Problem A. Gambling

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3.5 seconds
Memory limit:	512 mebibytes

Bytech and his friends secretly sneaked into the casino. The boys lined up at the slot machine called one-armed bandit with the aim to multiply their pocket money. The boys want everything to be fair and square, so they decided to play one after the other: after completing his gamble each boy moves to the end of the queue. Using the slot machine is dead simple. The gambler always bets one bythaler and pulls the lever to see if he has won. If so, the machine spits out two bythalers, and otherwise nothing happens. In other words, in a single game you can either gain or lose one bythaler.

Our juvenile gamblers are not aware that all their actions are under surveillance of the casino owner with the help of a hidden camera. He knows that the machine operates in a cycle of length m, i.e., the result of every m-th bet is always the same. What's more, the casino owner knows the exact shape of the machine's working cycle.

Presently the owner wonders whether to call the casino's security. He guesses that in case any of the boys loses all his money gambling, he will immediately leave the casino, and his friends would follow him solidarily (he also was once their age!). The owner now wants to see if it ever happens, and if so, how quickly. After all, if the boys leave the casino shortly, maybe it does not pay-off to call security – all the more so, if it turns out that during that time most of their pocket money would increase the casino's budget...

Input

The first line of the input contains a single integer n ($1 \le n \le 1\,000\,000$) denoting the number of youngsters visiting the casino (including Bytech). The second line contains n integers in the range $[1, 10^6]$, representing the sums of boys' pocket money listed in the same order in which they have lined up at the slot machine.

The third line contains one integer m ($1 \le m \le 1\,000\,000$) denoting the length of the working cycle of the machine. The fourth line is a string of m characters denoting the machine's working cycle: if its i-th character is `W`, then the boy wins the i-th gamble of the particular cycle, and if that character is `P` he loses the i-th gamble. The string contains at least one letter `W`.

Output

Your program should output a single integer representing the total number of gambles that will be carried out by the boys until one of them loses all his pocket money. In case this never happens, your program should output the number -1.

Examples

standard input	standard output
4 2 3 2 1 3 WPP	12

Problem B. Colourings

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

Let $G = (V, E)$ be an undirected graph. We call a function $c : V \to \mathbb{N}$ a *colouring* if for each edge $(u, v) \in E$, we have $c(u) \neq c(v)$.

We shall call a colouring c *beautiful* if for every $v \in V$, we have $c(v) \in \{1, 2, \ldots, k\}$. In other words, a colouring c is beautiful, if only colours that are numbers from 1 to k are used.

We shall call a colouring c *smart* if for every $v \in V$, there is a vertex $w \in V$, $w \neq v$, such that $c(v) = c(w)$. In other words, a colouring c is smart if every used colour is used at least twice.

Byteasar is looking for a suitable colouring for his graph. He has already found a beautiful colouring c_b, but it seemed too simple and not ambitious enough. Another time, he managed to find a smart colouring c_s, but after a while he could not stand to look at it any longer.

Byteasar lost hope that, on his way, he will meet a colouring beautiful and smart at the same time. Can you surprise him and find such a colouring?

Input

The first line of input contains three integers n, m, k ($1 \leq k \leq n \leq 200\,000$, $0 \leq m \leq 200\,000$). Number k describes which colourings are considered beautiful, while n and m are the numbers of vertices and edges of Byteasar's graph, respectively. Graph's vertices are numbered 1 through n.

The following m lines describe the edges of Byteasar's graph. The i-th of these lines contain two integers u_i, v_i ($1 \leq u_i < v_i \leq n$) indicating that the vertices numbered u_i and v_i are connected by an edge. The pairs (u_i, v_i) are distinct.

The next two lines contain descriptions of the colourings c_b and c_s, in that order. Colouring descriptions comprise n positive integers not greater than n: i-th of these numbers is the colour of the vertex i. The colouring c_b is beautiful, whereas the colouring c_s is smart.

