

Problem A. Ascending mountains

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

Isa, wanting to join the climbing group Climbing Camels, decided to start climbing mountains. In order to be accepted as a climbing camel, Isa must climb all the available N mountains. However, for Isa to be allowed to climb the mountain of a certain level, she must have climbed other easier mountains first.

Your mission is: given the number of mountains and pairs of mountains (A, B) where mountain of level A must be climbed before mountain of level B determine the lexicographically smallest order in which Isa can climb all the mountains and become a climbing camel.

An order A is lexicographically smaller than another order B if there exists an i such that $A_i < B_i$ and for all j with $1 \leq j < i$, $A_j = B_j$.

Input

The first line contains two integers N and M ($1 \leq N, M \leq 2 \cdot 10^5$).

In each of the next M lines, two integers A and B ($1 \leq A < B \leq N$) are given, where in order to climb the mountain of level B , Isa must first have climbed the mountain of level A . Certainly $A < B$, because a lower difficulty level is easier to climb .

Output

Print N numbers representing the lexicographically smallest order that Isa must follow to become a climbing camel, or -1 if there is no existing order that allows Isa to climb all the mountains.

Examples

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

Problem B. Boundless Deck

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

Henrique Ramos is very happy with his new purchase: a special deck with infinitely many cards!

Each card in this deck has a number written on it. When buying the deck, Henrique had to choose a positive integer n , which determines the numbers on the cards.

The number on the first card is n^0 , the second card has n^1 , the third has n^2 , and in general, the i -th card has the number n^i . This way, each card is unique, and no two cards have the same number.

Henrique is going to play a game with his friends using his new deck. In this game, a move is defined as a finite subset of cards from the deck, and the value of each move is the sum of the numbers on the selected cards.

For example, Henrique's deck for $n = 3$ contains the cards 1, 3, 9, 27, 81, and so on. In this same example, a few valid moves are:

- 1, with value 1;
- 1, 3, with value 4;
- 3, 243, 729, with value 975.

Henrique is interested in the moves with the smallest values, so he created a sequence of all possible move values, ordered by their value. For example, the first five terms of this sequence for $n = 3$ are: 1, 3, 4, 9, 10.

Henrique wants to know the k -th term of this sequence. However, he's currently too busy reading the problem *Collatz-Star of Pain and Suffering*. Help Henrique by calculating the value of the k -th term in this sequence.

Since the answer can be a very large number, print only the remainder when it is divided by $10^9 + 7$.

Input

The input is a single line containing two integers n and k ($2 \leq n \leq 10^9, 1 \leq k \leq 10^9$). n is the number that defines Henrique's deck, and k is the index Henrique is interested in.

Output

Output a single integer: the remainder when the k -th term of this sequence is divided by $10^9 + 7$.

Examples

standard input	standard output
3 4	9
4 5	17
3 98	975

Note

In the first test case, the first five terms of this sequence are: 1, 3, 4, 9, 10. The fourth term is 9.

In the second test case, the first five terms of this sequence are: 1, 4, 5, 16, 17. The fifth term is 17.

Problem C. Collatz-Star of Pain and Suffering

Input file: **standard input**
Output file: **standard output**
Time limit: **6.5 seconds**
Memory limit: **512 megabytes**

Since the latest Great War, the Galaxy has been ruled by the Galatic Empire. Lord Edsidious was finally able to conquer the status of Emperor, after centuries of planning. In his Empire, the following definitions and conventions apply:

- $X \% Y$ (both integers) represents the remainder when dividing X by Y if $X \geq 0$ and $Y > 0$. Additionally, $X \% Y = X$ if $X < 0$ and $Y > 0$, and $X \% Y = 0$ if $Y = 0$.
- $NCP(X, Y)$ is the smallest integer $K \geq X$ such that $\gcd(K, Y) = 1$.
- The Super Shift Function, SSF , is defined as: $SSF(X, Y, Z) = ((X - Z) \% (Y - Z)) + Z$.
- The Collatz Function, CF , is defined as:
$$CF(X) = \begin{cases} \frac{X}{2} & \text{if } X \text{ is even} \\ \frac{3 \cdot X + 1}{4} & \text{if } X \% 4 = 1. \\ 3 \cdot X + 1 & \text{if } X \% 4 = 3 \end{cases}$$
- About the value of a fraction mod X : consider the fraction $\frac{P}{Q}$ with Q coprime to X (if it's not, the value is not defined). The value of this fraction mod X is $(P \cdot Q^{-1}) \% X$, where Q^{-1} is the value such that $(Q \cdot Q^{-1}) \% X = 1$ and $0 \leq Q^{-1} < X$ (it can be proven a unique Q^{-1} exists if Q is coprime to X).
- About grid notation: a $X \times Y$ grid has X rows and Y columns. A cell (i, j) of the grid is at the i -th row and j column. A subgrid $Z = (Z_{x1}, Z_{y1}, Z_{x2}, Z_{y2})$ is the set of cells (i, j) with $Z_{x1} \leq i \leq Z_{x2}$ and $Z_{y1} \leq j \leq Z_{y2}$.

