

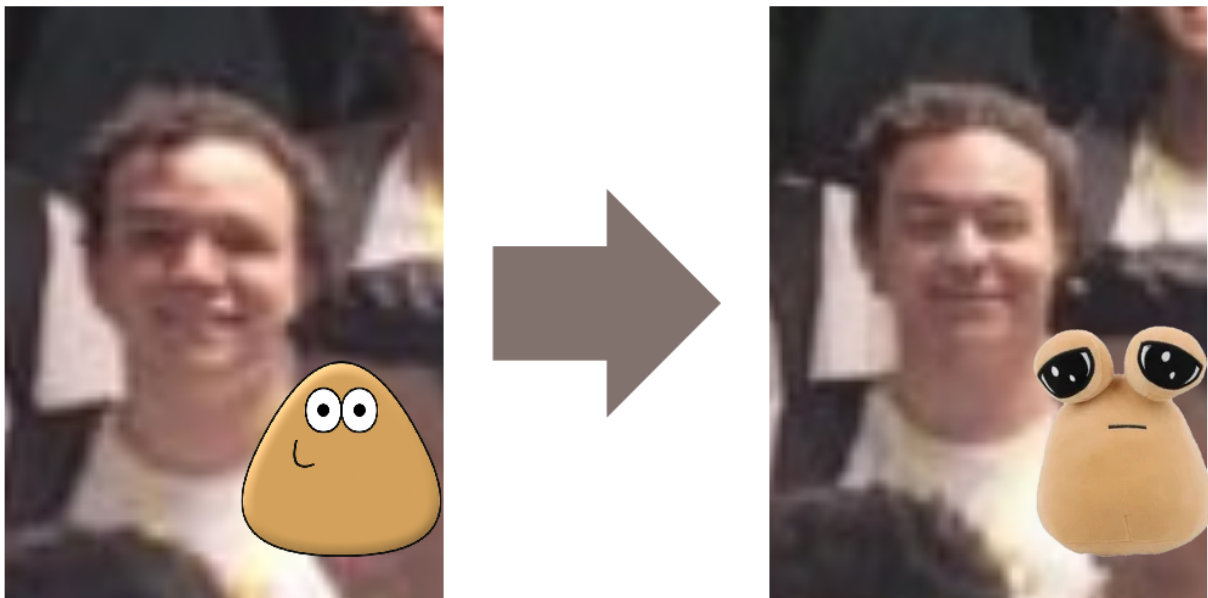
Problem A. Amusing Enhancements

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

Leon has just finished his talk and now wants to take a photo of the participants for his social media. The photo consists of a row of people arranged in a horizontal line. Ryei was chosen to assemble a photo with a total fun level of exactly D , where each person added to the photo increases the fun level by 1.

Ryei initially planned to just line up D normal people to reach the required fun level, but Leon wanted more variety. He decided that, when assembling the photo, any person added can be *enhanced* using AI, making them contribute 2 to the total fun level instead of 1.

Leon is now interested in the number of different *enhancement* configurations for photos that have fun exactly D with any number of people, considering that each person is indistinguishable; that is, the number of possible orders of normal and *enhanced* participants that result in a picture with fun level D . Since this number can be large, Leon wants it modulo 2026, because he's a modern entity.



Leon's AI enhancement

Since Ryei can't stop laughing whenever he sees an *enhanced* photo, he needs your help to write a program that counts what Leon wants.

Input

The input consists of a single integer D ($1 \leq D \leq 10^6$), the total fun level required by Leon.

Output

Your program should print a single integer on one line: the number of *enhancement* configurations that produce photos with fun exactly D , modulo 2026.

Examples

standard input	standard output
1	1
4	5
2026	1

Note

For the Second sample ($D = 4$), there are 5 possible configurations:

- $1 + 1 + 1 + 1$
- $2 + 1 + 1$
- $1 + 2 + 1$
- $1 + 1 + 2$
- $2 + 2$

Problem B. Balatro

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

Drex needed a break from implementing data structures, so he decided to relax with a few rounds of Balatro. He's on a big run. To win the game, he must defeat a powerful boss that can fight back and deal damage to him. Both Drex and the boss have hands consisting of n integers. Drex is allowed to rearrange the order of the integers in his hand before the fight begins.

