

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

Ladder

START

Count the number of different ways of climbing to the top of a ladder.

Programming language: C++ ▼

You have to climb up a ladder. The ladder has exactly N rungs, numbered from 1 to N . With each step, you can ascend by one or two rungs. More precisely:

- with your first step you can stand on rung 1 or 2,
- if you are on rung K , you can move to rungs $K + 1$ or $K + 2$,
- finally you have to stand on rung N .

Your task is to count the number of different ways of climbing to the top of the ladder.

For example, given $N = 4$, you have five different ways of climbing, ascending by:

- 1, 1, 1 and 1 rung,
- 1, 1 and 2 rungs,
- 1, 2 and 1 rung,
- 2, 1 and 1 rungs, and
- 2 and 2 rungs.

Given $N = 5$, you have eight different ways of climbing, ascending by:

- 1, 1, 1, 1 and 1 rung,
- 1, 1, 1 and 2 rungs,
- 1, 1, 2 and 1 rung,
- 1, 2, 1 and 1 rung,
- 1, 2 and 2 rungs,
- 2, 1, 1 and 1 rungs,
- 2, 1 and 2 rungs, and
- 2, 2 and 1 rung.

The number of different ways can be very large, so it is sufficient to return the result modulo 2^P , for a given integer P .

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

Write a function:

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

that, given two non-empty zero-indexed arrays A and B of L integers, returns an array consisting of L integers specifying the consecutive answers; position I should contain the number of different ways of climbing the ladder with A[I] rungs modulo $2^{B[I]}$.

For example, given L = 5 and:

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

the function should return the sequence [5, 1, 8, 0, 1], as explained above.

Assume that:

- L is an integer within the range [1..30,000];
- each element of array A is an integer within the range [1..L];
- each element of array B is an integer within the range [1..30].

Complexity:

- expected worst-case time complexity is O(L);
- expected worst-case space complexity is O(L), 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