To maintain his dominance, Lord Edsidious hired the famous botanic Collatz (the creator of the Collatz Function) to build him a Collatz-Star of Pain and Suffering, which is a massive device able to exterminate whole planets with a single blast. And to protect this strategic weapon, he assigned his most prestigious general: Darth Caetanus. Caetanus recently became aware of a weakness in the Star: it's energy board.

The energy board is composed of independent cells disposed in a $H \times W$ grid, and it is associated with a fixed integer $M > 2$. A cell (i, j) , at any time t (measured in some integer unit) has two positive integers associated with it: $A_{t,i,j}$ and $B_{t,i,j}$. These numbers describe the probability of a failure in the cell at that time, which is equal to $\frac{A'_{t,i,j}}{B'_{t,i,j}}$, where $B'_{t,i,j} = NCP(SSF(B_{t,i,j}, M, 2), M)$ and $A'_{t,i,j} = SSF(A_{t,i,j}, B'_{t,i,j}, 1)$. At any time, if a failure occurs in any cell, it shuts down, leaving the Collatz-Star with no power temporarily. These values A and B vary in time as follows: $A_{t+1,i,j} = CF(A_{t,i,j})$ and $B_{t+1,i,j} = CF(B_{t,i,j})$.

Vinicius Solo, an influent person in the fallen Galatic Republic, used his connections to find (at time 0) all the numbers $A_{0,i,j}$ and $B_{0,i,j}$ and the fixed number M . He also became aware that, at a certain unknown time T , Lord Edsidius will order the Collatz-Star to fire against the poor U.n.Balloon planet. To prevent its destruction, Vinicius will need the help of his ally Daniel Skywalker, who can Mystic Focus on subgrids of the energy board grid.

Let's consider that Daniel tries to Mystic Focus on a subgrid S in a period of time from t_l to t_r (inclusive), and let $Prob(S, t_l, t_r)$ be the probability that any cell in S shuts down at any time between t_l and t_r (which means the Collatz-Star would lose power in this time interval). Daniel can explode the cells in S using his ability, but only if the value of $Prob(S, t_l, t_r) \bmod M$ is greater than $\lfloor \frac{M}{2} \rfloor$ (it can be proven $Prob(S, t_l, t_r)$ is always representable by a fraction $\frac{P}{Q}$ with Q coprime to M). This explosion would provoke a chain reaction that would destroy the whole Collatz-Star!

However, saving U.n.Balloon will not be that easy. To protect the Collatz-Star, Darth Caetanús casted a spell which prevents Daniel from trying to Mystic Focus on subgrids S not contained in a special unknown subgrid K . Also, if Daniel tries to Mystic Focus on some subgrid S , he will be forced to do so from $t_l = S_{x1} + S_{y1} + K_{x1} + K_{y1}$ to $t_r = T - (S_{x2} + S_{y2} + K_{x2} + K_{y2})$.

Vinicius and Daniel don't know what is the actual time T or the special subgrid K , so they decided to prepare themselves. They will conduct Q studies: in the study i , they will count, for chosen values of T_i and K_i , the number of subgrids S_i that Daniel would be able to Mystic Focus on and explode, causing the destruction of the Collatz-Star. Help them write a program that calculates this!

Input

The first line of input contains the integers H, W ($1 \leq H, W \leq 30$) and M ($2 < M \leq 10^5$).

The next H lines contain W integers each: the values $A_{0,i,j}$ ($0 < A_{0,i,j} \leq 10^5$).

