

AVAILABLE  
LESSONS:*Lesson 1*  
Iterations*Lesson 2*  
Arrays*Lesson 3*  
Time Complexity*Lesson 4*  
**Counting  
Elements***Lesson 5*  
Prefix Sums*Lesson 6*  
Sorting*Lesson 7*  
Stacks and  
Queues*Lesson 8*  
Leader*Lesson 9*  
Maximum slice  
problem*Lesson 10*  
Prime and  
composite  
numbers*Lesson 11*

PAINLESS

## PermCheck

START

Check whether array A is a permutation.

Programming language: C++ ▼

A non-empty zero-indexed array A consisting of N integers is given.

A *permutation* is a sequence containing each element from 1 to N once, and only once.

For example, array A such that:

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

is a permutation, but array A such that:

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

is not a permutation, because value 2 is missing.

The goal is to check whether array A is a permutation.

Write a function:

```
int solution(vector<int> &A);
```

that, given a zero-indexed array A, returns 1 if array A is a permutation and 0 if it is not.

For example, given array A such that:

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

the function should return 1.

Sieve of  
Eratosthenes

*Lesson 12*

Euclidean  
algorithm

*Lesson 13*

Fibonacci  
numbers

*Lesson 14*

Binary search  
algorithm

*Lesson 15*

Caterpillar  
method

*Lesson 16*

Greedy  
algorithms

*Lesson 17*

Dynamic  
programming

*Lesson 90*

Tasks from  
Indeed Prime  
2016 challenge

*Lesson 99*

Future training

Given array A such that:

$A[0] = 4$

$A[1] = 1$

$A[2] = 3$

the function should return 0.

Assume that:

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

Complexity:

- expected worst-case time complexity is  $O(N)$ ;
- expected worst-case space complexity is  $O(N)$ , beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.

Copyright 2009–2016 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

## For programmers

[Lessons](#) [Challenges](#)  
[Terms](#) [FAQ](#)

## For companies

[Tour](#) [About us](#)  
[Pricing](#) [Jobs](#)  
[Blog](#) [Terms](#)  
[Privacy](#) [Cookies](#)  
[API](#)

## Sign up for our newsletter:

Information about upcoming challenges, solutions and lessons directly in your inbox.

[Sign up](#)

## Social:

[f](#) [t](#) [in](#)

## Contact us:

For customer support queries:

UK +44 (0) 208 970 78 68

US 1-415-466-8085  
[support@codility.com](mailto:support@codility.com)

For sales queries:

UK +44 (0) 208 970 78 67

US 1-415-466-8085  
[sales@codility.com](mailto:sales@codility.com)

---

© 2009–2016 Codility Ltd., registered in England and Wales (No. 7048726). VAT ID GB981191408. Registered office: 107 Cheapside, London EC2V 6DN