Problem A. Aho

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

All characters appearing in this and following statements are fictitious. Any resemblance to real persons, living or dead, is purely coincidental.

This is an interactive problem.

Alfred and Margaret always play some weird string-guessing games when they spend a charming evening in the coffee shop "Donald's place".

This time, the rules are the following:

- The first player writes down a string S of a predefined length N. The first player also has a string T that is initially empty. Both strings contain only **lowercase English letters**.
- The second player can not see these strings during the whole game. However, the second player is allowed to ask whether the characters on any pair of positions in any of the strings are equal. For example, a question can look like the following: "Is the second character of string S equal to the fifth character of string T?" Note that it is also allowed to compare two positions of the same string using this question.
- The game is played in M rounds. At the start of each round, the first player adds a single character to the end of string T.
- When a new character is added, the second player can ask **no more than five** questions. After that, the second player must say how many substrings of string T are equal to string S.

Margaret quickly noticed that Alfred always succeeds as the second player. She suspects there is a strategy that allows the second player to win regardless of what are the strings S and T. Could you figure it out?

Input

When your program starts, it must read two integers N and M from the standard input ($1 \le N, M \le 20\,000$).

Then M rounds of the game follow. In the i-th round, you can ask **up to five** questions in the form "`<position1> <position2>`". Here, the description of each position is either in the form "`s` x" where $1 \le x \le N$ (x-th character of the string S) or in the form "`t` y" where $1 \le y \le i$ (y-th character of the string T). The response from the jury program is "`Yes`" if the characters on these positions are equal, and "`No`" otherwise.

To finish each round, you must output the answer in the form "`$` k" where k stands for the number of occurrences of string S in the current string T. After that, the string T will be automatically expanded by some unknown character, or the game finishes if it is the last round.

Do not forget to flush the standard output after each operation. Once you already reported all the m required numbers, you should exit the program, otherwise the result of the judgement is **undefined**!

Examples

standard input	standard output
3 7 No Yes No Yes Yes Yes No No Yes Yes Yes	s 1 s 2 s 1 s 3 s 2 t 1 s 1 t 1 $ 0 s 2 t 2 $ 0 s 3 t 3 $ 1 s 2 t 4 s 1 t 4 $ 1 s 1 t 5 $ 1 s 2 t 6 $ 1 s 3 t 7 $ 2

Note

In the example above, the string S is initially "aba", and the string T is expanded sequentially by characters "a", "b", "a", "c", "a", "b", "a".

Problem B. Banach

Input file:	`banach.in`
Output file:	`banach.out`
Time limit:	3 seconds
Memory limit:	512 megabytes

Stefan, David and Felix are preparing an ACM-style programming contest. Stefan proposes the following task:

> Given N points on a plane and N movement vectors, find a one-to-one correspondence between vectors and points such that, if one moves (translates) each point by the corresponding vector, the distance between **every** pair of points will not decrease.

David managed to solve this problem pretty fast and claims it's too easy to be included in the problemset. To convince him, Felix proposed the following update: among all possible solutions, choose the one that maximizes the sum of squares of all resulting pairwise distances.

David still finds the problem to be easy, do you agree?

Input

The first line of the input contains a single integer N, the number of points and vectors ($1 \le N \le 500$).

Next N lines describe points. Each of them contains two integers px_i and py_i ($0 \le |px_i|, |py_i| \le 10\,000$).

Then follow N lines with vectors descriptions. Each vector is defined by two integers vx_i and vy_i ($0 \le |vx_i|, |vy_i| \le 10\,000$).

Output

If there exists a way to establish a one-to-one correspondence such that all pairwise distances will not decrease, print "`Yes`" on the first line of the output. On the next line, print N distinct integers from 1 to N, i-th of these numbers being the vector assigned to i-th point.

Do not forget to choose the answer with maximum possible sum of squares of all resulting pairwise distances. If there are still multiple answers, output any of them.