The other next H lines also contain W integers each: the values $B_{0,i,j}$ ($0 < B_{0,i,j} \leq 10^5$).

The next line of input contains the integer Q ($1 \leq Q \leq 100$).

Finally, the next Q lines contain five integers each: T_i ($0 \leq T_i \leq 10^5$) and the $x1, y1, x2$ and $y2$ (in this order) related to K_i ($1 \leq x1, x2 \leq H, 1 \leq y1, y2 \leq W$).

Output

Print Q lines: in the i -th line, the number of subgrids S_i that Daniel would be able to Mystic Focus on and explode, considering the given T_i, K_i and associated t_l and t_r for each subgrid (as described by the statement).

Examples

standard input	standard output
3 3 9 12 31 46 20 35 28 91 54 18 11 14 16 10 15 77 27 15 13 4 11 1 1 2 2 30 1 3 2 3 0 1 1 3 3 8 1 1 1 1	1 0 0 1
3 3 3 55 83 63 73 62 95 95 97 27 63 55 41 95 47 63 95 54 47 4 30 2 3 3 3 20 3 3 3 3 24 1 2 3 2 36 2 1 3 2	0 0 4 4

Note

Let's understand the first study of the first test case, with $T = 11$ and $K = (1, 1, 2, 2)$. The value of M is 9.

With these values, most subgrids have $t_l > t_r$, and thus cannot be targeted by Daniel's Mystic Focus. The only possible ones are $S_1 = (1, 1, 1, 1)$ ($t_l = 4, t_r = 5$), $S_2 = (1, 1, 2, 1)$, and $S_3 = (1, 1, 1, 2)$ (both with $t_l = t_r = 4$).

Starting with S_1 :

At time 4, we can verify that $A_{4,1,1} = 5$ and $B_{4,1,1} = 10$. Calculating $B'_{4,1,1} = NCP(SSF(10, 9, 2), 9)$, we have $NCP((8 \% 7) + 2, 9) = NCP(3, 9) = 4$. It follows that $A'_{t,i,j} = SSF(5, 4, 1) = (4 \% 3) + 1 = 2$ and that $Prob(S_1, 4, 4)$ is $\frac{2}{4}$.

At time 5, $A_{5,1,1} = 4$ and $B_{5,1,1} = 5$. Thus, we calculate $B'_{5,1,1} = NCP(SSF(5, 9, 2), 9)$ as $NCP((3 \% 7) + 2, 9) = NCP(5, 9) = 5$ and $A'_{t,i,j} = SSF(4, 5, 1)$ as $(3 \% 4) + 1 = 4$. This results in $Prob(S_1, 5, 5) = \frac{4}{5}$.

We can use $Prob(S_1, t_l = 4, t_r = 5) = Prob(S_1, 4, 4) \cup Prob(S_1, 5, 5)$ to calculate the probability as $Prob(S_1, 4, 4) + Prob(S_1, 5, 5) - Prob(S_1, 4, 4) \cdot Prob(S_1, 5, 5)$. Applying the values, we get $Prob(S_1, 4, 5) = \frac{2}{4} + \frac{4}{5} - \frac{2}{4} \cdot \frac{4}{5} = \frac{18}{20}$. The value of this fraction mod 9 is $(18 \cdot 20^{-1}) \% 9 = (18 \cdot 5) \% 9 = 0$. This is not greater than $\lfloor \frac{9}{2} \rfloor = 4$, so Daniel cannot use the Mystic Focus here.

Now with S_2 :

Let's take $S_2^1 = (1, 1, 1, 1)$ and $S_2^2 = (2, 1, 2, 1)$. Then, $Prob(S_2, t_l = 4, t_r = 4) = Prob(S_2^1, 4, 4) \cup Prob(S_2^2, 4, 4)$. We have already calculated $Prob(S_2^1, 4, 4) = Prob(S_1, 4, 4) = \frac{2}{4}$.