The total damage Drex takes is calculated as: $\sum_{i=1}^n |d_i - b_i|$, where d_i represents the i -th integer in Drex's hand (after rearranging) and b_i represents the i -th integer in the boss's hand.

Drex wants to rearrange his hand in a way that minimizes the total damage he receives.

Input

The first line contains an integer n ($1 \leq n \leq 2 \cdot 10^5$).

The second line contains n integers $d_1, d_2, \dots, d_n$ ($-10^9 \leq d_i \leq 10^9$), Drex's hand.

The third line contains n integers $b_1, b_2, \dots, b_n$ ($-10^9 \leq b_i \leq 10^9$), the boss's hand.

Output

Print a single integer, the minimum total damage Drex will take from the boss.

Example

standard input	standard output
3 1 2 3 3 1 2	0

Note

If Drex rearranges his hand to $[3, 1, 2]$, he will take $|3 - 3| + |1 - 1| + |2 - 2| = 0$ damage.

Problem C. Coach Calculations

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

An era has come to an end. With the closing of Leo's legendary legacy, the mission of assembling the MaratonaCIn teams for the Subregional fell on the shoulders of the new coach, Palão.

But peace in the lab didn't last long. Eol, Leo's eternal arch-nemesis, felt that life was too quiet without his old rival and decided to pick a fight with the successor. In the dead of night, he broke into the selection servers and sabotaged the evaluation system.

Palão's system maintained, for each competitor, a metric represented by a polynomial $R(x)$. To disguise it, Eol split the logic into two separate polynomials, $P(x)$ and $Q(x)$, so that the original metric can only be recovered by computing the composition $R(x) = P(Q(x))$. In other words, if $P(x) = p_0 + p_1x + p_2x^2 + \dots + p_nx^n$, then

$$R(x) = P(Q(x)) = p_0 + p_1 Q(x) + p_2 Q(x)^2 + \dots + p_n Q(x)^n$$

Eol knew that computing this composition naively would produce astronomical coefficients, frying the servers. To keep his sabotage working without blowing everything up (after all, he wants to have fun, not ruin the party), he forced all operations to be performed modulo a specific prime. And, being the arrogant villain he is, he didn't pick just any prime: he configured the system to use exactly his own WhatsApp number, 998244353.

Help Palão save the selection. Given the polynomials $P(x)$ and $Q(x)$, compute the coefficients of $R(x) = P(Q(x)) \pmod{998244353}$

Input

The first line contains two integers n and m ($1 \leq n, m \leq 100$): the degrees of P and Q , respectively.

The second line contains $n + 1$ integers $p_0, p_1, \dots, p_n$ ($0 \leq p_i < 998244353$): the coefficients of $P(x) = p_0 + p_1x + \dots + p_nx^n$.

The third line contains $m + 1$ integers $q_0, q_1, \dots, q_m$ ($0 \leq q_i < 998244353$): the coefficients of $Q(x) = q_0 + q_1x + \dots + q_mx^m$.

It is guaranteed that $p_n \neq 0$ and $q_m \neq 0$ (i.e., the polynomials have exactly the stated degrees).

Output

Print $n \cdot m + 1$ integers: the coefficients $r_0, r_1, \dots, r_{n \cdot m}$ of $R(x) = P(Q(x)) \pmod{998244353}$, separated by spaces.

Examples

standard input	standard output
2 1 1 2 1 1 1	4 4 1
1 2 0 1 1 0 1	1 0 1

Note

In the first sample, $P(x) = 1 + 2x + x^2$ and $Q(x) = 1 + x$

The composition $R(x) = P(Q(x))$ is computed by substituting x with $(1 + x)$:

$$R(x) = 1 + 2(1 + x) + (1 + x)^2 = 1 + 2 + 2x + 1 + 2x + x^2 = 4 + 4x + x^2$$

The resulting coefficients are 4 4 1.

