Problem A. Email Aliases

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

Polycarp has quite recently learned about email aliases. Of course, he used to suspect that the case of the letters doesn't matter in email addresses. He also learned that a popular mail server in Berland `bmail.com` ignores dots (characters '`.`') and all the part of an address from the first character "plus" ('`+`') to character "at" ('`@`') in a login part of email addresses.

Formally, any email address in this problem will look like "`login@domain`", where:

- a "`login`" is a non-empty sequence of lowercase and uppercase letters, dots ('`.`') and pluses ('`+`'), which starts from a letter;
- a "`domain`" is a non-empty sequence of lowercase and uppercase letters and dots, at that the dots split the sequences into non-empty words, consisting only from letters (that is, the "`domain`" starts from a letter, ends with a letter and doesn't contain two or more consecutive dots).

When you compare the addresses, the case of the characters isn't taken into consideration. Besides, when comparing the `bmail.com` addresses, servers ignore the dots in the `login` and all characters from the first character "plus" ('`+`') to character "at" ('`@`') in login part of an email address.

For example, addresses `saratov@example.com` and `SaratoV@Example.Com` correspond to the same account. Similarly, addresses `ACM.ICPC.@bmail.com` and `A.cmIcpc@Bmail.Com` also correspond to the same account (the important thing here is that the domains of these addresses are `bmail.com`). The next example illustrates the use of character '`+`' in email address aliases: addresses `polycarp+contest@BMAIL.COM`, `Polycarp@bmail.com` and `polycarp++acm+icpc@Bmail.Com` also correspond to the same account on the server `bmail.com`. However, addresses `a@bmail.com.ru` and `a+b@bmail.com.ru` are not equivalent, because '`+`' is a special character only for `bmail.com` addresses.

Polycarp has thousands of records in his address book. Until today, he sincerely thought that that's exactly the number of people around the world that he is communicating to. Now he understands that not always distinct records in the address book represent distinct people.

Help Polycarp bring his notes in order by merging equivalent addresses into groups.

Input

The first line of the input contains a positive integer n ($1 \le n \le 2 \cdot 10^4$) — the number of email addresses in Polycarp's address book.

The following n lines contain the email addresses, one per line. It is guaranteed that all of them are correct. All the given lines are distinct. The lengths of the addresses are from 3 to 100, inclusive.

Output

Print the number of groups k and then in k lines print the description of every group.

In the i-th line print the number of addresses in the group and all addresses that belong to the i-th group, separated by a space. It is allowed to print the groups and addresses in each group in any order.

Print the email addresses exactly as they were given in the input. Each address should go to exactly one group.

Examples

standard input	standard output
6 ICPC.@bmail.com p+con+test@BMAIL.COM P@bmail.com a@bmail.com.ru I.cpc@Bmail.Com a+b@bmail.com.ru	4 2 ICPC.@bmail.com I.cpc@Bmail.Com 2 p+con+test@BMAIL.COM P@bmail.com 1 a@bmail.com.ru 1 a+b@bmail.com.ru

Problem B. Layer Cake

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	512 megabytes

Dasha decided to bake a big and tasty layer cake. In order to do that she went shopping and bought n rectangular cake layers. The length and the width of the i-th cake layer were a_i and b_i respectively, while the height of each cake layer was equal to one.

From a cooking book Dasha learned that a cake must have a form of a rectangular parallelepiped constructed from cake layers of the same sizes.

Dasha decided to bake the biggest possible cake from the bought cake layers (possibly, using only some of them). It means that she wants the volume of the cake to be as big as possible. To reach this goal, Dasha can cut rectangular pieces out of the bought cake layers. She always cuts cake layers in such a way that cutting lines are parallel to the edges of that cake layer. Dasha isn't very good at geometry, so after cutting out a piece from the original cake layer, **she throws away** the remaining part of it. Also **she can rotate** a cake layer in the horizontal plane (swap its width and length).

Dasha wants her cake to be constructed as a stack of cake layers of the same sizes. Each layer of the resulting cake should be made out of only one cake layer (the original one or cut out from the original cake layer).

