Problem A. Brackets

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.5 seconds
Memory limit:	256 megabytes

A bracket symbol is one of the following: `()[]{}<>`. A correct *bracket expression* is any string consisting of bracket symbols, such that:

- Every left bracket has a matching right bracket of the same kind, and every right bracket is matched;
- No two pairs of matching brackets cross – for every two such pairs, they are either disjoint or one is contained inside the other.

For example, `([])<>` is a correct bracket expression, whereas `<{>}` is not, as the curly brackets and angle brackets cross each other.

You are given a graph of n vertices in which every (directed) edge is labeled with one of the bracket symbols. A path in this graph is *valid*, if its edges form a correct bracket expression. For some two vertices s and t, determine the length of a shortest valid path between s and t. We allow the path to pass multiple times through any vertex.

Input

On the first line of input there are four integers n, m, s, t ($1 \le n \le 200$, $0 \le m \le 2000$, $1 \le s, t \le n$) – the number of vertices, edges, starting and ending vertex, respectively. Each of the following m lines contains two integers x, y and a bracket symbol b ($1 \le x, y \le n$), which describe one graph edge. Note that there may be loops and multiple edges.

Output

Output a single line containing a single integer – the length of the shortest valid path between s and t. If there is no such path, output -1. You may assume that if a path exists, its length does not exceed 10^{18}.

Examples

standard input	standard output

`4 4 1 4`
`1 2 (`
`2 2 [`
`2 3 ]`
`3 4 )`
	`4`

`5 4 1 5`
`1 2 <`
`2 3`
`3 4 >`
`4 5`
	`-1`

Problem B. New Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

In SUA, there are n trees planted in the city park. Recently, Chiaki has planted a new tree in the park and she would like to protect it. To do so, she creates a protected area around the new tree by selecting three of the old trees, and encircling them with a band. The new tree must be strictly inside the protected area but no other tree is allowed to be there.

Chiaki has already selected one of the old trees. Help her find the other two.

Input

The first line contains two integers n and a ($3 \le n \le 10^5, 1 \le a \le n$) – the number of old trees and the identifier of the preselected old tree. The trees are identified by the numbers $1, 2, \ldots, n$.

The second line contains two integers x and y ($-10^6 \le x, y \le 10^6$) – the x-and y-coordinates of the new tree.

Each of the next n lines contains two integers x_i and y_i ($-10^6 \le x_i, y_i \le 10^6$) – the coordinates of an old tree.

No two trees will be planted in the same location.

Output

The first line of the output must contain two integers b and c separated by a single space, where b and c are old tree identifiers with the following property: if a is the identifier of the preselected old tree, then the triangle with nodes a, b and c (in counterclockwise order) forms a valid protected area. That is, there are no trees on the sides of the triangle other than a, b and c, and the only tree strictly inside the triangle is the new tree. If there is no solution then the output must be 0 0. If there are multiple solutions, your can output any of them.

Example

standard input	standard output
7 1 9 3 3 1 8 7 9 5 11 5 12 4 9 1 13 6	6 4

Problem C. Next Permutation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.5 seconds
Memory limit:	256 megabytes

A permutation $p_1, p_2, \ldots, p_n$ of the natural numbers $1, 2, \ldots, n$ is called $3-1-2$ pattern avoiding if there are no three indices $1 \le i < j < k \le n$ such that $p_i > p_j$, $p_i > p_k$ and $p_j < p_k$.

Chiaki has a $3-1-2$ pattern avoiding permutation. She would like to find the next $3-1-2$ pattern avoiding permutation according to the lexicographic ordering.

Input

The first line contains one integer n $(3 \le n \le 10000)$. The second line contains n positive integers separated by single spaces, a $3-1-2$ pattern avoiding permutation of the natural numbers $1, 2, \ldots, n$. The input will not be the decreasing sequence $n, n-1, \ldots, 1$.

Output

The first line of the output must contain the $3-1-2$ pattern avoiding permutation that follows the input permutation in the lexicographic ordering.

Example

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

Problem D. Posters

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 256 megabytes

Chiaki is a devoted fan of his favorite music band. She attends every concert and collects their posters. Each time, when she gets a new poster, she hangs it on the wall, above her bed. After many years of collecting posters, almost the whole wall has been covered with them and now Chiaki cannot find space for the new ones. She just got some new posters to hang and she needs your help to find the best place for them on the wall. For each poster and its placement, Chiaki would like to know how much this poster would cover other posters.