In the second sample, $P(x) = x$ and $Q(x) = 1 + x^2$.

$$R(x) = P(Q(x)) = Q(x) = 1 + x^2$$

The resulting coefficients are 1 0 1.

Problem D. Digits Duel

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

GabMei and TFG are two legends of MaratonaCIn, both World Finalists and now coaches of the next generation. Somewhere along the way, they picked up the nicknames “mastermei” and “the greatest” — nobody remembers who coined them, but the names stuck.

To settle who truly deserves the grander title, they devised a game. At the start of a match, there are N numbers on the table, corresponding to the values $2, 3, 4, \dots, N + 1$. Players take turns, and on each turn, the current player must choose a non-empty subset of the remaining numbers and remove it. A removal is only valid if the GCD of all chosen numbers is strictly greater than 1. If a player cannot make a valid move, they lose.

Whenever a subset of X numbers is removed, the remaining numbers are relabeled to be the smallest consecutive values starting from 2. That is, if there were Y numbers before the move, the numbers after the removal become $2, 3, 4, \dots, Y + 1 - X$.

Mastermei always goes first. Assuming both players play optimally, determine who wins.

Input

The first line contains a single integer T ($1 \leq T \leq 100$), the number of test cases.

Each test case consists of a single line containing one integer N ($1 \leq N \leq 5 \cdot 10^5$).

Output

For each test case, print a single line containing “mastermei” (without quotes) if the first player wins, or “the greatest” (without quotes) otherwise.

