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.

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.

Future training