Problem A. Final Exam

Input file:	*standard input*
Output file:	*standard output*
Time limit:	12 seconds
Memory limit:	256 mebibytes

Rikka is a talented student.

She spends almost every day on ICPC. But the final exam is approaching.

Rikka plans to grasp- the last minute to review the courses before the exam. She has up to M minutes for review and then takes n consecutive exams. If Rikka spends x minutes on the review for the i-th exam, she would get $f_i(x)$ points, where $f_i(x) = \max\{0, \min\{d_i, a_i x^2 + b_i x + c_i\}\}$ with the exam-specific parameters a_i, b_i, c_i, d_i.

Rikka wants to maximize the total score of her n exams. Note the minutes she spends in reviewing a certain course can be any non-negative real number. Also, she does not have to spend all of her M minutes on the review so that she can spend more time on ICPC.

Input

The first line contains an integer n and a real number M.

Each of the following n lines contains four real numbers a_i, b_i, c_i, d_i, denoting the parameters of all the n exams.

It is guaranteed that $1 \leq n \leq 100\,000$, $0 < M \leq 10^8$, $|a_i| \leq 10$, $|b_i| \leq 5000$, $0 \leq c_i \leq d_i \leq 5000$, and all real numbers in the input are given with exactly three decimal places.

It is guaranteed that there are at most 18 exams with $a_i > 0$.

Output

You need to output d, the maximum total score that Rikka can get. Assuming the correct result is d^*, you need to ensure that $\frac{|d-d^*|}{\max\{d^*,1\}} \leq 10^{-6}$.

Example

standard input	standard output
4 2.000 0.000 7.000 3.000 10.000 -1.000 10.000 3.000 10.000 -2.000 10.000 3.000 10.000 -3.000 10.000 3.000 10.000	29.5734198185

Problem B. Travel around China

Input file:	*standard input*
Output file:	*standard output*
Time limit:	8 seconds
Memory limit:	512 mebibytes

Rikka is a rich girl.

She would like to visit the beautiful cities of China. The city locations in China can be simply regarded as a grid that contains n rows and m columns. Rows are numbered by 1 to n from north to south, and columns are numbered by 1 to m from west to east. The city located in the i-th row and the j-th column is called (i, j).

There are some expressways that connect the whole country. City (i, j) has a direct expressway to city (x, y) if and only if $|i - x| + |j - y| = 1$. Because New Year is coming, expressways are opened to the public free of charge.

When Rikka travels from city (i, j) to (x, y), she can only travel through expressways. The cost of a travel route is the sum of the costs of all the cities she visits, including the starting and the ending cities. If the route includes some city, she will visit scenic spots, go shopping, and spend money. If the route includes city (i, j), she will spend $a_{i,j}$ yuan. And if she visits city (i, j) a total of k times, she will spend $k \cdot a_{i,j}$ yuan, because there are always enough shopping malls for her to spend money.

Rikka is a fanciful girl, she does not even set the starting and the ending city. She wants to know the sum of costs of all the cheapest routes with different starting and ending cities. In other words, let $f(i, j, x, y)$ be the minimal cost of the route that starts from city (i, j) and ends at city (x, y). She wants to know the value

$$\sum_{i=1}^{n}\sum_{x=1}^{n}\sum_{j=1}^{m}\sum_{y=1}^{m}[(i, j) \neq (x, y)]f(i, j, x, y).$$

Because the answer may be very large, you just need to tell her the answer modulo 1 000 000 007 (that is, $10^9 + 7$).

Input

The first line contains two integers n and m.

Each of the following n lines contains m integers. The j-th number in the i-th line is the value of $a_{i,j}$.

It is guaranteed that $n = 3$, $1 \leq m \leq 1.5 \cdot 10^5$, and $1 \leq a_{i,j} \leq 10^9$.

Output

Output a single line with a single integer, the answer modulo 1 000 000 007 (that is, $10^9 + 7$).

Example

standard input	standard output
3 3 1 1 1 1 100 1 1 1 1	1808

Problem C. Wandering

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

Rikka is a talented student.

She likes to wander in the corridor while solving ICPC problems. Specifically, she will do a random walk for n steps. In the i-th random step, she will choose one of the vectors (x, y) such that $x, y \in \mathbb{R}$ and $x^2 + y^2 \le R_i^2$ with equal probability. And then she will walk along the vector. In other words, if she stood at (A, B) before the random step, she will stand at $(A + x, B + y)$ afterwards. Before wandering, she stands at the door $(0, 0)$.

