Problem Cheesecake. Cuckoos

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	2 seconds
Memory limit:	512 megabytes

British scientists decided to study ornithology and investigate the life of one extraordinary kind of cuckoos. They have planted a large tree and put n nests on it. Each of the nests is inhabited with one cuckoo. The investigation involves checking if it is possible to put an extra egg to the certain nest or not.

Each egg may be raised only in some two of the nests. Each egg is defined as an unordered pair of distinct integers (x, y). The egg (x, y) may be incubated in the nest x or in the nest y and may not be incubated in any other nest. Notice that egg (x, y) is the same as the egg (y, x).

Now we will describe the process of putting an extra egg into a given nest. Suppose we want to put the egg (x, y) to the nest x. If the nest x is empty then (x, y) remains in that nest and the process is over. Otherwise, if there is already an egg (x, p) in the nest x, then cuckoo puts the egg (x, y) to the nest x and tries to put the egg (x, p) to the nest p in a similar manner, and the process continues.

Your task is to process scientists' queries of three types:

1. (Theoretical) Will the process of putting an egg (x, y) to the nest x eventually stop? Since the question is theoretical, this egg **is not added** actually, and the state of nests is not changed.
2. (Practical) Will the process of putting an egg (x, y) to the nest x eventually stop? If the answer is positive, the egg **is added** according to the description of the process above.
3. (Theoretical) How many **ordered** pairs of distinct integers (x, y) exist such that the egg (x, y) may be put into the nest x and the process will eventually stop? The answer for each pair is determined independently from any other pair.

Input

The first line of the input contains three integers n, m, and q, ($2 \le n \le 200\,000$, $0 \le m \le n$, $1 \le q \le 600\,000$), where n is the number of nests, m is the number of already possessed eggs, q is the number of queries to answer.

Each of the following m lines contains two integers x_i and y_i denoting that the nest x_i contains an egg (x_i, y_i). It is guaranteed that all x_i are distinct and that $x_i \neq y_i$ for all i.

The following q lines contain the queries. Queries are given in the order you have to process them. The first integer t_j of each query line defines the type of this query.

If $t_j = 1$ or $t_j = 2$, then there are two distinct integers x_j and y_j, defining an egg that appears in a corresponding query.

If $t_j = 1$, the egg should not be actually added to the current state.

If $t_j = 2$, the egg should be added if the process of putting an egg requires the finite number of egg transfers.

If $t_j = 3$, you are required to determine the number of ordered pairs (x, y), such that the egg (x, y) may be put to the nest x and the process will be finite. No eggs are actually added at this query, thus the state of each nest remain unchanged.

Output

For each query of the first and the second type output the only word "`Yes`" or "`No`" depending on whether the process is finite or not.

For each query of the third type output the number of ordered pairs that matches the statement above.

Example

input	output
5 3 8 1 2 5 1 2 4 1 1 2 3 2 1 2 3 2 4 2 2 5 3 3 1 4 5	Yes 20 Yes 8 No Yes 0 No

Note

In the sample, the initial state of the nests is the following: the first nest contains the egg $(1, 2)$, the second nest contains the egg $(2, 4)$, the fifth nest contains the egg $(5, 1)$, the third an the fourth nests are empty.

The egg $(1, 2)$ may be added, despite the fact that exactly the same egg is already present, as the existing egg $(1, 2)$ to the other nest.

Also, in the initial state it is possible to add any of the 10 possible eggs that exist for the tree with five nests, and each egg may be put in any of the two nests corresponding to it, such that the process will be finite. So, the answer for the second query is 20.

After the next query the egg $(1, 2)$ will be added, and the state of the eggs will be the following: the first nest contains an egg $(1, 2)$, the second also contains an egg $(1, 2)$, the fourth contains an egg $(2, 4)$ and the fifth contains an egg $(5, 1)$.

Now it is possible only to add the eggs $(1, 3)$, $(2, 3)$, $(4, 3)$ and $(5, 3)$, but still any of the specified eggs may be added to any of the corresponding nests, so the answer is 8.

An egg $(4, 2)$ may not be added to the tree in a finite number of steps, so the state of the nests won't change.

