Problem A. A Place For My Head

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

Artem has constructed a permutation p of n integers from 1 to n during the last Jinotega beer party. Kostya tried to guess it, but had no success. Then Artem gave some hints to Kostya: for each $i = 1, \ldots, n$, he told him two integers $l_i \le r_i$, meaning that element i is situated between positions l_i and r_i inclusively. Kostya is sure that the lexicographically smallest permutation satisfying these conditions is the one Artem has constructed. As it is too late, and Kostya's brain is not working as well as usually after several glasses of beer, you need to help him.

Input

The first line of input contains a positive integer n ($1 \le n \le 2 \cdot 10^5$), the length of the permutation.

Each of the next n lines contain two integers l_i and r_i ($1 \le l_i \le r_i \le n$).

Output

If it is impossible to reconstruct the permutation, print a line containing the integer -1.

Otherwise, print a line containing n space-separated integers $p_1, p_2, \ldots, p_n$: the desired permutation.

Examples

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

Problem B. New Divide

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2.5 seconds
Memory limit:	512 mebibytes

Consider an array of k integers $b_1, b_2, \ldots, b_k$. Let $x \oplus y$ be the bitwise exclusive OR of x and y. We shall say the *linear power* of the array b is

$$LP(b) = \max_{i=0,1,\ldots,k} (b_1 \oplus \ldots \oplus b_i) + (b_{i+1} \oplus \ldots \oplus b_k).$$

You are given an array a of n integers. Find the linear power of all its prefixes.

Input

The first line contains a positive integer n ($1 \le n \le 10^6$), the length of the array.

The second line contains n integers a_i ($0 \le a_i \le 10^6$).

Output

Output a single line containing n space-separated integers: $LP(a_1)$, $LP(a_1, a_2)$, …, $LP(a_1, a_2, \ldots, a_n)$.

Examples

standard input	standard output
5 1 2 3 4 5	1 3 6 10 9
10 11 13 14 14 9 8 0 10 10 7	11 24 20 24 15 23 23 17 23 30

Problem C. Lying From You

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

You are given n lines on the plane, defined by equations of the form $y = a_i x + b_i$. You can change the coefficients of a line from (a, b) to (a', b') at the cost of $|a - a'| + |b - b'|$ rubles. You can do this operation an arbitrary number of times with arbitrary lines, and the resulting coefficients can be any real numbers. Your goal is to make all the lines share a common point.

Let C be the set of total costs of operations leading to the goal. Find $\inf C$, that is, the tight lower bound for the total cost.

Input

The first line contains a positive integer n ($1 \le n \le 10^5$), the number of lines.

Each of the next n lines contains two integers a_i and b_i ($|a_i|, |b_i| \le 10^6$).

Output

Print the answer on a single line with absolute or relative error no more than 10^{-6}.

Examples

standard input	standard output
3 0 0 1 -1 -1 0	0.500000000000000
5 4 1 3 0 3 1 2 0 1 2	3.000000000000000

Note

In the first example, it is enough to change b of the first line to -0.5.

Problem D. Don't Stay

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

There are infinitely many lamps on a line, numbered with integers. Each lamp can be either turned on or turned off. There is a lamplighter, initially located at lamp number 0. He can walk one lamp left or right (that is, decrease or increase his coordinate by 1) and toggle the lamp at his position. We denote these actions with letters "L", "R" and "X", respectively.

A string of letters "L", "R" and "X" is called a program for the lamplighter. For example, if he is currently at position 0 and receives the string "RRXL", he goes right twice, toggles a lamp which is at position 2, and then goes left and ends up at position 1.

Initially, all lamps are turned off. In the end, you want to achieve a state where the lamps with coordinates $a_1, \ldots, a_n$ are turned on, and all other lamps are turned off. The lamplighter has some program s which he is going to execute, and he does not want to change his plans a lot. However, he agreed to help you a bit. If you give him some program t, he will start by executing all instructions from t. Then he will execute his own program s, as he planned before. However, after that, he will try to *cancel* his help by performing all instructions from t in reverse order, changing "L" to "R" and vice versa. See Notes section for an explanation by example.

