Problem A. Absolute Difference

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

Both Alice and Bob have a set of real numbers, and both sets are the union of some disjoint closed intervals. They will independently pick a real number uniformly at random from their own set, and you need to calculate the expected absolute difference between the two real numbers.

More formally, given a set of real numbers $S = \bigcup [l, r]$, picking a real number x from the set S uniformly at random means that $P(x \in [l_1, r_1]) = P(x \in [l_2, r_2])$ holds for any two intervals $[l_1, r_1], [l_2, r_2] \subseteq S$ with the same length, i.e., $r_1 - l_1 = r_2 - l_2$.

Input

The first line contains two integers n and m ($1 \le n, m \le 10^5$), the number of intervals that form Alice's set and Bob's set respectively.

Each of the following $n + m$ lines contains two integers l and r ($-10^9 \le l \le r \le 10^9$), describing a closed interval $[l, r]$. The first n intervals form Alice's set and the next m intervals form Bob's set. Note that an interval $[l, r]$ with $l = r$ is a degenerate interval that contains a single real number.

It is guaranteed that the intervals that form someone's set are pairwise disjoint.

Output

Output a single real number, indicating the expected absolute difference of the two real numbers picked by Alice and Bob separately.

Your answer is acceptable if its absolute or relative error does not exceed 10^{-9}. Formally speaking, suppose that your output is a and the jury's answer is b, your output is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-9}$.

Examples

standard input	standard output

1 1
0 1
0 1
	0.333333333333333

1 1
0 1
1 1
	0.5

Note

In the first sample case, both Alice and Bob can pick any real number from $[0, 1]$, and the expected absolute difference is $\int_0^1 \int_0^1 |x - y| \, \mathrm{d}x \, \mathrm{d}y = \frac{1}{3}$.

In the second sample case, Alice can pick any real number from $[0, 1]$ while Bob can only pick 1, and therefore the expected absolute difference is $\int_0^1 |x - 1| \, \mathrm{d}x = \frac{1}{2}$.

Problem B. Binary Substrings

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

Given an integer n, you need to find a string of length n containing only 0s and 1s that maximize the number of different nonempty substrings.

Input

The only line contains a single integer n ($1 \le n \le 2 \times 10^5$), the length of the 01-string.

Output

Output a single 01-string of length n that has the maximum number of different nonempty substrings among all the 01-strings of length n. If there are multiple solutions, you may output any.

Examples

standard input	standard output
2	01
5	00110

Note

In the first sample case, there are 3 different nonempty substrings "0", "1", and "01".

In the second sample case, there are 12 different nonempty substrings "0", "1", "00", "01", "11", "10", "001", "011", "110", "0011", "0110", and "00110".

Problem C. Clamped Sequence

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

Given an integer sequence $a_1, a_2, \ldots, a_n$ and a positive integer d, you need to clamp the sequence to a range $[l, r]$ satisfying $0 \le r - l \le d$ that maximize $\sum_{i=1}^{n-1} |a_i - a_{i+1}|$, where $|x|$ is the absolute value of x.

More specifically, clamping the sequence to the range $[l, r]$ makes each element

$$a_i := \begin{cases} l, & a_i < l; \\ a_i, & l \le a_i \le r; \\ r, & a_i > r. \end{cases}$$

Both l and r are arbitrary real numbers decided by you under the given constraints. It can be shown that the maximum sum is always an integer.

Input

The first line contains two integers n ($2 \le n \le 5\,000$) and d ($1 \le d \le 10^9$), denoting the length of the given sequence and the given parameter respectively.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$), denoting the given sequence.

Output

Output a line containing a single integer, denoting the maximum sum.

Example

standard input	standard output

`8 3`
`3 1 4 1 5 9 2 6`
	`15`

Note

In the sample case, you can clamp the given sequence to the range $[1, 4]$ to make it $[3, 1, 4, 1, 4, 4, 2, 4]$, and the resulting sum is the maximum 15.

Problem D. DRX vs. T1

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

The League of Legends Worlds Championship, Worlds for short, is the culminating event in the LoL Esports season and decides the ultimate global League of Legends Champion.

Worlds 2022 has been a great tournament thus far. We have seen the legend, Faker, rise once more and the surprise of the DRX run. There have also been massive collapses from other teams like Gen.G and JDG. Regardless, we are in for a compelling Worlds 2022 Finals, which kicks off on November 6 at 08:00 (UTC+8) in San Francisco as DRX and T1 face off.