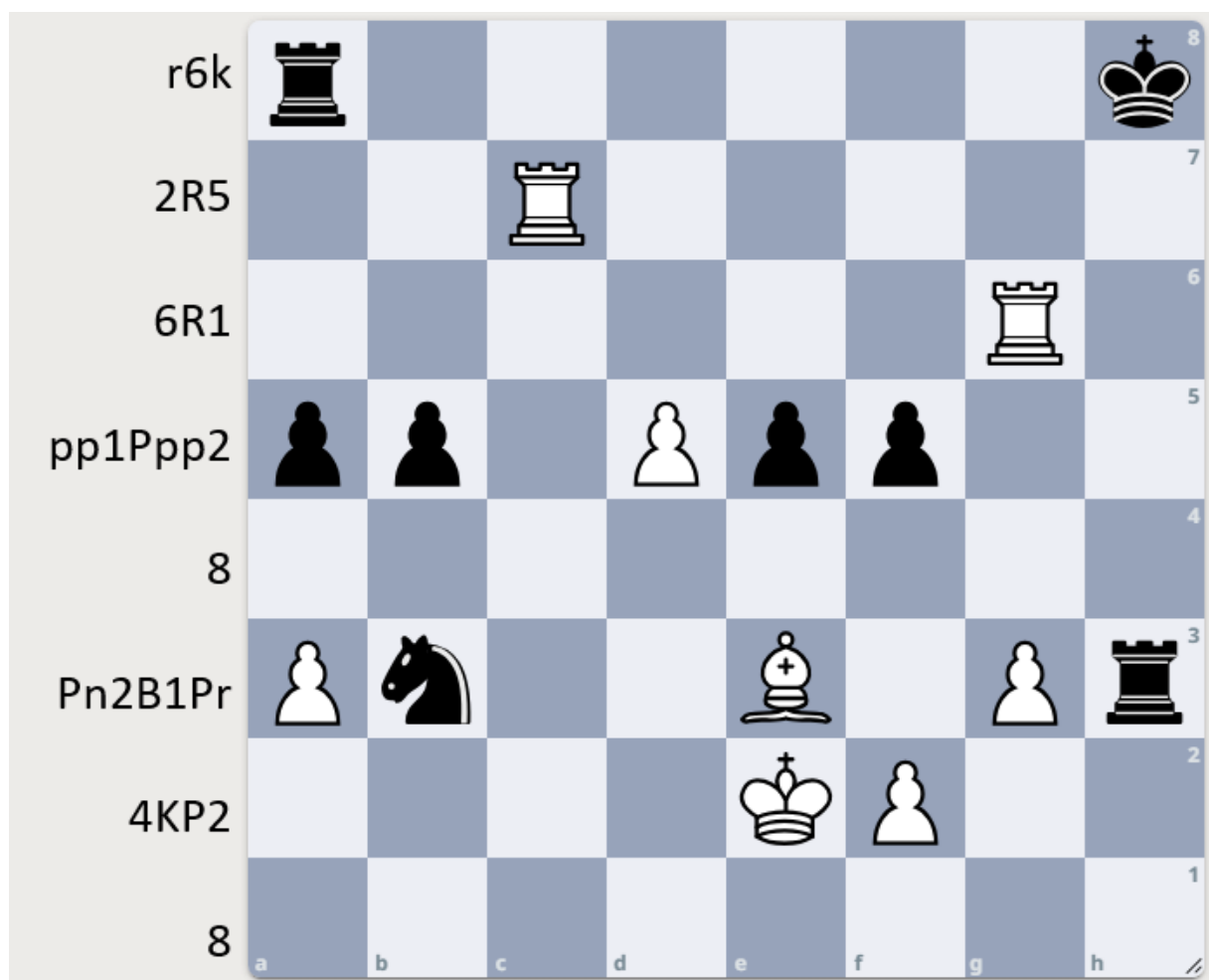
Examples

standard input	standard output
6	mastermei
1	the greatest
2	mastermei
3	mastermei
4	mastermei
5	the greatest
6	
6	mastermei
13	the greatest
62	mastermei
1000	mastermei
123456	the greatest
131070	the greatest
262142	

Problem E. Evaluation

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

Ftg, after losing several chess matches against Yrei, decides to develop a chess engine to assist him in his games. To do so, the engine must be able to receive a board position and indicate the best move to be played. After doing some research, Ftg discovered that the best way to share chess positions is by using the FEN (Forsyth-Edwards Notation) format, which works as follows: FEN describes the board from the 8th to the 1st row, using uppercase letters for white pieces and lowercase for black (e.g., P = White Pawn, n = Black Knight). Consecutive empty squares are represented by numbers (1 to 8), and each row is separated by a forward slash (/). The notation flows from left to right (file A to H), always totaling 8 units per line.



Note that both the FEN notation and the board ranks are read from 8 to 1.

Ftg has great programming skills and could code this quickly. However, Yrei has just challenged him to a match, and he will need to focus entirely on that. Could you help Ftg create a chess position converter for the FEN format?

Input

The input consists of exactly 8 lines, each containing exactly 8 characters. These lines represent the current state of the chessboard from the 8th row to the 1st row.

An empty square is represented by a dot (.). White pieces are represented by uppercase letters (P, N, B, R, Q, K), and black pieces are represented by lowercase letters (p, n, b, r, q, k).

Output

Output a single string — the FEN (Forsyth-Edwards Notation) representation of the given board position as described in the legend.

Examples

standard input	standard output
rnbqkbnr pppppppp pppppppp RNBQKBNR	rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR
r.bk...r p..pBpNp n...n.. .p.NP..PP. ...P.... P.P.K... q.....b.	r1bk3r/p2pBpNp/n4n2/1p1NP2P/6P1/3P4/P1P1K3/q5b1
R.....k ..R.....R. pp.Ppp.. Pn..B.PrKP..	R6k/2R5/6R1/pp1Ppp2/8/Pn2B1Pr/4KP2/8

Note

The first sample shows the standard starting position of a chess game. The first two rows are printed exactly as they are. The next four rows consist entirely of empty squares, so each row is converted to the number 8. The final two rows are printed exactly as they are.

In the second sample, the first row `r.bk...r` contains two groups of empty squares: one dot between `r` and `b` becomes 1, and three dots between `k` and `r` become 3, resulting in `r1bk3r`.

The third sample is the image.

Problem F. Fractions of a Stick

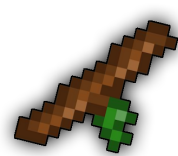
Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Toni was walking through the wild CIn parking lot, dodging beasts and trying to find where he parked his car when he stumbled upon a formidable object: a stick of exact integer length n .

