Problem Capybara. Zip-line

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Vasya has decided to build a zip-line on trees of a nearby forest. He wants the line to be as long as possible but he doesn't remember exactly the heights of all trees in the forest. He is sure that he remembers correct heights of all trees except, possibly, one of them.

It is known that the forest consists of n trees staying in a row numbered from left to right with integers from 1 to n. According to Vasya, the height of the i-th tree is equal to h_i. The zip-line of length k should hang over k ($1 \le k \le n$) trees $i_1, i_2, \ldots, i_k$ ($i_1 < i_2 < \ldots < i_k$) such that their heights form an increasing sequence, that is $h_{i_1} < h_{i_2} < \ldots < h_{i_k}$.

Petya had been in this forest together with Vasya, and he now has q assumptions about the mistake in Vasya's sequence h. His i-th assumption consists of two integers a_i and b_i indicating that, according to Petya, the height of the tree numbered a_i is actually equal to b_i. Note that Petya's assumptions are **independent** from each other.

Your task is to find the maximum length of a zip-line that can be built over the trees under each of the q assumptions.

In this problem the length of a zip line is considered equal to the number of trees that form this zip-line.

Input

The first line of the input contains two integers n and m ($1 \le n, m \le 400\,000$) — the number of the trees in the forest and the number of Petya's assumptions, respectively.

The following line contains n integers h_i ($1 \le h_i \le 10^9$) — the heights of trees according to Vasya.

Each of the following m lines contains two integers a_i and b_i ($1 \le a_i \le n$, $1 \le b_i \le 10^9$).

Output

For each of the Petya's assumptions output one integer, indicating the maximum length of a zip-line that can be built under this assumption.

Examples

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

Note

Consider the first sample. The first assumption actually coincides with the height remembered by Vasya. In the second assumption the heights of the trees are $(4, 2, 3, 4)$, in the third one they are $(1, 2, 3, 3)$ and in the fourth one they are $(1, 2, 3, 5)$.

Scoring

Tests for this problem are divided into seven groups. For each of the groups 1–5 you earn points only if your solution passes all tests in this group and all tests in all **previous** groups.

In the group 6 all tests are graded independently and each of them costs 1 point. The solution is tested on this group only if all tests of all previous are passed.

Subtask	Tests	Points	Constraints			Comment
			n	m	h_i, b_i	
0	1 – 2	0	–	–	–	Sample tests
1	3 – 32	10	$n \leq 15$	$m \leq 15$	$h_i, b_i \leq 100$	
2	33 – 51	10	$n \leq 500$	$m \leq 500$	$h_i, b_i \leq 500$	
3	52 – 70	20	$n \leq 2000$	$m \leq 3000$	$h_i, b_i \leq 100\,000$	
4	71 – 89	20	$n \leq 10\,000$	$m \leq 20\,000$	$h_i, b_i \leq 100\,000$	
5	–	20	$n \leq 75\,000$	$m \leq 75\,000$	–	**Offline evaluation**
6	–	20	–	–	–	**Offline evaluation**

Problem Echidna. Watchmen

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

Watchmen are in a danger and Doctor Manhattan together with his friend Daniel Dreiberg should warn them as soon as possible. There are n watchmen on a plane, the i-th watchman is located at point (x_i, y_i).

They need to arrange a plan, but there are some difficulties on their way. As you know, Doctor Manhattan considers the distance between watchmen i and j to be $|x_i - x_j| + |y_i - y_j|$. Daniel, as an ordinary person, calculates the distance using the formula $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.

The success of the operation relies on the number of pairs (i, j) ($1 \le i < j \le n$), such that the distance between watchman i and watchmen j calculated by Doctor Manhattan is equal to the distance between them calculated by Daniel. You were asked to compute the number of such pairs.

Input

The first line of the input contains the single integer n ($1 \le n \le 200\,000$) — the number of watchmen.

Each of the following n lines contains two integers x_i and y_i ($|x_i|, |y_i| \le 10^9$).

Output

Print the number of pairs of watchmen such that the distance between them calculated by Doctor Manhattan is equal to the distance calculated by Daniel.