Being the clear favorite for winning the Finals, T1 has performed dominantly in most games and has found its stride coming into the tournament. They have taken down RNG and JDG, two LPL heavyweights, and are now poised to add another trophy to their cabinet.

On the other side, DRX has practically come out of nowhere. From almost not making Worlds 2022, fighting their way through the Regionals and Play-ins to the Finals. It is a Cinderella run that DRX is on, and they have been doubted at every step of the tournament. With a lot to prove and Deft reaching the Finals in his swan song, winning the trophy might be the best send-off any player can wish for.

As we countdown to a battle for the all-new trophy, you will be given the predicted result of the Finals and required to find which team wins the championship under the prediction. A team must win three out of five games to win the championship, so the predicted result of the Finals is a string of length 5 containing only Ds, Ts, and ?s. The i-th character in the string points out who wins the i-th game, where D means DRX wins the game, T means T1 wins the game, and ? means no more games are needed since some team has already won 3 games and become the champion.

Input

The only line contains a single string of length 5 containing only Ds, Ts, and ?s, indicating the predicted result of the Worlds 2022 Finals.

It is guaranteed that the predicted result is a valid BO5 result, that is, no ?s are before any Ds and Ts, and there exists a team who wins exactly three games.

Output

Output a line containing a single string, indicating the team that wins the champion under the prediction. That is, if DRX wins the champion, output "`DRX`" (without quotes), and if T1 wins, output "`T1`" (without quotes).

Examples

standard input	standard output
`TDTT?`	`T1`
`DTDD?`	`DRX`

Note

In the first sample case, T1 wins the first game, DRX wins the second game, T1 wins the third and the fourth game and wins the champion, so the fifth game is not needed anymore.

Problem E. Graph Completing

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

Given a simple connected undirected graph with n vertices and m edges, you may add as many edges (possibly zero) as you want and count the number of different ways modulo $998\,244\,353$ to make the graph biconnected while keeping it simple. Two ways of adding edges are considered different, if and only if there exists at least an edge (u, v) added in one way and not added in the other.

Note that:

- A simple graph contains no self-loops and no multiple edges.
- For any two different vertices in a connected graph, there always exists at least a path from one vertex to the other.
- For any two different vertices in a biconnected graph, there always exist two or more paths sharing no common edges from one vertex to the other.

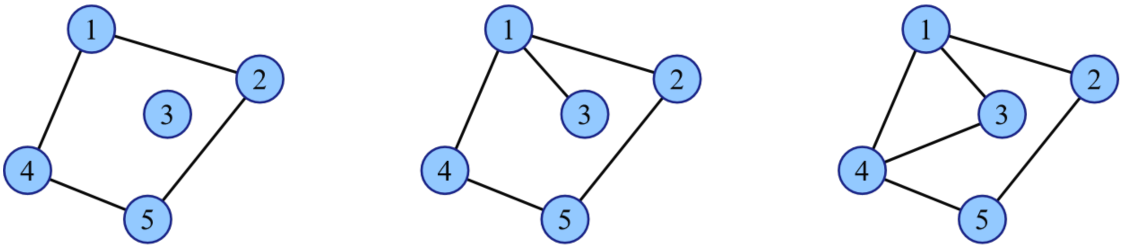


Figure: a simple graph, a connected graph, and a biconnected graph

As shown above, the graph on the left is simple but not connected because the 3rd vertex can't reach any other vertex by a path, while the graph in the middle is connected but not biconnected because it's impossible to find two paths sharing no common edges from the 3rd vertex to any other vertex.

Input

The first line contains two integers n ($2 \le n \le 5\,000$) and m ($n-1 \le m \le \min(\frac{n(n-1)}{2}, 10\,000)$), specifying the number of vertices and edges in the given graph.

Then m lines follow, the i-th of which contains two integers u and v ($1 \le u, v \le n$), indicating that the i-th edge connects the u-th and the v-th vertices.

Output

Output a line containing a single integer, indicating the number of different ways of adding edges modulo $998\,244\,353$.

