Problem A. Add One

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

Given n integers $a_1, a_2, \ldots, a_n$, you want to perform the following operation exactly $n-1$ times.

- Choose two integers x and y in the sequence, remove them, and add a number with the value $x \oplus y$.

Since this alone is just too boring, you can additionally choose a number and add one to it at any moment. You must perform the add-one operation **exactly once**.

Eventually, only one number will be left in this sequence, and you need to maximize this remaining number. Print the maximum value of the remaining number.

Input

The first line of the input contains a single integer n ($1 \le n \le 10^6$).

The next line of the input contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i < 2^{60}$).

Output

Output a single line containing a single integer: the maximum value of the remaining number.

Examples

standard input	*standard output*
4 1 2 1 2	7
5 1 2 3 4 5	14
6 1 2 4 7 15 31	47

Note

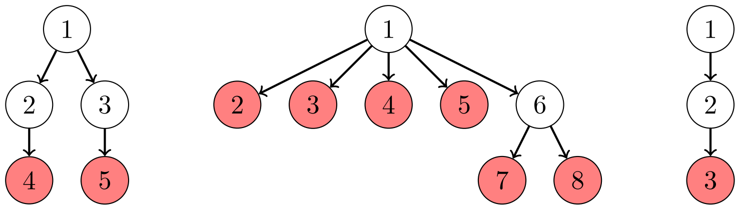
In the first example, the optimal strategy is:

- Choose 1 and 2: $[\mathbf{1}, \mathbf{2}, 1, 2] \to [1, 2, \mathbf{3}]$
- Choose 1 and 2: $[\mathbf{1}, \mathbf{2}, 3] \to [3, \mathbf{3}]$
- Add one to the number 3: $[\mathbf{3}, 3] \to [3, \mathbf{4}]$
- Choose 3 and 4: $[\mathbf{3}, \mathbf{4}] \to [\mathbf{7}]$

Problem B. Be Careful

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

You are given a rooted tree with n vertices, where the root is vertex 1. A vertex is a *leaf* if it is not the root vertex and its degree is exactly 1.



The figure corresponds to the sample tests, where the leaves are marked red.

Let $\operatorname{mex}(S)$ be the minimal non-negative integer that is not present in S. For example, $\operatorname{mex}\{0, 1, 3, 4\} = 2$, $\operatorname{mex}\{2, 3\} = 0$, $\operatorname{mex} \varnothing = 0$.

Let m be the number of leaves in the given tree. You will perform the following procedure:

1. For every **leaf vertex** u, write any integer from $\{0, 1, 2, \ldots, n\}$ to the vertex u.
2. For every **non-leaf vertex** u, the integer written in u will be the mex of the integers written in all the sons of vertex u.

For example, for the first tree which is described in the figure above, if we write integer 0 to vertex 4 and integer 3 to vertex 5, then:

- The integer written in vertex 2 will be $\operatorname{mex}\{0\} = 1$.
- The integer written in vertex 3 will be $\operatorname{mex}\{3\} = 0$.
- The integer written in vertex 1 will be $\operatorname{mex}\{1, 0\} = 2$.

In total, there are $(n + 1)^m$ ways to fill the tree. You would like to know, for all $k \in \{0, 1, 2, \ldots, n\}$, how many ways are there to fill the tree so that the number written in vertex 1 will be exactly k. Since the numbers can be huge, you only need to output them modulo 998 244 353.

Input

The first line of the input consists of a single integer n ($2 \le n \le 200$).

Each of the next $n - 1$ lines contains two integers x and y ($1 \le x, y \le n$, $x \ne y$), indicating that there is an edge between vertices x and y. It is guaranteed that the given graph is a tree.

Output

Output $n + 1$ lines. In the i-th line output a single integer, indicating the answer for $k = i - 1$, modulo 998 244 353.