Given the coordinates a_i and the lamplighter's program s, find such program t that in the end, lamps $a_1, \ldots, a_n$ are on and all other lamps are off, or determine that it is not possible.

Input

The first line of input contains a string s ($1 \le |s| \le 2 \cdot 10^5$) consisting of characters "L", "R" and "X": the initial program of the lamplighter.

The second line contains an integer n ($0 \le n \le 2 \cdot 10^5$), the number of lamps which should be turned on in the end. The third line contains n space-separated integers: the coordinates of the lamps. All given coordinates are distinct and do not exceed $2 \cdot 10^5$ by absolute value.

Output

Output the string t which you can give to the lamplighter to achieve your goal. If there are several possible answers, output any one of them. The answer may be empty. The length of the answer must not exceed $2 \cdot 10^6$ characters.

If achieving the goal is not possible, print a single word "NO" (without quotes).

Example

standard input	standard output
RXR 3 -2 0 2	XLLXR

Note

In the example, the lamplighter has the string "RXR", and the desired lamp positions are -2, 0 and 2. You can give the lamplighter the string "XLLXR". His combined program then becomes "XLLXR-RXR-LXRRX" (dashes for clarity).

The lamplighter will light the lamp number 0, then go to -2 and light it, go to 0 and turn it off, turn it on again, and finally go to 2 and turn it on. In the end, the state of all lamps is the one you wanted.

Problem E. In The End

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2.5 seconds
Memory limit:	512 mebibytes

There is a board which is infinite in one direction. The board consists of n rows and an infinite number of columns: there is the first (leftmost) column but no last column. Each column contains exactly one cell with a cake: the probability that the cake is in the i-th row is p_i, and all p_i sum up to 1. Positions of cakes in different columns are independent.

You control a robot which starts in some cell of the first column. On each step, if the robot is at cell (x, y) (which means x-th row and y-th column), the robot can move to $(x, y+1)$, $(x-1, y+1)$ or $(x+1, y+1)$, if such cell exists. Whenever the robot visits a cell with a cake in it, that cake is collected.

The robot wants to collect as many cakes as possible. Given the probabilities p_i, find the average number of cakes the robot will collect on each step.

Assume that the order of events is as follows. First, all cakes are placed on the board according to the given probabilities. Then, the configuration of the board is given to the robot. After that, the robot chooses a starting cell and a movement plan in order to maximize the average number of collected cakes.

Formally, consider the sequence of boards of sizes $n \times m$ for $m = 1, 2, \ldots$. For each such finite board, the robot receives the configuration and then chooses an optimal route. Let $f(m)$ be the expected average number of cakes collected on an $n \times m$ board. Your task is to calculate the limit

$$\lim_{m \to \infty} \frac{f(m)}{m}.$$

Input

The first line contains an integer n ($1 \le n \le 6$), the number of rows.

The next line contains n real numbers p_i ($0 \le p_i \le 1$) given with at most one digit after the decimal point: the probability distribution.

Output

It can be proven that the answer to the problem can be expressed as a fraction $\frac{P}{Q}$ for some positive integers P and Q. Find such P and Q, and print the number $(P \cdot Q^{-1}) \bmod (10^9 + 7)$.

Examples

standard input	standard output
2 0.5 0.5	1
3 0.3 0.3 0.4	804545461
4 0.2 0.3 0.2 0.3	84928504

Problem F. From The Inside

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 mebibytes

Consider a rectangular board of $n \times m$ cells. Each cell can be either black or white. Initially, all cells are white. Additionally, we fix an integer k such that $k \le n, m \le 3k$.

Alice and Bob play the following game: each player, in turn, picks a completely white square of $k \times k$ cells and paints it black. The player who cannot make a valid move loses. Alice goes first.

Eve wonders how many possible Alice's first moves lead to Alice's victory if both players continue optimally. Help her find this number.