Output

If there exists a graph colouring which is both smart and beautiful, your program should output the word "`TAK`" (Polish for *yes*) in the first line. The second line should contain n integers describing any such colouring. The description should be in the same format as the description of the colourings in the input.

If no such colouring exists, the only line of the output should contain the word "`NIE`" (Polish for *no*).

Examples

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

Problem C. Counter-manifestation

Input file: *standard-input*
Output file: *standard-output*
Time limit: 3.5 seconds
Memory limit: 256 mebibytes

As every year, *The P = NP Equality Parade* takes place in Bytebury. It is an event during which individuals sharing a belief that *for each language L, for which there exists a non-deterministic Turing machine recognizing L in a polynomial number of steps, there is also a deterministic Turing machine that recognizes the language in a polynomial number of steps*, manifest their views publicly.

Previous editions of the parade proceeded peacefully – participants shouted *"3-SAT is easy!"* at the most or raised banners presenting pseudo-code of the latest polynomial "algorithms" for finding the Hamiltonian cycle, however, without raising much interest among the passers-by. This year, parade organizers have decided to focus the attention of Bytebury residents and plan to chant more shocking slogans (which are somewhat true if $P = NP$) including *"Our money is not safe!"* and *"Our privacy is threatened!"*.

The Bytebury Security Bureau (BSB) fears that the slogans articulated by the parade participants may cause an overwhelming majority of Bytebury residents to withdraw money from the banks and delete their social media accounts that BSB utilises for population surveillance. Keeping it short, there is a suspicion that the situation in Bytebury might become quite unstable.

To prevent this destabilization, BSB officers intend to organize a counter-manifestation promoting the belief in the inequality $P \neq NP$. At the same time they plan to peacefully block the Parade. BSB intends to start the counter-manifestation suddenly, and this should take place at one of the junctions located on the parade's route. Unfortunately, the exact route of the $P = NP$ Equality Parade is kept secret to the very end and the Bureau needs to prepare the place for the counter-manifestation in advance. BSB only got tipped that the parade will start at some junction, will pass a non-zero number of routes to finally get back to the starting point. Your first task is the initial verification of this information, and so to check whether Bytebury road infrastructure allows for the existence of such a route. Moreover, agents are wondering whether there are any junctions, through which the Parade *has* to pass, provided the information is confirmed. They asked you to find all such junctions – one of them will be chosen as the most convenient location for the counter-manifestation (in case such a junction does not exist, BSB will proceed to Plan B).

There are n junctions in Bytebury. The junctions are connected with one-way roads. As motor vehicles are a part of the parade, we assume that the parade can move only along the one-way road's direction.

Input

The first line of input contains two integers n and m ($2 \leq n \leq 500\,000$, $1 \leq m \leq 1\,000\,000$) denoting the number of junctions and the number of roads in Bytebury, respectively. The junctions are numbered 1 through n. The following m lines comprise a description of Bytebury roads. The i-th of these lines contains two integers a_i and b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$), denoting that the i-th road leads from junction a_i to junction b_i. None of the ordered pairs (a_i, b_i) is repeated in the input.

Output

In case it is not possible to route the Parade in accordance with the information known to BSB, the output should contain a single line containing the word "`NIE`" (Polish for "no"). Otherwise, the output should contain two lines. The first of them should contain the number k denoting the number of junctions through which the Parade's route will certainly lead. The second line should contain a list of those intersections in ascending order (if $k = 0$, the second line should be left blank).

Examples

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

Problem D. Championships

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1.5 seconds
Memory limit:	256 mebibytes

The Computer Sports World Championships is the most important event in the calendar of every electronic entertainment fan. This year, the championships will be held in the kingdom of Byteotia. The organizing committee, appointed by the king Byteasar, is facing a difficult task – it has to decide in which Byteotian cities competitions will take place. There are n cities in Byteotia (numbered 1 through n) connected by m two-way roads.

