Problem A. Aircraft Seats Arrangement

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

The Byteland Airlines is investing billions of bytalers in a sophisticated algorithm that will assign better aircraft seats to passengers who purchased their tickets earlier.

The seats in the aircraft are arranged in r rows, where r is an even number. There are also three emergency exit rows that serve as access to exits in case of an emergency and do not contain seats. The first exit row is located at the very front of the aircraft (before the first seat row), the second is located exactly in the middle of the aircraft, and the third is at the end of the aircraft (after the last seat row). The rows of the aircraft are numbered with positive integers from 1 to $r + 3$, increasing from the front to the rear of the aircraft. That is, rows numbered 1, $r/2 + 2$, and $r + 3$ are exit rows, while all other rows contain seats.

The seat configuration in the aircraft is 3-3-3, meaning each seat row consists of three groups of three seats separated by passenger aisles. The seats in each row are labeled with consecutive letters according to the following pattern: `ABC.DEF.GHI` from left to right.

When a passenger purchases a ticket, the sophisticated algorithm assigns them a seat according to the following rules:

1. If there is an empty seat in a row directly behind an exit row, all other rows (that do not satisfy this property) are ignored in the next rule.
2. First, select the row with the largest number of empty seats. If there are multiple such rows, choose the one closest to an exit row (the distance between rows a and b is $|a-b|$). If there are still multiple such rows, choose the one with the smallest number.
3. Then, consider all empty seats in the selected row and choose the one with the highest *priority*. The order of seats by priority (from highest to lowest) is:
   (a) Seats next to the aisle in the middle group (D and F)
   (b) Seats next to the aisle in the first and third groups (C and G)
   (c) Window seats (A and I)
   (d) Middle seat of the middle group (E)
   (e) Remaining middle seats (B and H)

   If there are two empty seats with the same highest priority, the balance of the entire aircraft must be considered. Passengers on the left side of the aircraft sit in seats labeled A, B, C, and D, while passengers on the right side sit in seats labeled F, G, H, and I. The algorithm selects the seat on the side of the aircraft with more empty seats. If both sides have an equal number of empty seats, the algorithm selects the seat on the left side of the aircraft.

Assume that some seats in the aircraft are pre-reserved (not necessarily using the described algorithm), and your task is to assign seats for the next n passengers according to the described algorithm.

Input

The first line contains two integers r and n ($2 \le r \le 50$, $1 \le n \le 26$) from the problem text.

The next $r + 3$ lines represent the current state of the seats in the aircraft. Specifically, the j-th line consists of exactly 11 characters representing the state of the j-th row in the aircraft. Exit rows and passenger aisles are marked with the character '`.`', the character '`#`' represents a reserved seat, and the character '`-`' represents an empty seat.

You may assume that there are at least n empty seats in the aircraft.

Output

Print $r + 3$ lines representing the state of the seats in the aircraft after assigning seats to the next n passengers. The output format must match the input format, with the additional property that the seat assigned to the j-th passenger is marked with the j-th lowercase letter of the English alphabet.

Examples

standard input	standard output
2 17 .... ---.#--.--- .... ---.---.---	.... hnd.#lb.fpj .... kqg.cma.eoi ....
6 26 .... ---.---.### #-#.---.--- ---.###.--- .... ---.###.--- #--.#-#.--# #--.--#.#-#	.... gke.aic.### #-#.mzo.r-v x-p.###.n-t .... fjb.###.dlh #-s.#-#.w-# #-u.qy#.#-# ....

Problem B. Border on Map

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

The New Day Kingdom consists of two islands; each island has the shape of a convex polygon and can be represented by a sequence of coordinates of its consecutive vertices.

If the axis y represents the date change line, the first island lies entirely to the left of it (i.e., in the II and III quadrants), while the second lies entirely to the right of it (i.e., in the I and IV quadrants).

The old King decided to split the kingdom between his two twin sons. He plans to draw the line of the border on the map such that each island is split by this line into equal places.

Write a program that, given the coordinates of the vertices of both islands, determines the equation of the line drawn by the old King.

Input

The first line contains an integer n ($3 \le n \le 5\,000$), the number of vertices of the polygon representing the first island.

