

Candidate Report: training8XKS3R-R4P

Check out Codility training tasks

Test Name:

Summary

Timeline

Tasks summary

| Task                 | Time spent | Score |
|----------------------|------------|-------|
| TapeEquilibrium<br>C | 7 min      | 46%   |

Total score

46%

Tasks Details

|      |  |                   |                    |                    |
|------|--|-------------------|--------------------|--------------------|
| Easy | <b>1. TapeEquilibrium</b><br>Minimize the value $ A[0] + \dots + A[P-1]] - (A[P] + \dots + A[N-1]) $ . | Task Score<br>46% | Correctness<br>57% | Performance<br>33% |
|------|--|-------------------|--------------------|--------------------|

Task description

A non-empty array A consisting of N integers is given. Array A represents numbers on a tape.

Any integer P, such that  $0 < P < N$ , splits this tape into two non-empty parts:  $A[0], A[1], \dots, A[P - 1]$  and  $A[P], A[P + 1], \dots, A[N - 1]$ .

The *difference* between the two parts is the value of:  $|A[0] + A[1] + \dots + A[P - 1]] - (A[P] + A[P + 1] + \dots + A[N - 1])|$

In other words, it is the absolute difference between the sum of the first part and the sum of the second part.

For example, consider array A such that:

```
A[0] = 3
A[1] = 1
A[2] = 2
A[3] = 4
A[4] = 3
```



We can split this tape in four places:

- P = 1, difference =  $|3 - 10| = 7$
- P = 2, difference =  $|4 - 9| = 5$
- P = 3, difference =  $|6 - 7| = 1$
- P = 4, difference =  $|10 - 3| = 7$

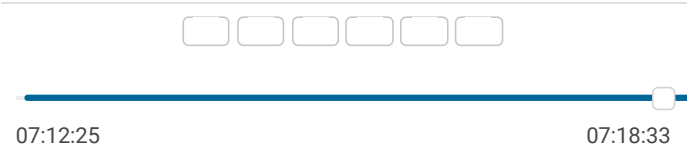
Write a function:

```
int solution(int A[], int N);
```

Solution

|                            |                 |   |
|----------------------------|-----------------|---|
| Programming language used: | C               |   |
| Total time used:           | 7 minutes       |  |
| Effective time used:       | 7 minutes       |  |
| Notes:                     | not defined yet |   |

Task timeline



Code: 07:18:33 UTC, c, final, [show code in pop-up](#)  
score: 46

```
1 // you can write to stdout for debugging purposes, e.g.
2 // printf("this is a debug message\n");
3
4 int solution(int A[], int N) {
5     // write your code in C99 (gcc 6.2.0)
6     int diff = 0, cnt = 0, tail_cnt = N-1;
7     for (int i=0; i<N; i++)
8         if (diff < 0)
```

that, given a non-empty array A of N integers, returns the minimal difference that can be achieved.

For example, given:

```
A[0] = 3
A[1] = 1
A[2] = 2
A[3] = 4
A[4] = 3
```

the function should return 1, as explained above.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [2..100,000];
- each element of array A is an integer within the range [-1,000..1,000].

Copyright 2009–2020 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Test results - Codility

```
9         diff+=A[cnt++];
10        else
11            diff-=A[tail_cnt--];
12
13        return diff<0? -diff : diff;
14    }
```

Analysis summary

The following issues have been detected: wrong answers.

For example, for the input [-2, -3, -4, -1] the solution returned a wrong answer (got 10 expected 0).

Analysis ?

| Example tests  |  |
|--|--|
| ▶ example<br>example test  | ✓ OK                                       |
| Correctness tests  |  |
| ▶ double<br>two elements   | ✓ OK                                       |
| ▶ simple_positive<br>simple test with positive numbers, length = 5               | ✓ OK                                       |
| ▶ simple_negative<br>simple test with negative numbers, length = 5               | ✗ WRONG ANSWER<br>got 27 expected 3        |
| ▶ simple_boundary<br>only one element on one of the sides                        | ✓ OK                                       |
| ▶ small_random<br>random small, length = 100                                     | ✗ WRONG ANSWER<br>got 13653 expected 39    |
| ▶ small_range<br>range sequence, length = ~1,000                                 | ✓ OK                                       |
| ▶ small<br>small elements  | ✗ WRONG ANSWER<br>got 200 expected 20      |
| Performance tests  |  |
| ▶ medium_random1<br>random medium, numbers from 0 to 100, length = ~10,000       | ✓ OK                                       |
| ▶ medium_random2<br>random medium, numbers from -1,000 to 50, length = ~10,000   | ✗ WRONG ANSWER<br>got 4731154 expected 196 |
| ▶ large_ones<br>large sequence, numbers from -1 to 1, length = ~100,000          | ✗ WRONG ANSWER<br>got 198 expected 0       |
| ▶ large_random<br>random large, length = ~100,000                                | ✗ WRONG ANSWER<br>got 108921 expected 1    |
| ▶ large_sequence<br>large sequence, length = ~100,000                            | ✓ OK                                       |
| ▶ large_extreme<br>large test with maximal and minimal values, length = ~100,000 | ✗ WRONG ANSWER<br>got 2000 expected 0      |

The PDF version of this report that may be downloaded on top of this site may contain sensitive data including personal information. For security purposes, we recommend you remove it from your system once reviewed.