The 2021 icpc Asia Jinan Regional Contest

Contest Session

November 14, 2021

Problems

A. Space Station
B. Monitored Area
C. Optimal Strategy
D. Arithmetic Sequence
E. Insidemen
F. Neural Network Counting
G. Happy Alice
H. Game Coin
I. Permutation Pair
J. Determinant
K. Search for Mafuyu
L. Strange Series
M. Coloring Rectangles

Do not open before the contest starts.

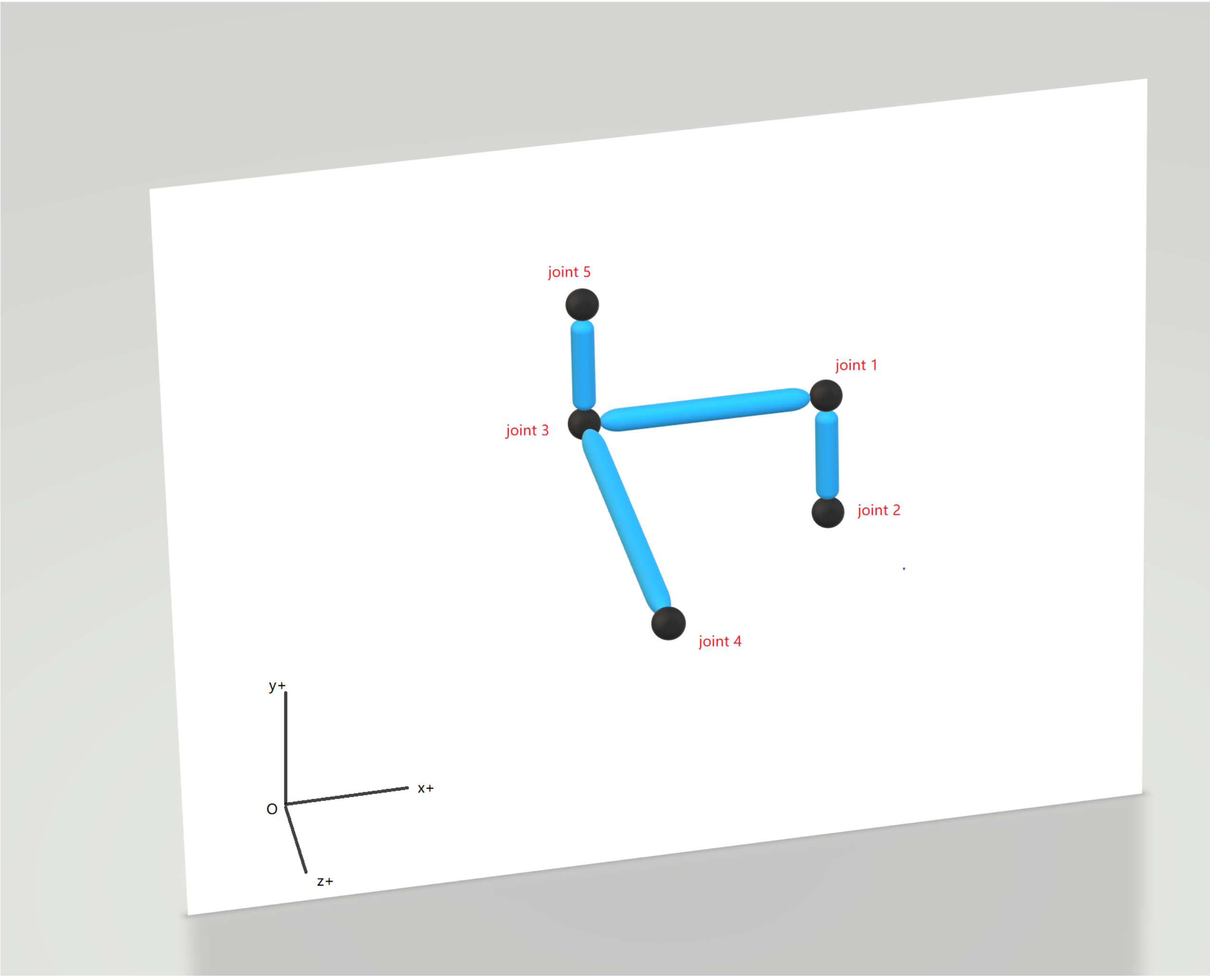
Prepared by Peking University

Problem A. Space Station

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	512 megabytes

"TianTing" is a huge space station. It consists of several thin cylindrical capsules. There is a joint on each end of a capsule. One joint may connect to several capsules.

If we take joints as nodes and take capsules as edges the structure of the space station can be seen as a tree.



The structure of space station corresponding to the sample input.

The shape of the space station can be shifted by rotating capsules around joints. At the beginning directions, (from one end to the other) of all the capsules are parallel to the 3D coordinate axes. A capsule can rotate around a joint of its either end as pivot arbitrarily as long as its final direction is still parallel to one of the axes. Note all the capsules which are connected to the other end (directly or indirectly) will also rotate at the same time. You do not need to worry about any collisions or overlapping of capsules. Assume they can "pass through" each other safely.