Examples

standard input	standard output
5 1 2 1 3 2 4 2 5	55 127 34 0 0 0
8 1 2 1 3 1 4 1 5 1 6 6 7 6 8	69632 265534 133905 47790 12636 1944 0 0 0
3 1 2 2 3	1 3 0 0

Problem C. Counting Sequence

Input file:	*standard input*
Output file:	*standard output*
Time limit:	16 seconds
Memory limit:	1024 mebibytes

We are given integers n and c.

A sequence $a_1, a_2, \ldots, a_m$ is *good* if and only if:

- $a_i > 0$ for all $1 \le i \le m$,
- $|a_{i+1} - a_i| = 1$ for all $1 \le i \le m - 1$,
- $\sum_{i=1}^{m} a_i = n$.

For a good integer sequence $a_1, a_2, \ldots, a_m$, let us define

$$f(a) = \sum_{i=1}^{m-1} [a_i > a_{i+1}].$$

That is, $f(a)$ denotes the number of indices i that satisfy $a_i > a_{i+1}$ among all $1 \le i \le m-1$. We define the *weight* of the sequence a as the value of $c^{f(a)}$.

Your task is to calculate the sum of the weights of all good sequences, modulo 998 244 353.

Input

The first line contains two integers n and c ($1 \le n \le 3 \cdot 10^5$, $0 \le c < 998\,244\,353$).

Output

Output the answer modulo 998 244 353.

Examples

standard input	*standard output*
5 3	8
1 0	1
2022 39	273239559

Note

In the first example, all good sequences are as follows:

a	$f(a)$	$c^{f(a)}$
$[5]$	0	1
$[2,3]$	0	1
$[3,2]$	1	3
$[2,1,2]$	1	3

So the answer is $1 + 1 + 3 + 3 = 8$.

Problem D. DS Team Selection

Input file:	*standard input*
Output file:	*standard output*
Time limit:	18 seconds
Memory limit:	1024 mebibytes

The 34th *International Olympiad in Data Structures* will take place soon! In order to qualify, you need to pass the team selection contest in your country. As a member of the *Cat* team, you have to solve this problem in the *Cat* Team Selection contest.

There are infinitely many points **with integer coordinates** on an infinite plane, each of which can be represented as (x, y). Initially, the weights of all points are 0. You need to perform q operations, each of which takes the form:

- `1` x y d w: For all points (X, Y) that satisfy $|X - x| < d$ and $|Y - y| < d$, increase their point weights by $w \cdot (d - \max(|X - x|, |Y - y|))$.
- `2` x_1 x_2 y_1 y_2: Print the sum of the weights of points (x, y) that satisfy $x_1 \le x \le x_2$ and $y_1 \le y \le y_2$. Since the sum can be large, output it modulo 2^{30}.

Input

The first line contains a single integer m $(1 \le m \le 10^5)$, indicating the number of the operations.

The next m lines contains several integers in one of the following forms:

- `1` x y d w $(1 \le x, y, d, w \le 10^8)$
- `2` x_1 x_2 y_1 y_2 $(1 \le x_1 \le x_2 \le 10^8,\ 1 \le y_1 \le y_2 \le 10^8)$

Output

For each operation of type 2, print a single line containing an integer: the desired sum of the weights modulo 2^{30}.

Example

standard input	*standard output*
4 1 3 4 5 1 2 1 4 3 5 1 2 4 2 2 2 4 5 3 5	46 21

Problem E. Exciting Travel

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 mebibytes

After finishing the *IODS* team selection, you want to travel to a new country to relax. The country contains n cities, which are connected by $n-1$ bidirectional roads. There is a unique simple path between any two cities.

In the next m days, you wish to explore the country. On each day you have set k cities $x_1, x_2, \ldots, x_k$ that you wish to visit **in order**. At the beginning of each day, you can choose any city as the starting city of your trip, and then you need to reach city x_1, then city x_2, ..., and finally city x_k to complete your day's travel.

To add to the fun of the trip, you don't want to pass through a city more than once **in a day**. At any given moment, you can choose to follow a road from this city to another city, or choose to take a yacht to any city.