After wandering, she was curious about the expectation of the square of Euclidean distance to point $(0, 0)$. In other words, she wants to know the expected value of $x^2 + y^2$, if she stands at (x, y) after all n random steps.

Input

The first line contains an integer n, the number of random steps.

The second line contains n positive integers R_i, the parameter of the i-th random step.

It is guaranteed that $1 \le n \le 50\,000$ and $1 \le R_i \le 1000$.

Output

You need to output d, the expected value of $x^2 + y^2$. Assuming the correct result is d^*, you need to ensure that $\frac{|d-d^*|}{\max\{d^*,1\}} \le 10^{-6}$.

Example

standard input	standard output
3 1 2 3	7.000000000000000

Problem D. Insects

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

MianKing is playing a game. In this game, he has n insects, and each insect has two integer attributes: type and level. The type and level of the i-th insect are $type_i$ and $level_i$, respectively.

Initially, each of these n insects has a "seed" buff. When an insect with a "seed" buff is eliminated, let L denote the highest level among the remaining insects (with the seed buff or not) of the same type as the eliminated insect. Then MianKing can choose an integer type D from $[1, n]$ arbitrarily and add a new insect of type D and level L. And this new insect does not have the "seed" buff.

Notice that if there are no other insects of the same type as the eliminated insect, no new insect can be added.

Now MianKing wants to maximize the total level of all insects on the field by eliminating some insects. The total level is the sum of levels of individual insects. You need to help him to calculate ans_K, the maximum total level that he can get by eliminating at most K insects.

Input

The first line has one integer n ($1 \le n \le 10^5$).

Then there are n lines, where the i-th line contains two integers $type_i$ and $level_i$ ($1 \le type_i \le n$, $1 \le level_i \le 10^7$).

Output

Output n lines, such that the i-th line has one integer ans_i.

Examples

standard input	standard output
4 1 5 1 6 2 2 3 1	15 20 24 24
6 1 1 2 2 3 3 4 4 5 5 5 5	20 24 27 29 30 30
4 1 1 2 2 3 3 4 4	10 10 10 10

Problem E. Minimum Spanning Tree

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

MianKing has a graph with n nodes and m edges, where the i-th edge (x_i, y_i) has an edge weight of w_i.

The Minimum Spanning Tree of the graph is a spanning tree with the minimum sum of edge weights.

MianKing forgot the weights $w_{1...m}$, but he still remembers that $w_{1...m}$ are a permutation of $\{1...m\}$ and that the edge set of the Minimum Spanning Tree of this graph consists of the first $n-1$ edges.

Now you need to help MianKing to calculate how many $w_{1...m}$ satisfy the conditions above. The answer may be very large, so you only need to output the answer modulo 998 244 353.

Input

The first line contains two integers n and m ($2 \le n \le 20$, $n-1 \le m \le 100$).

Then there are m lines, where the i-th line contains two integers x_i and y_i ($1 \le x_i, y_i \le n$).

It is guaranteed that the edges (x_1, y_1), …, (x_{n-1}, y_{n-1}) form a tree with n nodes.

Note that the graph may have multiple edges and self-loops.

Output

Output the answer modulo 998 244 353.

Examples

standard input	standard output
3 3 1 2 2 3 3 1	2
4 5 1 2 2 3 2 4 1 4 1 2	25
3 6 1 2 2 3 1 1 1 1 1 1 1 1	720

Problem F. Minimal Cut

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

Today Rikka got an undirected graph G with n vertices and m edges. The vertices are numbered by integers from 1 to n. The i-th edge connects vertices u_i and v_i, and its weight is w_i.

Rikka likes Hamiltonian graphs: the ones that have a Hamiltonian cycle. Therefore, Rikka constructs a graph based on G that is surely Hamiltonian. She does so by inserting n extra edges: the i-th edge connects vertices i and $(i \bmod n + 1)$, and its weight is 10^9.

Let $c(i, j)$ be the value of the minimal cut between the i-th and the j-th vertices. Rikka wants you to calculate

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} c(i, j).$$

Given a graph $G_0 = \langle V, E \rangle$, a set of edges $C \subseteq E$ is a *cut* between vertices i and j if and only if in graph $G_1 = \langle V, E \setminus C \rangle$, vertices i and j are not (indirectly or directly) connected. The *minimal cut* between i and j is the cut with the minimal sum of edge weights. The *value* $c(i, j)$ of the minimal cut is this minimal sum itself.

