

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*

RESPECTABLE

MaxCounters

START

Calculate the values of counters after applying all alternating operations:
increase counter by 1; set value of all counters to current maximum.

Programming language: C++ ▼

You are given N counters, initially set to 0, and you have two possible operations on them:

- *increase(X)* – counter X is increased by 1,
- *max counter* – all counters are set to the maximum value of any counter.

A non-empty zero-indexed array A of M integers is given. This array represents consecutive operations:

- if $A[K] = X$, such that $1 \leq X \leq N$, then operation K is *increase(X)*,
- if $A[K] = N + 1$ then operation K is *max counter*.

For example, given integer $N = 5$ and array A such that:

```
A[0] = 3
A[1] = 4
A[2] = 4
A[3] = 6
A[4] = 1
A[5] = 4
A[6] = 4
```

the values of the counters after each consecutive operation will be:

```
(0, 0, 1, 0, 0)
(0, 0, 1, 1, 0)
(0, 0, 1, 2, 0)
(2, 2, 2, 2, 2)
(3, 2, 2, 2, 2)
(3, 2, 2, 3, 2)
(3, 2, 2, 4, 2)
```

The goal is to calculate the value of every counter after all operations.

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

Write a function:

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

that, given an integer N and a non-empty zero-indexed array A consisting of M integers, returns a sequence of integers representing the values of the counters.

The sequence should be returned as:

- a structure Results (in C), or
- a vector of integers (in C++), or
- a record Results (in Pascal), or
- an array of integers (in any other programming language).

For example, given:

```
A[0] = 3  
A[1] = 4  
A[2] = 4  
A[3] = 6  
A[4] = 1  
A[5] = 4  
A[6] = 4
```

the function should return [3, 2, 2, 4, 2], as explained above.

Assume that:

- N and M are integers within the range [1..100,000];
- each element of array A is an integer within the range [1..N + 1].

Complexity:

- expected worst-case time complexity is O(N+M);
- 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
Pricing us
Blog Jobs
Privacy Terms
API Cookies

**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

For sales queries:

UK +44 (0) 208 970
78 67

US 1-415-466-8085
sales@codility.com

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