Problem A. After Two Hares

Input file:	stdin
Output file:	stdout
Time limit:	3 seconds
Memory limit:	256 megabytes

Pavel and Alexey play a strange game. The game takes place on the rectangle field of size $n \times m$ cells, each cell is either empty or a wall.

The rules of the game are the following: Alexey chooses two empty cells on the field, and a hare appears on both of these cells, then Pavel throws a bomb on some empty cell (x, y) (maybe in that cell where the hare just appeared). The bomb explodes immediately, producing the blast waves in four directions: up, down, left and right. Blast waves destroy everything on their way until they reach the wall or the field's border. Of course, the cell (x, y) itself is also exploded. If the blast waves can kill two hares at once, Pavel wins, otherwise hares disappear from the game field and Alexey becomes the winner. Note that Alexey can summon both the hares in the same cell (nobody knows why he's doing that).

Nikita has seen his friends playing this strange game and decided to give hints to Pavel. He won't say anybody the cell (x, y), but determining the possibility of Pavel's victory quickly and correctly is easy to him: Nikita says «YES» if Pavel can win, and «NO» otherwise.

In total q rounds of the game were played. Repeat Nikita's answers in each of them.

Input

The first line contains two space-separated integers n and m ($1 \le n, m \le 2500$) — numbers of rows and columns on the game field.

Each of the next n lines contains m characters describing the game field. If the character is equal to «#», the corresponding cell is a wall, and if the character is «.», then this cell is empty.

Next line contains one integer q ($1 \le q \le 800000$) — number of played rounds.

Each of the next q lines contains four integers r_1, c_1, r_2, c_2 ($1 \le r_1, r_2 \le n$, $1 \le c_1, c_2 \le m$) — coordinates of the cells where the first and the second hare were summoned.

Output

Write q lines. i-th line should be «YES» (without the quotes) if Pavel can win in the i-th round, and «NO» (without the quotes) otherwise.

Examples

stdin	stdout
5 4 .#.. ..#. #... ..#. .#.. 7 1 1 1 1 2 2 4 2 4 1 2 2 1 3 5 4 4 1 2 4 1 3 3 2 3 4 5 1	YES YES YES YES NO NO NO

Problem B. Birthday Cake

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

Alexey has a birthday soon. Pavel and Nikita want to celebrate that significant event so they've bought a birthday cake.

Nobody knows why, but the cake is flat and has the form of a strictly convex polygon. Nikita suggested to divide it into two equal parts — one for Pavel and one for himself. Pavel declined that suggestion because it was, in his opinion, insufficiently beautiful from a mathematical point of view — and made another suggestion — to choose two vertices of the polygon to make a cut along the straight line that connects them.

Nikita realized that it's not always possible to make this cut to obtain two equal parts. That's why he offer to cut the cake in such a way that the difference between the areas of the resulting parts is minimal. Your task is to help Pavel and Nikita to find such a cut.

Input

The first line contains the only integer n ($4 \le n \le 200000$) — number of vertices in the cake-polygon.

Each of the next n lines contains two space-separated integers x_i and y_i ($-10^8 \le x_i, y_i \le 10^8$) — coordinates of i-th vertex. Vertices are listed in counter-clockwise order.

It's guaranteed that the polygon is strictly convex.

Output

Write two integers — numbers of vertices to be chosen to make a cut along the straight line that connects them. Difference between the areas of the resulting parts should be minimal. If there are many possible answers, output any of them.

Examples

stdin	stdout
4 0 0 1 0 1 1 0 1	1 3
5 0 1 0 0 2 -1 2 1 1 2	2 4

Problem C. Graph Restoration

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

Square matrix $\{d_{ij}\}$ of size $n \times n$ is called a shortest path matrix of the weighted directed graph with n vertices, if each its element d_{ij} is the length of the shortest path from vertex i to vertex j.

You are given a square matrix in which all the elements on the main diagonal are zeroes, and the remaining elements are positive integers. Determine if that matrix can be a shortest path matrix of some weighted directed graph without self-loops and multiple edges.

Input

The first line contains one integer n ($1 \le n \le 300$) — the size of the matrix.

Then n lines are following. i-th line contains n space-separated integers $d_{i1}, \ldots, d_{in}$ — elements of the i-th row of the matrix. All the elements on the main diagonal are zeroes, and all the other elements are integers from 1 to 10^9.

Output

If there is no graph with such shortest path matrix, write «-1».