Examples

standard input	standard output
3 1 1 7 5 1 5	2
6 0 0 0 1 0 2 -1 1 0 1 1 1	11

Note

In the first sample, the distance between watchman 1 and watchman 2 is equal to $|1 - 7| + |1 - 5| = 10$ for Doctor Manhattan and $\sqrt{(1 - 7)^2 + (1 - 5)^2} = 2 \cdot \sqrt{13}$ for Daniel. For pairs $(1, 1)$, $(1, 5)$ and $(7, 5)$, $(1, 5)$ Doctor Manhattan and Daniel will calculate the same distances.

Scoring

Tests for this problem are divided into three groups. For each group you earn points only if your solution passes all tests in this group and all tests in all **previous** groups.

Group	Tests	Points	Additional constrains		Comment
			n	X_i, Y_i	
0	$1 - 2$	0	–	–	Samples
1	$3 - 23$	50	$1 \leqslant n \leqslant 1000$	$-10\,000 \leqslant x_i, y_i \leqslant 10\,000$	–
2	–	50	–	–	–

Problem Python. Clockwork Bomb

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2.5 seconds
Memory limit:	256 megabytes

My name is James diGriz, I'm the most clever robber and treasure hunter in the whole galaxy. There are books written about my adventures and songs about my operations, though you were able to catch me up in a pretty awkward moment.

I was able to hide from cameras, outsmart all the guards and pass numerous traps, but when I finally reached the treasure box and opened it, I have accidentally started the clockwork bomb! Luckily, I have met such kind of bombs before and I know that the clockwork mechanism can be stopped by connecting contacts with wires on the control panel of the bomb in a certain manner.

I see n contacts connected by $n-1$ wires. Contacts are numbered with integers from 1 to n. Bomb has a security mechanism that ensures the following condition: if there exist $k \geq 2$ contacts $c_1, c_2, \ldots, c_k$ forming a circuit, i. e. there exist k **distinct** wires between contacts c_1 and c_2, c_2 and c_3, …, c_k and c_1, then the bomb immediately explodes and my story ends here. In particular, if two contacts are connected by more than one wire they form a circuit of length 2. It is also prohibited to connect a contact with itself.

On the other hand, if I disconnect more than one wire (i. e. at some moment there will be no more than $n-2$ wires in the scheme) then the other security check fails and the bomb also explodes. So, the only thing I can do is to unplug some wire and plug it into a new place ensuring the fact that no circuits appear.

I know how I should put the wires in order to stop the clockwork. But my time is running out! Help me get out of this alive: find the sequence of operations each of which consists of unplugging some wire and putting it into another place so that the bomb is defused.

Input

The first line of the input contains n ($2 \leq n \leq 500\,000$), the number of contacts.

Each of the following $n-1$ lines contains two of integers x_i and y_i ($1 \leq x_i, y_i \leq n$, $x_i \neq y_i$) denoting the contacts currently connected by the i-th wire.

The remaining $n-1$ lines contain the description of the sought scheme in the same format.

Output

The first line should contain k ($k \geq 0$) — the minimum number of moves of unplugging and plugging back some wire required to defuse the bomb.

In each of the following k lines output four integers a_i, b_i, c_i, d_i meaning that on the i-th step it is neccesary to unplug the wire connecting the contacts a_i and b_i and plug it to the contacts c_i and d_i. Of course the wire connecting contacts a_i and b_i should be present in the scheme.

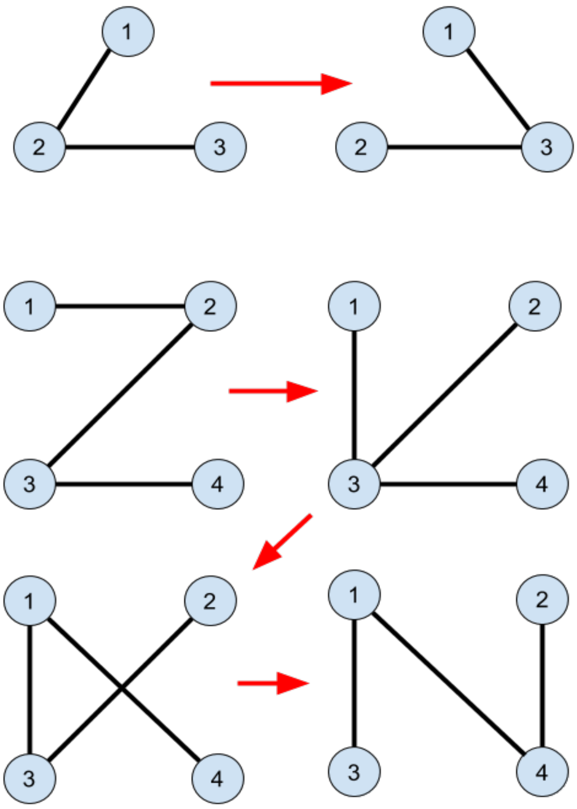
If there is no correct sequence transforming the existing scheme into the sought one, output `-1`.