We have $A_{4,2,1} = 1$ and $B_{4,2,1} = 1$. Thus, $B'_{4,2,1} = NCP(SSF(1, 9, 2), 9)$, which results in $NCP((-1) \% 7 + 2, 9) = NCP(-1 + 2, 9) = 1$, and $A'_{4,2,1} = SSF(1, 1, 1)$, which gives $(0 \% 0) + 1 = 0 + 1 = 1$. With this, we obtain $Prob(S_2^2, 4, 4) = \frac{1}{1}$. This results in $Prob(S_2, t_l = 4, t_r = 4) = \frac{2}{4} + \frac{1}{1} - \frac{2}{4} \cdot \frac{1}{1} = \frac{4}{4}$, which has a value of 1 in mod 9 (still not greater than $\lfloor \frac{9}{2} \rfloor = 4$).

Lastly, with S_3 :

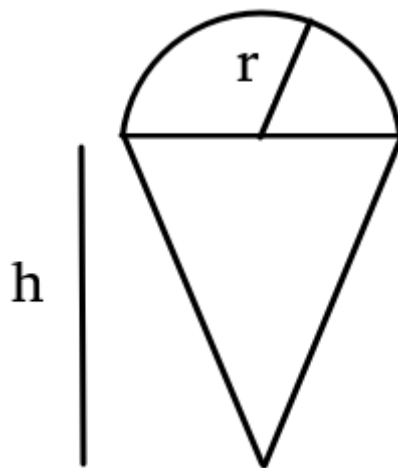
Similarly to S_2 , let's take $S_3^1 = (1, 1, 1, 1)$ and $S_3^2 = (1, 2, 1, 2)$. $A_{4,1,2} = 71$, $B_{4,1,2} = 34$, $B'_{4,1,2} = NCP(SSF(34, 9, 2), 9) = NCP(6) = 7$ and $A'_{4,1,2} = SSF(71, 7, 1) = 5$. Then, $Prob(S_3^2, 4, 4) = \frac{5}{7}$ and $Prob(S_3, 4, 4) = \frac{2}{4} + \frac{5}{7} - \frac{2}{4} \cdot \frac{5}{7} = \frac{24}{28}$. This has a value of 6 in mod 9, so Daniel would be able to use the Mystic Focus here!

In total, we find that there is only one subgrid of K (our S_3) in which Daniel can use his ability. Thus, the result for this study is 1.

Problem D. Delirium at Unballoon

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

The members of Unballoon gathered at Isa's house after a marathon to celebrate and eat ice cream. Most people wanted to serve their ice cream in cones and then eat them. However, Lucas Sala does not like cones and therefore preferred to serve his ice cream in his water bottle. This caused a great controversy, as some of the marathon runners believed that Sala's bottle could hold more ice cream than a cone, but they were not sure.



Cone with height h and base radius r , which is also the radius of the semi-sphere.

After several calculations with integrals and some crimes against basic physics, the programmers could not reach a conclusion; therefore, you must help them solve this mystery. For this, consider that the ice cream cones are circular base cones. Additionally, the cones will be filled with ice cream uniformly up to their top, where a solid semi-sphere of ice cream with a radius equal to the radius of the cone's base will be placed. Finally, assume that Lucas Sala's bottle is a cylinder, which will also be filled with ice cream uniformly to its top. **A solid semi-sphere of ice cream will not be placed in Lucas Sala's bottle.**

Input

The first and only line of input contains four integers $1 \leq r_L, h_L, r_C, h_C \leq 10^2$: the radius of the base of Lucas Sala's bottle, the height of the bottle, the radius of the base of the cone representing his ice cream cone, and its height, respectively. All values are given in centimeters.

Output

If Lucas Sala's bottle can hold a strictly greater volume of ice cream than the cone, print "Injusto". Otherwise, print "Justo".

Examples

standard input	standard output
1 4 1 10	Justo
27 100 45 18	Justo
78 57 60 40	Injusto

Problem E. Elementary Magical School of Words

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

In a distant kingdom called Alfabétia, wizards did not use common spells — they used magical words. Each word was formed by lowercase letters of the alphabet, and its power did not lie in the letters themselves, but in the way they repeated.

The wisest of the wizards, Archmagus MagePetrus, noticed something curious: two words could be equally powerful, even if they had different letters, as long as they had the same pattern of repetition.

For example:

- The word “aba” had a pattern where the first and third letters were the same, and the second was different.
- The word “cbc” had the same pattern!
- Even the strange “z fz” was like that too!

MagePetrus called this letter symmetry. He decided that, for magical purposes, words with the same pattern of letter repetition were equivalent — it did not matter which letters were used, only how they repeated.