Each of the next n lines contains two real numbers x and y ($-1\,000 \le x < 0$, $-1\,000 \le y \le 1\,000$), representing the coordinates of the vertices of the first island.

The following line contains an integer m ($3 \le m \le 5\,000$), the number of vertices of the polygon representing the second island.

Each of the following m lines contains two real numbers x and y ($0 < x \le 1\,000$, $-1\,000 \le y \le 1\,000$), representing the coordinates of the vertices of the second island.

For both polygons, the vertices are given in counterclockwise order. The coordinates are specified with exactly three decimal places. You are assumed that the polygons are strictly convex, i.e., no two adjacent edges of the polygon are parallel.

Output

Print two real numbers a and b such that $y = ax + b$ is the equation of the line drawn by the King.

An absolute or relative error of up to 10^{-3} is acceptable. You may assume that a solution always exists and is unique.

Examples

standard input	standard output
3 -6.000 1.000 -2.000 2.000 -5.000 6.000 5 1.000 -1.000 3.000 -2.000 6.000 0.000 4.000 3.000 1.000 2.000	-0.320 1.556
4 -5.000 -1.000 -3.000 -1.000 -3.000 6.000 -5.000 6.000 4 3.222 2.000 5.000 1.000 5.000 4.000 3.222 3.000	-0.000 2.500

Problem C. Conducting An Experiment

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

The scientists conduct the following experiment. The rat is placed in a cage whose floor is divided into an $n \times n$ grid of squares. Each square is either black or colored. The rows are numbered from 1 to n from top to bottom, and the columns are numbered from left to right.

The rat can move from any square to an adjacent square sharing a common side, but it is afraid of colored squares, so it will never step on a colored square at any time.

After the rat spent some time in the cage and learned where it could move, the scientists covered it with a box of size $k \times k$ squares, aligned such that its sides are parallel to the sides of the cage. Although the rat is now in the dark, it still curiously moves only on black squares.

While the rat moves inside the box, its movements cannot be observed from the outside. The only visible movement is that of the box, which occurs when the rat is next to the edge of the box and moves in the direction of the box. In this case, the box also moves one square in that direction.

The scientists recorded the initial position of the box and, for each of its movements, wrote the character '`L`' if the box moved left, '`R`' if it moved right, '`U`' if it moved up, or '`D` if it moved down.

Write a program that, based on the recorded data, determines the minimum number of steps the rat could have taken.

Input

The first line contains two integers n and k ($2 \le k \le 10$, $k < n \le 100$) as described in the problem.

Each of the next n lines contains n characters; each character is either a lowercase '`b`' or a lowercase '`c`'. The letter '`b`' represents a black square, while '`c`' represents a colored square.

The next line contains two integers r and s ($1 \le r, s \le n - k + 1$), representing the row and column of the top-left corner of the box at the start of the experiment.

The next line contains an integer p ($1 \le p \le 10^6$), the number of box movements.

The next line contains a sequence of p characters. Each character is one of the four uppercase letters '`L`', '`R`', '`U`', or '`D`', corresponding to the direction of the box's movement.

Output

Print the smallest possible number of steps the rat could have taken.

Examples

standard input	standard output
5 3 bbbbb bcbcb bbbbb bcccb bbbbb 3 3 2 LU	3
5 3 bbbbb bbccb bccbb bbbbb bbbbb 3 1 4 URRU	10
6 4 bbbbbc bbbccc bcbbbb bccccb bbbbcb bbbbbb 1 1 4 DRUR	18

Note

In Sample 2, the rat could have been located at square (3, 1) and then taken the following 10 steps: up, down, down, right, right, right, up, right, up, up.

In Sample 3, the rat could have been located at square (4, 1) and then taken the following 18 steps: down, up, up, up, right, right, down, right, right, left, left, up, up, down, down, right, right, right.

Problem D. Divide And Conspire

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

The famous gangsters Bytieri have once again escaped from the notorious Al-Gorithm prison!

At the first opportunity, they stopped a train and found a sequence of m NFTs. The i-th NFT has a value of a_i. They decided to divide these NFTs in an unusual way. There are m people in the gang, and they take turns picking one NFT, either from the beginning or the end of the sequence, in a cyclic order. They repeat this process until all NFTs are taken. Since they are not very sophisticated, each brother wonders how much money he could earn at most if the remaining brothers conspired against him and made decisions to minimize his profit (regardless of whether this is optimal for their individual gains).