The committee hopes that the championship will attract crowds of fans from all over the world. Obviously, the fans will travel frequently between the cities to watch the competitions of various e-sport types. The priority is therefore that the set of cities, hosting the championships events, is *well connected*.

We call a set of cities S well connected, if:

(1) From every city of the set S there are at least d direct connections to other cities of S.

(2) Between any two cities of S there exists a route running only through the cities belonging to the set S.

Additionally, to minimize the average number of visitors in each city, the committee would prefer the chosen set to be possibly large.

Input

The first line of the input contains three integers n, m and d ($2 \le n \le 200\,000$, $1 \le m \le 200\,000$, $1 \le d < n$) denoting the the number of cities, the number of roads in Byteotia and the parameter d, respectively. Next m lines describe the Byteotian roads. The i-th of these lines contains two integers a_i and b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$) indicating that the i-th road connects the cities numbered a_i and b_i. Each pair of cities is connected by at most one direct road.

Output

If it is not possible to choose a set of cities of Byteotia that is well connected, the only line of the output should contain the word "NIE" (Polish for *no*).

Otherwise, the output should contain the most numerous set of cities that is well connected, in the following format. The first line should contain the number k denoting the size of the found set. The second line should contain k numbers representing the cities belonging to the set, in ascending order.

In case there are multiple solutions, your program can output any of them.

Examples

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

Problem E. Neon

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Byteasar is a well known Byteotian prankster – he earns a living arranging various funny situations, which he then documents in the form of videos posted on the internet. This time he has targeted a huge neon sign on the roof of a prestigious hotel with a very long name.

The neon sign w displaying the name of the hotel, contains n letters. Byteasar intends to break in onto the hotel roof during the night to switch off some of the neon letters so that remaining letters, read from left to right, comprise some very funny, m-letter word s. In order to obtain an attractive effect, the position of the last lit letter and the first lit letter must not differ by less than k. The prankster wonders what is the number of ways he can switch off some of the letters in order to achieve his goal.

Formally, he is interested in the number of ways in which it is possible to choose indices $j_1, j_2, \ldots, j_m$ from the range $[1, n]$, so that $j_1 < j_2 < \ldots < j_m$, $j_m - j_1 \geq k$ and $w_{j_1} w_{j_2} \ldots w_{j_m} = s$, where w_i denotes i-th letter of the string w. The indices $j_1, j_2, \ldots, j_m$ correspond to the positions of letters which will remain lit.

Input

The first line of the input contains three integers n, m, k ($1 \leq k \leq n \leq 100\,000$, $1 \leq m \leq 10$). The second line contains an n-letter word w displayed on the hotel's roof. The third line contains an m-letter word s to be displayed after switching off some of letters. Words w and s consist solely of lower-case letters of the English alphabet (`a-z`).

Output

The only line of the output should contain the sought number of ways Byteasar can achieve his goal, modulo $10^9 + 7$.

Examples

standard input	standard output
13 3 5 longlonghotel lol	5

Note

In the example, Byteasar can leave the lit letters on one of the following position sets: $\{1, 2, 13\}$, $\{1, 6, 13\}$, $\{1, 10, 13\}$, $\{5, 6, 13\}$, $\{5, 10, 13\}$.

Problem F. Robots

Input file:	`standard input`
Output file:	`standard output`
Time limit:	10 seconds
Memory limit:	256 mebibytes

Captain Byteasar supervises the colonization of a natural asteroid BA-1T which is rich in raw materials. His job is controlling the robot-miners extracting ardanium. The space weather forecast predicts a meteor shower, and it would be better if all the robots were hidden in armoured strongholds once the shower happens.

Unfortunately, the miners control system leaves much to be desired. The only thing one can do is to input a single non-negative integer k into it, which will result in sending k "Move up!" commands to every robot.