Help Dasha to calculate the maximum possible volume of the cake she can bake using given cake layers.

Input

The first line contains an integer n ($1 \le n \le 4000$) — the number of cake layers that Dasha can use.

Each of the following n lines contains two integer numbers a_i and b_i ($1 \le a_i, b_i \le 10^6$) — the length and the width of i-th cake layer respectively.

Output

The first line of the output should contain the maximum volume of cake that can be baked using given layers.

The second line of the output should contain the length and the width of the resulting cake. If there are many solutions with maximum possible volume, print any of them.

Examples

standard input	standard output
5 5 12 1 1 4 6 6 4 4 6	96 6 4
2 100001 900000 900001 100000	180000000000 900000 100000

Note

In the first example Dasha doesn't use the second cake layer. She cuts 4×6 rectangle from the first cake layer and she uses other cake layers as is.

In the second example Dasha cuts off slightly from the both cake layers.

Problem C. Polycarp's Masterpiece

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

Inspired by success of Joanne Rowling and her fantasy series of Harry Potter, Polycarp decided to create his own masterpiece. He has already picked a name for his future bestseller — string s.

Polycarp faced a number of difficulties while he was working on the main story of his book so he decided to become Malevich in a literature and create his own "Black Square".

Chapters of his masterpiece were written by Polycarp in n days. He started from writing down the string s. Every day (during the following n days) he did the following procedure: on the i-th day he added to the right of current text t its k_i-th circular shift. It means that every day content of his masterpiece doubled in size.

Circular shift of a string r is defined as a string resulted after movement of the last character of r to the first position in the string. As an example: circular shift of «`masterpiece`» is «`emasterpiec`». The i-th circular shift of a string r is defined as a string resulted after applying i subsequent circular shifts of the string r (e.g., the third circular shift of «`masterpiece`» is «`ecemasterpi`»).

After n days of tedious work Polycarp managed to write a very long text deserving the efforts made. However, text of his masterpiece was so long that it was nearly impossible to do any analysis of its content.

Help Polycarp to answer m requests about his masterpiece: for the j-th request you will need to find how many times a letter c_j is contained in a substring of his masterpiece starting from position l_j and ending at r_j inclusively.

Input

The first line contains a name of the masterpiece — string s. The string s contains only lowercase latin letters and its length is between 1 and 100 inclusively.

The second line contains a pair of integer numbers n and m ($1 \le n \le 10^5$, $1 \le m \le 10^5$) — a number of days Polycarp worked on his masterpiece and a number of requests you need to answer for the text respectively.

The next line consists of n integer numbers k_i ($0 \le k_i \le 100$) — circular shifts used during n days of work.

Following m lines contains two integer numbers l_j, r_j and a lowercase latin letter c_j each — requests description. For each request you need to count number of times letter c_j is contained in a substring of the masterpiece from position l_j to r_j inclusively. Positions are numerated starting from 1. It is guaranteed that $1 \le l_j \le r_j \le 10^{18}$ and both l_j and r_j do not exceed length of the masterpiece.

Output

Output should contain m lines, where the j-th line contains the answer on the j-th request.

Examples

standard input	standard output
masterpiece 1 3 3 1 22 m 9 14 e 8 15 p	2 4 0
polycarp 1 2 0 2 15 p 1 16 p	2 4

Problem D. Boulevard

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

Welcoming autumn evening is the best for walking along the boulevard and n people decided to do so.

The boulevard can be represented as the axis Ox. For every person there are three parameters characterizing the behavior: t_i, s_i, f_i — the moment of time when the i-th person starts walking, the start point and the end point of the walk respectively. Each person moves in a straight line along the boulevard from s_i to f_i with a constant speed of either 1 or -1 depending on the direction.

When the i-th person appears on the boulevard at the point s_i she immediately starts walking towards the point f_i.

If two or more persons meet at the boulevard (they are at the same point at the same time, no matter which directions they are going) they all greet each other. Like in the normal life, every pair of people greet each other at most once.