If there is no way to meet the desired requirement, print "`No`" on the single line of the output.

Examples

banach.in	banach.out
2 0 0 1 0 2 0 -2 0	Yes 2 1
2 2 2 2 2 -1 -1 -1 -1	Yes 1 2
3 -2 -3 3 1 -3 -3 -3 5 -2 -4 4 -1	Yes 2 3 1

Note

In the first example, there are only two possible ways to establish a one-to-one correspondence between points and vectors. In both correspondences, "`1 2`" and "`2 1`", the distance between every pair of points does not decrease. In the first case the sum of squares of all resulting pairwise distances is equal to 9, but in the second one it is equal to 25. So, the only correct answer is "`2 1`".

In the second example, there are again only two possible ways to establish a correspondence. In both cases, the distance between every pair of points does not decrease, and the sum of squares of all resulting pairwise distances is equal to 0. So, both of the answers are correct.

Problem C. Conway

Input file:	`conway.in`
Output file:	`conway.out`
Time limit:	4 seconds
Memory limit:	512 megabytes

There are M light bulbs in a room and N switches outside of it. For the purpose of this problem, N is guaranteed to be **odd**. Each bulb has its own power p_i and can only be fully on or fully off. If the bulb is on, it adds p_i units to the illumination of the room.

Some switches and bulbs are connected by cords. Toggling a single switch changes the state of all bulbs it is connected to. There are no restrictions on connections, which means that any switch can be connected to any subset of bulbs, and vice versa.

John invented a new game and invited his friends Roland and Patrick to play it. Roland likes light and tries to make the illumination as high as possible, and the Patrick's goal is strictly opposite: to reduce the illumination as low as possible. There is a value K chosen by John to determine the winner. At the end of the game, if the total power of all bulbs turned on is greater than or equal to K, then Roland is considered the winner, otherwise, the winner is Patrick. The game process looks as follows:

- Switches are numerated from 1 to N.
- Each player makes exactly $\frac{N-1}{2}$ moves, one after another. Roland makes his move first.
- When making his i-th move, Roland can toggle switches numbered $2 \cdot i - 1$ and $2 \cdot i$. These are first and second switches on his first move, third and fourth on the second move, and so on. Note that he can choose to toggle any subset of these two switches (one of them, both or none). If any bulb is connected to both switches, it changes its state once for each toggled switch.
- Rules for Patrick are absolutely the same, the only difference is in the indices of switches he can toggle. On his i-th move, he can toggle switches $2 \cdot i$ and $2 \cdot i + 1$. These are second and third for his first move, fourth and fifth for the second move, and so on.

John likes to watch his friends playing while he already knows the result. He asks you to write a program that will determine the winner for each of the T games, assuming that both Roland and Patrick play optimally.

Input

The first line of the input contains single integer T — number of test cases ($1 \le T \le 5$). Each test case describes a separate game.

The first line of each test case contains three integers N, M and K ($3 \le N \le 33$, N is odd, $1 \le M \le 32$, $0 \le K \le 2 \cdot 10^9$) — the number of switches, the number of bulbs and illumination value to determine the winner.

The second line of each test case contains M integers l_i ($1 \le l_i \le 5 \cdot 10^7$) — the power of each bulb.

The next N lines of each test case describe connections between switches and bulbs. Each of them contains M characters. If i-th switch is connected to j-th bulb, then j-th character of i-th line equals '`1`', otherwise it equals '`0`'.

Output

For each test case, print a line with the name of the winner, that is, either "`Roland`" or "`Patrick`".

Examples

conway.in	conway.out
2 3 2 10 10 10 01 00 11 3 5 1 1 2 3 4 5 01011 11000 10011	Roland Patrick

Note

In the first game, one of the optimal strategies for Roland is the following one: on his first turn, he toggles the first and the second switches. For any possible Patrick's move, there will be exactly one bulb that is turned on at the end of the game, so Roland wins anyway.