You want to know, for each day of travel, the minimum number of yacht rides required in order to avoid passing through the same city more than once. Note that each day's travel is independent: a city passed on the previous day can still be passed on the next day.

Input

The first line contains two integers n and m ($1 \le n \le 2 \cdot 10^5$, $0 \le m \le 5 \cdot 10^4$).

Each of the next $n-1$ lines contains two integers x and y ($1 \le x, y \le n$, $x \ne y$), indicating that there is a bidirectional edge between vertices x and y. It is guaranteed that the given graph is connected.

Each of the next m lines describes queries in the format $k\ x_1\ x_2\ \ldots\ x_k$. It is guaranteed that $1 \le x_i \le n$ and $x_i \ne x_j$ for all $1 \le i < j \le k$.

The sum of k in one test case does not exceed $2 \cdot 10^5$.

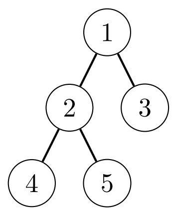
Output

For each day, output a single line containing a single integer: the minimum number of yacht rides.

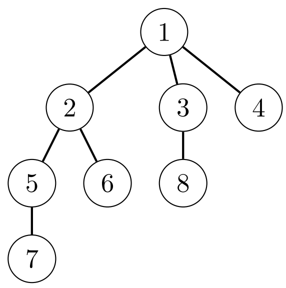
Examples

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

Note



The figure corresponds to the first sample test case



The figure corresponds to the second sample test case

Problem F. Flower's Land

Input file:	*standard input*
Output file:	*standard output*
Time limit:	10 seconds
Memory limit:	2048 mebibytes

The Kingdom of Flowers consists of n cities, and the i-th city grows a_i flowers. There are $n-1$ roads, where the i-th road connects cities u_i and v_i. It is guaranteed that for any two cities there is a path connecting them.

Now, the Kingdom of Flowers wants to hold a flower exhibition. To do that, you need to first choose a city z to build an exhibition hall, and then select exactly k cities $x_1, x_2, \ldots, x_k$ and transport the flowers from those k cities to the city z.

To avoid upsetting people in cities along the path, the organizers stipulated that if city x was selected, then all cities on the path from x to z had to be selected as well. In particular, this means that city z must be selected.

For each $z = 1, 2, \ldots, n$, find the maximum number of flowers that can be transported if city z is chosen to build the exhibition hall.

Input

The first line of the input contains two integers n and k ($1 \le n \le 40\,000$, $1 \le k \le \min(n, 3000)$).

The next line of the input contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 5 \cdot 10^5$).

Each of the next $n-1$ lines contains two integers x and y ($1 \le x, y \le n$, $x \ne y$), indicating that there is an edge between vertices x and y. It is guaranteed that the given graph is a tree.

Output

Output a single line containing n integers $f_1, f_2, \ldots, f_n$, where f_i denotes the answer for $z = i$.

Examples

standard input	*standard output*
5 3 6 10 4 3 4 3 4 4 2 2 5 5 1	20 20 17 17 20
7 4 1 3 2 1 7 12 17 4 6 1 4 5 1 2 5 7 6 3 2	31 13 13 31 21 31 31

Problem G. Games

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

You are given a tree of n vertices. Since you feel that there is nothing to do, you want to play a game on this tree.

Before the game, you need to assign a label $\ell_u \in \{0, 1, 2, \ldots, m-1, m\}$ to each vertex u.

The game consists of $m+1$ stages enumerated from 0 to m. In the i-th stage, all vertices u that satisfy $\ell_u \leq i$ will be painted black. If at this point, for every pair of **uncolored** vertices x and y, there exists a path from x to y that does not go through **any of the colored vertices**, then the game continues. Otherwise, you will lose and the game ends immediately. You win if the game continues after all stages.

You find that your ability to win the game depends only on how you initially assign labels to the vertices on this tree. In the next q days, you want to re-label the vertices and play the game. On the i-th day, you initially give the vertex a_i the label b_i. Then, you want to calculate how many ways are there to assign labels to the remaining vertices that allow you to win the game. Since the number could be large, you only need to output the answer modulo 998 244 353.