Input

The first line contains two integers n and m ($3 \le n \le 20\,000$, $0 \le m \le 20\,000$).

Then m lines follow. Each of them contains three integers u_i, v_i, and w_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$ and $1 \le w_i \le 10\,000$).

Note that the graph has no self-loops, but may contain multiple edges.

Output

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

Example

standard input	standard output
4 2 1 3 2 2 4 2	21067776

Problem G. Revenue

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

There is a seller who has n items for sale to a single buyer. The buyer has a *valuation profile* $\bar{v} = (v_1, \ldots, v_n)$, where $v_j \geq 0$ denotes her value for item j.

The seller can set a *pricing* $\bar{p}$, that is, a vector of item prices $(p_1, \ldots, p_n)$. Given a pricing $\bar{p}$, the *utility* of buying item j is $v_j - p_j$. The buyer will purchase a single item j that maximizes her utility, or nothing if her utility from purchasing any item would be negative. If there are multiple items with the same maximal utility, she will choose the one with the minimal price. The *revenue* of the seller is defined as the price of the item that the buyer buys, and if the buyer buys nothing, the revenue is 0.

Now we know that the valuation profile $\bar{v}$ is drawn from a *joint distribution* F which defines the probability of every possible value of $\bar{v}$. Unfortunately, we **do not** know F. Instead, we know the *marginal distributions* $F_1, F_2, \ldots, F_n$: distribution F_i defines the probability of $v_i = x$ for every possible x. But we do not know how they are correlated: the values are not necessarily independent, so the individual probabilities of $v_i = x$ and $v_j = y$ don't define the probability of both happening simultaneously. Note that the joint distribution F is over the valuation profile $\bar{v}$ and that the marginal distribution F_i is over the value v_i of item i.

Given the pricing $\bar{p}$ and the marginal distributions $F_1, F_2, \ldots, F_n$, we are now asked to compute the minimal expected revenue among all possible joint distributions. Formally, let $\mathcal{F}$ be the set of joint distributions over valuation profiles $\bar{v}$ whose marginal distributions for the individual item values are just $F_1, F_2, \ldots, F_n$. Let $\mathrm{Rev}(\bar{p}, F)$ be the seller's expected revenue from setting a pricing $\bar{p}$, if the valuation profile $\bar{v}$ is drawn from a joint distribution F. We are asked to compute

$$\min_{F \in \mathcal{F}} \mathrm{Rev}(\bar{p}, F).$$

Input

The first line contains a single integer n ($1 \leq n \leq 10^5$), the number of items for sale.

The second line contains n non-negative integers $p_1, p_2, \ldots, p_n$ ($0 \leq p_i \leq 10^5$), the pricing vector $\bar{p}$.

Next n lines describe the marginal distributions $F_1, F_2, \ldots, F_n$. Each line starts with an integer k: the support size (number of different values) of F_i. Then follow k pairs of numbers q_j and v_j ($0 \leq q_j \leq 1$, $0 \leq v_j \leq 10^6$), meaning that F_i has probability of q_j to have value v_j. The values v_j may be given as decimal fractions or in scientific notation. It is guaranteed that $\sum_{j=1}^{k} q_j = 1$.

The total sum of the values of k on these n lines will not exceed $3 \cdot 10^5$. The total size of the input will not exceed 5 mebibytes.

Output

Output a single real number: the minimal expected revenue among all possible joint distributions. Your answer will be considered correct if and only if its absolute or relative error does not exceed 10^{-6}.

Examples

standard input	standard output
2 2 5 4 0.254 5 0.227 8 0.269 10 0.25 9 4 0.274 9 0.272 9 0.223 8 0.231 7	2.0000000000
2 7 7 2 0.5 1 0.5 0 2 0.3 5 0.7 1	0.0000000000
1 5 1 1.0 5	5.0000000000

Problem H. Longest Loose Segment

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

A list A is called *loose* if $\max(A) + \min(A) > \text{len}(A)$.

Today Rikka got a list A of length n. She wants to find the longest segment $[l, r]$ in A such that list $[A_l, A_{l+1}, \ldots, A_r]$ is loose.

Rikka will make m turns with list A. On each turn, Rikka will perform one or more given operations in sequence. Each operation is swapping two elements in list A. Your task is to calculate the length of the longest loose segment of A and the resulting list after each turn.

Note that the operations on turn i are performed on the list that was the result of turn $(i-1)$.