In the second game, regardless of the Ronald's move, Patrick can turn off all the bulbs.

Problem D. Dirichlet

Input file:	`dirichlet.in`
Output file:	`dirichlet.out`
Time limit:	3 seconds
Memory limit:	512 megabytes

Four friends Johann, Peter, Gustav and Lejeune are planning to build a brick wall. They already decided it should be a parallelepiped with one decimeter width and N decimeters height, but they are still not sure about its length.

There were T types of bricks available in the nearest store, and as they have no project of the wall, they simply bought c_i bricks of type i for each i from 1 to T. A brick of type i is a parallelepiped of size $1 \times 1 \times l_i$ decimeters.

Due to the stability issues, all of the bricks should be put horizontally. That means each brick will be used in only one level of the wall. To build a wall of height N, one needs to form N sets of bricks. Every set can contain arbitrary number of bricks of any type, but the total length of all bricks in any set should be equal among all the sets.

Now friends wonder, whether it is possible to build a wall using **all** bricks they bought in the store.

Input

The first line of the input contains two integers T and N — the number of brick types and the desired height of the wall respectively ($1 \le T \le 5$, $1 \le N \le 10$).

Then follow T lines describing the types. The i-th of these lines contains two integers c_i and l_i — the number of bricks of type i bought by the friends and the length of a single brick of this type ($1 \le c_i \le 64$, $1 \le l_i \le 10^6$).

Output

If it is possible to build a wall without breaking any requirements, print "`Yes`" on a single line of the output. Otherwise, print "`No`".

Examples

dirichlet.in	dirichlet.out
3 2 1 7 2 5 5 1	Yes
3 10 2 14 2 5 1 1	No
3 2 1 7 2 5 1 1	No

Note

In the first example, the only possible way to construct the wall is to make one level from the only brick of the first type and four bricks of the third type, and another level from all bricks of the second type and one brick of the third type.

Please note that the time limit is harsh. Some extra optimizations may be required to get the problem accepted.

Problem E. Euclid

Input file:	euclid.in
Output file:	euclid.out
Time limit:	3 seconds
Memory limit:	512 megabytes

Alexander the Great is dead, and there is no legitimate heir. The empire is going to be divided into pieces between his generals. Nobody wants to start a bloody war, as they were friends just a while ago, so the generals are trying to invent some fair ways to draw the borders.

The empire could be represented as a rectangle on a plane with sides parallel to the coordinate axes. There are N generals situated in different points inside this rectangle. Their first idea was to give everybody an area that is closer to him than to any other general. However, those of them who are now somewhere in the Middle East desert sharply rejected this way of division.

Next, the generals came up with an idea to ask a famous scientist how to distribute the land. They chose Euclid, as he was an outstanding mathematician of the ancient times. His first suggestion was to give everybody an area that is **farther** from him than from any other general. Help them to check this idea, calculate for each general the area of the land he will receive according to this distribution rule. Actually, generals are not interested in the precise values of the areas itself, but what is the percentage each general is going to receive.

Input

The first line of the input describes an empire by two pairs of integers L, D, and R, U, denoting bottom left and upper right corners of the rectangle respectively ($0 \le |L|, |R|, |U|, |D| \le 10^6$, $L < R$, $D < U$).

The second line contains a single integer N, the number of generals who want to cut a piece from the great legacy ($1 \le N \le 100\,000$).

The next N lines describe the locations of generals. Each of these lines contains x_i and y_i — coordinates of the point on a plane where i-th general is located ($L \le x_i \le R$, $D \le y_i \le U$). All the points are distinct.