Input

The first line contains three integers n, m and q ($1 \leq n, q \leq 10^5$, $1 \leq m \leq 30$).

Each of the next $n-1$ lines contains two integers x and y ($1 \leq x, y \leq n$, $x \neq y$), indicating that there is an edge between vertices x and y. It is guaranteed that the given graph is a tree.

Each of the next q lines contains two integers a_i and b_i ($1 \leq a_i \leq n$, $0 \leq b_i \leq m$), indicating a query.

Output

For each query, output a single line containing a single integer, indicating the answer modulo 998 244 353.

Examples

standard input	standard output
3 2 5 1 2 3 2 3 1 1 0 2 0 2 1 2 2	7 9 5 8 9
6 6 6 6 1 1 3 2 6 4 2 5 2 2 1 3 2 5 3 5 6 5 3 1 1	896 2820 2433 1218 2433 2272

Problem H. Half Plane

Input file:	`standard input`
Output file:	`standard output`
Time limit:	12 seconds
Memory limit:	1024 mebibytes

This problem might be well-known in some countries, but how do other countries learn about such problems if nobody poses them.

There are n points on the plane, where the i-th point (x_i, y_i) has value $\mathbf{d}_i \in D$. Two sets D and O are given, with the following properties:

- There exists a special element ε_D in D.
- There exists a special element ε_O in O.
- A binary operation $+ : D \times D \to D$ is given with the following properties:
  - $\forall \mathbf{a}, \mathbf{b} \in D$, $\mathbf{a} + \mathbf{b} = \mathbf{b} + \mathbf{a}$
  - $\forall \mathbf{a}, \mathbf{b}, \mathbf{c} \in D$, $(\mathbf{a} + \mathbf{b}) + \mathbf{c} = \mathbf{a} + (\mathbf{b} + \mathbf{c})$
  - $\forall \mathbf{x} \in D$, $\mathbf{x} + \varepsilon_D = \varepsilon_D + \mathbf{x} = \mathbf{x}$
- A binary operation $\cdot : O \times D \to D$ is given with the following properties:
  - $\forall \mathbf{a}, \mathbf{b} \in O, \mathbf{x} \in D$, $(\mathbf{a} \cdot \mathbf{b}) \cdot \mathbf{x} = \mathbf{a} \cdot (\mathbf{b} \cdot \mathbf{x})$
  - $\forall \mathbf{a} \in O, \mathbf{x}, \mathbf{y} \in D$, $\mathbf{a} \cdot (\mathbf{x} + \mathbf{y}) = \mathbf{a} \cdot \mathbf{x} + \mathbf{a} \cdot \mathbf{y}$
- A binary operation $\cdot : O \times O \to O$ is given with the following properties:
  - $\forall \mathbf{a}, \mathbf{b}, \mathbf{c} \in O$, $(\mathbf{a} \cdot \mathbf{b}) \cdot \mathbf{c} = \mathbf{a} \cdot (\mathbf{b} \cdot \mathbf{c})$
  - $\forall \mathbf{x} \in O$, $\mathbf{x} \cdot \varepsilon_O = \varepsilon_O \cdot \mathbf{x} = \mathbf{x}$

In this problem, we treat D as the set of all 3×1 matrices over $\mathbb{F}_p$ and O as the set of all 3×3 matrices over $\mathbb{F}_p$, where $p = 10^9 + 7$. That is, you can treat the above operations as the usual matrix addition and matrix multiplication modulo $10^9 + 7$.

Now, m queries are given in the form `a b c o`:

- Let $\mathbf{s} = \varepsilon_D$.
- For all points i with $ax_i + by_i < c$, modify $\mathbf{s}$ to $\mathbf{s} + \mathbf{d}_i$, then modify $\mathbf{d}_i$ to $\mathbf{o} \cdot \mathbf{d}_i$.
- Return $\mathbf{s}$ as the answer of the query.

As a data structure master, you need to perform all queries and find the answer.