Examples

standard input	standard output

3 2
1 2
2 3
	1

4 4
1 2
2 3
3 4
4 1
	4

Problem F. Half Mixed

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

Given two integers n and m, you need to construct a matrix M satisfying the following constraints:

- The number of rows and columns of M are n and m respectively.
- The matrix contains only 0s and 1s, namely, $M_{i,j} \in \{0, 1\}$ for all $1 \leq i \leq n$ and $1 \leq j \leq m$.
- The number of mixed subrectangles equals to the number of pure subrectangles. The subrectangles that contain both 0s and 1s are considered as mixed subrectangles, otherwise pure subrectangles. Note that a subrectangle is an intersection of some consecutive rows and some consecutive columns.

If multiple solutions exist, print any one of them. If there is no solution, report it.

Input

The first line contains an integer T ($1 \leq T \leq 10^5$), denoting the number of test cases.

For each test case, the only line contains two integers n and m ($1 \leq n, m \leq 10^6$, $1 \leq n \times m \leq 10^6$), denoting the number of rows and columns respectively.

It is guaranteed that the sum of $n \times m$ over all test cases does not exceed 5×10^6.

Output

For each test case, if there is no solution, print "`No`" (without quotes) in one line. If the solution exists, print "`Yes`" (without quotes) in the first line. Then print n lines, the i-th of which contains m integers $M_{i,1}, M_{i,2}, \ldots, M_{i,m}$, describing the i-th row of the matrix.

Example

`standard input`	`standard output`

`2`
`2 3`
`1 1`
	`Yes`
	`0 1 1`
	`1 1 0`
	`No`

Note

In the first sample case, the number of mixed subrectangles and pure subrectangles are both 9.

In the second sample case, the only subrectangle is the whole matrix that must be pure, so the number of mixed subrectangles and the number of pure subrectangles must be 0 and 1 respectively, which are not equal.

Problem G. Meet in the Middle

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

Byteland has n cities numbered from 1 to n, and $n-1$ bidirectional roads and $n-1$ bidirectional railways connecting them. For each pair of cities, one can always arrive one from another one through the roads only, and the same thing also holds for the railways.

Alice and Bob are planning q long journeys in Byteland. In each journey, Alice starts from the a-th city and then visits some other cities through the roads only, while Bob starts from the b-th city and then visits some other cities through the railways only. Both of them will finally reach the **same** destination city without visiting any cities more than once. You need to find the maximum total length of their journey routes among all the selections of the destination city.

Input

The first line contains two integers n $(2 \le n \le 10^5)$ and q $(1 \le q \le 5 \times 10^5)$, indicating the number of cities in Byteland and the number of journey plans.

Each of the next $n-1$ lines contains three integers u, v $(1 \le u, v \le n)$ and w $(1 \le w \le 10^9)$, indicating a road of length w connecting the u-th and v-th cities.

Each of the next $n-1$ lines contains three integers u, v $(1 \le u, v \le n)$ and w $(1 \le w \le 10^9)$, indicating a railway of length w connecting the u-th and v-th cities.

Each of the next q lines contains two integers a and b $(1 \le a, b \le n)$, indicating a journey where Alice starts from the a-th city and Bob starts from the b-th city.

Output

Output q lines, each of which contains a single integer, indicating the maximum total length of Alice's and Bob's journey routes among all the selections of the destination city.

Example

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

Problem H. P-P-Palindrome

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

Given n strings $S_1, S_2, \ldots S_n$, you need to calculate the number of different *P-P-Palindromes* given by these n strings.

A palindrome is a string that can be read the same from left to right and from right to left. For example, "`a`", "`level`", and "`otto`" are palindromes, while "`aab`" and "`icpc`" are not.

A *P-P-Palindrome* is an ordered pair of **nonempty** palindromes (P, Q) such that both P and Q are the substrings of some in $S_1, S_2, \ldots S_n$ and $P + Q$ is also a palindrome. Here $P + Q$ means concatenating P and Q in order, or more specifically, let $P = p_1p_2 \ldots p_{|P|}$ and $Q = q_1q_2 \ldots q_{|Q|}$, then $P + Q = p_1p_2 \ldots p_{|P|}q_1q_2 \ldots q_{|Q|}$, where $|S|$ is the length of string S.

Note that two *P-P-Palindromes* are considered different if and only if P differs or Q differs.

Input

The first line contains an integer n $(1 \le n \le 10^6)$, indicating the number of given strings.

Then n lines follow, the i-th of which contains a string S_i $(1 \le |S_i| \le 10^6)$ consisting of lowercase English letters only.