Given the shape of the space station at the beginning and a sequence of rotations afterwards can you tell the Euclidean distance between two given joints?

Input

The first line contains two integers N and Q ($2 \le N \le 10^5, 1 \le Q \le 10^5$) – the number of joints and the number of operations.

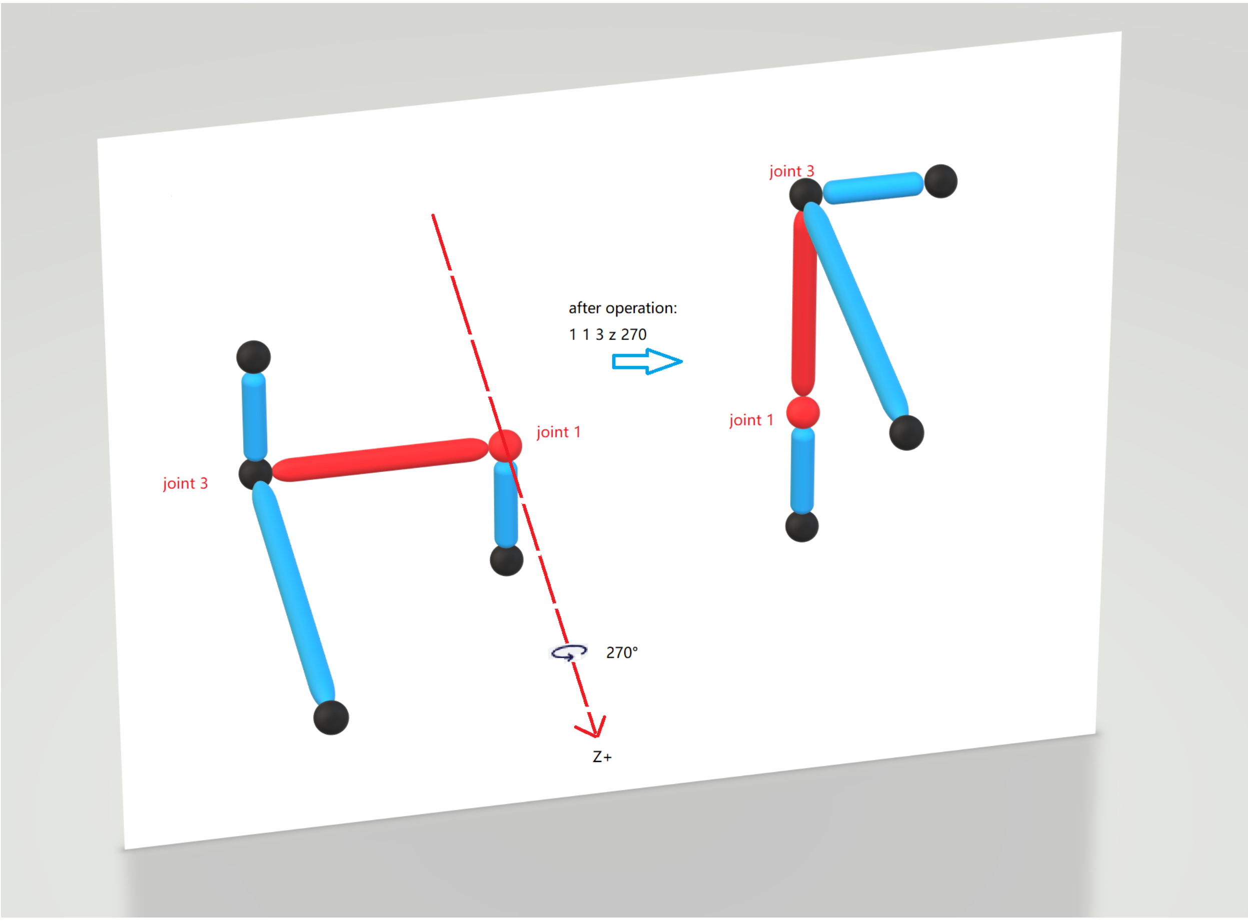
Each of the following $N-1$ lines contains three integers u, v, w ($1 \le u, v \le N, u \ne v, 1 \le w \le 10^6$) and two characters which means there is a capsule of length w between joint u and joint v. The two characters are one of "x+", "x-", "y+", "y-", "z+", "z-" which represents the direction of the capsule from joint u to joint v. It is guaranteed that the given structure is a tree.

Each of the following Q lines contains an operation. There are two types of operation: rotation of capsule and query on Euclidean distance of two given joints.