Input

The first line of the input contains a single integer n ($1 \leq n \leq 3 \cdot 10^5$), indicating the number of points.

Each of the following n lines contains **five** integers $x_i, y_i, d_{i0}, d_{i1}, d_{i2}$, indicating the coordinates of the i-th point and its value $\mathbf{d}_i = \begin{bmatrix} d_{i0} \\ d_{i1} \\ d_{i2} \end{bmatrix}$.

The next line of the input contains a single integer m ($1 \leq m \leq 1.5 \cdot 10^4$), indicating the number of the queries.

Each of the following m lines contains **twelve** integers $a, b, c, o_{00}, o_{01}, o_{02}, o_{10}, \ldots, o_{22}$. Note that the real $\mathbf{o} = \begin{bmatrix} o_{00} & o_{01} & o_{02} \\ o_{10} & o_{11} & o_{12} \\ o_{20} & o_{21} & o_{22} \end{bmatrix}$.

It is guaranteed that:

- $|x_i| \le 10^6$, $|y_i| \le 10^6$.
- $|a_i| \le 10^3$, $|b_i| \le 10^3$, $b_i \ne 0$, $|c_i| \le 10^6$.
- All matrix elements are from 0 to $10^9 + 6$ inclusive.
- For all $1 \le i \le m$ and $1 \le j \le n$, $a_i x_j + b_i y_j \ne c_i$.
- For all $1 \le i \le m$ and $1 \le j \le m$, $\left(\frac{a_i}{b_i}, \frac{c_i}{b_i}\right) \ne \left(\frac{a_j}{b_j}, \frac{c_j}{b_j}\right)$.

Output

For each query, output a single line containing three integers s_0, s_1, s_2, indicating $\mathbf{s} = \begin{bmatrix} s_0 \\ s_1 \\ s_2 \end{bmatrix}$.

Example

standard input	*standard output*
5 1 1 2 3 4 12 12 4 6 1 1 12 5 1 2 12 1 1 5 5 6 6 2 0 3 3 1 1 4 1 1 2 3 4 5 2 3 4 1 1 400 1 3 4 2 1 2 3 4 5 -1 -1 -10 3 2 1 4 6 5 4 3 2	2 3 4 25 50 40 92 58 139

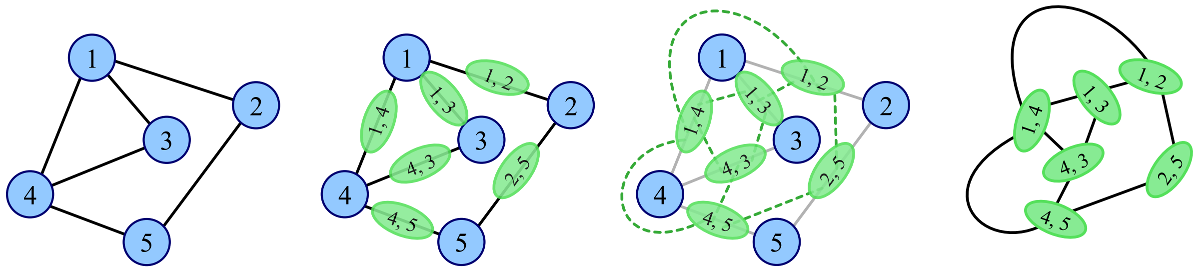
Note

Note that the solution does not depend on other properties of matrix addition/multiplication than those mentioned in the statements. Defining D and O as sets of matrices is only for testing convenience (since we can't use the graders or interaction libraries).

Problem I. Inverse Line Graph

Input file:	*standard input*
Output file:	*standard output*
Time limit:	7 seconds
Memory limit:	1024 mebibytes

In the mathematical discipline of graph theory, the line graph of an undirected graph G is another graph $L(G)$ that represents the adjacencies between edges of G. $L(G)$ is constructed in the following way: for each edge in G, make a vertex in $L(G)$; for every two edges in G that have a vertex in common, make an edge between their corresponding vertices in $L(G)$. (*From Wikipedia*)



Example of line graph construction

You have solved many tasks based on the line graphs, but you like line graphs so much that you want to solve one more task!