Intrigued by the possibilities, MagePetrus proposed a challenge to the young apprentices of Alfabétia:

Given a number n , how many different magical patterns (i.e., how many distinct classes of words by letter symmetry) can be formed with words of size n , using letters from “a-z”?

We consider two strings equal, by letter symmetry, if one can be obtained from the other through a permutation of its letters. Formally, let s be the string formed by the concatenation of the characters $s_1, s_2, \dots, s_n$ in that order and $A = \{a, b, \dots, z\}$ the set of lowercase letters of the Latin alphabet. We say that s is equal to a string t by letter symmetry if and only if there exists an injective function $f : A \rightarrow A$ such that t is the string formed by the concatenation of the characters $f(s_1), f(s_2), \dots, f(s_n)$. The function f being injective means that, for all characters $x, y \in A$, if $x \neq y$, then $f(x) \neq f(y)$.

For example, the strings “aba”, “cbc”, and “z fz” are considered equivalent, as they all have the same structure of letter repetition, while the strings “aa” and “ab” are different.

Input

The input consists of a single integer n ($1 \leq n \leq 10^6$), representing the size of the strings to be considered.

Output

The output should be a single integer, representing the number of different strings of size n that can be formed, taking into account the symmetry of letter permutations. Since the answer can be enormous, find it modulo $(10^9 + 7)$.

Examples

standard input	standard output
3	5
1	1
5	52
1000000	281893449

Note

For $n = 3$:

The strings “aaa”, “aab”, “aba”, “abb”, and “abc” are examples of the distinct strings that can be formed.

Problem F. Fibonacci

Input file: **standard input**
Output file: **standard output**
Time limit: 0.25 seconds
Memory limit: 256 megabytes

The Fibonacci sequence (f_n) is defined as follows:

$$f_n = \begin{cases} 0 & n = 0, \\ 1 & n = 1, \\ f_{n-1} + f_{n-2} & n \geq 2. \end{cases}$$

For example, the first 6 terms of this sequence are 0, 1, 1, 2, 3, and 5, and $f_9 = 34$.

Given two integers a and b ($1 \leq a, b \leq 10^9 + 9$) and an integer M ($1 \leq M \leq 10^9 + 9$), compute the remainder of the division of f_{a^b} by M . In other words, find the unique integer A such that $f_{a^b} - A$ is a multiple of M and $0 \leq A < M$.

Input

The first and only line of input consists of three integers a, b , and M , satisfying $1 \leq a, b, M \leq 10^9 + 9$.

Output

Print a single integer: the remainder of the division of f_{a^b} by M .

Examples

standard input	standard output
3 2 100	34
2 3 6	3
314159265 271828182 987654321	28908313

Note

In the second test case, note that $f_{2^3} = f_8 = 21$. The answer is the remainder of the division of 21 by 6, which is 3.

Problem G. Graph of Love

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

Luiz Eduardo is one of the last remnants of an almost extinct race on Earth: the romantics. He believes that everyone in the world will eventually find their one and only true love, whoever they may be (in fact, their true love could even be themselves!).

He took a sabbatical year and traveled the world, talking to people and hearing about their true loves. Carrying his notebook, he wrote down the names of the people he spoke to or heard about and the corresponding name of their true love. After his long journey, he showed his notes to his colleague Iasmim (to protect people's privacy, he replaced names with numbers).

Iasmim did not find Luiz's notes very interesting, so she decided to turn them into something nicer: a graph! More specifically, she constructed a graph where there is a directed edge from A to B if and only if person A truly loves person B .

She then challenged Luiz with a problem: what is the maximum number of partially happy monogamous marriages (P.H.M.M.s) that could happen simultaneously in this graph? A P.H.M.M. consists of two people, A and B , such that:

- $A \neq B$
- There is an edge in Iasmim's graph from A to B or from B to A .
- Neither A nor B are in another P.H.M.M.

Both Luiz and Iasmim have terrible memories of the last time they heard the words marriage and graph in the same sentence. Therefore, help them solve this problem.

Input

The first line of the input contains the number N , the number of people in Luiz's notes ($1 \leq N \leq 10^5$).

The second line of the input contains N integers A_i , each representing the true love of person i ($1 \leq i, A_i \leq N$, it is possible that $A_i = i$).