Input

The first line of the standard input contains two integers n and m, representing the number of NFTs in the sequence and the number of people in the Bytieri gang, respectively ($1 \leq n, m \leq 10^4$).

The second and last line of the standard input contains the sequence a consisting of n integers a_i ($1 \leq a_i \leq 10^5$).

Output

Print m integers. The i-th integer represents the profit of the i-th gang member if the remaining gangsters conspire against him.

Examples

standard input	standard output
3 3 1 2 3	3 2 1
13 4 12 6 4 7 3 9 12 5 4 9 12 7 12	26 23 19 19

Note

In Sample 1, since $n = m$, each gangster will receive one NFT:

- The first gangster is the first to move and can take the third NFT, which has a value of 3.
- The second gangster plays after the first. To prevent the second gangster from taking the NFT with a value of 3, the first gangster will pick it in the first move. The second gangster will then be able to take the NFT with a value of 2.
- The third gangster is last. The first two gangsters will choose NFTs in such a way that the third gangster is left with the NFT of minimal value, i.e., the NFT with a value of 1.

Problem E. Evacuation of Martian Rabbits

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

As is well known, Martian rabbits have no whiskers, are blind, have three times larger ears, and live in caves. In one particular cave, there are n chambers, each containing one rabbit. Some chambers are connected by passages. There are $n-1$ such pairs of chambers. The i-th pair indicates that chamber u_i is directly above chamber v_i. It's possible to move from chamber v_i to chamber u_i through this passage. Chamber 1 is special: there are no chambers above it, only the cave exit. Every other chamber has exactly one chamber above it. It's guaranteed that from any chamber, one can reach chamber 1 by moving through the passages.

Due to magnetic pole shifts, unexpected seismic activity has occurred. The cave is experiencing an earthquake, and all rabbits must be evacuated. The rabbits are numbered from 1 to n, with the i-th rabbit located in chamber i and having strength i. To escape, rabbits can jump upward. Specifically, they can jump to any chamber directly above their current one, or to any chamber above that, and so on. The constraint is that the i-th rabbit with strength i can jump at most i chambers upward in a single leap. Rabbits can make multiple jumps but need to escape in the fewest jumps possible. If there is no chamber "above" to jump to, the rabbit can jump strongly enough to exit the cave directly.

Complicating matters, the earthquake has affected chamber accessibility. While all chambers remain passable, not all have jumping platforms. Only certain chambers contain these platforms, and rabbits can only land on chambers with platforms. However, rabbits can always jump from any chamber (even those without platforms) to chambers above them.

The rabbits have agreed to evacuate in order from rabbit 1 to rabbit n. Before rabbit i begins its escape, one of the following $n+1$ things may happen due to cave instability:

- The state of chamber 1 changes;
- The state of chamber 2 changes;
- ...
- The state of chamber n changes;
- Nothing changes.

A state change means that if a chamber previously had a platform, it's now gone, and vice versa.

For each rabbit, we need to determine the minimum number of jumps required to escape the cave. If escape is impossible, the rabbit will remain trapped (indicated by -1 in the output).

Input

The first line of the input contains one integer n — the number of the chambers and rabbits ($2 \le n \le 5 \cdot 10^5$).

Each of the following $n-1$ lines contains two integers u_i and v_i, indicating that the chamber u_i is directly above v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$). The following line contain n integers b_i ($0 \le b_i \le 1$), where $b_i = 1$ means that the chamber i initially has a platform. The last line contains n integers x_i ($0 \le x_i \le n$) where $x_i = 0$ means no change before rabbit i's turn; otherwise, it means that the state of chamber x_i changes before rabbit i's turn.

Output

Print n integers where the i-th integer is the minimum jumps for rabbit i to escape, or -1 if impossible.

Examples

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

Problem F. Fast Delivery

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

During the day, the cargo company needs to complete n tasks along a street where there are ℓ shops, sequentially labeled from 1 to ℓ from the beginning to the end of the street. An additional interesting fact is that every two neighboring shops on the street are exactly 1 meter apart. Each of the tasks involves picking up an important package at one shop and delivering it to another shop.