Output

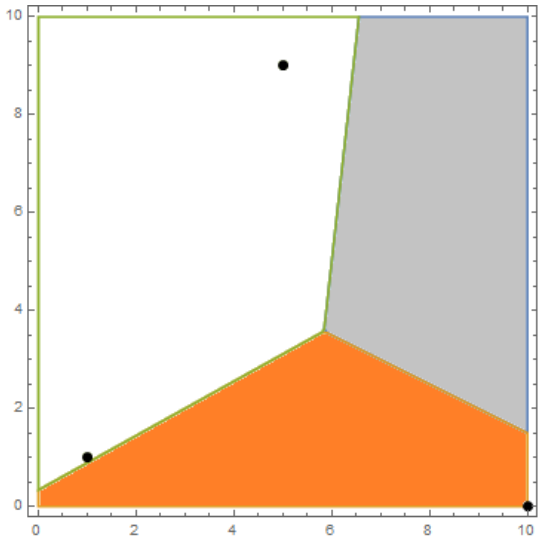
Print N lines, i-th of them containing a single real value, denoting the share of i-th general, which is defined as the area of the part that i-th general will own according to the distribution suggested by Euclid, divided by the total area of the great empire. Your answer will be considered correct if its absolute error does not exceed 10^{-6}.

Formally, if your answer for some general is A and the jury's answer is B, the checker will accept your answer if $|A - B| \le 10^{-6}$.

Examples

euclid.in	euclid.out
0 0 10 10 3 1 1 5 9 10 0	0.2872954595 0.2198690620 0.4928362497

Note



The picture corresponds to the example case.

Problem F. Fulkerson

Input file:	`fulkerson.in`
Output file:	`fulkerson.out`
Time limit:	5 seconds
Memory limit:	512 megabytes

Lester and Delbert already know how to disconnect a railroad network by exploding the minimum possible number of railways. That turned out to be pretty easy, so they went on with another interesting problem: how to rob the Federal Bank. Now they need your assistance to prepare an escape plan.

The Bank is situated in the city that has N intersections and $N - 1$ bidirectional roads connecting them. All roads are of the same length. It is possible to get from any intersection to any other using the road network.

To ensure a safe escape, Lester and Delbert plan to hire K alert groups and place them at different intersections to collect information about police movements. Once the placement is defined, for each of the intersections they calculate the minimum possible number of roads one needs to travel to reach some of the alert groups. The maximum of this value among all intersections is called the *threat level* of the current placement.

Lester wants to hire only one alert group, while Delbert prefers to use N of them. As the resulting solution will most likely lie somewhere in between, they ask you to calculate the minimum possible threat level that can be achieved for every possible value of K.

Input

The first line of input contains a single integer N, the number of intersections in the city ($1 \le N \le 150\,000$). The next $N - 1$ lines contain descriptions of the roads. Each description consists of two integers u and v, denoting that there is a road between these intersections ($1 \le u, v \le N$, $u \ne v$). It is guaranteed that it is possible to get from any intersection to any other using the road network.

Output

The output must consist of N integers. The i-th of these integers must be equal to the minimal possible threat level that could be achieved by using exactly i alert groups.

Examples

fulkerson.in	fulkerson.out
10 7 3 6 9 9 7 1 7 10 7 8 5 4 1 5 9 2 5	3 2 2 1 1 1 1 1 1 0

Note

number of alert groups	one of the optimal placements	optimal distance
1	7	3
2	7, 9	2
3	1, 5, 6	2
4	1, 5, 7, 9	1
5	1, 5, 7, 9, 10	1
6	1, 5, 7, 8, 9, 10	1
7	1, 5, 6, 7, 8, 9, 10	1
8	1, 4, 5, 6, 7, 8, 9, 10	1
9	1, 3, 4, 5, 6, 7, 8, 9, 10	1
10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	0

This table describes the example given on the previous page.

Problem G. Galois

Input file:	galois.in
Output file:	galois.out
Time limit:	1 second
Memory limit:	512 megabytes

Third time Evariste is attempting the entrance examination for the École Polytechnique. He managed to solve all the tasks very quickly, but the examiner accuses him of lack of explanations. He tries to make Evariste fail and gives him the following task: for the given permutation p of size N, count the number of **even** permutations q of size N such that $p_{q_i} = q_{p_i}$ for all i from 1 to N. As this number can be very large, the examiner wants to know it modulo $10^9 + 7$.