Output

Print a single integer: the maximum number of partially happy monogamous marriages (P.H.M.M.s) that could happen simultaneously in Iasmim's graph.

Example

standard input	standard output
12 4 1 5 1 9 7 9 12 9 11 12 10	5

Note

In the sample test case, to generate the maximum number of P.H.M.M.s, one of multiple possibilities is to pair the people in the following marriages: $(1, 4)$, $(3, 5)$, $(6, 7)$, $(8, 12)$, $(10, 11)$.

Although person 2 and person 9 remain outside of P.H.M.M.s, it is not possible to pair them because there are no edges between them. It can be verified that it is not possible to generate more than 5 P.H.M.M.s simultaneously.

Problem H. High-Speed Collision

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

Pedro Pacoca enjoys playing an online car racing game and recently discovered that his colleague Pedro Gallo also plays the game. Excited by this, the two Pedros decided to help each other.

In the game, each player drives a car around a circuit, and the player who completes the course in the shortest time wins. To achieve this, it is crucial for players to avoid crashing their cars into others.

Knowing this, the two Pedros decided to share the initial positions of their cars, their speeds, and the directions they were moving. They also agreed to maintain both their speeds and directions constant.

Now, they need your help to determine if there will be a collision between the two cars. Given the initial positions and the speeds in the x and y directions for each car, determine the moment in time when a collision will happen. If no collision occurs, output -1 .

For the purposes of collision detection in this problem, each car is modeled as a triangle on the plane.

Input

The input consists of four lines. The first two lines describe the first car, and the last two lines describe the second car.

Each car is described by two lines:

- The first line contains six integers x_1, y_1, x_2, y_2, x_3 , and y_3 ($-10^5 \leq x_i, y_i \leq 10^5$). (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) are the coordinates of the vertices of the collision triangle of the car.
- The second line contains two integers v_x and v_y ($-10^5 \leq v_x, v_y \leq 10^5$). (v_x, v_y) is the speed of the car in the x and y directions.

It is guaranteed that the provided vertices form a non-degenerate triangle. Furthermore, it is also guaranteed that collision triangles do not intersect. That is, there is no point p that belongs to both triangles simultaneously.

Output

Print a single line containing a real number: the moment in time when a collision will occur between the cars, or -1 , if there is no collision.

Your answer will be considered correct if the absolute or relative error does not exceed 10^{-6} .

Examples

standard input	standard output
0 0 1 0 0 1 1 0 1 1 2 1 1 2 0 -1	0.5
0 0 1 0 0 1 1 0 5 0 5 -1 4 0 -3 0	0.75
2 2 2 0 7 3 4 2 3 3 8 7 10 8 1 0	-1

Problem I. Inventing Names

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

Quirino is an aspiring Pokémon professor. He has a specialization in Taxonomy, and one of his responsibilities as an assistant to the renowned Professor Rowan is to decide the names of new Pokémon that are discovered in the Sinnoh region.

Different Pokémon cannot have the same name, so Quirino received a list of the last N names given to Pokémon to avoid repetitions. Additionally, due to restrictions in the naming system, names must have a maximum of K characters.

Quirino is methodical and wishes to choose the lexicographically smallest possible name for every Pokémon he will name. However, since he climbed Mount Coronet to uncover the secrets of the Creation Trio, he has not been seen again. In the name of Arceus, help him write a program that names new Pokémon according to his restrictions and preferences while he is away!

Note: a word A is lexicographically smaller than another B if it would come before it in the dictionary. That is, if $A \neq B$ and one of the two holds:

1. A is a prefix of B .
2. At the first position i (from left to right) where they differ, $A_i < B_i$ (with P_i being the i -th character of P from left to right).

Input

The first line of the input contains the numbers N and K : the number of names in the list and the maximum size of a name ($1 \leq N, K \leq 2 \cdot 10^5$).

The next N lines each contain one of the names from the list: a word composed of characters from a to z having between 1 and K characters.

Additional constraints:

- There are no two identical names in the list.
- The total number of characters in the list's names does not exceed $2 \cdot 10^5$.
- There exists some name (with between 1 and k characters from a to z) that is not in the list.

Output

Print, in a single line, the lexicographically smallest name (with between 1 and k characters from a to z) that is not in the list given in the input.

Examples

standard input	standard output
2 10 torterra pikachu	a
2 2 a aa	ab

Problem J. Joining Xegos

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 1024 megabytes

Tiago DFS is going to China!!