The management knows all the tasks it needs to complete that day in advance and will provide this list to the delivery robot. The order in which the tasks are completed does not matter to the robot.

Note that:

- **The robot is strong**, so it can carry any number of important packages at the same time.
- **The robot is strong and smart**, so it wants to complete all tasks while covering the smallest possible total distance. It doesn't matter which shop it starts at, nor does it matter which shop it ends at.

Write the program that calculates the minimum distance covered by the robot.

Input

The first line of the input contains two integers ℓ ($1 \le \ell \le 10^9$) and n ($1 \le n \le 10^5$) as described in the problem.

Each of the next n lines contains two integers a_i and b_i ($1 \le a_i, b_i \le \ell$, $a_i \ne b_i$), indicating that the i-th task involves picking up a package at the shop a_i and delivering it to the shop b_i.

Output

Print the required minimum distance as described in the problem, in a single line.

Examples

standard input	standard output
10 6 1 4 3 5 6 7 2 1 9 4 8 5	14
100 3 11 50 50 49 36 35	42

Problem G. Great Desert

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

In the Great Desert, there is a river and two cities located at its ends. The river is a polygonal line that starts in one city and ends in the other.

Each straight segment of the river flows strictly from south to north, or in the direction of increasing y-coordinate.

We need to build a road connecting these two cities. The road can run alongside the river, but it cannot simply cross it: a bridge must be built at each crossing.

The cost of building one meter of road on land is 1 unit, and the cost of building each bridge (crossing) is t.

Within the cities, traffic is already resolved, so the road can start and end on either side of the river.

Input

The first line of the input contains two numbers:

- An integer n ($2 \le n \le 1500$), representing the number of river vertices (including the cities).
- A real number t ($0 < t \le 10^6$), representing the cost of one bridge. t is given with at most two decimal places.

Each of the following n lines contains two integers x_i and y_i ($|x_i|, |y_i| \le 10^5$).

This pair represents the coordinates of the i-th river vertex. They satisfy $y_i < y_{i+1}$ for $1 \le i < n$.

There are no three collinear points in the input. The road must start at point (x_1, y_1) and end at point (x_n, y_n).

Output

Print a single integer representing the minimum possible cost of the project. The answer will be accepted if the absolute or relative error is at most 10^{-6}.

Example

standard input	standard output
5 1 0 0 -1 2 4 3 -3 4 1 5	6.8416192530

Problem H. Hairband and Nails

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 1024 megabytes

Alice found a wooden board and n nails in the old attic. She quickly hammered the nails into the board. The board can be represented as a coordinate plane, with the nails as points on it. No two nails share the same x-coordinate or the same y-coordinate.

To further entertain herself with her newfound items, Alice stretched the hairband around all the nails and then let it go. Naturally, the hairband contracted around the outermost nails. Alice then repeats the following process until the number of nails on the board is greater than 2:

1. She writes down the area of the shape described by the hairband on a piece of paper.
2. She chooses one of the following nails: the leftmost, rightmost, topmost, or bottommost.
3. She removes the selected nail from the board, and the hairband contracts again around the remaining outer nails.

Write a program that calculates the numbers written on the paper, given which nail Alice chose in each step.

Input

The first line contains an integer n $(3 \leq n \leq 300,000)$ as described in the problem.

Each of the next n lines contains two positive integers x and y, representing the coordinates of each nail. All coordinates will be less than 10^9, and no two nails will share the same x or y coordinate.

The next line contains a sequence of $n - 2$ characters, each being '`L`', '`R`', '`U`', or '`D`', indicating that Alice chose the leftmost, rightmost, topmost, or bottommost nail, respectively, in each step.

Output

Output $n - 2$ integers representing the areas Alice wrote down on the paper in order. Each area should be printed with exactly one decimal place after the point.

Examples

standard input	standard output
5 1 4 2 2 4 1 3 5 5 3 LUR	9.0 6.5 2.5
8 1 6 2 4 3 1 4 2 5 7 6 5 7 9 8 3 URDLUU	34.0 24.0 16.5 14.0 9.5 5.0

Problem I. Installation Disks

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

Gill Bates needs to make space in his overcrowded office. He usually grabs a stack of old floppy disks and throws them away. This time, he found a set of n disks containing his favorite operating system — The Doors 95.