It is guaranteed that the total length of the given strings does not exceed 10^6.

Output

Output a line containing a single integer, indicating the number of different *P-P-Palindromes* given by the n strings.

Examples

standard input	standard output

`2`
`aaaa`
`aaa`
	`16`

`3`
`abaaa`
`abbbba`
`bbbaba`
	`28`

Problem I. Quartz Collection

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

There is a shop selling n types of quartz in Byteland. Only two pieces of each type of quartz will be on sale every day, and the second piece will be on sale **only after** the first piece has been sold.

Two wizards Alice and Bob are collecting these n types of quartz to strengthen their wand. Due to the quartz shortage, they reach an agreement that either of them buy only one piece of each type each day.

Both of them wants to minimize their own cost each day. To reflect fairness, Alice buys one piece of quartz first, and then Bob and Alice buy two pieces in turn until only one piece remains. The one who still hasn't collected all types of quartz buys the last piece.

In each of the following m days, the prices of the two pieces of one type of quartz will change **permanently**, and Alice wants to know the minimum cost for her to collect all types of quartz if both Alice and Bob take the best strategy on the first day and each of the following m days.

Input

The first line contains two integers n and m ($1 \le n, m \le 10^5$), indicating the number of types of quartz and the number of following days.

Then n lines follow, the i-th of which contains two integers a and b ($1 \le a, b \le 10^5$), indicating the prices of the first piece and the second piece of the i-th type of quartz on sale respectively.

Then m lines follow, each line contains three integers t ($1 \le t \le n$), x and y ($1 \le x, y \le 10^5$), indicating that the prices of the first piece and the second piece of the t-th type of quartz on sale will change to x and y respectively.

Output

Output $m + 1$ lines, each containing a single integer, indicating the minimum cost for Alice to collect all types of quartz on the first day and each of the following m days.

Example

standard input	standard output

4 5
2 4
5 7
1 7
2 1
4 5 2
1 6 2
4 4 3
2 1 3
3 6 6
	13
	14
	15
	14
	10
	13

Note

In the sample case, one of the best strategies on the first day is as follows:

- Alice buys the first piece of the third type of quartz.
- Bob buys the first piece of the first type and the second type.
- Alice buys the second piece of the first type and the second type.
- Bob buys the second piece of the third type and the first piece of the fourth type.
- Alice buys the second piece of the fourth type, which is also the last piece of quartz.

Problem J. Referee Without Red

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

Night falls, neon flickers, and the crowd shouts. As the grand show in the Moon Theater begins, Calatonia, a city of assorted anthropomorphic animals, feels ready for a sleepless night. Over the years since its revival, the annual show has never failed to ignite animals' passion for music, mainly due to the various genres it includes. So this time, it adds more attractive elements, such as dance!

If you inquire of some creature in Calatonia about the best dance troupe, they are most likely to answer: Icy Cold Predators Crew (ICPC). Its performance will be a highlight of tonight's show without doubt. And you may overhear how Referee — well, that's his name — led the troupe to its peak. As the director, Referee starts every day by selecting background music and gets busy with choreographing in the afternoon. Before sleep, he tends to make dazzling costumes which are of little use in such a world actually. The routine is exhausting, but he somehow enjoys it.

Referee considers himself a born perfectionist. That's true when it comes to formation design. Offstage, he arranged the whole crew into a matrix of n rows and m columns, numbering rows $1, 2, \ldots, n$ from front to back and columns $1, 2, \ldots, m$ from left to right. The species of each dancer can be labeled as a positive integer not more than 3×10^6, where there may be animals of identical species. Two arrangements are different if and only if there exists at least a position where the species of dancers in the two arrangements are different.

Unsatisfied with the initial arrangement, Referee intends to reorder the crew without breaking its matrix formation. He thinks it's a good idea to adjust by showing cards. Certainly not red cards — Referee, after all, does not serve as a referee. Instead, he prepared white cards with $1, 2, \ldots, n$ written on them respectively and black cards with $1, 2, \ldots, m$, and commanded the crew to do as follows:

- Once Referee shows the white card with k ($1 \le k \le n$), all dancers in the k-th row count off from 1 to m from left to right. Those who count an even number immediately move to the right end of the row. Lastly, align to keep the matrix formation.
- Once Referee shows the black card with k ($1 \le k \le m$), all dancers in the k-th column count off from 1 to n from front to back. Those who count an even number immediately move to the back end of the column. Lastly, align to keep the matrix formation.