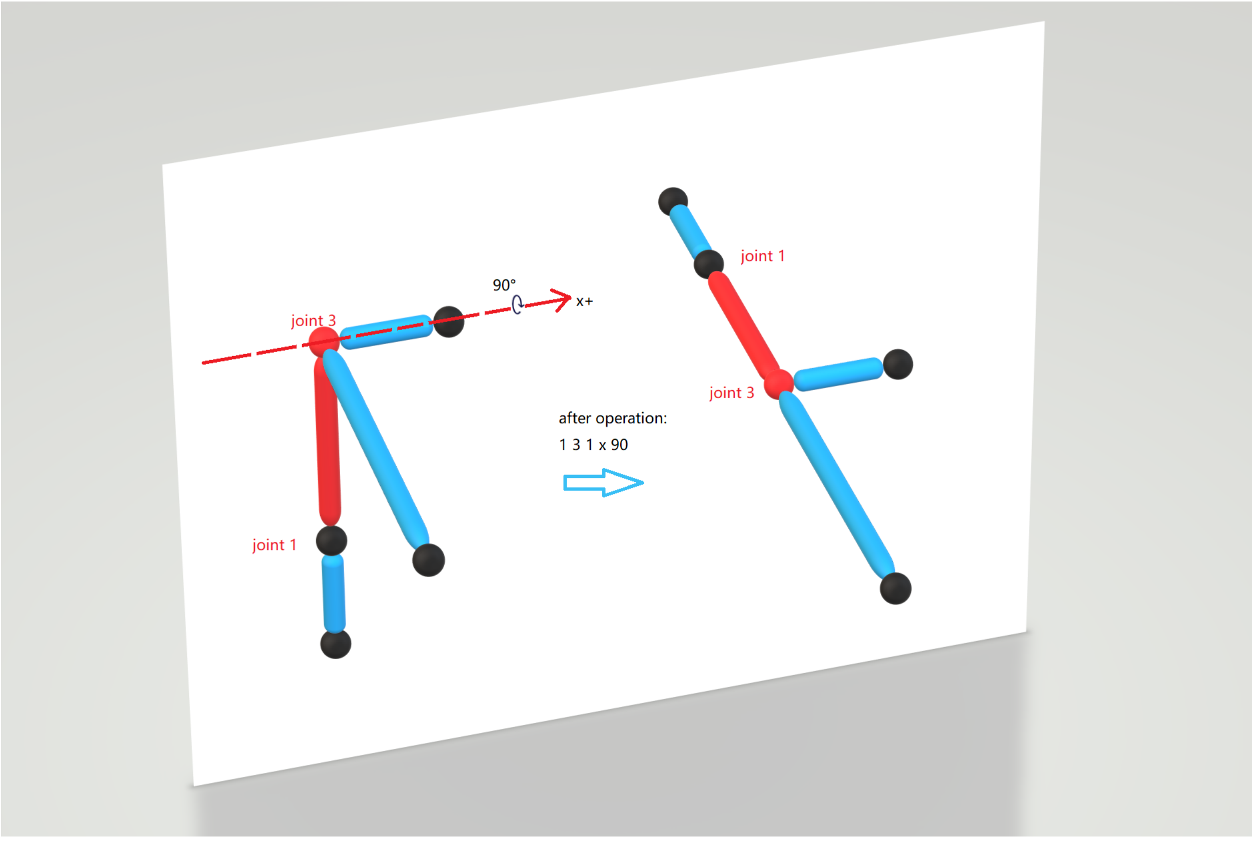
The first type is described as follows:

1 u v $axis$ $degree$

Where u and v ($1 \le u, v \le N, u \ne v$) are two joints, $axis$ is one of "x", "y" or "z" and $degree$ is an integer either 90, 180 or 270. It means we rotate the capsule connecting joint u and joint v $degree$ degrees following **right-hand rule**, using the line which points to the **positive** direction of $axis$ through joint u as pivot. It is guaranteed that joint u and joint v are connected by a capsule. See the following pictures for more information about rotation.



Rotation operation: `1 1 3 z 270`



Rotation operation: `1 3 1 x 90`

The second type is described as follows:

$2\ u\ v$

This means you should tell the Euclidean distance between joint u and joint v ($1 \le u, v \le N$).

Again, you do not need to worry about any collisions or overlapping of capsules. Assume they can "pass through" each other safely.

Output

For each second type operations output one line contains the distance. Your answer should have an absolute or relative error less than 10^{-9}. Namely, if your answer is a and the jury's answer is b, then your answer is accepted if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-9}$.

Example

standard input	standard output
5 10 2 1 1 y+ 1 3 2 x- 3 4 3 z+ 5 3 1 y- 2 2 3 2 2 5 1 1 3 z 270 2 2 3 2 2 5 1 3 1 x 90 2 4 2 2 1 4 2 5 5 2 5 2	2.236067977499790 2.828427124746190 3.000000000000000 3.162277660168379 6.000000000000000 5.000000000000000 0.000000000000000 3.162277660168379

Note

Please read the online statement for the color figures.