Like any good problem-solver, Toni's mind immediately thought of a "gambiarra" (a makeshift solution). He needs to build a support base for his new hardware project. For the base to be stable, he needs to form a perfect triangle using pieces of this stick.

However, Toni is exhausted after a coding marathon. He decides to cut the stick at two distinct points, chosen completely at random. He only has energy for these two cuts.

Your mission is to calculate the chances of Toni's plan working, so he knows whether or not to give up and go grab a coffee.



Formally, we have a line segment of length n . The set of possible cut points consists of the internal integer coordinates $\{1, 2, \dots, n-1\}$.

Toni chooses two distinct cut points from this set, uniformly at random. This divides the original stick into three pieces with positive integer lengths a , b , and c (such that $a + b + c = n$).

What is the probability that these three resulting pieces (a, b, c) can form a non-degenerate triangle (i.e., a triangle with an area greater than zero)?

Since the answer can be an irreducible fraction $\frac{p}{q}$, you must print the value of $p \cdot q^{-1} \pmod{10^9 + 7}$, where q^{-1} is the modular inverse of q modulo $10^9 + 7$.

Input

The first line of the input contains a single integer t ($1 \leq t \leq 10^5$) — the number of test cases. Each of the following t lines contains a single integer n ($3 \leq n \leq 10^9$), representing the length of the stick Toni found in that specific scenario.

Output

For each test case, print a single integer on a new line: the probability of the makeshift solution working, modulo $10^9 + 7$.

Example

standard input	standard output
5	1
3	0
4	500000004
5	800000006
7	636363641
12	

Problem G. Gabmei, the hacker

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

Gab “gabmei” mei is known for his hacks in Codeforces contests. He once hacked three marathonists in the same contest!

gabmei	matheuslafayette 324872687 , all	D - 1709	View test	Successful hacking attempt (wa)
gabmei	miguel 324899845 , all	D - 1709	View test	Successful hacking attempt (wa)
gabmei	SamuellH12 324915997 , all	D - 1709	View test	Successful hacking attempt (rt)

Triple kill!

Now, he is planning to make a comeback. This time, he is coming to the UFPE Starters Final Try-Outs 2026 and plans to “hack” the competitors’ balloons by throwing his ultimate darts, which he bought from Quincy Jr., son of Quincy. The competitors’ computers can be represented as lattice points on a plane, and their balloons are attached to those points.

His goal is to pop all competitors balloons. However, he wants to use the minimum amount of darts possible. Gabmei can throw a dart from any position he chooses and, once he throws a dart, it continues infinitely in the direction it was thrown, popping all balloons along its path. Help him find the minimum amount of darts he needs to buy from Quincy Jr. to pop all the competitors’ balloons.

Note that a dart thrown by Gabmei acts as an infinite line across the plane, since he can throw it from anywhere.

Input

The first line contains an integer n ($1 \leq n \leq 20$), the number of balloons.

The next n lines will contain a pair of integers separated by a space, x_i and y_i , ($-10^6 \leq x_i, y_i \leq 10^6$), the coordinates of the i -th balloon.

It is guaranteed that all balloon coordinates are distinct, that is, $(x_i, y_i) \neq (x_j, y_j)$ for all $i \neq j$.

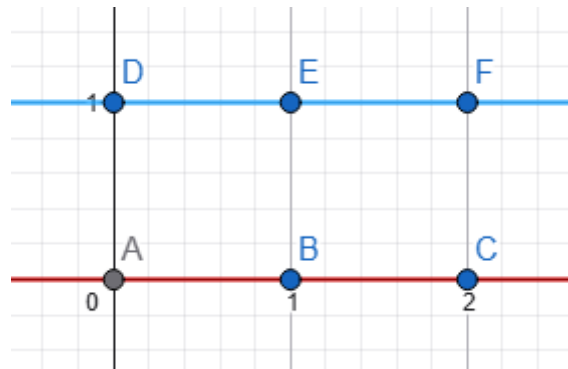
Output

Print a single line containing an integer denoting the minimum amount of darts needed to pop all balloons.

