Codeforces Round #167 (Div. 2)

A. Dima and Friends

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Dima and his friends have been playing hide and seek at Dima's place all night. As a result, Dima's place got messy. In the morning they decided that they need to clean the place.

To decide who exactly would clean the apartment, the friends want to play a counting-out game. First, all the guys stand in a circle, and then each of them shows some number of fingers on one hand (one to five), and then the boys count in a circle, starting from Dima, the number of people, respective to the total number of fingers shown. The person on who the countdown stops will clean the apartment.

For example, if Dima and one of his friends played hide and seek, and 7 fingers were shown during the counting-out, then Dima would clean the place. If there were 2 or say, 8 fingers shown, then his friend would clean the place.

Dima knows how many fingers each of his friends will show during the counting-out. Now he is interested in the number of ways to show some number of fingers on one hand (one to five), so that he did not have to clean the place. Help Dima.

Input

The first line contains integer n ($1 \le n \le 100$) — the number of Dima's friends. Dima himself isn't considered to be his own friend. The second line contains n positive integers, not exceeding 5, representing, how many fingers the Dima's friends will show.

The numbers in the lines are separated by a single space.

Output

In a single line print the answer to the problem.

Examples

input
1 1
output
3

input
1 2
output
2

input
2 3 5
output
3

Note

In the first sample Dima can show 1, 3 or 5 fingers. If Dima shows 3 fingers, then the counting-out will go like that: Dima, his friend, Dima, his friend.

In the second sample Dima can show 2 or 4 fingers.

B. Dima and Sequence

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Dima got into number sequences. Now he's got sequence $a_1, a_2, \ldots, a_n$, consisting of n positive integers. Also, Dima has got a function $f(x)$, which can be defined with the following recurrence:

- $f(0) = 0$;
- $f(2 \cdot x) = f(x)$;
- $f(2 \cdot x + 1) = f(x) + 1$.

Dima wonders, how many pairs of indexes (i, j) $(1 \le i < j \le n)$ are there, such that $f(a_i) = f(a_j)$. Help him, count the number of such pairs.

Input

The first line contains integer n $(1 \le n \le 10^5)$. The second line contains n positive integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^9)$.

The numbers in the lines are separated by single spaces.

Output

In a single line print the answer to the problem.

Please, don't use the `%lld` specifier to read or write 64-bit integers in *C++*. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Examples

input
3 1 2 4
output
3

input
3 5 3 1
output
1

Note

In the first sample any pair (i, j) will do, so the answer is 3.

In the second sample only pair $(1, 2)$ will do.

C. Dima and Staircase

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Dima's got a staircase that consists of n stairs. The first stair is at height a_1, the second one is at a_2, the last one is at a_n ($1 \le a_1 \le a_2 \le \ldots \le a_n$).

Dima decided to play with the staircase, so he is throwing rectangular boxes at the staircase from above. The i-th box has width w_i and height h_i. Dima throws each box vertically down on the first w_i stairs of the staircase, that is, the box covers stairs with numbers $1, 2, \ldots, w_i$. Each thrown box flies vertically down until at least one of the two following events happen:

- the bottom of the box touches the top of a stair;
- the bottom of the box touches the top of a box, thrown earlier.

We only consider touching of the horizontal sides of stairs and boxes, at that touching with the corners isn't taken into consideration. Specifically, that implies that a box with width w_i cannot touch the stair number $w_i + 1$.

You are given the description of the staircase and the sequence in which Dima threw the boxes at it. For each box, determine how high the bottom of the box after landing will be. Consider a box to fall after the previous one lands.

Input

The first line contains integer n ($1 \le n \le 10^5$) — the number of stairs in the staircase. The second line contains a non-decreasing sequence, consisting of n integers, $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$; $a_i \le a_{i+1}$).

The next line contains integer m ($1 \le m \le 10^5$) — the number of boxes. Each of the following m lines contains a pair of integers w_i, h_i ($1 \le w_i \le n$; $1 \le h_i \le 10^9$) — the size of the i-th thrown box.