You are given a simple undirected graph G with n vertices and m edges. Your task is to find another simple undirected graph H, such that G is the line graph of H.

Input

There are multiple test cases. The first line contains a single integer T ($1 \le T \le 3 \cdot 10^5$), indicating the number of test cases. The test cases follow, for each test case:

The first line contains two integers n and m ($1 \le n \le 3 \cdot 10^5$, $0 \le m \le 3 \cdot 10^6$), indicating the number of the vertices and edges in the graph G.

Each of the following m lines contains two integers u and v ($1 \le u, v \le n$, $u \ne v$), indicating a bidirectional edge between vertex u and vertex v in G.

It is guaranteed that $1 \le \sum n \le 3 \cdot 10^5$ and $0 \le \sum m \le 3 \cdot 10^6$, and the given graph does not contain multiple edges or self-loops.

Output

For each test case, if such a graph H does not exist, output a single line with the word "`No`".

Otherwise, output a line with the word "`Yes`", followed by a line containing two integers n' and m' indicating the number of vertices and the number of edges of H ($0 \le n' \le 10^6$, $m' = n$).

Each of the following m' lines must contain two integers u and v ($1 \le u, v \le n'$, $u \ne v$), indicating a bidirectional edge between vertex u and vertex v in H.

Note that the edges in H will be numbered $1, 2, \ldots, m'$ **in the order you output them**. You need to make sure that the numbering of the edges corresponds to the numbering of the vertices in G.

If there are multiple possible solutions, you can output any one of them.

Example

standard input	*standard output*
6 5 6 1 2 1 3 1 4 3 4 2 5 4 5 1 0 2 1 1 2 3 3 1 2 1 3 2 3 4 3 1 2 1 3 1 4 5 6 1 2 2 3 2 4 3 4 3 5 4 5	Yes 5 5 3 4 1 3 4 5 2 4 1 2 Yes 2 1 1 2 Yes 3 2 2 3 1 2 Yes 4 3 2 3 2 4 1 2 No Yes 5 5 4 5 3 4 1 3 2 3 1 2

Problem J. Just Another Number Theory Problem

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

Given are n **prime** numbers $1 < p_1 < p_2 < \ldots < p_n < 10^{18}$ with $p_1 \leq 100$. We say that the number x is *good* if x is divisible by at least one p_i.

Take all *good* numbers $a_1, a_2, \cdots, a_m$ in $[0, p_1 \cdot p_2 \cdot \ldots \cdot p_n]$ and sort them in order ($a_1 < a_2 < \ldots < a_m$). Your task is to calculate $\sum_{i=1}^{m-1}(a_{i+1} - a_i)^2$. As the sum could be very large, you should output it modulo 998 244 353.

Input

The first line of the input contains a single integer n ($1 \leq n \leq 10^5$).

The next line of the input contains n integers $p_1, p_2, \ldots, p_n$ ($1 < p_1 < p_2 < \ldots < p_n < 10^{18}$). It is guaranteed that $2 \leq p_1 < 100$ and each p_i ($1 \leq i \leq n$) is a prime number.

Output

Output a single line with a single integer, indicating the answer modulo 998 244 353.

Examples

standard input	*standard output*
2 2 5	18
3 5 7 233	31275

Note

In the first example, the list of good numbers is:

- $a_1 = 0$
- $a_2 = 2$
- $a_3 = 4$
- $a_4 = 5$
- $a_5 = 6$
- $a_6 = 8$
- $a_7 = 10$

Thus, the answer is $(2-0)^2 + (4-2)^2 + (5-4)^2 + (6-5)^2 + (8-6)^2 + (10-8)^2 = 18$.

Problem K. Matrix Counting

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	1024 mebibytes

We call an $n \times n$ matrix containing only `0`s and `1`s *bad* if and only if it contains exactly one `1` in each row and column.