A permutation is said to be *even* if it contains an even number of inversions. An *inversion* of a permutation p is a pair of indices (i, j) such that $i < j$ and $p_i > p_j$

Unfortunately for the examiner, who has no idea about the correct answer, Galois managed to solve the problem in only one second. You should help examiner and tell him the answer, or young Evariste will be denied again.

Input

The first line of the input contains a single integer N, which denotes the length of the permutation ($1 \le N \le 500\,000$).

The second line describes the permutation p itself and contains N integers p_i ($1 \le p_i \le N$, $p_i \ne p_j$ for all $i \ne j$).

Output

Count the number of even permutations q that satisfy the condition $p_{q_i} = q_{p_i}$, and output it modulo $10^9 + 7$.

Examples

galois.in	galois.out
3 3 1 2	3
3 2 1 3	1

Note

In the first example there are three appropriate permutations: (1, 2, 3), (2, 3, 1), (3, 1, 2). All of them are even.

In the second example there are two appropriate permutations: (1, 2, 3), (2, 1, 3). However, only first of these permutations is even.

Problem H. Harary

Input file:	harary.in
Output file:	harary.out
Time limit:	1 second
Memory limit:	512 megabytes

Frank enjoys counting graphs with given properties so much that, he even plans to write a book about it one day. He likes to invent new complicated problems in this area, however, coding the solutions is not his favorite part.

Right now, Frank is preparing many versions of a single task for an exam. He wants students to find the number of valid topological orderings of a given directed graph with N nodes.

A *topological ordering* of a directed graph is a permutation of its vertices such that, for every arc from vertex u_i to vertex v_i, u_i comes before v_i in the permutation. Two orderings are considered different if there exists a vertex which has different positions in the first ordering and in the second one, in other words, if they differ as permutations.

Frank considers the task to be easy if the answer is one, medium if the answer is two, and hard if the answer is three. For larger answers, he thinks the task is almost impossible to solve. He also doesn't treat all his students equally, as some of them have visited all the classes, and some will see him for the first time on the exam. However, he doesn't want students to notice his small revenge, so everybody will have the same value of N in their versions of the task.

Now Frank asks you to count, for a fixed N, how many versions of the task he could prepare for each difficulty level. As these numbers can be very large, Frank wants to know them modulo $10^9 + 7$. Note that the number of arcs can be arbitrary, but self-loops and multiple edges are not allowed.

Input

The input contains a single integer N, the number of vertices ($1 \le N \le 100$).

Output

Output three integers: the number of directed graphs with N vertices that have exactly one, two, and three topological orderings. Give the answers modulo $10^9 + 7$.

Examples

harary.in	harary.out
3	12 6 6

Problem I. Ito

Input file:	`ito.in`
Output file:	`ito.out`
Time limit:	1 second
Memory limit:	512 megabytes

Math can be very interesting and fascinating, no doubt it deserves a life to be devoted to it. However, if you plan to end up your days somewhere in Caribbeans, feeling warm sand with your feet and watching beautiful women bringing Pina Colada one by one, you really should find another application of your skills.

That's why Kiyoshi decided his impact to math is already great, so he can go and make some money. As you probably heard, an economical crisis is quickly approaching and everyone cares about efficient way to save their money. While trying to follow new market trends, Kiyoshi opened a consulting agency that helps clients to invest their savings properly. Investment process in this problem consists of three easy steps: go to the market and buy some valuables, wait a year, then go to the market again to sell valuables.

There are N different valuables available for the investment. For each of them the current price c_i wooddollars for one measure unit is given. You may assume that all the valuables are unlimited in quantity and arbitrary divisible, i.e. it's possible to buy any non-negative **real** amount of any valuable. Also, for each valuable two limitations a_i and b_i are known, determining that next year the cost of one measure unit will be continuously and uniformly distributed between these values.