The numbers in the lines are separated by spaces.

Output

Print m integers — for each box the height, where the bottom of the box will be after landing. Print the answers for the boxes in the order, in which the boxes are given in the input.

Please, do not use the `%lld` specifier to read or write 64-bit integers in *C++*. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Examples

input
5 1 2 3 6 6 4 1 1 3 1 1 1 4 3
output
1 3 4 6

input
3 1 2 3 2 1 1 3 1
output
1 3

input
1 1 5 1 2 1 10 1 10 1 10 1 10

output
1 3 13 23 33

Note

The first sample are shown on the picture.

D. Dima and Two Sequences

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Dima has two sequences of points with integer coordinates: sequence $(a_1, 1), (a_2, 2), \ldots, (a_n, n)$ and sequence $(b_1, 1), (b_2, 2), \ldots, (b_n, n)$.

Now Dima wants to count the number of distinct sequences of points of length $2 \cdot n$ that can be assembled from these sequences, such that the x-coordinates of points in the assembled sequence will **not decrease**. Help him with that. Note that each element of the initial sequences should be used exactly once in the assembled sequence.

Dima considers two assembled sequences $(p_1, q_1), (p_2, q_2), \ldots, (p_{2 \cdot n}, q_{2 \cdot n})$ and $(x_1, y_1), (x_2, y_2), \ldots, (x_{2 \cdot n}, y_{2 \cdot n})$ distinct, if there is such i $(1 \le i \le 2 \cdot n)$, that $(p_i, q_i) \ne (x_i, y_i)$.

As the answer can be rather large, print the remainder from dividing the answer by number m.

Input

The first line contains integer n $(1 \le n \le 10^5)$. The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^9)$. The third line contains n integers $b_1, b_2, \ldots, b_n$ $(1 \le b_i \le 10^9)$. The numbers in the lines are separated by spaces.

The last line contains integer m $(2 \le m \le 10^9 + 7)$.

Output

In the single line print the remainder after dividing the answer to the problem by number m.

Examples

input
1 1 2 7
output
1

input
2 1 2 2 3 11
output
2

Note

In the first sample you can get only one sequence: $(1, 1), (2, 1)$.

In the second sample you can get such sequences : $(1, 1), (2, 2), (2, 1), (3, 2)$; $(1, 1), (2, 1), (2, 2), (3, 2)$. Thus, the answer is 2.

E. Dima and Horses

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Dima came to the horse land. There are n horses living in the land. Each horse in the horse land has several enemies (enmity is a symmetric relationship). The horse land isn't very hostile, so the number of enemies of each horse is at most 3.

Right now the horse land is going through an election campaign. So the horses trusted Dima to split them into two parts. At that the horses want the following condition to hold: a horse shouldn't have more than one enemy in its party.

Help Dima split the horses into parties. Note that one of the parties can turn out to be empty.

Input

The first line contains two integers n, m $(1 \le n \le 3 \cdot 10^5; 0 \le m \le \min(3 \cdot 10^5, \frac{n(n-1)}{2}))$ — the number of horses in the horse land and the number of enemy pairs.

Next m lines define the enemy pairs. The i-th line contains integers a_i, b_i $(1 \le a_i, b_i \le n;\ a_i \ne b_i)$, which mean that horse a_i is the enemy of horse b_i.

Consider the horses indexed in some way from 1 to n. It is guaranteed that each horse has at most three enemies. No pair of enemies occurs more than once in the input.

Output

Print a line, consisting of n characters: the i-th character of the line must equal "0", if the horse number i needs to go to the first party, otherwise this character should equal "1".

If there isn't a way to divide the horses as required, print -1.

Examples

input
3 3 1 2 3 2 3 1
output
100

input
2 1 2 1
output
00

input
10 6 1 2 1 3 1 4 2 3 2 4 3 4
output
0110000000