Examples

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

Note

Picture with the clarification for the sample tests:



Scoring

Tests for this problem are divided into six groups. For each group you earn points only if your solution passes all tests in this group and all tests in all **previous** groups.

Group	Tests	Points	Additional constraints	Comment
			n	
0	1 – 2	0	—	Sample tests
1	3 – 18	20	$n \leq 50$	It is guaranteed that the answer exists and requires no more than one operation
2	19 – 37	20	$n \leq 50$	
3	38 – 53	20	$n \leq 5000$	
4	54 – 69	20	$n \leq 100\,000$	
5	–	20	–	**Offline evaluation**

Problem Unicorn. Table Compression

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

Little Petya is now fond of data compression algorithms. He has already studied *gz*, *bz*, *zip* algorithms and many others. Inspired by the new knowledge, Petya is now developing the new compression algorithm which he wants to name *dis*.

Petya decided to compress tables. He is given a table a consisting of n rows and m columns that is filled with positive integers. He wants to build the table a' consisting of positive integers such that the relative order of the elements in each row and each column remains the same. That is, if in some row i of the initial table $a_{i,j} < a_{i,k}$, then in the resulting table $a'_{i,j} < a'_{i,k}$, and if $a_{i,j} = a_{i,k}$ then $a'_{i,j} = a'_{i,k}$. Similarly, if in some column j of the initial table $a_{i,j} < a_{p,j}$ then in compressed table $a'_{i,j} < a'_{p,j}$ and if $a_{i,j} = a_{p,j}$ then $a'_{i,j} = a'_{p,j}$.

Because large values require more space to store them, the maximum value in a' should be as small as possible.

Petya is good in theory, however, he needs your help to implement the algorithm.

Input

The first line of the input contains two integers n and m ($1 \leqslant n, m$ and $n \cdot m \leqslant 1\,000\,000$), the number of rows and the number of columns of the table respectively.

Each of the following n rows contain m integers $a_{i,j}$ ($1 \leqslant a_{i,j} \leqslant 10^9$) that are the values in the table.

Output

Output the compressed table in form of n lines each containing m integers.

If there exist several answers such that the maximum number in the compressed table is minimum possible, you are allowed to output any of them.

Examples

standard input	standard output
2 2 1 2 3 4	1 2 2 3
4 3 20 10 30 50 40 30 50 60 70 90 80 70	2 1 3 5 4 3 5 6 7 9 8 7

Note

In the first sample test, despite the fact $a_{1,2} \neq a_{21}$, they are not located in the same row or column so they may become equal after the compression.

Scoring

Tests for this problem are divided into eight groups. For each group you earn points only if your solution passes all tests of this group and all tests of some of the previous groups. **Offline evaluation** means that your submission will be evaluated on the tests of the group only after the end of the contest.

Group	Tests	Points	Additional constraints			Required groups	Comment
			n	m	$a_{i,j}$		
0	1 – 2	0	–			–	Sample tests
1	3 – 19	10	$n \leqslant 1000$	$m = 1$	–	–	
2	20 – 39	15	$n, m \leqslant 100$		All $a_{i,j}$ are distinct	–	
3	40 – 74	15	$n, m \leqslant 100$		–	0, 2	
4	75 – 84	15	$n, m \leqslant 400$		All $a_{i,j}$ are distinct	2	
5	85 – 102	15	$n, m \leqslant 400$		–	0, 2, 3, 4	
6	103 – 112	15	–		All $a_{i,j}$ are distinct	2, 4	
7	–	15	–			0 – 6	**Offline evaluation**