You task is to calculate for every person how many people she greets while walking along the boulevard.

Please, pay attention to the fact that i-th person may meet and greet any other person at points s_i and f_i. After a person achieves the destination point f_i she moves out of the boulevard and cannot greet anyone else. The same rule applies to the start of the walk: a person cannot greet anyone until she appears on the boulevard.

Input

In the first line there is an integer n ($2 \le n \le 1000$) — the number of people who decided to go for a walk.

The following n lines contain parameters for n people. In the i-th line there are three positive integers t_i, s_i, f_i ($1 \le t_i, s_i, f_i \le 10^6$, $s_i \ne f_i$), where t_i, s_i, f_i — the moment of time when the i-th person starts walking, the start point and the end point of the walk respectively.

Output

The single line of the output should contain a sequence of n integers $r_1, r_2, \ldots, r_n$ separated by a space, where r_i denotes the number which the i-th person greets other people while walking along the boulevard.

Examples

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

Problem E. Training with Doors

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

Col. Sheffard is going to make a training for Cpt. Plice. The colonel has a polygon containing infinite number of closed doors placed one by one in a row. Initially Cpt. Plice stands in front of the first door. The captain is able to perform the following two commands:

- open the first closed door and go to the front of the next closed door (mark this command as '('),
- go to the front of the last of currently opened doors and close it (mark this command as ')').

A sequence of commands is valid if the following conditions are met:

- there is at least one opened door when the command ')' should be performed (in particular, the first command shouldn't be ')'),
- all the doors are closed after performing all commands from the sequence (the captain stands in front of the first door).

Col. Sheffard received a template — a rectangular grid with n rows and m columns where each cell contains one of the commands, i.e. '(' or ')'. The colonel has to choose a training plan for the captain — select a non-empty rectangle from the grid, each row of the rectangle will be an independent training for the captain. The training plan is valid, if *each* row of the selected rectangle is a valid sequence of commands.

You are to find the number of ways to select a valid training plan for a given grid. The selected rectangle should be continuous, i.e. without holes. Plans are different if they have different positions of their corners even if their contents are the same.

Input

The first line contains two integers n and m ($1 \le n, m \le 50000$) — number of rows and number of columns in the template respectively. Each of the following n lines contains a string with m symbols '(' or ')' — commands of the template. The total number of elements in the template doesn't exceed 10^6.

Output

Output a single integer — the number of rectangles on the template, corresponding to a valid training plan. Two rectangles with equal content but different positions of corners should be counted both.

Examples

standard input	standard output
1 5 (()()	3
3 2 () () ()	6
4 4 ()() ()())()(()()	13

Problem F. Gourmet and Banquet

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

A gourmet came into the banquet hall, where the cooks suggested n dishes for guests. The gourmet knows the schedule: when each of the dishes will be served.

For i-th of the dishes he knows two integer moments in time a_i and b_i (in seconds from the beginning of the banquet) — when the cooks will bring the i-th dish into the hall and when they will carry it out ($a_i < b_i$). For example, if $a_i = 10$ and $b_i = 11$, then the i-th dish is available for eating during one second.

The dishes come in very large quantities, so it is guaranteed that as long as the dish is available for eating (i. e. while it is in the hall) it cannot run out.

The gourmet wants to try each of the n dishes and not to offend any of the cooks. Because of that the gourmet wants to eat each of the dishes for **the same amount of time**. During eating the gourmet can instantly switch between the dishes. Switching between dishes is allowed for him only at integer moments in time. The gourmet can eat no more than one dish simultaneously. It is allowed to return to a dish after eating any other dishes.

The gourmet wants to eat as long as possible on the banquet without violating any conditions described above. Can you help him and find out the maximum total time he can eat the dishes on the banquet?

Input

The first line of input contains an integer n ($1 \le n \le 100$) — the number of dishes on the banquet.