Input

The only line of input contains three positive integers n, m and k ($1 \le n, m, k \le 10^9$). There is an additional **important condition**: $k \le n, m \le 3k$.

Output

Print a single line with a single integer: the answer to the problem.

Examples

standard input	standard output
`2 3 1`	`0`
`3 3 2`	`4`

Note

In the first example, there are exactly six moves in total, no matter how the players act. In the second example, Alice can place her square wherever she wants, and Bob will lose immediately.

Problem G. Numb

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

You are given an even integer n. Construct a binary number $a = \overline{a_1 a_2 \dots a_n}$ consisting of n binary digits such that it is divisible by n, and all numbers $\overline{a_1 a_2 \dots a_i}$ (the prefixes of a in binary notation) for $i = 1, 2, \dots, n$ have different remainders modulo n.

Input

The only line of input contains an integer n ($2 \le n \le 1000$, n is even).

Output

Print the desired number $\overline{a_1 a_2 \dots a_n}$ as a string of n binary digits. Leading zeroes are disallowed. If there are several possible answers, print any one of them. It is guaranteed that at least one answer exists under these constraints.

Examples

standard input	standard output
2	10
4	1100

Problem H. One Step Closer

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Consider a table of size $n \times m$ consisting of characters "`+`" and "`-`". A single operation with the table is to choose a cell and invert all characters in its row and column simultaneously (the cell itself is also inverted). Inverting a character means replacing "`+`" by "`-`" and vice versa.

Vasya wants to fill all cells with "`-`". To achieve that, he uses the following algorithm:

- If all cells are "`-`", end the algorithm.
- Otherwise, remember the current positions of "`+`". Assume they are (r_1, c_1), (r_2, c_2), $\ldots$, (r_k, c_k). After that, he performs one operation for each cell (r_i, c_i) and proceeds to step 1.

Sasha is interested whether Vasya can reach his goal. If this is the case, he wants to know the total number of operations Vasya will perform before he stops.

Input

The first line of input contains three integers n, m and q ($1 \le n, m \le 10^5$, $0 \le q \le 10^5$).

Then follows the description of the initial configuration which looks as follows. The i-th of the next q lines contains four integers $x_{i,1}$, $y_{i,1}$, $x_{i,2}$ and $y_{i,2}$ ($1 \le x_{i,1} \le x_{i,2} \le n$, $1 \le y_{i,1} \le y_{i,2} \le m$). They define a rectangle containing all such cells (x, y) that $x_{i,1} \le x \le x_{i,2}$ and $y_{i,1} \le y \le y_{i,2}$. A cell (x, y) is initially "`+`" if and only if it is contained in an odd number of rectangles.

See Notes section for further explanation.

Output

If Vasya will never finish performing operations, print "`-1`". Otherwise, print the number of operations he will perform.

Examples

standard input	standard output

`3 3 2`
`1 1 2 2`
`2 2 3 3`
	`6`

`3 3 2`
`1 2 1 3`
`2 1 3 1`
	`-1`

Note

In the first example, the board initially looks as follows:

```
++-
+-+
-++
```

After the only iteration of the algorithm, which consists of six operations, the board is filled with "`-`".

In the second example, the board looks as follows and stays the same after each iteration of the algorithm:

```
-++
+--
+--
```

Problem I. Invisible

Input file:	*standard input*
Output file:	*standard output*
Time limit:	20 seconds
Memory limit:	512 mebibytes

This is an interactive problem.

You are given an array $a_1, \ldots, a_n$ of n integers. You have to process queries of two types:

- "`1` x y": change value of a_x to y.
- "`2` l r": find and print a number that occurs an odd number of times in the segment $a_l, a_{l+1}, \ldots, a_r$, or determine that there is no such number and print -1 instead. If there are several suitable numbers, you can output any of them.

Interaction Protocol

Your program must read from standard input and write to standard output.

The input starts with two lines: the first line contains an integer n ($1 \le n \le 10^5$), and the second line contains n space-separated integers a_1, a_2, $\ldots$, a_n ($1 \le a_i \le 10^5$).