Input

The first line contains two integers n and m ($1 \le n \le 10^6$ and $1 \le m \le 30$).

The second line contains n integers A_i ($-10^6 \le A_i \le 10^6$) that constitute the initial list A.

Then follow m descriptions of the turns. For each turn, the first line contains a single integer k ($1 \le k \le 10^6$), the number of swaps. Then k lines follow: each of them contains two integers u_i and v_i ($1 \le u_i, v_i \le n$ and $u_i \ne v_i$) such that Rikka will swap A_{u_i} and A_{v_i} in this operation.

It is guaranteed that $\sum k \le 10^6$.

Output

On the first line, output a single integer: the length of the longest loose segment of A.

Then output m lines. On each of them, print a single integer: the length of the longest loose segment of the resulting list after each turn.

Example

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

Problem I. Color

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

Big Horse is the God of Mathematics. He has drawn a complete undirected graph with n vertices. Each edge has one of the m colors, numbered $1, \ldots, m$. Big Horse has a big ambition to extend this graph to a maximal possible complete graph, such that any two edges with the same endpoint have different colors. He finds out that obviously, the graph has at most $m + 1$ vertices. So he asks you if he can extend his graph to $m + 1$ vertices.

Input

In the first line there are two integers n and m ($1 \le n \le 200$, $1 \le m \le 200$, and $n \le m + 1$).

Then there are $n - 1$ lines. In the i-th of these lines, there are $n - i$ numbers. The j-th number in the i-th line indicates the color of the edge connecting vertex i and $i + j$. All colors are integers from 1 to m.

Output

In the first line, output "`Yes`" (without quotes) if you can extend the graph, or "`No`" otherwise.

If the first line is "`Yes`", output m extra lines. In the i-th of these lines, print $m + 1 - i$ numbers. The j-th number in the i-th line indicates the color of the edge connecting vertices i and $i + j$. The edges which were given in the input must be colored as in the input. Any two edges with the same endpoint must have different colors. If there are several possible answers, print any one of them.

Examples

standard input	standard output
3 5 1 2 4	Yes 1 2 4 3 5 4 3 5 2 5 1 3 2 1 4
4 5 1 2 3 3 2 1	No

Problem J. Horses

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

Big Horse is the God of Horses. He has n different kinds of horses. Since his eyes are not so good, he cannot distinguish between the horses of the same kind.

Now he wants to arrange m horses in a queue. But the horses are so active that they may change their positions at will. However, Big Horse noticed that only two adjacent horses may swap, and this may happen only if the two **kinds** of horses are friends. Since the horses can swap their positions at any time, Big Horse considers two queues equivalent if and only if one can be reached from the other one by a finite number of swaps.

Now Big Horse has a queue $a = (a_1, \ldots, a_m)$ of horses. He wants to add some other horses to the left of the queue. However, Big Horse cannot tell left from right. So he wants to add a queue $b = (b_1, \ldots, b_k)$ such that b commutes with a: in other words, $b_1, \ldots, b_k, a_1, \ldots, a_m$ is equivalent to $a_1, \ldots, a_m, b_1, \ldots, b_k$.

However, the number of such b may be too large. Big Horse only cares about the "minimal" such queues b. Specifically, he is interested in b such that:

- b commutes with a,
- b is not equivalent to $c_1, \ldots, c_{k'}, d_1, \ldots, d_{k''}$ such that c commutes with a and d commutes with a,
- b is lexicographically the least among all queues equivalent to it.

He found out that there are at most n minimal queues. He asks you to help him find them.

Input

In the first line, there is an integer n ($1 \le n \le 200$).

Then follow $n-1$ lines. In the i-th of these lines, there are $n-i$ integers. The j-th integer in the i-th of these lines is 1 if a horse of kind i can swap with a horse of kind $i+j$, and 0 otherwise.

The next line contains an integer m ($1 \le m \le 300\,000$).

The last line contains m integers $a_1, \ldots, a_m$: the kinds of horses in the queue ($1 \le a_i \le n$).

Output

Output the minimal queues, one per line. Since a queue may be too long, when the minimal queue is b, you only need to print the hash value $b_1 + b_2 \cdot (n+1) + \ldots + b_k \cdot (n+1)^{(k-1)}$ modulo $998\,244\,353$.

You should output the minimal queues in lexicographical order (order them before hashing).

Example

standard input	standard output
3 1 1 0 5 2 1 3 2 3	1 14

Note

The two minimal queues in the example are (1) and $(2, 3)$.