The following n lines contain information about availability of the dishes. The i-th line contains two integers a_i and b_i ($0 \le a_i < b_i \le 10000$) — the moments in time when the i-th dish becomes available for eating and when the i-th dish is taken away from the hall.

Output

Output should contain the only integer — the maximum total time the gourmet can eat the dishes on the banquet.

The gourmet can instantly switch between the dishes but only at integer moments in time. It is allowed to return to a dish after eating any other dishes. Also in every moment in time he can eat no more than one dish.

Examples

standard input	standard output
3 2 4 1 5 6 9	6
3 1 2 1 2 1 2	0

Note

In the first example the gourmet eats the second dish for one second (from the moment in time 1 to the moment in time 2), then he eats the first dish for two seconds (from 2 to 4), then he returns to the second dish for one second (from 4 to 5). After that he eats the third dish for two seconds (from 6 to 8).

In the second example the gourmet cannot eat each dish for at least one second because there are three dishes but they are available for only one second (from 1 to 2).

Problem G. Hiring

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

The head of human resources department decided to hire a new employee. He created a test exercise for candidates which should be accomplished in at most m working days. Each candidate has to pass this test exercise. During the j-th day a candidate is allowed to be in the office for at most t_j units of time.

Overall, n candidates decided to apply for the job and sent out their resumes. Based on data received the head has defined two parameters describing every candidate: d_i and r_i. The parameter d_i is the time to get prepared for work which the i-th candidate spends each morning. This time doesn't depend on day. The parameter r_i is the total working time needed for the i-th candidate to accomplish the whole test exercise.

Thus the time spent in the office in the j-th day consists of d_i units of time to get prepared and some units of time to proceed with the exercise. A candidate can skip entire working day and do not come to the office. Obviously in this case he doesn't spend d_i units of time to prepare.

To complete the exercise a candidate should spend exactly r_i units of time working on the exercise (time to prepare is not counted here).

Find out for each candidate what is the earliest possible day when he can fully accomplish the test exercise. It is allowed to skip working days, but if candidate works during a day then he must spend d_i units of time to prepare for work before he starts progressing on the exercise.

Input

The first line contains two integer numbers n, m ($1 \le n, m \le 2 \cdot 10^5$) — the number of candidates and the maximum number of working days to do the test exercise.

The second line contains m integer numbers $t_1, t_2, \ldots, t_m$ ($1 \le t_j \le 10^6$) — the durations of working days in time units.

The following n lines contain two integers each: d_i, r_i ($0 \le d_i \le 10^6$, $1 \le r_i \le 10^6$) — how much time in the beginning of a day is required for i-th candidate before he starts his work on the test exercise and how much time it is needed for him to accomplish this task.

Output

Output a sequence of n integer numbers $b_1, b_2, \ldots, b_n$, where b_i is the earliest day when the i-th candidate can finish the test exercise.

In case the i-th candidate cannot finish the test exercise in m days output $b_i = 0$.

Days in this problem are numbered from 1 to m in the order they are given in the input.

Examples

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

Problem H. Tourist Guide

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

It is not that easy to create a tourist guide as one might expect. A good tourist guide should properly distribute flow of tourists across the country and maximize the revenue stream from the tourism. This is why there is a number of conditions to be met before a guide may be declared as an official tourist guide and approved by Ministry of Tourism.

Ministry of Tourism has created a list of k remarkable cities out of n cities in the country. Basically, it means that in order to conform to strict regulations and to be approved by the ministry a tourist guide should be represented as a set of routes between remarkable cities so that the following conditions are met:

- the first and the last city of every route are distinct remarkable cities,
- each remarkable city can be an endpoint of at most one route,
- there is no pair of routes which share a road.

Please note that a route may pass through a remarkable city. Revenue stream from the tourism highly depends on a number of routes included in a tourist guide so the task is to find out a set of routes conforming the rules of a tourist guide with a maximum number of routes included.

Input

The first line contains three integer numbers n, m, k ($1 \le n \le 50000$, $0 \le m \le 50000$, $1 \le k \le n$) — the number of cities in the country, the number of roads in the country and the number of remarkable cities correspondingly.