Otherwise write its adjacency matrix. Each its element a_{ij} should be zero if there is no edge from vertex i to vertex j, otherwise it should be equal to weight of this edge. All non-zero elements a_{ij} should satisfy the condition $1 \le a_{ij} \le 10^9$.

If there are many possible solutions, write any of them.

Examples

stdin	stdout
3 0 1 2 1 0 3 2 3 0	0 1 2 1 0 4 2 5 0
4 0 2 4 3 5 0 3 1 4 2 0 2 2 1 4 0	-1

Problem D. Insomnia

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

Recently Alexey discovered a new way to control the insomnia. He takes a positive integer k and tries to divide it without a remainder to get number 1. At each step Alexey can divide the current number by any its divisor which is not equal to 1.

For example, if $k = 12$, there are many such ways: $12 \to 6 \to 1$, or $12 \to 4 \to 2 \to 1$, or even $12 \to 1$. And the total number of such ways is 8.

Alexey is interested how many ways exist to get 1 from k by such consecutive divisions without the remainder. Two ways $k \to a_1 \to \cdots \to a_n \to 1$ and $k \to b_1 \to \cdots \to b_m \to 1$ are considered different, if $n \neq m$ or if exist such i that $a_i \neq b_i$.

Input

The input contains the only integer k ($1 \leq k \leq 10^6$).

Output

Write the number of ways to get 1 from k by consecutive divisions without the remainder. It's guaranteed that this number does not exceed 10^9.

Examples

stdin	stdout
1	1
4	2
7	1
10	3
12	8

Problem E. Counterfeiters

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

Two counterfeiters each made a coin: one of these coins has the probability of getting heads $p\%$, and another one — $q\%$.

For some reason the counterfeiters bet that if one chooses the random of these two coins, he gets heads two times in a row. They do so: choose the random coin and throw it. At the first throw they really got heads. Now they want to know: what is the probability to getting heads at the second throw of the chosen coin?

Input

The only line of the input contains two integers p and q ($0 \le p, q \le 100$, $p + q > 0$) — probabilities (in percent) of getting heads on the first and on the second coin correspondingly.

Output

Write one real number — the probability of getting heads at the second throw. Absolute or relative error of the answer should not exceed 10^{-6}.

Examples

stdin	stdout
50 50	0.500000000000000
33 66	0.550000000000000
80 40	0.666666666666667

Problem F. Asperger Syndrome

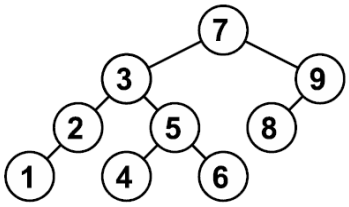
Input file:	`stdin`
Output file:	`stdout`
Time limit:	2 seconds
Memory limit:	256 megabytes

World famous scientist Innokentiy has Asperger syndrome. Therefore, his entertainments are not like the usual people's ones. For example, now he is fascinated by the following game: he write out a permutation of numbers from 1 to n and then consequentially inserts the elements of permutation into the unbalanced binary search tree.

Binary search tree is a binary tree which has the following properties:

- For every node in the tree both the left and right its subtrees are binary search trees.
- The left subtree of each node x contains only nodes with keys *less* than the key of node x.
- The right subtree of each node x contains only nodes with keys *no less* than the key of node x.

For example, the following picture illustrates the binary search tree with 9 elements:



When the element is being added into the unbalanced binary search tree, it descends from the root of the tree to its leaves. At every step of this descent it is being compared to the value of the current node. If new element is less than this value, it will be added into the left subtree, otherwise — into the right subtree. Finally, if the current node is the empty tree, it becomes a new node with value of new element.

Innokentiy wondered how many comparisons will be made during the insertion of all the elements of permutation into the tree. Unfortunately, processing some permutations takes a long time. Knowing the permutation, help Innokentiy to find the number he wondered very quickly.

Input

The first line contains the only integer n ($1 \le n \le 200000$) — number of elements of the permutation.

The second line contains n pairwise distinct integers $p_1, \ldots, p_n$ ($1 \le p_i \le n$) separated by spaces — the elements of the permutation.

Output

Write one integer — the total number of comparisons that will be made during the insertion of elements of the permutation into the unbalanced binary search tree.

Examples

stdin	stdout
3 2 1 3	2
3 1 2 3	3
4 2 4 3 1	4

Problem G. The Last Wish

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

One day Pavel felt asleep and got inside the computer program. He saw a huge amount of its different inhabitants: vectors, strings and even garbage collector. It turned out that each of these objects has its own mind, has its own family, place, feelings... However, it doesn't matter.