Bad	**Bad**	**Bad**	**Not Bad**	**Not Bad**	**Not Bad**
$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

Define B to be a *subrectangle* of an $n \times n$ matrix A if and only if there exist $1 \le l_1 \le r_1 \le n$ and $1 \le l_2 \le r_2 \le n$ such that

- B is a $(r_1 - l_1 + 1) \times (r_2 - l_2 + 1)$ matrix.
- $B_{i,j} = A_{l_1+i-1, r_1+j-1}$ $(1 \le i \le r_1 - l_1 + 1, 1 \le j \le r_2 - l_2 + 1)$

A	B	**Explanation**
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & \mathbf{0} & \mathbf{0} \\ 0 & \mathbf{0} & \mathbf{1} \\ 0 & 1 & 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \mathbf{1} & \mathbf{0} & 0 \\ \mathbf{0} & \mathbf{0} & 1 \\ 0 & 1 & 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	Not a subrectangle

Given two integers n and m, you want to calculate how many $n \times n$ matrices M containing only `0`s and `1`s are there such that:

1. M is *bad*,
2. all its subrectangles of size $k \times k$ $(k = m+1, m+2, \ldots, n-1)$ are not *bad*.

Since the answer can be large, output it modulo 998 244 353.

Input

The first line contains two integers n and m $(1 \le m < n \le 10^5)$.

Output

Output a single line containing a single integer, indicating the answer modulo 998 244 353.

Examples

standard input	*standard output*
`3 2`	`6`
`4 2`	`4`
`300 20`	`368258992`
`100000 1`	`91844344`

Note

In the first example, there are 6 *bad* matrices. The second condition does not matter since $m + 1 = 3 > n - 1 = 2$. So the answer is 6.

In the second example, there are 4 matrices satisfying the conditions:

$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Problem L. No!

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	1024 mebibytes

Given are n walls, the i-th wall has height h_i and strength s_i. You need to arrange them in a line from left to right.

For the next q days, the wind will blow every night. Due to the inaccuracy of the weather forecast, you cannot be sure of the strength of the wind blowing each night.

On the morning of the k-th day, the security center will build an infinitely strong wall of height h'_k on the far left side of the ground. You can then arrange the n walls you have in any order to the right of this wall. Note that the wall built by the security center each day is only available for that day. The original one will be removed before a new wall is built the next day.

In the evening, the wind with strength v will blow from the far left. For each i, the i-th wall from the left will receive the impact of $\Delta_i = v \cdot \max(0, h_i - \max_{0 \le j < i} h_j)$. Here, wall 0 is the wall that the security center built: its height on the k-th day is h'_k, and its strength is infinite. If $\Delta_i > s_i$ for some i, then the i-th wall will collapse and you will suffer a heavy economic loss.

You want to know, for each day, the maximum strength of the wind you can sustain without suffering economic losses if you arrange your n walls optimally.

It can be shown that the answer is either infinite or can be expressed in the form $r = a/b$, where a and b are coprime positive integers. You only need to output the values of a and b.

Input

The first line of the input consists two integers n and q ($1 \le n, q \le 5 \cdot 10^5$).

Each of the next n lines contains two integers h_i and s_i ($1 \le h_i \le 4 \cdot 10^6$, $1 \le s_i \le 10^9$), indicating the height and strength of the i-th wall.

Each of the next q lines contains an integer h'_k ($1 \le h'_k \le 4 \cdot 10^6$), indicating the height of the infinitely strong wall that the security center built on the k-th day.

Output

Output q lines, each in the format of an irreducible fraction "a/b", with no spaces and a slash in between, indicating that the answer has the value $r = a/b$. If the walls can withstand wind of any strength, print the infinite answer as "`1/0`".

Examples

standard input	*standard output*
4 5 3 5 4 3 2 5 6 3 5 2 3 7 2	3/1 3/2 3/2 1/0 3/2
5 6 10 6 3 7 6 15 7 6 8 15 5 3 9 7 7 9	3/1 3/1 6/1 3/1 3/1 6/1