There are n sectors marked out on the surface of the planet. Some of the sectors are the strongholds, and the other ones are the open pit ardanium mines. Robot-miners are equipped with quantum brains and therefore they work nondeterministically. For each sector s, Byteasar knows a non-empty set A_s, such that any robot located in the sector s will move to one of the sectors of A_s after receiving a command. It is generally not known which sector of A_s will be picked; one can neither count on any repeatability – even if a certain robot has been in the sector s a few times already, next time it can move to a sector completely different than before.

Now Byteasar wonders whether there exists such k, that after issuing k "Move up!" commands, each robot will be in one of the strongholds for sure.

Input

The first line of the input contains three integers n, b and r ($2 \le n \le 200$, $1 \le b, r \le n$), denoting the number of sectors, the number of strongholds and the number of robot-miners, respectively. The sectors are numbered 1 through n and the sectors numbered 1 through b are the strongholds.

This is then followed by n lines containing descriptions of possible transitions after receiving the "Move up!" command. The i-th of these lines contains a string of n digits from the set $\{0, 1\}$; the j-th of these numbers is equal to 1 if and only if a miner can move from sector i to sector j after receiving a command. At least one digit in a line is equal to 1.

The last line of the input contains an increasing sequence of r numbers ranging from 1 to n, indicating the sectors where the robot-miners are initially present.

Output

In case a number k, sought by Byteasar, does not exist, you should output the number -1. Otherwise we guarantee that there exists a non-negative integer satisfying Byteasar's requirements and having at most 200 digits (in decimal notation). Then, you should output any such number.

Examples

standard input	standard output
4 2 2 0100 0010 1001 1000 3 4	2
4 2 2 0100 0010 1001 1000 2 3	-1

Problem G. Equation

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

For a positive integer n, let $f(n)$ be the sum of squares of its digits in the decimal notation. Given are three integers k, a, b. Your task is to determine the number of such natural numbers n, that $a \le n \le b$ and n is the solution of the equation

$$k \cdot f(n) = n.$$

Input

The first and only line of the input contains three integers of the task statement: k, a, b, $(1 \le k, a, b \le 10^{18}, a \le b)$.

Output

Your program should output a single integer: the number of integral solutions of the given equation contained in the range $[a, b]$.

Examples

standard input	standard output
51 5000 10000	3

Note

In the example, the only positive integers n from the range $[5000, 10000]$ satisfying the equation for $k = 51$ are 7293, 7854 and 7905.

Problem H. Hay

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	128 mebibytes

Byteasar, a farmer, bought a field of area equal to n ares, where he intends to sow grass seeds. Mown and dried hay will be used as a fodder for livestock kept on Byteasar's farm.

A mix of n grass species will be sown in the field – each species will take up a total of 1 ares. For the i-th species it is known that its grass blades grow by a_i centimetres each day, regardless of weather conditions or soil fertility. It is also known that as a result of mowing one are of any grass type by one centimetre, exactly 1 kilogram of hay is obtained.

Byteasar has a mower that can be set in such a way that it cuts grass to any height b. With this setting each blade of grass higher than b centimetres will be trimmed down to the height of exactly b centimetres.

Byteotian law requires that after each grass mowing the amount of hay obtained is to be documented. Byteasar faced a choice: he must either buy a weighing machine, or write a program that based on the dates of mowing and mowers settings, could estimate the weight of the hay obtained after each mowing. The second of these options seemed more convenient and cheaper to him.

We assume that the grass will be sown on day 0 at midnight and always be mown at midnight. We also assume that the time it takes to mow the grass in the field to any height b is negligible.

Input

The first line of input contains two integers n and m ($1 \leq n, m \leq 500\,000$), denoting Byteasar's field area in ares (and at the same time the number of grass species sown) and the number of times the grass has been mown. The second line contains a sequence of n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^6$) indicating the rate of growth of successive species of grass.

