

Candidate Report: trainingE9KQV2-E9Z

[Check out Codility training tasks](#)

Test Name:

Summary Timeline

Tasks summary

Task	Time spent	Score
PassingCars Java 8	83 min	100%

Total score

100%

?

 Identity verification
Add another layer of security with Identity Verification, [learn more](#).

Tasks Details

Easy	1. PassingCars	Task Score	Correctness	Performance
	Count the number of passing cars on the road.	100%	100%	100%

Task description

A non-empty array A consisting of N integers is given. The consecutive elements of array A represent consecutive cars on a road.

Array A contains only 0s and/or 1s:

- 0 represents a car traveling east,
- 1 represents a car traveling west.

The goal is to count passing cars. We say that a pair of cars (P, Q), where $0 \leq P < Q < N$, is passing when P is traveling to the east and Q is traveling to the west.

For example, consider array A such that:

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

We have five pairs of passing cars: (0, 1), (0, 3), (0, 4), (2, 3), (2, 4).

Write a function:

```
class Solution { public int solution(int[] A); }
```

that, given a non-empty array A of N integers, returns the number of pairs of passing cars.

The function should return -1 if the number of pairs of passing cars exceeds 1,000,000,000.

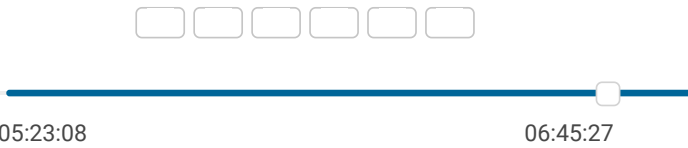
For example, given:

A[0] = 0
A[1] = 1

Solution

Programming language used:	Java 8	
Total time used:	83 minutes	?
Effective time used:	83 minutes	?
Notes:	not defined yet	

Task timeline



Code: 06:45:26 UTC, java, [show code in pop-up](#)
final, score: 100

```
1 // you can also use imports, for example:
2 // import java.util.*;
3
4 // you can write to stdout for debugging purposes,
5 // System.out.println("this is a debug message");
6
7 class Solution {
8     public int solution(int[] A) {
9         // write your code in Java SE 8
10         int n=0;
11         int p=0;
12         for (int i=0;i<A.length;i++){
```

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

the function should return 5, as explained above.

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

- N is an integer within the range [1..100,000];
- each element of array A is an integer that can have one of the following values: 0, 1.

Copyright 2009–2021 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

```
13         if(A[i]==0){
14             n++;
15         }
16         else{
17             p=p+n;
18         }
19     }
20     if(p>1000000000 || p<0){
21         return -1;
22     }
23     return p;
24 }
25 }
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: **O(N)**

expand all	Example tests	
▶ example		✓ OK
example test		
expand all	Correctness tests	
▶ single		✓ OK
single element		
▶ double		✓ OK
two elements		
▶ simple		✓ OK
simple test		
▶ small_random		✓ OK
random, length = 100		
▶ small_random2		✓ OK
random, length = 1000		
expand all	Performance tests	
▶ medium_random		✓ OK
random, length = ~10,000		
▶ large_random		✓ OK
random, length = ~100,000		
▶ large_big_answer		✓ OK
0..01..1, length = ~100,000		
▶ large_alternate		✓ OK
0101..01, length = ~100,000		
▶ large_extreme		✓ OK
large test with all 1s/0s, length = ~100,000		

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.