M clients want Kiyoshi to create a good investment plan for them. i-th of them has s_i wooddollars. From the one hand, each client wants to maximize the expected amount of money he will possess after selling all valuables. From the other hand they are afraid of loosing too much, so they want to have at least l_i wooddollars in the worst case.

For each of the clients Kiyoshi needs to choose how much of each of the valuables client should buy. The client can spend any amount of wooddollars, but no more than he owns. Of course, he will possess all the unspent money next year.

The data is too large, so Kiyoshi asks you to help. Tell each client the maximum expected sum of woddollars he could have, without risking too much.

Input

The first line of the input contains two integers N and M ($0 \le N, M \le 100\,000$) — the number of valuables available and the number of the clients.

The following N lines describe the valuables available to buy on the market. Each of them contains three integers c_i, a_i and b_i, denoting the current cost of the valuable and the limitations on the future cost a_i and b_i ($1 \le c_i \le 100\,000$, $0 \le a_i \le b_i \le 100\,000$).

Then follow M lines describing Kiyoshi's clients. For each client integers s_i and l_i are given — the amount of money he has and the minimal amount of money he can agree to have in the worst case ($1 \le l_i \le s_i \le 100\,000$).

Output

Print M lines, i-th of them containing a single real value — the maximum expected amount of wooddollars i-th client could have, without risking too much . Your answer will be considered correct, if it's absolute or relative error won't exceed 10^{-6}.

Formally, if your answer is A and the jury's answer is B, the checker will accept your answer if $\frac{|A-B|}{\max(1,B)} \le 10^{-6}$.

Examples

ito.in	ito.out
3 3 10 5 31 10 10 12 1 0 5 10 1 10 5 10 10	23.6000000000 18.0000000000 11.0000000000

Note

In the example above, the first client can buy 0.1 units of the second valuable and 9 units of the third valuable.

The second client can buy 0.4 units of the first valuable, 0.3 units of the second valuable and 3 units of the third valuable.

The third client can just buy 1 unit of the second valuable.

Problem J. Jordan

Input file:	jordan.in
Output file:	jordan.out
Time limit:	1 second
Memory limit:	512 megabytes

Michael likes to spend time with numbers... wait a minute, the statement should have a story about a mathematician!

Camille likes to spend time with numbers. This morning, he drew on a whiteboard an infinite number line and marked on it a finite set of distinct points with real coordinates. Also, for each of these points he chose some real non-negative value, named it *weight* of the point, and started to have fun. N times Camille selected a segment $[l_i; r_i]$ on the number line and counted the sum of weights of all points lying inside this segment. In case you forget, point x lies inside the segment $[l; r]$ if $l \le x \le r$. He wrote all these calculations down to his notebook and left the room for a lunch.

Henri likes to tease Camille, and that was not an opportunity he could miss. While Camille was enjoying frog legs and young Bordeaux, Henri took a sponge and cleared the board completely. When the prank revealed, Camille's disappointment was out of description!

You accidentally witnessed this crime and now feel a little ashamed for not preventing it. Moreover, you still remember nice days spent together with Bernhard and Camille, when you were freshmen, so you decided to help him. Given his records, calculate minimum and maximum possible sum of weights of all the points. Assume that every marked point belongs to at least one segment selected by Camille.

Input

The first line of the input contains a single integer N — the number of records in Camille's notebook ($1 \le N \le 1000$).

Next N lines describe the calculations he performed. Each of them contains three integers. First two of them are l_i and r_i — borders of the segment, the last one is the sum of weights of all points lying inside this segment s_i ($-10^9 \le l_i \le r_i \le 10^9$, $0 \le s_i \le 10^9$).

Output

Print the minimum and maximum possible total weight of all points drawn by Camille. If the data is inconsistent print two numbers -1 instead.