Then follow query block descriptions.

Each query block description starts with a line containing the number of queries q_i. If this number is 0, the solution must immediately terminate gracefully. Otherwise, q_i query descriptions follow, one per line.

Each query description is either "`1` x y" ($1 \le x \le n$, $1 \le y \le 10^5$) or "`2` l r" ($1 \le l \le r \le n$).

For each query of type 2, print the answer on a separate line. You must answer all queries of type 2 in the current block, if there are any, to receive the next block. Don't forget to flush your output after printing the answers, for example, using `fflush (stdout)` in C or C++ and `System.out.flush ()` in Java. Otherwise, the most likely outcome will be "`Idleness Limit Exceeded`".

The total number of queries in all blocks does not exceed 10^5. It is guaranteed that there is at least one query of type 2.

Example

standard input	standard output
5 1 2 2 3 3 2 2 1 5 1 1 3 3 2 1 4 1 5 4 2 2 5 0	1 -1 3

Problem J. Leave Out All The Rest

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

You are given two integer arrays: an array a of length n and an array b of length m. All integers in both arrays are **pairwise distinct**.

An *interleaving* of the two arrays is an array c of size $n + m$ such that arrays a and b are its disjoint subsequences. Formally, there exist indices $i_1 < i_2 < \ldots < i_n$ such that $c_{i_1} = a_1$, $c_{i_2} = a_2$, $\ldots$, $c_{i_n} = a_n$, and indices $j_1 < j_2 < \ldots < j_m$ such that $c_{j_1} = b_1$, $c_{j_2} = b_2$, $\ldots$, $c_{j_m} = b_m$. For these indices, $i_x \neq j_y$ for all $x = 1, 2, \ldots, n$ and all $y = 1, 2, \ldots, m$.

It is clear that there are usually many ways to interleave arrays a and b. Find such a way that maximizes the length of the longest increasing subsequence of c.

Input

The first line of input contains integer n ($1 \le n \le 5 \cdot 10^5$) — the length of array a.

The second line contains n integers a_i ($1 \le a_i \le 10^9$).

The third line of input contains integer m ($1 \le m \le 5 \cdot 10^5$) — the length of array b.

The fourth line contains m integers b_j ($1 \le b_j \le 10^9$).

It is guaranteed that the numbers in both arrays are pairwise distinct: $a_i \neq a_j$ for $i \neq j$, $b_i \neq b_j$ for $i \neq j$ and $a_i \neq b_j$ for all valid i and j.

Output

Output one integer: the maximum length of the longest increasing subsequence in an interleaving of a and b.

Examples

standard input	standard output
2 1 7 3 6 10 11	5
3 7 1 5 3 9 8 6	3

Problem K. Faint

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 mebibytes

Consider the set of integers $\{1, 2, \ldots, n\}$ and all its subsets of size k. Let us write each such subset as $a_1, a_2, \ldots, a_k$ where $a_1 < a_2 < \ldots < a_k$. We write them down in lexicographical order row by row, obtaining a table with $\binom{n}{k}$ rows and k columns. Let the j-th integer in the i-th row be $A_{i,j}$.

Your task is to calculate the sum of absolute differences between consecutive numbers in a given column. Formally, given a positive integer m $(1 \le m \le k)$, calculate the sum

$$\sum_{i=1}^{\binom{n}{k}-1} |A_{i,m} - A_{i+1,m}|.$$

As the answer can be very large, print it modulo $10^9 + 7$.

Input

The only line of input contains three positive integers n, k and m $(1 \le n \le 10^6, 1 \le m \le k \le n)$.

Output

Print a single line with a single integer: the answer to the problem modulo $10^9 + 7$.

Examples

standard input	standard output
4 2 2	4
5 3 2	4

Note

In the first example, the table looks as follows.

$$\begin{matrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 2 & 3 \\ 2 & 4 \\ 3 & 4 \end{matrix}$$

The answer is $|2 - 3| + |3 - 4| + |4 - 3| + |3 - 4| + |4 - 4| = 4$.