sum of zero elements is zero) 7 is not an equilibrium index, because it is not a valid index of sequence A.

If you still have doubts, this is a precise definition: the integer k is an equilibrium index of a sequence A[0], A[1], ..., A[n - 1] if

and only if  $0 \leq k < n$  and  $\sum_{i=0}^{k-1} A[i] = \sum_{i=k+1}^{n-1} A[i]$ .

Assume the sum of zero elements is equal zero. Write a function  
`int equi(int[] A);`  
that given a sequence, returns its equilibrium index (any) or -1 if no equilibrium indexes exist. Assume that the sequence may be very long.

All Rights Reserved. Unauthorized copying or publication prohibited.

### Solution

Time used: 1 min.

2009-11-16 16:16:30, score: 75.00

Submitted at 2009-11-16 16:16:30, c, score: 75.00

```
01. int equi(int arr[], int n) {
02.     if (n==0) return -1;
03.
04.     int i;
05.
06.     for(i=0;i<n;i++) {
07.         long long sum_left = 0;
08.         long long sum_right = 0;
09.         int j;
10.         for(j=0;j<i;j++) sum_left += (long long)
11.             arr[j];
12.         for(j=i+1;j<n;j++) sum_right += (long long)
13.             arr[j];
14.         if (sum_left == sum_right) return i;
15.     }
16.     return -1;
17. }
```

### Analysis

Detected time complexity:

## O(n^2)

test	time	result
example Test from the task description	0.004 s.	OK
extreme_empty Empty array	0.004 s.	OK
extreme_first	0.004 s.	OK
extreme_large_numbers Sequence with extremely large numbers testing arithmetic overflow.	0.000 s.	OK
extreme_last	0.000 s.	OK
extreme_single_zero	0.004 s.	OK
extreme_sum_0 sequence with sum=0	0.004 s.	OK
simple	0.004 s.	OK
single_non_zero	0.004 s.	OK
combinations_of_two multiple runs, all combinations of {-1,0,1}^2	0.000 s.	OK
combinations_of_three	0.000 s.	OK

test	time	result
large_long_sequence_of_ones	1.000 s.	<b>TIMEOUT ERROR</b> running time: >1.00 sec., time limit: 0.10 sec.
large_long_sequence_of_minus_ones	1.008 s.	<b>TIMEOUT ERROR</b> running time: >1.01 sec., time limit: 0.10 sec.
medium_pyramid	1.004 s.	<b>TIMEOUT ERROR</b> running time: >1.00 sec., time limit: 0.10 sec.
large_pyramid Large performance test, $O(n^2)$ solutions should fail.	1.012 s.	<b>TIMEOUT ERROR</b> running time: >1.01 sec., time limit: 0.22 sec.