The disks are neatly labeled with natural numbers from 1 to n, in the order they should be inserted during installation. Gill recalled all the happy moments and decided not to throw these disks away until he sorts them from 1 to n.

First, he arranged all the disks in a sequence where he can perform only one type of operation called *ms-swap*. One ms-swap operation works as follows:

1. Divide the entire sequence of disks into four consecutive subsequences (intervals) of arbitrary sizes (including empty subsequences), labeled A, B, C, and D in order.
2. Rearrange these subsequences such that the new sequence consists of C, A, D, and B in this order.

Determine the minimum number of ms-swap operations needed to sort the sequence of disks from 1 to n. Note that the subsequences in different ms-swap operations do not need to be identical.

Input

The first line contains an integer n ($1 \le n \le 10$) as described in the problem statement.

The next line contains a permutation of integers from 1 to n representing the initial sequence of disks.

Output

Print the minimum number of ms-swap operations needed to sort the given sequence.

Example

standard input	standard output
9 3 4 8 9 1 2 5 6 7	1

Problem J. Jumbo And Food

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

The elephant Jumbo is very hungry and hopes to find food in an $r \times c$ matrix where each cell is either cyan ('`C`') or black ('`B`').

Jumbo starts in the top-left corner of the matrix and wants to reach the bottom-right corner where the food is located. In each step, he must move to an adjacent cell (up, down, left, or right). If the cell he moves to is black, the following occurs: from all black cells in the matrix, one is chosen uniformly at random, and Jumbo teleports to it (possibly staying on the same cell).

After this, the process continues with the next step. Thus, in each step, Jumbo first moves to an adjacent cell, and then at most one teleportation occurs (if the cell is black). The quest ends when Jumbo reaches the bottom-right corner.

Determine the expected number of steps Jumbo will take to reach the bottom-right corner (assuming he moves optimally). The top-left and bottom-right cells are always cyan.

Input

The first line of the input contains an integer t ($1 \le t \le 1000$), the number of scenarios. For each scenario:

- The first line contains two integers r and c ($1 \le r, c \le 1000$).
- The next r lines contain c characters each ('`C`' for cyan or '`B`' for black), describing the matrix.
- You may assume that the first character in the first line and the last character in the last line are equal to '`C`'.

The total number of cells across all scenarios will not exceed 10^6.

Output

For each scenario, print the expected value as a completely reduced fraction in the form `numerator/denominator`.

Example

standard input	standard output
3 2 2 CC CC 2 2 CB BC 1 12 CBCCCCCCCCBC	2/1 2/1 4/1

Problem K. King Byteazar's Road

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

The main road of the Kingdom of Byteotia, connecting the old and new capitals of the kingdom and named in honor of the legendary King Byteazar, is heavily used, so the current King orders the repair of the road.

The Royal Minister of Roads has surveyed the highway and marked all potholes that need repair. The repair process works as follows: first, starting from a point before the first damage, the road is divided into segments of equal length. Then, for each segment containing damage, one excavator with its crew is dispatched.

Due to the insufficient number of excavators in the country, traffic experts need to determine how to divide the road into segments of a predetermined length such that the number of segments with damage is minimized.

The road has n potholes, each represented by an integer indicating its distance from the start of the road. The length of each segment is predetermined as the positive integer m. The first m meters of the road are undamaged. The road is divided into segments by choosing a starting point for the first segment, which must lie within the first m meters. If the first segment starts at the k-th meter, then the i-th segment starts at $k+(i-1)\cdot m$ meters. One excavator can cover all potholes from the start of a segment (inclusive) to the start of the next segment (exclusive).

Write a program to determine the minimum number of excavators required to repair the highway and all possible starting positions for the first segment.

Input

The first line contains two integers m and n ($1 \le m, n \le 100\,000$) — the length of the segment and the number of potholes, respectively.

The second line contains n integers $x_1, x_2, \ldots, x_n$ ($m < x_1 < x_2 < \ldots < x_n \le 2 \cdot 10^9$), representing the positions of all potholes on the road.

Output

In the first line, print the minimum number of excavators required to repair the highway.