Once upon a time non-empty string that consisted of only lowercase Latin letters lived. Unfortunately, there are no references to it anymore and it should be the next garbage collector's prey. Before being sacrificed, it asked Pavel to replace the minimal number of symbols in it such that all its symbols remained lowercase Latin letters, but were pairwise distinct. Help sleeping Pavel to fulfill the string's last wish.

Input

The input contains the only non-empty string consists of only lowercase Latin letters. Its length does not exceed 200000.

Output

If Pavel is not able to fulfill the last wish, write «IMPOSSIBLE» (without the quotes).

Otherwise write the modified string. All its symbols should be pairwise distinct and be the lowercase Latin letters. Number of changes in the string should be minimal. If there are many possible solutions, write any of them.

Examples

stdin	stdout
abcba	abcde
xxxxyyyy	xabcydef
theanswerforthistestisimpossible	IMPOSSIBLE

Problem H. Game with the Stones

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

Constantine and Mike like to sort things out with the help of strange games. This time they play the following game: n stone piles, i-th of which contains a_i stones, lie on the table. Players move in rotation. Each player in his turn splits each pile that contains two or more stones into two non-empty parts. If before player's turn each pile contains exactly one stone, this player loses.

Constantine moves first. Determine who will win if both players play optimally.

Input

The first line contains the only integer n ($1 \le n \le 100$) — number of stone piles on the table.

The second line contains n space-separated integers $a_1, \ldots, a_n$ ($1 \le a_i \le 10^9$) — numbers of stones in the piles.

Output

Write «Constantine» (without the quotes) if Constantine will win, and «Mike» (without the quotes) if the winner will be Mike.

Examples

stdin	stdout
4 2 4 6 8	Constantine
4 7 5 3 1	Mike

Problem I. Retakes

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

It's well known that student Nikita is a procrastinator. He has failed n exams, therefore he has to retake them. It's possible to retake no more than one exam per day.

i-th exam can be retaken at any day starting from the day t_i. However, lecturers don't want to see students doing nothing during the term. That's why they take money for retakes.

Retake of i-th exam costs $c_i + d$ roubles, where c_i is the initial cost of the retake and d is the number of days passed from the day when this exam became available to retake. Thus, if student retakes the exam just as it becomes available, he expends c_i roubles, if he retakes it at the next day — he expends $c_i + 1$ roubles, and so on.

Nikita is poor, lives at the dormitory and, in general, works for food, so he has no money. So every time when he needs money, he borrows from Pavel. Namely, every time when he decides to retake the exam, he comes to Pavel, borrows exactly that sum that he needs to pay for retake, no more and no less, and goes to retake.

Pavel has the special form of amnesia: he remembers only maximal debt. For example, if one borrows from him 100 roubles, then 200 roubles, then 200 roubles one more time, and, finally, 150 roubles, Pavel will be absolutely sure that the debt is equal to 200 roubles. Knowing this fact, make the retake schedule for Nikita in such a way that the sum of money to refund to Pavel is minimal. That is, for every exam find a day when Nikita should retake it. It's known that Nikita retakes any exam at the first attempt.

Input

The first line contains the only integer n ($1 \le n \le 200000$) — number of exams that Nikita should retake.

Each of the following n lines contains two integers t_i and c_i ($1 \le t_i \le 10^6$, $0 \le c_i \le 10^9$) separated by space — the first day the i-th exam can be retaken from, and its initial cost. Numbers $t_1, \ldots, t_n$ are given in non-descending order, i.e. $t_i \le t_j$ for all $i \le j$.

Output

Write n integers separated by spaces. i-th number should be a day when Nikita should retake i-th exam. If there are many schedules such that the debt is minimal, output any of them.

Examples

stdin	stdout
3 1 3 1 1 2 4	1 3 2
4 3 5 4 3 4 6 5 3	3 5 4 6

Problem J. Product Innovation

Input file:	`stdin`
Output file:	`stdout`
Time limit:	2 seconds
Memory limit:	256 megabytes

Everybody knows about pointless discoveries of British scientists. Everybody knows how Japanese scientists like huge battle humanoid robots. This problem tells about French scientists who have managed to unite both of these: recently robot «Renatus Cartesius», though not battle, but huge and totally pointless, was released.

«Renatus Cartesius» can move on the integer array and insert new elements on both sides of itself. Science community decided to check how effective the new creation of French scientists is and prepared for the robot q commands to execute.