Example

standard input	standard output
6 0 0 1 0 2 0 0 1 1 1 2 1	2

Note



A solution with 2 darts to the first sample

Problem H. Hungry pou

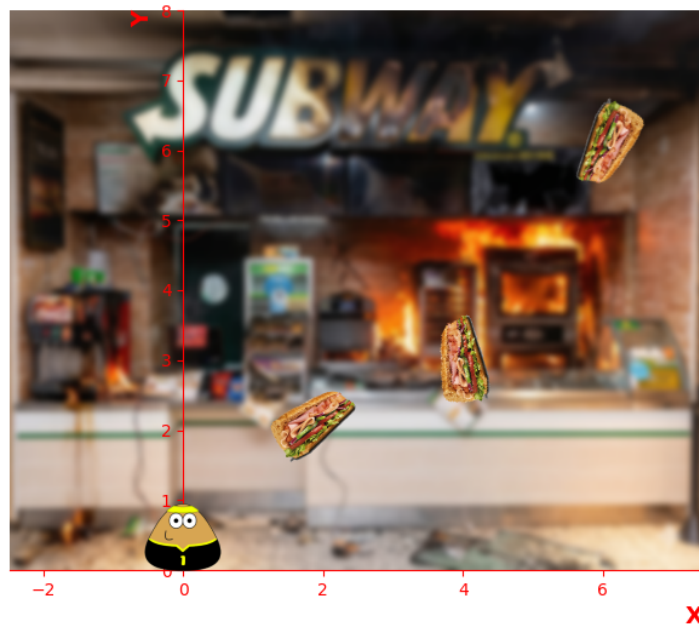
Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

Pou, Grisi, and Guilherme were starving, so they decided to have dinner at Subway. While they sat waiting for their meals and playing Brawl Stars, the kitchen exploded out of nowhere, sending food flying everywhere! Pou, being the only one quick on his feet, decided to take advantage of the situation to satisfy his hunger. He closed his game and prepared to attack!

All food items and Pou are located on a two-dimensional plane, where $y = 0$ represents the ground. At time $t = 0$, Pou is at position $(0, 0)$ and the i -th food is at position (x_i, y_i) . Each food falls at a constant vertical speed of 1 unit per second, while Pou can move left or right at a constant speed of 1 unit per second.

In other words, if in a time t a food item is at (x, y) , at time $t + 1$ it will be at $(x, y - 1)$. Simultaneously, if in a time t Pou is at $(x, 0)$, at time $t + 1$ he can be at $(x, 0)$, $(x + 1, 0)$, or $(x - 1, 0)$.

Pou can eat food item i if and only if he is at position x_i at the exact moment the food hits the ground ($y = 0$). For hygiene reasons, he will not eat any food that has already touched the ground before his arrival.



Grisi and Guilherme were so stunned by the explosion that they forgot how many food items Pou ate, but they remember the initial positions of all the food items. Knowing that Pou ate the maximum possible amount of food, your mission is to determine that number.

Input

The input consists of several lines. The first line contains n ($1 \leq n \leq 2 \cdot 10^3$), the number of food items.

The following n lines contain two integers x and y ($1 \leq x, y \leq 10^6$), the i -th line represents the initial position of food i .

It is guaranteed that no two food items occupy the same initial position, that is:
 $\forall i \forall j ((i \neq j) \implies (x_i, y_i) \neq (x_j, y_j))$.

Output

The output must contain a single integer: the maximum number of food items Pou could have eaten.

Examples

standard input	standard output
3 2 2 4 3 6 6	2
3 2 3 2 4 727 67	2

Problem I. Infinity Money Glitch

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

Tiago was playing the famous Directed Bank game with his friends Leonardo, Gabriel, and Paulo.

As everyone knows, the Directed Bank board has N positions and M possible moves, with the i -th move represented by the tuple (U_i, V_i, W_i) ; that is, if the player is on position U_i , they can move to position V_i , adding W_i to their balance. It's possible for the balance to become negative during the game.