Adding an egg of $(5, 3)$ requires 5 egg transfers, and after that no new egg may be added in a finite number of steps.

Scoring

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

Let t_1 denote the number of queries of the first type, t_2 denote the number of queries of the second type and t_3 denote the number of queries of the third type.

Group	Points	Additional constraints				Required groups	Comment
		n	t_1	t_2	t_3		
0	0	–	–	–	–	–	Sample tests
1	13	$n \le 2000$	$t_1 \le 2000$	$t_2 = 0$	$t_3 = 0$	–	
2	14	$n \le 2000$	$t_1 \le 2000$	$t_2 = 0$	$t_3 \le 1$	1	
3	12	$n \le 2000$	$t_1 \le 2000$	$t_2 \le 2000$	$t_3 \le 2000$	0 – 2	
4	12	–	$t_1 \le 2 \cdot 10^5$	$t_2 = 0$	$t_3 = 0$	1	
5	18	–	$t_1 \le 2 \cdot 10^5$	$t_2 = 0$	$t_3 \le 1$	1 – 2, 4	
6	31	–	$t_1 \le 2 \cdot 10^5$	$t_2 \le 2 \cdot 10^5$	$t_3 \le 2 \cdot 10^5$	0 – 5	**Offline evaluation**

Problem Halva. Gleb and Two Numbers

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	2 seconds
Memory limit:	512 megabytes

When Gleb is not busy with writing long problem statements he enjoys playing with numbers. He picks two integers l and r and tries to find integers a and b such that $l \le a \le b \le r$ and the Hamming distance between a and b is maximum possible.

The *Hamming distance* between two integers x and y is defined as the number of decimal places at which they are different. If these integers are of different length, the shorter one is prepended with leading zeroes.

Input

The first line of the input contains a single integer l and the second line contains a single integer r ($1 \le l \le r \le 10^{1\,000\,000}$).

Output

Print the maximum possible Hamming distance between a pair of integers in range from l to r.

Examples

input	output
11 17	1
1 11	2

Note

In the first sample, one can choose integers 12 and 16. In the second sample, 1 and 10 form an optimal answer.

Scoring

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

Group	Points	Additional constraints	Comment
		l, r	
0	0	–	Sample tests
1	19	$l, r \le 1000$	
2	21	$l, r \le 1\,000\,000$	
3	32	$l, r \le 10^{18}$	
4	28	–	

Problem Strudel. Efficient Evaluation

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	2 seconds
Memory limit:	512 megabytes

Starting from year $20xx$ organizers of all Olympiads in Informatics for high school students decided to conduct their competitions online. An Organization of Online Olympiads (OOO) was created to control and monitor that no one thinks of announcing an onsite competition. Of course, such a powerful and respectful organization needs its own contest managing system. Efficient managers were already hired, sticks and blue tape bought.

In order to increase the efficiency of solutions evaluation a new architecture was introduced. Initially, there are m tests located in the testing queue in order from test 1 to test m. Scheduling module consequently performs n moves. On the i-th move it picks a segment of elements of the queue consisting of elements on positions from l_i to r_i inclusive (positions are numbered starting with 1) and runs the solution on tests on the positions of $l_i, l_i + 2, l_i + 4, \ldots, r_i$ (its guaranteed that l_i and r_i are of the same parity). After this is done, all the tests that solution have already been run on are removed from the testing queue and the remaining tests are shifted such that no empty space is left between them. For example, if the testing queue contains tests with initial indices $2, 3, 4, 5, 10, 12, 13, 20$ and scheduling module applies operation $l_i = 3$, $r_i = 7$, the solution will run on tests at position 3, 5 and 7, that have initial indices 4, 10 and 13. After this the testing queue will look like $2, 3, 5, 12, 20$.

Your goal is to implement the part of this module that for each of the n moves determines the minimum and the maximum initial index of the test that solution was run on at this move.

Input

The first line of the input contains two integers n and m ($1 \le n \le 100\,000$, $1 \le m \le 10^{18}$) — the number of moves that the scheduling module is going to make and the number of tests initially located in the testing queue.

