

AVAILABLE
LESSONS:Lesson 1
IterationsLesson 2
ArraysLesson 3
Time ComplexityLesson 4
Counting
ElementsLesson 5
Prefix SumsLesson 6
SortingLesson 7
Stacks and
QueuesLesson 8
LeaderLesson 9
Maximum slice
problemLesson 10
Prime and
composite
numbers

Lesson 11

PAINLESS

CountSemiprimes

START

Count the semiprime numbers in the given range [a..b]

Programming language: C++ ▼

A *prime* is a positive integer X that has exactly two distinct divisors: 1 and X . The first few prime integers are 2, 3, 5, 7, 11 and 13.

A *semiprime* is a natural number that is the product of two (not necessarily distinct) prime numbers. The first few semiprimes are 4, 6, 9, 10, 14, 15, 21, 22, 25, 26.

You are given two non-empty zero-indexed arrays P and Q , each consisting of M integers. These arrays represent queries about the number of semiprimes within specified ranges.

Query K requires you to find the number of semiprimes within the range $(P[K], Q[K])$, where $1 \leq P[K] \leq Q[K] \leq N$.

For example, consider an integer $N = 26$ and arrays P, Q such that:

$$P[0] = 1 \quad Q[0] = 26$$
$$P[1] = 4 \quad Q[1] = 10$$
$$P[2] = 16 \quad Q[2] = 20$$

The number of semiprimes within each of these ranges is as follows:

- (1, 26) is 10,
- (4, 10) is 4,
- (16, 20) is 0.

Write a function:

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

that, given an integer N and two non-empty zero-indexed arrays P and Q consisting of M integers, returns an array consisting of M elements specifying the consecutive answers to

**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
2015 challenge*Lesson 91*Tasks from
Indeed Prime
2016 challenge*Lesson 92*Tasks from
Indeed Prime
2016 College
Coders
challenge*Lesson 99*

all the queries.

For example, given an integer $N = 26$ and arrays P, Q such that: $P[0] = 1 \quad Q[0] = 26$ $P[1] = 4 \quad Q[1] = 10$ $P[2] = 16 \quad Q[2] = 20$ the function should return the values $[10, 4, 0]$, as explained above.

Assume that:

- N is an integer within the range $[1..50,000]$;
- M is an integer within the range $[1..30,000]$;
- each element of arrays P, Q is an integer within the range $[1..N]$;
- $P[i] \leq Q[i]$.

Complexity:

- expected worst-case time complexity is $O(N \cdot \log(\log(N)) + M)$;
- expected worst-case space complexity is $O(N + M)$, 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
Cookies
API

**Sign up for our
newsletter:**

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

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