The next m lines contain the details of how Byteasar mowed the grass: the i-th mowing is described by two integers d_i and b_i ($1 \leq d_i \leq 10^{12}$, $0 \leq b_i \leq 10^{12}$) contained in the i-th of these lines and indicating that on day d_i Byteasar trimmed the grass to the length of b_i centimetres. You can assume that this data is listed in chronological order, i.e., $d_1 < d_2 < \ldots < d_m$.

Furthermore, you can also assume that Byteasar would never allow a situation when the grass has height exceeding 10^{12} centimetres at any place in the field.

Output

The output should contain exactly m lines. The i-th of these lines should include a total weight of hay (in kilograms) obtained after the i-th grass mowing.

Examples

standard input	standard output
4 4 1 2 4 3 1 1 2 2 3 0 4 4	6 6 18 0

Note

Consider the example. The heights of grass species $1, 2, 3, 4$ before and after the subsequent mowings are shown in the following table.

Day	Before mowing	After mowing
1	$1, 2, 4, 3$	$1, 1, 1, 1$
2	$2, 3, 5, 4$	$2, 2, 2, 2$
3	$3, 4, 6, 5$	$0, 0, 0, 0$
4	$1, 2, 4, 3$	$1, 2, 4, 3$

Problem I. Gym

Input file:	*standard input*
Output file:	*standard output*
Time limit:	10 seconds
Memory limit:	256 mebibytes

Byteasar is an owner of a newly opened gym. Because market competition is high, he decided to approach every aspect of his business professionally and offer an advanced booking system for his customers.

The gym is equipped with k various fitness machines. When placing a reservation the customer declares his desire to use a certain gym device for one hour at his chosen time interval. The system then determines the exact times when customers will be using gym equipment, so that no device is assigned to two different reservations at the same time.

Byteasar already received all n reservations for the near future. He noted rightly, that if at some point no person uses the gym, the lights and the air conditioning could be turned off, and the supplements outlet could be closed. As a part of the savings plan, he would like to arrange the exercise sessions in such a way that the total number of hours when the gym is used by at least one person is minimized. Help him with this task.

Input

The first line of the input contains two integers n and k ($1 \le n \le 1\,000\,000$, $1 \le k \le 10^9$) denoting the number of reservation requests and the number of devices in the gym, respectively. The gym devices are numbered using integers 1 through k; for simplicity also the hours are numbered using consecutive integers starting from 1.

The next n lines describe the placed reservations: the i-th of these lines contains three integers a_i, b_i and p_i ($1 \le a_i \le b_i \le 10^9$, $1 \le p_i \le k$) denoting that the i-th customer expressed his desire to use the device p_i for one hour in a time interval ranging from hour a_i to hour b_i, inclusive.

Output

If it is possible to plan the exercise sessions, so that all reservations are met and no piece of equipment is used at the same time by two people, the output should contain $n+1$ lines. The first line should contain the minimum number of hours when the gym is used by at least one person. The i-th of the next n lines should contain a single integer $t_i \in [a_i, b_i]$ indicating that within the i-th reservation the device p_i is used during the hour t_i.

If it is not possible to schedule the exercise in accordance with these requirements, you should output the word `NIE` (Polish for *no*).

Examples

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

Problem J. Triangles

Input file:	*standard input*
Output file:	*standard output*
Time limit:	10 seconds
Memory limit:	256 mebibytes

n points are given on a plane. We are interested in the number of right-angled triangles with vertices at these points and area contained in the range $[A, B]$.

Input

The first line contains three integers n, A, B ($1 \le n \le 2000$, $1 \le A \le B \le 10^{18}$). The following n lines describe the individual points. The i-th of these lines contains two integers x_i, y_i ($-10^9 \le x_i, y_i \le 10^9$) which are the coordinates of the i-th point. All the given points are distinct.

Output

The only line of the output should contain the number of triangles with vertices at the given points and area in the range $[A, B]$.

Examples

standard input	standard output
7 5 25 0 0 2 0 0 2 10 0 0 10 3 3 3 -3	3