Each of the following n lines contains two integers l_i and r_i ($1 \le l_i \le r_i \le m$) — parameters of the i-th move of testing module. It's guaranteed that before the module applies the i-th move there are at least r_i tests in the queue and that l_i and r_i are of the same parity.

Output

For each of the n moves of the testing module print two integers — the minimum and the maximum initial index of the test that will be used to test solution (and then removed) during the i-th move.

Examples

input	output
2 10 2 8 1 3	2 8 1 5
4 6 1 1 1 1 1 1 2 2	1 1 2 2 3 3 5 5

Note

Consider the first sample.

0. Initially, there are all tests from 1 to 10 in the queue, it looks like $1, 2, 3, 4, 5, 6, 7, 8, 9, 10$.

1. During the first move tests $2, 4, 6, 8$ will be removed. The queue will look like $1, 3, 5, 7, 9, 10$.
2. During the second move tests 1 and 5 will be removed and the queue will transform to $3, 7, 9, 10$.

Scoring

Tests for this problem are divided into six groups. For each of the groups you earn points only if your solution passes all tests in this group and all tests in 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	Points	Additional constraints		Required groups	Comment
		n	m		
0	0	–	–	–	Sample tests
1	10	$n \le 100$	$m \le 100$	0	
2	9	$n \le 10\,000$	$m \le 10\,000$	0, 1	
3	13	–	$m \le 1\,000\,000$	0 – 2	
4	15	$n \le 1000$	–	0, 1	
5	17	$n \le 10\,000$	–	0 – 2, 4	
6	36	–	–	0 – 5	**Offline evaluation**

Problem Tiramisu. Vanya and Jackets

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	2 seconds
Memory limit:	512 megabytes

Vanya attempted to change his life once again and decided to create a schedule of jackets he is going to wear during the next n days.

He read several manuals on jackets operation and found out that different jackets are designed for different temperature ranges. For each of his m jackets he determined values of l_i and r_i, the minimum and maximum temperature value that admits wearing the i-th jacket.

Vanya knows the weather forecast for the next n days, namely there will be the temperature of a_j during the j-th day. Since Vanya is sane enough, he will choose the appropriate jacket for each temperature, that is, on the j-th day he will wear any jacket i such that $l_i \le a_j \le r_i$. Also, Vanya tries to be really fashionable, so he never wears the same jacket for two consecutive days.

Given the fact that Vanya's mother does not allow him leaving home without the jacket or wearing several jackets, create a schedule of which jackets he should wear during the next n days satisfying all the requirements of Vanya.

Input

The first line of the input contains two integers n and m ($1 \le n, m \le 100\,000$), the number of days and the number of jackets in Vanya's wardrobe respectively.

The second line of the input contains n integers a_i ($0 \le a_i \le 10^9$), the temperature at the i-th day.

Then m lines follow, i-th of them contains two integers l_i and r_i ($0 \le l_i \le r_i \le 10^9$), defining the temperature range of the i-th jacket.

Output

If there exists a way of choosing a jacket for each of the n days, output the word "`Yes`" (without the quotes) in the first line, and n numbers b_i in the second line, where b_i is the index of a jacket Vanya should wear in the i-th day. Otherwise output the only line containing the word "`No`" (without the quotes). Jackets are indexed starting from one in the order they appear in the input.

If there are several satisfying schedules, you are allowed to output any of them.

Examples

input	output
4 4 25 25 30 50 10 40 20 30 70 100 50 50	Yes 2 1 2 4
4 2 30 40 50 60 30 40 50 60	No

Note

In the first sample the answer "`2 1 2 4`" is not the only possible, another correct answer is "`1 2 1 4`".

In the second sample there is no schedule satisfying the requirements of Vanya, since he is obligated to wear the first jackets for both the first and the second days.

Scoring

Tests for this problem are divided into four groups. For each of the groups you earn points only if your solution passes all tests in this group and all tests in all **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	Points	Additional constraints	Comment
		n, m	
0	0	–	Sample tests
1	20	$n, m \leq 7$	
2	19	$n, m \leq 100$	
3	21	$n, m \leq 4000$	
4	40	–	**Offline evaluation**