At the beginning of the experiment «Renatus Cartesius» is on the p-th position of the array of n integers ($1 \le p \le n$). It can execute commands of 5 types:

- `print` — print the element the robot is now standing on;
- `moveLeft` — move one position to the left if there are elements there;
- `moveRight` — move one position to the right if there are elements there;
- `insertLeft` x — insert a new element x to the left (between positions $p-1$ and p, where p is a position the robot is now standing on);
- `insertRight` x — insert a new element x to the right (between positions p and $p+1$, where p is a position the robot is now standing on).

Unfortunately, community members themselves don't know the expected results of execution this sequence of commands by the robot. That's why they asked you to help.

Knowing the sequence of commands, write correct answers to the `print` commands.

Input

The first line contains two space-separated integers n and p ($1 \le n \le 200000$, $1 \le p \le n$) — number of elements in the array and position where the robot initially stands.

The second line contains n space-separated integers $a_1, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$) — elements of the array on which the robot stands.

The third line contains an integer q ($1 \le q \le 200000$) — number of commands which the robot should execute.

Each of the next q lines contains a command. The descriptions of the commands are given above. Value of x in commands `insertLeft` and `insertRight` is in the range from -10^9 to 10^9.

Output

For each `print` command write the array's element on which the robot is standing at that moment. Each answer should be written on the separate line.

Examples

stdin	stdout
3 2 2 3 5 15 moveLeft insertLeft 1 moveLeft print moveLeft moveRight print moveRight insertRight 4 moveRight moveRight print moveRight moveLeft print	1 2 5 4

Problem K. Bracket Sequence

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

Let us define a regular bracket sequence in a following way:

- empty string is a regular bracket sequence;
- if s is a regular bracket sequence, then «(s)» is a regular bracket sequence;
- if s и t are regular bracket sequences, then «st» is a regular bracket sequence.

You are given a bracket sequence. Make it regular by adding minimal number of symbols to its end. If it's impossible, write «IMPOSSIBLE».

Input

The input contains the only one non-empty string. It consists of brackets «(» and «)» and its length does not exceed 200000.

Output

If it's impossible to make the sequence regular by adding symbols to its end, write «IMPOSSIBLE».

Otherwise write a regular bracket sequence obtained from the initial sequence by adding minimal number of symbols to the end.

Examples

stdin	stdout
((	(())
(()())	(()())
())(()	IMPOSSIBLE
((()((	((()(()()))

Problem L. Hard Problem

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

Array $s_1, \ldots, s_n$ is called as array of prefix sums of array $a_1, \ldots, a_n$, if $s_1 = a_1$, $s_2 = a_1 + a_2$, $\ldots$, $s_n = a_1 + \cdots + a_n$.

Pavel had an array $a_1, \ldots, a_n$. He calculated its array of prefix sums but, since his memory is not infinite, he forgot the initial array. Help Pavel to restore the initial array.

Input

The first line contains one integer n ($1 \le n \le 200000$) — number of elements in the array.

The second line contains n integers $s_1, \ldots, s_n$ ($-2 \cdot 10^{14} \le s_i \le 2 \cdot 10^{14}$) separated by space — elements of array of prefix sums.

Output

Output n integers separated by space — elements of the initial array $a_1, \ldots, a_n$. It is guaranteed that all a_i are in range from -10^9 to 10^9.

Examples

stdin	stdout
4 1 3 6 10	1 2 3 4
5 -4 -1 3 2 4	-4 3 4 -1 2
3 1000000000 2000000000 3000000000	1000000000 1000000000 1000000000

Problem M. Jumping along the Hummocks

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

Crazy Frog lives on the swamp where n hummocks of different colors stay along the straight line. Each hummock has a color that is an integer from 1 to 200000.

At the beginning Crazy Frog stands at the first hummock and wants to reach the last, n-th hummock. He can jump:

- just one hummock to the right;
- if he stands on a hummock of color c, he can jump to closest hummock of color c that is to the right of him, if such a hummock exist.

What is the minimal number of jumps that Crazy Frog should make to reach the n-th hummock?

Input

The first line contains the only integer n ($1 \le n \le 200000$) — number of hummocks on the swamp.

The second line contains n integers $c_1, \ldots, c_n$ ($1 \le c_i \le 200000$) separated by space. They are colors of the hummocks.

Output

Output one integer — the minimal number of jumps required to reach the n-th hummock.

Examples

stdin	stdout
5 1 2 3 4 5	4
6 1 8 1 1 7 1	3
6 1 2 3 1 3 2	2