Problem B. Monitored Area

Input file:	`standard input`
Output file:	`standard output`
Time limit:	10 seconds
Memory limit:	512 megabytes

Curator George owns a gallery full of valuable artworks. To monitor and protect the artworks in this gallery, there are many CCTV cameras installed.

The gallery can be viewed as a simple polygon. And the cameras can be viewed as points inside the polygon.

We say that one point in the gallery is monitored if there exists a camera such that the segment between the camera and the point is inside the gallery.

Now George asks for your help to calculate the area of the monitored points.

Input

The first line contains an integer n ($3 \le n \le 50$) — the number of vertices of the polygon.

Then n lines follow. The i-th line contains two integer x, y ($|x|, |y| \le 100$) — the coordinates of the i-th polygon vertex. The vertices are given in clockwise order, and no three consecutive vertices are collinear.

The next line contains an integer m ($1 \le m \le 50$) — the number of cameras.

Then m lines follow. The i-th line contains two integer x, y ($|x|, |y| \le 100$) — the coordinate of the i-th camera.

It is guaranteed that the cameras are strictly inside the gallery.

Output

Output the area of the monitored points. Your answer should have an absolute or relative error less than 10^{-6}. Namely, if your answer is a and the jury's answer is b, then your answer is accepted if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-6}$.

Example

standard input	standard output
8 0 0 0 5 5 5 5 10 10 10 10 5 50 5 50 0 1 8 7	79.166666667

Problem C. Optimal Strategy

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Ena and Mizuki are playing a game.

There are n items in front of them, numbered from 1 to n. The value of the i-th item is a_i. Ena and Mizuki take turns to move, while Ena moves first. In a move, the player chooses an item that has not been taken and takes it away. The game ends when all items are taken away. The goal of either player is to maximize the sum of values of items they have taken away.

Given that both players move optimally, how many possible game processes are there? Since the number may be too large, you should output it modulo 998 244 353.

Two processes are considered different if there exists some integer i $(1 \le i \le n)$ such that the indices of items taken away in the i-th move are different.

Input

The first line contains an integer n $(1 \le n \le 10^6)$.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le n)$.

Output

Output the answer.

Examples

standard input	standard output
3 1 2 2	4
6 1 3 2 2 3 1	120
12 1 1 4 5 1 4 1 9 1 9 8 10	28800

Note

In the first example, there are four possible processes:

- $[1, 2, 3]$.
- $[1, 3, 2]$.
- $[2, 3, 1]$.
- $[3, 2, 1]$.

Here $[a, b, c]$ means that in the first move Ena takes away the a-th item, in the second move Mizuki takes away the b-th item, and in the final move Ena takes away the c-th item.

Note that $[2, 1, 3]$ is not a possible process, since the second move is not optimal for Mizuki.

Problem D. Arithmetic Sequence

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

Alice received a sequence of n integers as her birthday gift. As she likes arithmetic sequences, she wants to turn her gift into an arithmetic one. In an arithmetic sequence, the difference between one term and the next is a constant.

She can use her magical power and cast spells on a sequence. And she can cast two types of spells. The first type is the "increment spell": When she uses this spell, she can choose a number in this sequence and add this number by one. The other type, as you have guessed, is the "decrement spell": She can choose a number in this sequence and subtract this number by one. Casting either type of spell costs her 1 Mana (the unit of magical power).

Now she wonders about the minimum Manas she would use to make her gift arithmetic. Alice is not proficient in math skills, so she asks for your help.

Input

The first line consists of an integer n ($1 \leq n \leq 2 \times 10^5$) — the length of the sequence.

The second line consists of n integers a_i ($0 \leq |a_i| \leq 10^{13}$ and $1 \leq i \leq n$) separated by space — the initial sequence.

Output

Output only one integer — the minimum Manas she would use to make the sequence arithmetic.

Example

standard input	standard output
5 2 4 7 9 9	3

Note

The best way is $(2, 4, 7, 9, 9) \to (2, 4, 6, 9, 9) \to (2, 4, 6, 8, 9) \to (2, 4, 6, 8, 10)$, which cost Alice 3 Manas.

Problem E. Insidemen

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

There are N wizards arranged clockwise on the circular magic circle O, numbered from 1 to N according to their positions, and M magic links are drawn between their positions.

If two magic links (i, j) and (p, q) intersect and the intersection is located strictly inside the circle, then $w = (i + j)(p + q)$ gold coins are generated at the intersection. If one of the wizards among i, j, p, or q does not cast a spell, they cannot generate gold coins.

There are 2 insidemen who cannot cast spells among the N wizards. In the best case, how many gold coins can they get? (It is sure that the more coins, the better.)

Input

The first line contains two integers N and M.

The following M lines have two integers a and b in each line, indicating the magic link between wizard a and b.

It is guaranteed that $N \le 10^3$, $M \le 10^5$, and there are no repeated links.

Output

A single integer — the maximum possible number of gold coins generated.