You are given the coordinates of the posters which already hang on the wall and the coordinates of the posters which do not hang yet, but Chiaki considers hanging them. For each new poster, find the area of the parts of hanging posters which would be covered directly by this poster.

The posters on the wall may overlap, and if their intersection is covered, you shouldn't count its area twice.

Input

On the first line of input there is one integer n ($1 \le n \le 10^5$) – the number of posters which hang on the wall.

In the next n lines, there are descriptions of those posters. In $n+2$ line there is one integer m ($1 \le m \le 10^5$) – the number of new posters which Chiaki would like to hang on the wall. In the next m lines, there are descriptions of those posters.

Each poster is a rectangle with edges parallel to axis. It is described by four integers x_1, y_1, x_2, y_2 ($0 \le x_1 < x_2 \le 10^9, 0 \le y_1 < y_2 \le 10^9$), denoting the coordinates of the bottom left corner and the top right corner.

Output

For each new poster output one line containing an integer – the answer for the Chiaki's problem. The answers should be printed in the same order as the posters were given in the input.

Example

standard input	standard output
2 0 1 3 5 2 3 6 6 2 1 0 5 4 4 2 7 7	8 6

Problem E. Sorting

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Chiaki has some files containing integers. She wants to sort integers in each file in ascending order. Chiaki is an IT specialist so, of course, she tried to find a command line tool that would do his task. The name of a tool wasn't hard to guess, but it didn't work as Chiaki expected – after sorting the files, she realized that this tool was treating every integer as a string and it sorted them lexicographically.

She knew that such a thing could happen, but she was surprised anyway – some files were still sorted in ascending order.

Now, Chiaki wonders how lucky she was and how was even possible that integers from these files could have had same lexicographical and numerical order.

Given a range of integers $[A, B]$, determine the number of subsets of those integers, that their lexicographical and numerical orders are equal.

Input

On the first and only line of input there are two integers A and B ($1 \le A \le B \le 10^{18}, B - A \le 10^5$).

Output

Output a single line with integer M, where M is the number of subsets of set $\{A, A+1, \dots B\}$, which keep specified condition. As the answer may be really big, output it modulo $10^9 + 7$.

Example

standard input	standard output
98 101	7

Note

Those subsets are: $\{\}, \{98\}, \{99\}, \{100\}, \{101\}, \{98, 99\}, \{100, 101\}$.

Problem F. Robot Race

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

A robot race on a maze will be held in SUA. The maze has a shape of a rectangle and is divided into $n \times m$ fields arranged in n rows and m columns. Each of the fields is either empty or contains an obstacle.

The contestants register their robots for the competition and come to the maze one after another. Each of the contestants gets random coordinates of the initial field and the target field. The robot is then placed in the initial field and must get to the target field using a sequence of steps.

To make the game more challenging, the rules specify that in each step the robot can only move one field right or one field down in the maze. Moving the robot in any other direction is not allowed.

The contestant whose robot gets fastest from the initial field to the target field wins the competition. If the robot does not get to the target field in the time limit, the contestant in disqualified.

The organizers of the competition realized that if the contestant gets bad coordinates, the robot will not be able to get to the target field using any sequence of moves. In that case they would like to give the contestant another pair of coordinates.

You are given a map of a maze of size $n \times m$ and q pairs of coordinates of the initial and target fields. Determine for each pair of coordinates whether it is possible to get from the initial field to the target field using a sequence of steps right or down.

Input

The first line contains three integers n, m and q $(1 \le n, m \le 1000, 1 \le q \le 10^6)$ – the number of rows and columns of the maze and the number of pairs of coordinates.

Each of the following n lines contains m characters describing the fields of the maze. The character `.` represents an empty field and the character `#` a field with an obstacle.

Then q lines follow describing the pairs of coordinates. The i-th of them contains four integers r_{1i}, c_{1i}, r_{2i}, c_{2i} $(1 \le r_{1i}, r_{2i} \le n$, $1 \le c_{1i}, c_{2i} \le m)$ – the row and the column of the initial field and the row and the column of the target field. For all $i \in \{1, 2, \ldots, q\}$ the fields in the maze with coordinates (r_{1i}, c_{1i}) and (r_{2i}, c_{2i}) are empty.

Output

Output q lines. The i-th line should contain the word `YES` if the robot can get from the i-th initial field to the i-th target field, or the word `NO` otherwise.

Example

standard input	standard output
3 4 5 .#.. .##. 1 1 3 4 1 3 3 4 1 1 1 1 1 1 2 4 2 1 2 4	YES YES YES NO NO