Each of the following m lines contains two integer numbers a_i and b_i ($1 \le a_i, b_i \le n$) — meaning that cities a_i and b_i are connected by a bidirectional road. It is guaranteed that a_i and b_i are distinct numbers and there is no more than one road between a pair of cities.

The last line contains k distinct integer numbers — a list of remarkable cities. All cities are numbered from 1 to n.

Output

The first line of the output should contain c — the number of routes in a tourist guide. The following c lines should contain one tourist route each. Every route should be printed in a form of "t v_1 v_2 $\dots$ v_{t+1}", where t is a number of roads in a route and v_1, v_2, $\dots$, v_{t+1} — cities listed in the order they are visited on the route.

If there are multiple answers print any of them.

Examples

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

Problem I. Lottery

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

Today Berland holds a lottery with a prize — a huge sum of money! There are k persons, who attend the lottery. Each of them will receive a unique integer from 1 to k.

The organizers bought n balls to organize the lottery, each of them is painted some color, the colors are numbered from 1 to k. A ball of color c corresponds to the participant with the same number. The organizers will randomly choose one ball — and the winner will be the person whose color will be chosen!

Five hours before the start of the lottery the organizers realized that for the lottery to be fair there must be an equal number of balls of each of k colors. This will ensure that the chances of winning are equal for all the participants.

You have to find the minimum number of balls that you need to repaint to make the lottery fair. A ball can be repainted to any of the k colors.

Input

The first line of the input contains two integers n and k ($1 \le k \le n \le 100$) — the number of balls and the number of participants. It is guaranteed that n is evenly divisible by k.

The second line of the input contains space-separated sequence of n positive integers c_i ($1 \le c_i \le k$), where c_i means the original color of the i-th ball.

Output

In the single line of the output print a single integer — the minimum number of balls to repaint to make number of balls of each color equal.

Examples

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

Note

In the first example the organizers need to repaint any ball of color 2 to the color 1.

In the second example the organizers need to repaint one ball of color 1 to the color 2 and two balls of the color 1 to the color 3.

Problem J. Cleaner Robot

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

Masha has recently bought a cleaner robot, it can clean a floor without anybody's assistance.

Schematically Masha's room is a rectangle, consisting of $w \times h$ square cells of size 1×1. Each cell of the room is either empty (represented by character '`.`'), or occupied by furniture (represented by character '`*`').

A cleaner robot fully occupies one free cell. Also the robot has a current direction (one of four options), we will say that it looks in this direction.

The algorithm for the robot to move and clean the floor in the room is as follows:

1. clean the current cell which a cleaner robot is in;
2. if the side-adjacent cell in the direction where the robot is looking exists and is empty, move to it and go to step 1;
3. otherwise turn 90 degrees clockwise (to the right relative to its current direction) and move to step 2.

The cleaner robot will follow this algorithm until Masha switches it off.

You know the position of furniture in Masha's room, the initial position and the direction of the cleaner robot. Can you calculate the total area of the room that the robot will clean if it works infinitely?

Input

The first line of the input contains two integers, w and h ($1 \le w, h \le 10$) — the sizes of Masha's room.

Next w lines contain h characters each — the description of the room. If a cell of a room is empty, then the corresponding character equals '`.`'. If a cell of a room is occupied by furniture, then the corresponding character equals '`*`'. If a cell has the robot, then it is empty, and the corresponding character in the input equals '`U`', '`R`', '`D`' or '`L`', where the letter represents the direction of the cleaner robot. Letter '`U`' shows that the robot is looking up according to the scheme of the room, letter '`R`' means it is looking to the right, letter '`D`' means it is looking down and letter '`L`' means it is looking to the left.

It is guaranteed that in the given w lines letter '`U`', '`R`', '`D`' or '`L`' occurs exactly once. The cell where the robot initially stands is empty (doesn't have any furniture).