The game always starts from position number 1, and in a round, the player can make a valid move or wait on the same spot, without losing or gaining anything.

Being an expert Directed Bank player, Tiago soon discovers a ~~bug~~ feature in the game! It's possible to accumulate infinite money! But it's not so clear how this could happen, and Leonardo and Gabriel disagreed with the statement, demanding proof. Tiago said he would demonstrate with an example of infinite moves during the game, which would also take an infinitely long time.

But since the pizza they ordered was almost arriving, there was only enough time for Tiago to play a finite number of rounds, and the problem remained open.

Knowing that you are a great programmer and that you are training for the UFPE Programming Contest, Paulo proposed you a challenge: prove or disprove Tiago!

But since infinity is a very long time, you only need to simulate $10^{998244353}$ rounds to convince the group of friends. That is, given the description of the board, state the **largest possible average value per round** that can be obtained on this board after $10^{998244353}$ rounds.

Input

The first line of input contains two integers $2 \leq N \leq 2000$, $1 \leq M \leq 2000$, representing the number of positions on the board and the number of possible valid moves.

The following M lines contain three integers U_i, V_i, W_i , with $1 \leq U_i, V_i \leq N$ and $-10^9 \leq W_i \leq 10^9$, where $U_i \neq V_i$. This represents the list of valid moves.

Note that multiple edges between the same pair of positions are allowed.

Output

Print a real number, the answer to the problem. Your solution will be considered correct if it has a relative or absolute error less than or equal to 10^{-6} .

Examples

standard input	standard output
2 2 1 2 2 2 1 -1	0.5
3 3 1 2 -5 2 3 5 3 1 -5	0

Note

In sample 1, a valid strategy is to go from square 1 to square 2, return to square 1, then go to square 2, and so on $(1, 2, 1, 2, 1, 2, \dots)$. After infinite rounds, the average points accumulated per round will be 0.5.

In sample 2, a valid strategy is to never leave the first square, thus never losing points in the balance.

Problem J. Journey for Grapes

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

In Miguel's beach house, there is a magic room: whoever solves the room's riddle is granted a wish. Noticing that his friend Walfrido was sad, Thawan decided to enter the room to cheer him up. His goal is to ask for a generous portion of grapes with margarine, Walfrido's favorite snack!

However, the room will only grant the wish if Thawan can solve the following problem:

Consider a regular polygon with N vertices, numbered from 0 to $N - 1$ in a clockwise direction. Thawan starts his journey at vertex 0. In each move, he makes a jump of size S , advancing exactly S vertices clockwise. That is, if he is currently at vertex x , the jump will take him to vertex $(x + S) \bmod N$.

Knowing that Thawan can continue jumping indefinitely, determine the maximum number of distinct vertices he will be able to visit along his path. Help Thawan secure Walfrido's snack!

Input

The only input line contains two integers, N and S ($1 \leq S \leq N \leq 10^9$), representing the number of vertices of the polygon and the size of Thawan's jump, respectively.

Output

Print a single integer representing the number of distinct vertices Thawan can reach.

Examples

<code>standard input</code>	<code>standard output</code>
2 1	2
14 8	7

Problem K. K-Places

Input file: `standard input`
Output file: `standard output`
Time limit: 1.5 seconds
Memory limit: 256 megabytes

After qualifying for the ICPC Worlds, the N MaratonaCIn members decided to celebrate with a trip together. They found K tourist hotspots, conveniently numbered from 1 to K . There is, however, a problem: maratonistas (the members of MaratonaCIn) are notoriously picky. The i -th member has m_i places they will refuse to go.

Wanting to make everyone happy, they are willing to split into groups as long as each group is a contiguous interval of the original list. In order to help Coach Palão plan the trip, answer Q queries: what places can be visited if he brings the members in a given interval $[l, r]$?

Input

The first line contains two integers N ($1 \leq N \leq 10^5$) and K ($1 \leq K \leq 60$).