And, taking advantage of the trip, he decides to go to the largest ancestral store in the city to buy the pieces missing from his xego collection.

The xego section can be represented by an immense shelf with N xegos available, in positions numbered from 1 to N , for purchase.



But the eccentric Chinese owner of the store chooses a continuous interval $[L, R]$ every day and allows customers to buy only the items within that interval. Additionally, he only allows the customer to enter the store once a day, so always choose wisely which items you will buy before leaving the store.

After hours, the owner himself always restocks the items that were purchased that day with the same items.

Moreover, each xego piece has an identifier, which can be represented by a unique positive integer a_i .

And everyone knows that if we combine xegos, we can form other xegos, which can also, in turn, be combined into other xegos.

After playing a lot with xegos, Tiago noticed a curious property: the combination of two xegos A and B into another xego C can be exactly described as the XOR operation of their identifiers. That is, $A \oplus B = C$.

Knowing that he will have Q free days to buy xegos, help Tiago to answer, for each day, how many distinct combinations of shelf positions he can buy to form the xego X , knowing that on that day the owner allowed the sale only of the continuous interval $[L, R]$.

Input

The first line contains a positive integer N ($1 \leq N \leq 3 \cdot 10^5$) indicating the number of xegos on the shelf.

The second line contains N positive integers a_i ($1 \leq a_i \leq 10^{18}$), representing the xegos arranged on the shelves.

The third line contains a positive integer Q ($1 \leq Q \leq 3 \cdot 10^5$) representing the number of days Tiago will go shopping.

Then, in the next Q lines, three positive integers L, R ($1 \leq L \leq R \leq N$), and X ($1 \leq X_i \leq 10^{18}$) will be given, representing the limits of the interval and the xego of the day.

Output

For each day Q , respond with how many ways Tiago can form the xego X , knowing that only the continuous interval $[L, R]$ is available for purchase. Knowing that the value can be quite high, print the answer as the remainder of this value divided by 998244353.

Examples

standard input	standard output
5	1
1 2 4 8 16	1
3	0
1 3 7	
1 5 19	
4 4 16	
4	2
2 1 1 2	4
3	0
1 3 3	
1 4 2	
1 4 4	

Note

Test Case 1:

For day 1, Tiago can choose all the xegos in the interval $[1, 3]$, obtaining $1 \oplus 2 \oplus 4 = 7$.

For day 2, Tiago can choose the xegos at positions #1, #2, and #5, obtaining $1 \oplus 2 \oplus 16 = 19$.

For day 3, Tiago can only choose the item at position #4, which is of type 8, and cannot generate 16.

Test Case 2:

For day 1, there are 2 valid combinations #1, #2 and #1, #3.

For day 2, there are 4 valid combinations #1, #4, #1, #2, #3 and #2, #3, #4.

For day 3, there are no valid combinations.

Problem K. Koga needs you

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

The great Pokémon trainer Koga has decided that his team of Poison-type Pokémon needs more variety. To achieve this, Koga needs to understand a bit more about Pokémon types. Koga has 3 different Pokémon: Torterra, Staraptor, and Luxray, and he is going to face one of these three Pokémon. You must help him by telling which Pokémon Koga should use.

Torterra has an advantage over Luxray, Luxray has an advantage over Staraptor, and Staraptor has an advantage over Torterra.

Input

A line is given in the following format: “Vamos la, *S*!” (without quotation marks). *S* can be Torterra, Staraptor, or Luxray.

Output

Print a line in the following format: “*T*, eu escolho voce!” (without quotation marks). *T* can be Torterra, Staraptor, or Luxray.

Example

<code>standard input</code>	<code>standard output</code>
Vamos la, Torterra!	Staraptor, eu escolho voce!

Note

Staraptor has an advantage over Torterra, so if *S* is Torterra, *T* must be Staraptor.

Problem L. Lagrange's Legacy

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

Lórien, the last of the wizards of the Kingdom of Lagrange, found a secret formula carved into the walls of an ancient tower. It represents the absolute value of a powerful polynomial that, according to legends, possesses magical properties when applied correctly. The formula is:

$$f(x) = |(A + C)x^3 - Bx^2 + BCx + (x + C)(x - A)|.$$

To protect this knowledge, Lórien needs to apply the function to several values and reveal only the final result of a disguise spell: the XOR ($\oplus$) of all the results of the evaluations $f(x)$ at these values.