Output

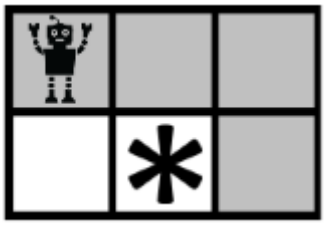
In the first line of the output print a single integer — the total area of the room that the robot will clean if it works infinitely.

Examples

standard input	standard output
2 3 U.. .*.	4
4 4 R... .**. .**.	12
3 4 ***D ..*. *...	6

Note

In the first sample the robot first tries to move upwards, it can't do it, so it turns right. Then it makes two steps to the right, meets a wall and turns downwards. It moves down, unfortunately tries moving left and locks itself moving from cell $(1, 3)$ to cell $(2, 3)$ and back. The cells visited by the robot are marked gray on the picture.



Problem K. Task processing

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

Vasya wants to create a computing system to process arbitrary tasks. As a part of the system he needs an algorithm which will choose an order of task execution.

He came up with the following algorithm:

- There is a single execution queue. For task to be completed, it needs to be added to the queue.
- For each task two values are known: l_i and t_i — the number of seconds that it takes to complete the task and the moment of time when this task is added to the execution queue.
- If at some moment of time T the algorithm has to select a task for execution, it picks the one with the minimum value of $l_i - (T - t_i)^2$. In case of a tie, algorithm picks a task with the lowest index. Then for the following l_i seconds the algorithm will wait for the task to be completed.

In order to test the algorithm Vasya wants you to simulate it.

You are given n tasks. For each task, you know the number of seconds l_i that it takes to complete the task and the moment of time t_i when this task is added to the execution queue.

For each task find out the moment of time when it will be completed.

Input

The first line contains an integer number n ($1 \le n \le 10^5$) — the number of task to process. The next n lines contain two integers each: l_i, t_i ($1 \le l_i \le 10^5$, $0 \le t_i \le 10^5$).

Output

Print n space-separated integers. The i-th integer is the moment of time when the i-th task was completed.

Examples

standard input	standard output
1 10 5	15
3 3 0 4 3 5 2	3 7 12
3 3 0 4 2 5 1	3 12 8
6 3 0 5 1 4 2 5 18 4 19 5 14	3 8 12 24 28 19

Problem L. Agricultural Archaeology

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

Recently Berland archaeologists found ancient text describing agricultural field of ancient people. It had a form of $n \times m$ rectangle divided into $n \cdot m$ unit cells. Each cell was sown with some kind of a food plant. There were 90 kinds of food plants popular in ancient Berland.

As written in the ancient text, region of each kind of food plant that was sown formed a single perfect square without any holes. Two square regions are adjacent if they share at least one unit cell border. The ancient text does not mention the sizes of the squares. But it provides something about the adjacent squares. For each square, it is known which kind of plants were sown to the each side of this square. Formally, for each food kind there are four lists: top neighbour kinds, right neighbour kinds, bottom neighbour kinds and left neighbour kinds. Food plants are written in arbitrary order in each list of neighbours.

Help archaeologists to reconstruct any possible agricultural field given the information from the ancient text.

Input

The first line contains three integer numbers n, m, k ($1 \le n, m \le 300$, $1 \le k \le 90$) — the sizes of the field and the number of the sown kinds of food plants.

The sown food plants are encoded with characters which ASCII codes are from 33 (‘`!`’) to 122 (‘`z`’) inclusively.

Each of the following lines describes one sown food plant (square) and has the format: “`char top right bottom left`”, where `char` is the character encoding the plant, and `top`, `right`, `bottom`, `left` are the strings of ASCII characters with codes from 33 to 122 — the lists of corresponding neighbours (all characters in each list are unique and written in arbitrary order). The empty list of neighbours is given by the character ‘`~`’ which ASCII code is 126. All characters which encode the sown food plants are different.

Output

Print the field in the form of $n \times m$ matrix of characters with ASCII codes from 33 to 122 — possible field corresponding to the given input. If there are many solutions, print any of them. It is guaranteed that at least one solution exists.