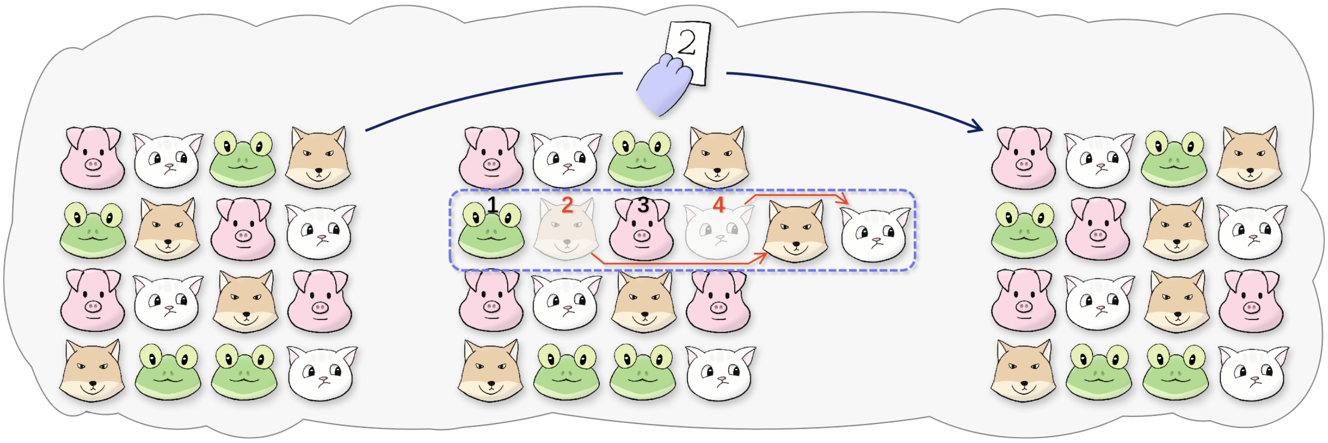


Figure: Referee shows the white card with 2

Referee can show any card any number (possibly zero) of times he wants, and your task is to calculate the number of different arrangements he may obtain, modulo 998 244 353 since it could be extremely large.

Input

The first line contains an integer T ($1 \le T \le 10^5$), denoting the number of test cases.

For each test case:

The first line contains two integers n and m ($2 \le n, m \le 10^6$), denoting the number of rows and columns respectively.

Then n lines follow, the i-th of which contains m integers $a_{i,1}, a_{i,2}, \ldots, a_{i,m}$ ($1 \le a_{i,j} \le 3 \times 10^6$), where $a_{i,j}$ denotes the species of the dancer in the i-th row and the j-th column initially.

It is guaranteed that the sum of $n \times m$ over all test cases does not exceed 3×10^6.

Output

For each test case, output a line containing a single integer, indicating the number of different arrangements modulo 998 244 353.

Example

standard input	standard output

2
4 4
1 2 3 4
3 4 1 2
1 2 4 1
4 3 3 2
3 9
1 8 1 1 8 1 1 8 1
1 8 8 8 8 8 8 8 1
1 1 1 8 8 8 1 1 1
	96
	6336

Problem K. Security at Museums

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

Recently a museum of arts in Byteland is open to the public. The museum is modeled as a polygon, and there is an art placed on each vertex of the polygon.

A group of thieves are coveting the arts in the museum and they have chosen at least two arts to steal from the museum at night when all is still. After sneaking into the museum, each of them will work on stealing one of the chosen art simultaneously.

To be on the safe side, the thieves have chosen the arts in such a manner that every two thieves can see each other, i.e., every point of the straight line segment between them lies in the interior or on the boundary of the museum, when they are working on stealing the arts.

You, the administrator of the museum, need to anticipate the thieves' actions, i.e., compute the number of different sets of the arts that might be chosen by the thieves, to ruin their plans. The answer could be extremely large, so you only need to output it modulo 998 244 353.

Input

The first line contains an integer n ($3 \le n \le 200$), specifying the number of vertices of the polygon.

Then n lines follow, each containing two integers x and y ($-10^6 \le x, y \le 10^6$) that give the coordinates (x, y) of the vertices of the polygon in counter-clockwise order.