Examples

standard input	standard output
6 7 6 4 4 1 4 5 2 1 1 5 5 2 2 3	70
10 30 4 5 8 4 10 3 8 3 6 5 7 9 8 5 2 7 9 10 3 4 1 10 1 9 6 3 5 7 10 8 5 9 5 10 1 6 2 1 1 5 2 5 3 1 9 3 3 5 7 1 6 7 10 7 6 8 6 10 9 6	8336

Problem F. Neural Network Counting

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	512 megabytes

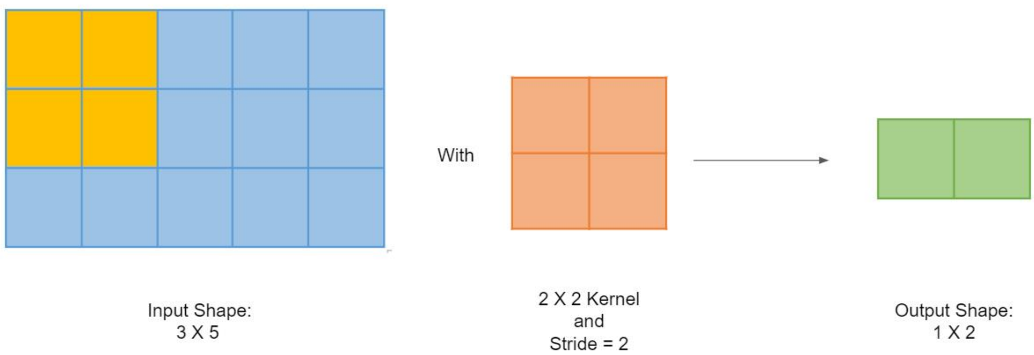
An artificial neural network could be modelled as a directed acyclic graph. The edges in the graph represent bundles of artificial neural signals, where a bundle of neural signals can be represented by a tenor in mathematics (or informally, a multi-dimensional array). The nodes in the graph represent the processing operations for the bundles of neural signals.

In this problem, the neural network only processes bundles of neural signals in two-dimensional tensors (i.e., matrices), and it only contains three types of processing operations: input operation, multiply operation, and convolution operation. We will explain them in detail. A processing operation consumes one or two bundles of signals and generates one bundle of signals. In our problem, the whole neural network contains one or more input operations and only allows one single bundle of output signals (neural signals not processed by any operations).

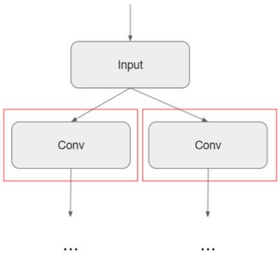
The input operation generate a tensor from the actual input stream of the neural network (which is a sequence of tensors). The actual input data are not modelled as an edge in the graph; instead, the node for the input operation has a zero in-degree and has one out-edge to model or replicate the bundle of input signals as a tensor. The input operation will produce the tensor one by one from the input stream.

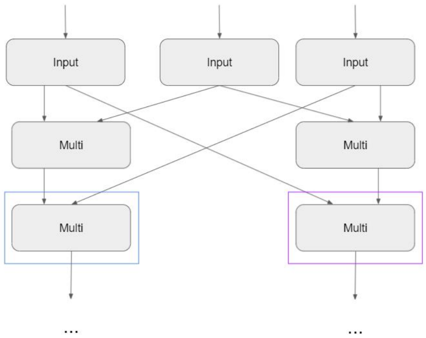
The multiply operation takes two tensors as its inputs, and then it multiplies these two tensors and generates a product tensor. In our problem, all tensors are matrices, so the sizes of the two input matrices must be legal for the matrix multiplication. Note that the order of inputs matters, unless both input tensors are virtually the same one; in other words, A multiplies B is likely different from B multiplies A, if A and B are not the same.

The convolution operation takes one tensor as its input and contains some attributes such as kernel size and stride. To simplify the problem, here we only allow the convolution operation with a 2×2 kernel size, a stride of 2, and no padding (do not add 0 outside). What we concern about is the tensor shape: if the input tensor's shape is (x, y) (the size of the first dimension is x and the size of the second dimension is y), then its output tensor's shape is $(\lfloor \frac{x}{2} \rfloor, \lfloor \frac{y}{2} \rfloor)$. Note that we do not allow the tensor with empty size, so each dimension of the output tensor should be greater than 0.



Besides, in this problem, we do not allow the neural network contains repeated nodes. Two nodes are called repeated only if their input(s) are actually the same node(s). For example, the left neural network contains repeated nodes (the nodes in red box). Note, as another an example in right neural network, the node in blue box and the node in purple box are different, because their inputs are different nodes.





We call the input format of a neural network the shape of its input tensor(s). For example, if a neural network has two input tensor(s), the shape of the first tensor is (x_1, y_1), and the shape of the second tensor is (x_2, y_2), then we call the input format of this neural network as $(x_1, y_1), (x_2, y_2)$. We call two neural networks are different if they have different input formats or their graphs models are different.