Examples

standard input	standard output
5 4 6 a ~ b zc ~ z ba ~ ~ dce e d z ~ ~ b ~ ~ z a c a z d ~ d c z e ~	aabb aabb czzz dzzz ezzz

Problem M. Taxi in Berland

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

Polycarp came to the capital of Berland to attend a major conference. In order not to be late, he called a taxi and asked the driver to take him as quickly as possible from the railway station to the building, which will host the conference. It is about to start in a few minutes!

The capital of Berland is a rectangular field with the width of w and the length of l. From a plane, this field is viewed as a rectangle with leftmost bottom corner at the point (0, 0) and the rightmost top corner at the point (w, l). Roads in the capital of Berland are continuous vertical and horizontal segments which are parallel to the coordinate axes. There are $w + 1$ vertical roads and $l + 1$ horizontal roads. The coordinates of the ends of i-th vertical road ($0 \le i \le w$) are in points (i, 0) and (i, l). The coordinates of the ends of j-th horizontal road ($0 \le j \le l$) are in points (0, j) and (w, j). Thus, each point within the rectangular field with integer coordinates is a crossroads.

The railway station is located at the point of the field (x_1, y_1) with integer coordinates and the taxi will take Polycarp from this point. The major conference will be held in a building that is at the point (x_2, y_2) with integer coordinates.

Cars in the capital of Berland can travel only by roads, that is, from every intersection the car can go either left or right or up or down to the nearby crossroads (cars cannot leave the field boundaries during driving).

The taxi, which arrived for Polycarp, can accelerate and slow down with the acceleration which does not exceed a by the absolute value and the maximum speed of the taxi is v_{max}. The taxi can turn (even rotate to reverse the direction) without reducing the speed.

The order is strictly preserved in the capital of Berland and there are n police cars with speed radar equipped at crossroads of the city. The maximum speed allowed in the capital of Berland is v_p so if a policeman is at the crossroads then the taxi driver will not pass this crossroads with a speed greater than v_p (a policeman located at crossroads (x_i, y_i) measures the speed of a car in the moment when the car passes this crossroads). The taxi driver agrees to drive with any speed in any other point of city because there is no risk to be arrested.

The taxi starts from the point (x_1, y_1) with zero speed. Polycarp is so afraid of being late, that he agreed to literally jump out of the taxi at the moment when it is at the point (x_2, y_2). Therefore, the taxi may arrive to the end point of the path with any speed not exceeding v_{max}.

It is guaranteed that no two policemen share the same position. There is no policeman at points (x_1, y_1) and (x_2, y_2). Points (x_1, y_1) and (x_2, y_2) are distinct.

You have to find the minimum time after which the taxi driver will be able to arrive the destination without passing a policeman with speed exceeding v_p.

Input

The first line of the input contains six numbers w, l, n, a, v_{max}, v_p ($1 \le w, l \le 100$, $0 \le n \le 100$, $0.01 \le a \le 5.00$, $1 \le v_{max}, v_p \le 100$). All given numbers are integers, except the acceleration a, which is a float number and is given with exactly two digits after the decimal point.

The second line of the input contains four integers x_1, y_1, x_2, y_2 ($0 \le x_1, x_2 \le w$, $0 \le y_1, y_2 \le l$) — where (x_1, y_1) are the coordinates of the starting point (the railway station) and (x_2, y_2) are the coordinates of the end point (the building where the conference will take place).

The following n lines contains two integers x_i, y_i ($0 \le x_i \le w$, $0 \le y_i \le l$) — the coordinates of crossroads occupied by the police.

It is guaranteed that no two policemen share the same position. There is no policeman at points (x_1, y_1)

and (x_2, y_2). Points (x_1, y_1) and (x_2, y_2) are distinct.

Output

The output should contain the only float number — the minimum time required for the taxi driver to drop off Polycarp from the railway station to the building hosting the conference. Relative or absolute error of the answer should not exceed 10^{-6}.

Examples

standard input	standard output
5 5 1 0.50 3 1 2 1 4 1 3 1	2.8284271247