The polygon is simple, i.e., its vertices are distinct and no two edges of the polygon intersect or touch, except that consecutive edges touch at their common vertex. In addition, no two consecutive edges are collinear.

Output

Output a line containing a single integer, indicating the number of different sets of arts that might be chosen by the thieves modulo 998 244 353.

Examples

standard input	standard output

7
0 20
40 0
40 20
70 50
50 70
30 50
0 50
	56

3
0 2022
-2022 -2022
2022 0
	4

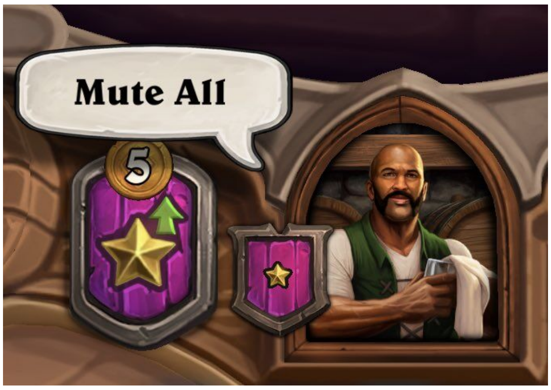
Note

In the second sample case, all the sets that contain at least two arts are counted.

Problem L. Tavern Chess

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

One day, Alice and Bob are playing an online chess game named "Tavern Chess", where each player can choose at most 7 minions to build a team.



Welcome to Tavern

Each minion has its Hit Points (HP) and Attack (ATK), and **the HP is the same as the ATK initially**. A minion with positive HP is alive and can take attacks, and it dies immediately if its HP becomes zero or lower after an attack.

The battle begins after both Alice and Bob finish building their team and lasts until all the minions from one team are dead and the other team wins. In case the last alive minions from the two teams die simultaneously, the battle ends in a tie.

Alice and Bob's teams take turns to attack, and the team that has more minions takes the first attack. In case of a tie, it will be decided by a coin toss — 50% probability for Alice taking the first attack and the remaining 50% probability for Bob taking the first attack.

When a team takes the attack, **the leftmost minion with the minimum number of taking attacks** from the team attacks one of the alive minions from the other team uniformly at random, and then the other team takes the attack.

If a minion with the HP h_1 and the ATK a_1 attacks another minion with the HP h_2 and the ATK a_2, each minion deals damages equal to its ATK to the other one, so the attacker minion will have the HP $h_1 - a_2$ and the attacked minion will have the HP $h_2 - a_1$ after the attack.

Given Alice's team and Bob's team, you need to calculate the probabilities that Alice's team wins the battle, Bob's team wins the battle, or the battle ends in a tie, respectively.

Input

The first line contains two integers n and m ($1 \le n, m \le 7$), indicating the number of minions in Alice's team and Bob's team respectively.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$), the i-th of which is the HP as well as the ATK of the i-th minions from the left from Alice's team.

The third line contains m integers $b_1, b_2, \ldots, b_m$ ($1 \le b_i \le 10^9$), the i-th of which is the HP as well as the ATK of the i-th minions from the left from Bob's team.

Output

Output three lines, each containing a single real number, indicating the probabilities that Alice's team wins the battle, Bob's team wins the battle, or the battle ends in a tie, respectively.

Your answer is acceptable if its absolute error does not exceed 10^{-9}. Formally speaking, suppose that your output is a and the jury's answer is b, your output is accepted if and only if $|a - b| \leq 10^{-9}$.

Examples

standard input	standard output

2 3
2 5
3 4 1
	0.125
	0.75
	0.125

6 6
1 1 4 5 1 4
1 1 4 5 1 4
	0.241867283950617
	0.241867283950617
	0.516265432098765

Problem M. Vulpecula

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

In the center of the Summer Triangle nestles a little fox, a constellation named Vulpecula, to be precise. It is situated in a zone with numerous brilliant stars, but it itself is relatively faint. Even the brightest star in the constellation has an apparent magnitude of merely 4.44. However, despite its dimness, Vulpecula captures the attention of many stargazers for various reasons. Say, Mu, the protagonist of our problem, regards Vulpecula as his favorite constellation because there, he can always see his own reflection.