In the next line, print the positions of all possible starting points for the first segment. These numbers should be listed in ascending order and separated by a single space.

Examples

standard input	standard output
3 5 4 5 7 8 9	2 1
4 3 7 14 15	2 1 2 4
2 10 3 4 7 8 12 13 14 15 20 21	7 1 2

Problem L. Looking For Palindromic Path

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

The graph contains n vertices. Initially, no two vertices are connected. You need to process q queries of two types:

- 1 u v c — an edge with the symbol c is added between vertices u and v.
- 2 u v — you need to answer whether there is currently a path (not necessarily vertex-simple or edge-simple) between vertices u and v, such that the symbols on the edges in the path form a palindrome.

Input

The first line of input contains two integers n and q — the number of vertices and the number of queries, respectively ($1 \le n \le 1000$, $1 \le q \le 3 \cdot 10^5$).

Each of the following q lines contains queries in the format described in the statement ($1 \le u, v \le n$, $u \ne v$, c — a lowercase English letter).

It is guaranteed that at any moment in time, any two vertices are connected by no more than one edge.

Output

For each query of type 2, print in a new line 1 if the path with the property from the statement exists at the time of the query, and 0 otherwise.

Example

standard input	standard output
5 6 1 1 2 a 1 2 3 b 2 1 3 2 2 3 1 3 4 a 2 1 4	0 1 1

Problem M. Median On Path

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	1024 megabytes

Given a connected graph with n vertices and m edges, where each vertex i has a value x_i.

Define $S(a,b)$ as the sequence of values x_v for all vertices v that lie on at least one simple path from vertex a to vertex b. A simple path from vertex a to vertex b is a sequence of distinct vertices $v_1, v_2, \ldots, v_k$, where $v_1 = a$, $v_k = b$, and for every $1 \le i \le k-1$, vertices v_i and v_{i+1} are connected by an edge.

Given q queries of the form (a,b), find the median of $S(a,b)$. The median of a sequence c is defined as the element $c'_{\lceil m/2 \rceil}$, where c' is c sorted in non-decreasing order, with m being the length of c. Here, $\lceil y \rceil$ denotes the ceiling of y, i.e., the smallest integer greater than or equal to y.

The queries are generated as follows. Let ans_i be the answer to the i-th query, and $ans_0 = 0$. For each query, two numbers a_1 and b_1 are given. Then, the actual query parameters are computed as:

$$a = (a_1 + ans_{i-1} \cdot p - 1)\%n + 1,$$

$$b = (b_1 + ans_{i-1} \cdot p - 1)\%n + 1,$$

where i is the query number and p is a given parameter.

Input

The first line of the input contains four integers n, m, q, and p, representing the number of the vertices in the graph, the number of edges, the number of queries, and the parameter for query generation, respectively ($1 \le n, m, q \le 5 \cdot 10^5$, $0 \le p \le 1$).

The second line contains n integers x_i, where the i-th integer represents the value of the vertex i ($1 \le x_i \le 10^9$).

The next m lines each contain two integers u and v, indicating an edge between vertices u and v ($1 \le u, v \le n$, $u \ne v$). Note that there may be multiple edges connecting the same vertices. You may assume that the graph is connected.

The following q lines each contain two integers a_1 and b_1, which are used to compute the actual query parameters a and b as described above ($1 \le a_1, b_1 \le n$).

Output

Print q lines, where the i-th line contains the answer to the i-th query.

Examples

standard input	standard output
6 6 7 0 1 2 3 4 5 6 1 2 2 4 4 3 3 1 5 4 4 6 1 1 2 1 1 4 4 6 5 6 5 1 3 5	1 2 2 4 5 3 3
6 7 3 1 7 8 9 10 11 12 1 2 1 3 2 3 3 4 4 5 4 6 5 6 1 6 1 2 5 3	9 11 8

Note

In Sample 1, the only path from vertex 1 to itself is the trivial path containing only vertex 1. Thus, $S(1,1) = \{1\}$, and its median is 1.

From vertex 2 to vertex 4, there are two paths: $2 \to 4$ and $2 \to 1 \to 3 \to 4$. Thus, $S(2,4) = \{1,2,3,4\}$, and its median is 2.