For a given number N and D, assuming the upper bound of tensor's dimension size is 2^D-1, we would like to know the number of different neural networks with N nodes. As this number may be large, output its remainder when divided by 1000000007.

Input

The first line of the input contains one integer T $(1 \le T \le 10)$ — the number of test cases. Then T test cases follow. Each test case contains one line with two integers N and D ($N \le 7$ and $D \le 50$).

Output

For each test case, print one line as its corresponding answer.

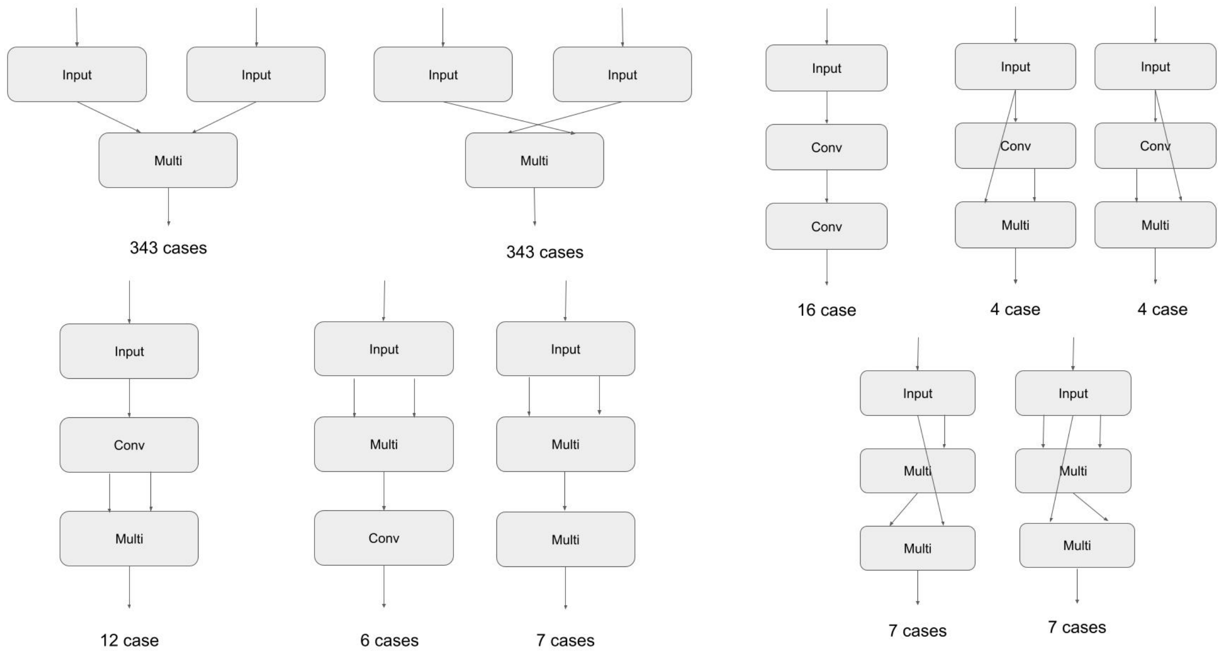
Example

standard input	standard output
3 2 2 3 3 4 3	7 749 2391

Note

For the first test case, it contains the following different neural networks: an input node connects with a multiply node with input tensor's shape as 1×1, or 2×2, or 3×3; an input node connects with a convolution node with input tensor's shape as 2×2, or 2×3, or 3×2, or 3×3.

For the second test case, it contains the following different neural networks, the number below each network represent the number of input formats on that graph.


Input
Input
Multi
343 cases
Input
Input
Multi
343 cases
Input
Conv
Conv
16 case
Input
Conv
Multi
4 case
Input
Conv
Multi
4 case
Input
Conv
Multi
12 case
Input
Multi
Conv
6 cases
Input
Multi
Multi
7 cases
Input
Multi
Multi
7 cases
Input
Multi
Multi
7 cases

Problem G. Happy Alice

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3.5 seconds
Memory limit:	512 megabytes

Alice is an excellent doll maker with lots of dolls. She has n dolls, labelled from 1 to n. The color of the dolls is either red or black. Each doll also has a distinct value.

Every day, Alice chooses two dolls from the label range $[l, r]$ to perform a play. If she can find two dolls with labels i and j, such that 1) both doll i and doll j have the same color with their values $v_i < v_j$, and 2) there exists not a doll $k \in [l, r]$ in a different color with its value $v_i < v_k < v_j$, then Alice gets a score of $v_j - v_i$ for the play. Or she can do nothing and get a score of zero for the play.

Now Alice needs to maximize the score of the plays each day. Could you help her to calculate the maximum scores?

Input

The first line contains an integer n ($1 \le n \le 10^4$) — the number of dolls.

The next line contains n integers, the i-th of which is c_i ($c_i \in \{0, 1\}$) — the polar (1 for red and 0 for black) of doll i.