Then, N lines follow. The i -th line starts with an integer m_i ($0 \leq m_i \leq K$), the amount of places the i -th person refuses to go. After that, m_i integers p_j ($1 \leq p_j \leq K$), the numbers of the places.

Then one integer Q ($1 \leq Q \leq 2 \cdot 10^5$), the number of queries.

Finally, Q lines containing two integers l, r , ($1 \leq l \leq r \leq N$).

Output

For every query, output the places the group consisting of every maratonista from l to r can go, **in ascending order**. If there isn't anywhere they can go, output -1 instead.

Example

standard input	standard output
5 6	3
1 1	4 6
3 2 4 6	1 3 4 6
2 2 5	-1
0	
2 1 3	
4	
1 3	
3 5	
3 4	
1 5	

Note

Note that the output must be in strictly ascending order or your answer might not be accepted.

Problem L. Linked Letters

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

Lua is from Vitória da Conquista, a small city in the countryside of Bahia, or so she claims. Ryei has never bought this story: the city spans over three thousand square kilometers, roughly seventeen times the area of Recife. “Small” and “countryside” are not exactly the words that come to mind.

During a particularly slow afternoon in the lab, Lua came up with a linguistic theory: two words are connected if you can transform one into the other through a chain of mutations, where each mutation swaps a word for another that shares at least one common letter with it. She was convinced this could settle the debate: if she could show that “CONQUISTA” reaches “INTERIOR” through a chain of mutations over a given dictionary, the link between her city and the countryside would be mathematically proven.

Ryei, naturally, wants to disprove every connection she finds. The argument quickly escalated; now the entire lab is involved, and everyone keeps submitting new queries to test.

Given a dictionary of N words and Q queries, determine for each query whether word a can reach word b through one or more mutations.

Input

The first line contains two integers N and Q ($2 \leq N \leq 2 \cdot 10^5$, $1 \leq Q \leq 2 \cdot 10^5$): the size of the dictionary and the number of queries.

Each of the next N lines contains a single string s_i ($1 \leq |s_i| \leq 50$) consisting of lowercase English letters. All words are distinct.

Each of the next Q lines contains two strings a and b , separated by a space: a query asking whether word a can reach word b through one or more mutations. It is guaranteed that both a and b appear in the dictionary and that $a \neq b$.

Output

For each query, print one line containing “LUA” (without quotes) if word a can be transformed into word b through a chain of one or more mutations, or “RYEI” (without quotes) otherwise.

Examples

standard input	standard output
6 4 conquista interior recife capital lua ryei conquista interior recife capital conquista recife lua ryei	LUA LUA LUA LUA
5 5 lua palao ryei grisi recife lua palao ryei grisi lua ryei palao recife grisi palao	LUA LUA RYEI RYEI RYEI
5 4 ab bc cd da xyz ab cd ab xyz da xyz ab da	LUA RYEI RYEI LUA

Note

Note that two words share a common letter if they both contain that letter anywhere in the word, regardless of position. For example, “ab” and “bc” share the letter b, so they are connected.

Problem M. Meet Me Halfway

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

Palao and his friends live in the same condominium indexed from 0 to $m - 1$. The condominium has a circular layout because the architect, Peppy, loves circles. From building i , you can go to building $(i + 1) \bmod m$ or to building $(i - 1) \bmod m$.

They want to host a LAN party to play Counter-Strike in one of the condominium buildings halls, h . Each person will travel to h from where they live using the shortest path possible along the circle.

Help Palao determine the minimum possible sum of distances traveled by him and his friends over all possible choices of the hosting building h .

Input

The first line of input contains two numbers n, m ($1 \leq n, m \leq 2 \cdot 10^5$), the number of persons that will play at the LAN party and the number of buildings in the condominium respectively.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq m - 1$) where the i -th person lives.

Output

Output the minimum possible sum of distances traveled by the players over all possible choices of the hosting building h .

Examples

standard input	standard output
3 10 0 0 9	1
5 5 3 4 3 3 0	3

Note

In the first sample, if $h = 0$, a_3 can go in one move to $(9 + 1) \bmod 10 = 0$.