The XOR between two numbers a and b is another number c . To calculate $a \oplus b = c$, write the binary representation of a and b . Each bit (digit in binary representation) of c will be 1 if the corresponding bits in a and b are different, and 0 otherwise.

Your mission is to help him calculate this final value.

Input

The first line contains three integers A, B and C ($-100 \leq A, B, C \leq 100$).

The second line contains an integer Q ($2 \leq Q \leq 1000$), the number of values.

The third line contains Q integers $X_1, X_2, \dots, X_Q$ ($-100 \leq X_i \leq 100$), the values for which the function f should be evaluated.

Output

Print a single integer: the XOR of all values $f(X_i)$, for $1 \leq i \leq Q$.

Examples

standard input	standard output
1 2 3 3 0 1 2	34
13 100 -17 4 1 -10 5 -5	7509

Note

In the first test case, we have $f(0) = |-3| = 3$, $f(1) = |8| = 8$, and $f(2) = |41| = 41$. The binary representations of $f(0)$, $f(1)$, and $f(2)$, respectively, are as follows:

000011

001000

101001

. Thus, the XOR of these values in binary is 100010, which is 34 in decimal.

Problem M. Minimum Path

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

After a successful expedition for a marathon at UFG, the competitors AlbertoTDNeto, duduFreire, EYZ, henriqueros.ql, and MagePetrus made a pact: every time the 5 of them meet, they must roll a D100 (100-sided die). If the rolled number is 100, they must immediately go to Goiânia.

After many meetings, this finally happened, and the competitors will keep their promises. They all get into Henrique's car, which will drive them to Goiânia. The marathon runners want to take the shortest possible route from Brasília to Goiânia, of course. Additionally, since the car is quite heavy, Henrique is worried about the possibility of running out of fuel in the middle of the road. So, among the shortest paths, he wants a path that minimizes the longest distance the car will travel between two gas stations.

The gas stations are numbered from 1 to N . Henrique will start from Brasília at gas station number 1 and will finish his journey at gas station number N , in Goiânia. There are M roads connecting pairs of distinct gas stations. For each i from 1 to M , road number i starts from gas station a_i , leads to gas station b_i , and has a length of w_i kilometers. It is only possible to travel this road in the direction from a_i to b_i . It is guaranteed that there is a path from 1 to N .

Help Henrique find a shortest path that minimizes the longest distance traveled between two consecutive gas stations.

Input

The first line contains two integers $2 \leq N, M \leq 2 \cdot 10^5$: the number of gas stations and the number of roads connecting pairs of distinct gas stations, respectively.

Next, there are M input lines. The i -th of these line contains three integers a_i, b_i , and w_i , with $1 \leq a_i, b_i \leq N$ and $1 \leq w_i \leq 10^5$, representing a one-way road from gas station a_i to gas station b_i with a length of w_i kilometers.

Output

Print two integers, in this order: the length of the shortest path and the length of the longest edge traveled in a shortest path that minimizes the longest edge traveled.

Examples

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

Problem N. Na zdrowie

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

Frank and his friends are playing a game at the little gas station. The game has a list of n distinct languages $a_1, a_2, \dots, a_n$. Each language is represented by a number from 1 to n and appears only once. Frank does not trust the competitive programming skills of Brazilians, so he wants you to answer q queries. Each query contains two integers l and r , and you need to tell Frank if the sublist $a_l, a_{l+1}, \dots, a_r$ is a **good drink**. Since Frank only speaks Polish, respond with “TAK” if the list is a **good drink** and “NIE” otherwise.

A list of size m is a **good drink** if it is a permutation, that is, it contains all the numbers from 1 to m exactly once.

Input

The first line contains two integers n and q ($1 \leq n, q \leq 10^5$), the size of the list and the number of queries, respectively.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq n$), representing the elements of the list. It is guaranteed that the list is a permutation.

The next q lines contain two integers l and r ($1 \leq l \leq r \leq n$), representing the queries.

Output

For each query, print a line with the word “TAK” if the sublist $a_l, a_{l+1}, \dots, a_r$ forms a permutation, or “NIE” otherwise.

Examples

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