Mu lives a stressful life and only by viewing stars can he find some relief. Every night, he would climb onto the roof, watching skyward with his telescope and identifying constellations of all shapes. While admiring the cosmic wonder, he would occasionally write down some thoughts that crossed his mind. And this night started with a strange question: "*What in the world are dreams for?*"

Thoughtfully, Mu looked up at the starry fox as the past experience came reeling through his head. Although recent years have been even rougher, his striving for dreams has never diminished all the way. He gave up money because he should not overvalue it. He gave up prestige because he could not maintain it. He gave up the place he longed to settle because he did not deserve it. He gave up everything but...

"*Dreams are like the stars, distant and sometimes insignificant, yet enough to spark up the darkest night.*" Mu jotted it down on a note and squeezed a smile, before immersing himself in the joy of stargazing again.

The Constellation Vulpecula

Now, let's get back to the problem. People have associated constellations with animals and mythological characters since ancient times. Giving free rein to their imagination, they drew straight lines between the stars on a chart and conceived the story behind every constellation. Suppose there are n stars numbered from 1 to n in Vulpecula with $n-1$ imaginary lines connecting them. Namely, for each pair of stars (s,t), there always exists a sequence of stars $u_0, u_1, \ldots, u_k$ where $u_0 = s$ and $u_k = t$, satisfying that star u_{i-1} and star u_i are linked by a line for each $i = 1, 2, \ldots, k$, and the *distance* between star s and star t is defined as the minimum k among all such sequences.

Mu is prepared to view those stars in Vulpecula, and he will focus his telescope on one of the n stars. Detailed observation, however, requires advanced technology as there will be several obstacles.

One is that some stars may be too tiny to observe. To tackle it, Mu can screw a knob to zoom in or out. Specifically, after selecting the focused star, he will adjust the focal distance to an integer d between 0 and $n-1$ and see all stars whose *distance* from the focused star is not more than d.

The other is that some stars may be too faint to observe. Luckily, Mu can use filters to regulate the rays of a star. Let a_i represent the *visibility* of star i which is initially 0. A filter denoted as (i, x) only applies to star i, and when Mu installs it, a_i will become $a_i \oplus x$, where $\oplus$ denotes the bitwise exclusive-OR operation. Mu can pick a set of filters he has got and install them in arbitrary order. He may also pick no filters or two or more identical filters.

For a pleasant stargazing experience, Mu will make the *visibilities* of the stars in sight equal and maximized. Playing with a telescope is quite easy, whereas your task is not so simple. Suppose $f(d)$ is the maximum visibility under a certain focal distance d. Can you calculate the sum of $f(d)$ over $d = 0, 1, \ldots, n-1$ modulo 2^{64} for each star if it is the focused star?

Input

The first line contains an integer n ($2 \le n \le 5 \times 10^4$), denoting the number of stars in Vulpecula.

The second line contains $n-1$ integers $p_1, p_2, \ldots, p_{n-1}$ ($1 \le p_i \le i$), indicating that there is an imaginary line between star p_i and star $i+1$ for each $i = 1, 2, \ldots, n-1$. It is guaranteed that the given $n-1$ imaginary lines connect all the stars.

In the i-th of the following n lines, an integer m_i ($m_i \ge 0$) comes first, denoting the number of filters that apply to star i. Then m_i integers $x_{i,1}, x_{i,2}, \ldots, x_{i,m_i}$ ($0 \le x_{i,j} < 2^{64}$) follow, the j-th of which denotes a filter $(i, x_{i,j})$. It is guaranteed that the sum of m_i over $i = 1, 2, \ldots, n$ does not exceed 2×10^6.

Output

Output n lines, the i-th of which contains a single integer, indicating the sum of $f(d)$ over $d = 0, 1, \ldots, n-1$ modulo 2^{64} if star i is the focused star.

Examples

standard input	standard output

2
1
2 2 3
2 1 1
	4
	2

5
1 2 2 3
3 83 75 58
4 125 124 58 16
4 39 125 71 112
3 69 66 5
4 48 73 69 6
	171
	125
	183
	142
	243

Note

In the first sample case, if star 1 is the focused star, Mu can install filter $(1, 3)$ to reach $f(0) = 3$ because star 1 is the only star in sight when $d = 0$. He can successively install filter $(1, 2)$, filter $(1, 3)$ and either filter $(2, 1)$ to reach $f(1) = 1$. Thus, the sum for star 1 is $(3 + 1) \bmod 2^{64} = 4$.