And the next line contains n integers, the i-th of which is v_i ($1 \le v_i \le 10^9$ and all v_i's are distinct) — the value of doll i.

Then the next line contains an integer m ($1 \le m \le 10^6$) — the number of days.

For the next m line, each line contains two numbers l_i and r_i ($1 \le l_i \le r_i \le n$) — the label range $[l_i, r_i]$ of the candidate dolls on day i.

Output

The output contains m lines with an integer in each line.

The integer in the i-th line is the maximum score Alice could get on day i.

Specifically, if Alice cannot find such dolls i and j, you should output 0.

Example

standard input	standard output
5 1 0 0 0 1 4 5 3 1 2 5 1 3 2 4 2 5 1 5 1 1	0 4 2 0 0

Note

Please use fast input and output.

Problem H. Game Coin

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	512 megabytes

Bob recently becomes obsessed with a new game called Glory of Kings.

He has a plan to use game coins to buy hero skins. On the ith day, Bob needs a_i game coins to buy hero skins. There are two ways to get coins:

1. Each game coin can be purchased directly for t yuan.

2. Bob can buy a coin card to get some free temporary coins each day for a valid period. For example, after purchasing a coin card on day i with w free temporary coins each day and a valid period of d days, Bob can get w temporary coins each day from day i to day $i+d-1$. Note that the temporary coin means that if Bob gets the temporary coins on day x, he can only use it on day x, and the temporary coins will expire on day $x+1$.

There are n different types of coin cards in the Glory of Kings. A coin card of ith type with w_i free temporary coins each day and a valid period of d_i days costs c_i yuan.

Each card can be purchased unlimited times on any day. But any new coin card will override the previous one even if it is still valid. For example, if Bob doesn't have any coin card at the begining of day x and he buys one coin card i and one coin card j **in turn** on day x , he will have w_i+w_j temporary coins in day x, and he will only have one coin card j at the end of day x (Card i will be overridden by Card j).

Bob wants to know the minimum money he has to pay to achieve his plan, provided that he takes the best strategy.

Input

The first line of input contains three integers m, n, t ($1 \le m \le 10^5$, $1 \le n \le 400$, and $1 \le t \le 10^9$) — the number of days, the number of coin card types, and the price of a game coin.

Then follow line contains m integers $a_1, a_2, ..., a_m$ ($0 \le a_i \le 5\times10^5$ and $\sum_{i=1}^{m} a_i \le 5\times10^5$) — the numbers of coins that Bob needs on the m individual days.

The last n lines describe the n types of coin cards. The i-th line contains three integers c_i, w_i, d_i ($1 \le c_i, w_i, d_i \le 10^9$) — the price, the number of free temporary coins each day, and the valid period of coin card of type i.

Output

Print one integer in a line — the minimum money Bob has to pay if he adopts the best strategy.

Examples

standard input	standard output
3 2 9 2 7 4 10 2 3 20 4 3	39
3 2 8 10 23 10 20 10 3 10 2 2	58

Problem I. Permutation Pair

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

PTY got a positive integer N and an array A with indices from 1 to N. He needs to count the total number of **ordered permutation pair** (P, Q) (i.e., an ordered pair (P, Q), where both P and Q are permutations of $\{1, 2, \cdots, N\}$) that satisfies the following conditions:

1. M is an $N \times N$ matrix where $M_{i,j} = \begin{cases} 1 & \exists k \in \{1, 2, \cdots, N\}, i = P_k \wedge j = Q_k \\ 0 & \text{otherwise} \end{cases}$.

2. For every $M_{i,j} = 1$, there is **NO** $(A_j + 1) \times (A_j + 1)$ submatrix of M (i.e., a subregion of M in consecutive $(A_j + 1)$ rows and $(A_j + 1)$ columns) that **covers** $M_{i,j}$ and is an identity matrix or horizontal mirror of the identity matrix.

Because of the large answer, you just need to tell PTY the answer mod 998244353. Can you help him?

Input

Two lines. The first line contains an integer N ($1 \le N \le 5000$). The second line contains N positive numbers A_j ($1 \le A_j \le N$) for j in $1, 2, .., N$.

Output

A single line with the answer mod 998244353.

Examples

standard input	standard output
3 1 2 3	12
4 4 2 3 2	432
5 1 4 4 4 3	8400

Problem J. Determinant

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Alice uses the excellent property of the matrix A^TA to find the determinant of the matrix A^TA. Recall that the determinant of A, denoted by $\det(A)$, satisfies $\det(A^TA) = \det(A)^2$. Alice uses this property to find the absolute value $|\det(A)|$. But unfortunately, when $|\det(A)| \neq 0$, this method does not work out whether $\det(A)$ is positive or negative.

Now that you know the matrix A and the absolute value of $\det(A)$, determine whether it is positive or negative.

Input

An integer T ($1 \leq T \leq 100$) in the first line represents the number of test cases.

For each test case, the first line has an integer n ($1 \leq n \leq 100$), where the size of matrix A is $n \times n$.