Examples

jordan.in	jordan.out
2 1 2 1 2 3 1	1 2
2 1 2 1 1 2 2	-1 -1

Note

In the first example, one can obtain the total weight 1 by having just one point with coordinate $x = 2$ and weight 1. Also, one can obtain the total weight 2 by having two points with weights 1 and coordinates $x = 1.5$ and $x = 2.5$ respectively.

In the second example, the input data is obviously inconsistent.

Problem K. Kolmogorov

Input file:	`kolmogorov.in`
Output file:	`kolmogorov.out`
Time limit:	1 second
Memory limit:	512 megabytes

Andrey cares a lot about smart children in the remote parts of Russia, who can't receive good education just because there are no math schools nearby. He dreams of opening a special school with a hostel inside, there gifted children could live and study together, with professors from Moscow State University working as teachers and tutors. He decided to travel around the country to speak with different people and see if they like this idea.

He has just arrived to Nsk, where he should visit the only school of this town and tell an inspiring and motivational speech. He is short in time, so he wants to get from his current location to the school as soon as possible.

There are N intersections in the Nsk, connected by M bidirectional roads. It takes exactly one minute to walk between any pair of intersections directly connected by a road. Every road connects two different intersections, no pair of intersections is directly connected by more than one road, and every pair of intersections is directly or indirectly connected.

Andrey starts his journey at intersection 1 and goes to intersection N, where the school is located. Could you help him and calculate the minumum number of minutes he needs to walk to the target? Of course you could, but that's not the whole statement.

Headmaster of this school isn't happy of this visit and plans to make Andrey's lecture as short as possible. He knows that famous mathematician has a very strong fear of the dark, so to obtain his cruel goal headmaster turned off almost all the street lighting. Now, there is only one road that is still lighted at every moment of time. Moreover, what road is lighted may change every minute. To be more detailed, the following happens:

- At the beginning of every minute one road is randomly and uniformly chosen among all M possible roads.
- This road will be lighted for the current minute.
- If Andrey stands at intersection incident to the lighted road, he may choose to use this road to get to another end. He may also choose to stand still for this minute. Andrey can't use other roads except the lighted one.

Compute the expected number of minutes Andrey needs to reach the goal, assuming he knows everything about the graph and acts optimally.

Input

The first line of the input contains the number of intersections N and the number of bidirectional roads M ($1 \le N, M \le 100\,000$).

Next M lines contain two integers a_i and b_i each, denoting a pair of intersections connected by this road ($1 \le a_i, b_i \le N$, $a_i \ne b_i$). It's guaranteed that it's possible to reach any intersection from any other, and there are no loops or multiple edges.

Output

Print one real number — the expected number of minutes Andrey needs to go from intersection 1 to intersection N, if he acts optimally. Your answer will be considered correct, if it's absolute or relative error won't exceed 10^{-6}.

Formally, if your answer is A and the jury's answer is B, checker will accept your answer if $\frac{|A-B|}{\max(1,B)} \le 10^{-6}$.

Examples

kolmogorov.in	kolmogorov.out
3 2 1 2 2 3	4.0000000000
4 4 1 2 2 4 1 3 3 4	6.0000000000

Note

In the first example, Andrey can act using the following strategy:

- Initially he stays at the intersection 1.
- He waits until the road $(1, 2)$ is lighted, and use it to get to intersection 2. The waiting time equals 2.0 in average.
- Standing at the intersection 2, he waits until the road $(2, 3)$ is lighted, and use it to get to intersection 3. It also requires 2.0 minutes to wait in average, so the expected time to reach the destination is 4.0 minutes.

In the second example, Andrey can act in the following way:

- Initially he stays at the intersection 1.
- He waits until one of the roads $(1, 2)$ or $(1, 3)$ is lighted, and use it to get to the intersection 2 or 3 respectively. It requires 2.0 minutes to wait in average.
- Now he stays at the intersection 2 or 3, and in both cases he should just wait until the road to the intersection 4 is lighted. It requires 4.0 minutes to wait in average, so the expected time to reach the school is 6.0 minutes;