The second line has a large number $|\det(A)|$. And it can be proved that $|\det(A)|$ has no more than 10^4 bits under the conditions in this problem.

The third to the $(n+2)$-th lines, each with n numbers, describe this matrix. It is ensured that the absolute value of each number does not exceed 10^9.

Output

For each test case, output a single line with a character "+" or "-", indicating whether the determinant is positive or negative.

Example

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

Problem K. Search For Mafuyu

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Mafuyu has hidden in Sekai, and Kanade is searching for her.

In Sekai, there is nothing but a lot of rooms. There are n rooms in Sekai, numbered from 1 to n. Besides, $n-1$ pairs of rooms are directly connected by corridors, such that it is possible to move from one room to any other one, using one or more corridors. In other words, rooms in Sekai form a tree.

Kanade is at room 1, and she knows that Mafuyu may hide in any room except room 1, with uniform probability. In one second, Kanade can move to a room adjacent to the room she is currently in. Once Kanade is in the same room with Mafuyu, she immediately finds her. What is the minimum expected time for Kanade to find Mafuyu, if Kanade is taking the optimal strategy?

Input

The first line contains an integer t ($1 \le t \le 1\,000$) — the number of test cases.

The first line in each test case contains an integer n ($2 \le n \le 100$) — the number of rooms.

Each of the following $n-1$ lines contains two integers a_i, b_i ($1 \le a_i, b_i \le n$) — the rooms connected by the i-th corridor. It is guaranteed that it is possible to move from one room to any other one, using one or more corridors.

Output

Output t real numbers. For each test case, output the minimum expected time. Your answer is considered correct if the absolute or relative error is less than 10^{-9}.

Example

standard input	standard output
4 2 1 2 5 1 2 2 3 3 4 1 5 7 1 2 1 3 2 4 2 5 3 6 3 7 10 1 2 2 3 3 4 1 5 5 6 6 7 1 8 8 9 9 10	1.0000000000 3.2500000000 5.3333333333 8.0000000000

Problem L. Strange Series

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

Given are an integer n and a polynomial $P(x) = a_0 + a_1x + \ldots + a_nx^n$, where a_0, a_1, $\ldots$, a_n are all non-negative integers.

Consider the series $S = \sum_{m=0}^{\infty} \frac{P(m)}{m!}$. One can prove that S is well-defined and we can always represent $S = pe$, where p is a non-negative integer and $e = \sum_{m=0}^{\infty} \frac{1}{m!} \approx 2.71828$ is Euler's number. You should calculate p modulo 998244353.

Input

Each test contains multiple test cases. The first line contains the number of test cases T ($1 \le T \le 100$). The description of each test case follows.

The first line of each test case contains an integer n ($0 \le n \le 10^5$).

The second line of each test case contains $n + 1$ space-separated integers a_0, a_1, $\ldots$, a_n ($0 \le a_i < 998244353$).

It is guaranteed that $n > 1000$ for at most 4 test cases.

Output

For each test case output $p \bmod 998244353$ where p is defined in the statements.

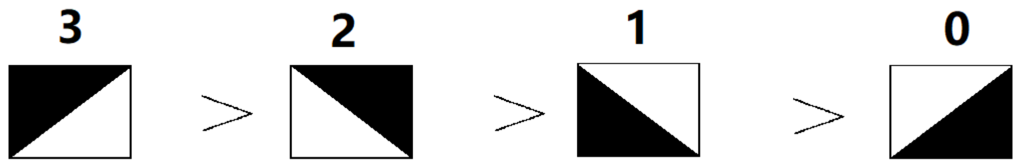
Example

standard input	standard output
2 1 1 2 5 1 1 4 5 1 4	3 258

Problem M. Coloring Rectangles

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

Berlinetta has got many white rectangles on a plane in the Cartesian coordinate system. She wants to color half of each white rectangle black along the main diagonal or the second diagonal. After coloring, she wants to have all black triangles not overlapped (but can share edges or points) with each other. Help her find a feasible coloring pattern that is lexicographically **largest** or tell her the sad truth that it is impossible. To rank the coloring patterns lexicographically, we generate a string of digits from the sequence of rectangles, and each half-black rectangle is encoded into a digit 3, 2, 1, or 0 as below



Input

The first line contains an integer n ($1 \le n \le 200$) — the number of rectangles Berlinetta have.

The following n lines give a sequence of n rectangles. Each line contains 4 integers x_1, y_1, x_2, y_2 ($x_1 < x_2$, $y_1 < y_1$, and $|x_1|, |y_1|, |x_2|, |y_2| \le 10^6$) — a rectangle $[x_1, x_2] \times [y_1, y_2]$ in the sequence to be colored.

Output

If there is a solution, output a sequence of integers, each of which shows how we color the i-th rectangle, separated by space. Otherwise, just print "no solution".

Example

standard input	standard output
2 1 1 4 4 3 2 8 8	3 2