

Judging Notes

1. Very few clarifications are issued during the contest, and they are only for ambiguities. Please read the problem statement and examine the sample test cases carefully before submitting any request for clarifications.
 2. The following guidelines apply to handling input/output in programs:
 - All input comes from standard input.
 - All output goes to standard output.
 - Unless the problem explicitly states otherwise, the input for a problem consists of a single test case. If the input contains multiple test cases, then the problem will also state an upper bound on the number of test cases.
 - Your program may be run on multiple input files. Note that this means that if your program has more than one error (say, Time Limit Exceeded and Wrong Answer), then you can get either error as judgment.
 - Output formatting should follow the specification and sample output in the problem statement.
 3. For problems with floating point output, the judging system will accept a range of answers as correct as long as they satisfy the constraints described in the problem statement. These constraints will be specified as an absolute and/or relative tolerance, which will be given.
 4. There is no such thing as “Presentation Error” or “Format Error.” If you misspell the word “impossible,” for example, and the problem requires that word as output, then your submission will be judged as “Wrong Answer.”
 5. Unless a problem specifically indicates that uppercase or lowercase letters are important, then either will be accepted. For example, “Yes” or “yes” would be treated the same, but “yse” is Wrong Answer.
 6. You should follow the sample output format, but extra whitespace within reason is acceptable. For example, if you print out a gigabyte of blanks, then the judging system will treat that as Wrong Answer; however an extra blank at the end of a line or an extra blank line between test cases is acceptable.
 7. During the contest, input size constraints on test cases will be given as part of the problem statements.
 8. If you submit a solution that has a Compile Error, then you will be notified of it (just as any other error). However, Compile Errors do not count toward penalty time.
-

Problem A : Final Price

Neverland has recently experienced a rapid rise in the inflation rate. The value of money is continuously decreasing, and citizens' purchasing power is lowered daily. The government is trying to control the inflation rate by testing various methods, such as reducing the amount of money in the economy by increasing interest rates and promoting investment, even in purchasing cars to be delivered in an unforeseeable future.

Despite these efforts, the inflation rate is still above 50%, and the prices are jumping up and down drastically every now and then. The Statistical Center of Neverland provides detailed information on the inflation rate and the average price change over time for a basket of goods commonly purchased by households. However, it is hard to calculate the actual price of a specific good at a given point in time using that information.



The ICPC (International Center for Price Changes) is asking you to help the citizens of Neverland to calculate the prices for their desired goods after a specific period of time. The information provided to you is the initial price of a good at the start of a time period, and the daily price change for that good over the observed time period.

Input

The first input line contains an integer n ($1 \leq n \leq 1000$), indicating the number of days the prices are observed. The second line contains a positive integer indicating the initial price of the desired good on day 1. The next $n - 1$ lines contain $n - 1$ integers showing the daily change in the price of the good, in order from day 2 to day n . You can assume that the price of the good is always positive and never exceeds 1,000,000 in the observed period.

Output

In the output, print the final price of the good on day n .

Example

Standard Input	Standard Output
2 11 68	79

Standard Input	Standard Output
4 110 -5 0 27	132

Problem B : Flower Festival

Today is the Flower Festival day. The festival is held in Rose Square, at the end of Flower Street. People are heading towards the festival on Flower Street with n cars, numbered 1 through n . Soroush, an expert traffic analyst, wants to know which car will arrive at Rose Square first. Using the traffic cameras on Flower Street, he has gathered the current location of all cars, along with their speeds. Each car maintains a constant speed throughout their journey. Also, the location of a car is defined as its distance from the start of Flower Street. Help Soroush find the first car that arrives at the festival. It is guaranteed that no two cars reach Rose Square at the same time.

Input

The first line of input contains two space-separated integers n ($1 \leq n \leq 100$) and f ($1 \leq f \leq 10,000$), the number of cars and the length of Flower Street, respectively. The $(i + 1)$ -th line (for $1 \leq i \leq n$) contains the information of car numbered i , two space-separated integers x_i ($0 \leq x_i < f$) and v_i ($1 \leq v_i \leq 100$) indicating its observed location and speed, respectively.

Output

Print the number of the car which will arrive at Rose Square first.

Example

Standard Input	Standard Output
3 200 0 1 10 5 40 1	2

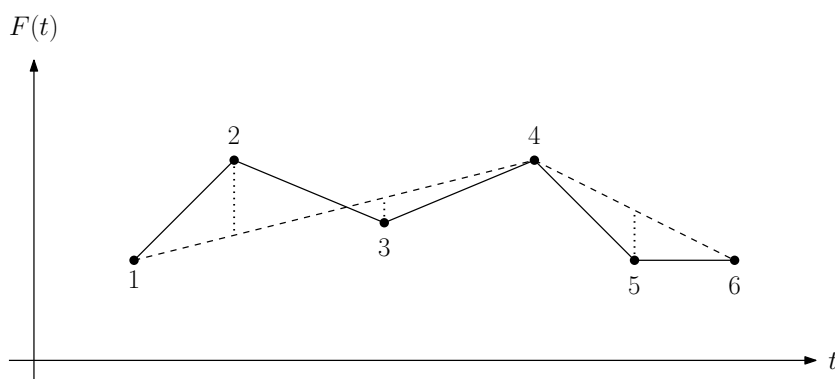
Standard Input	Standard Output
5 100 0 1 10 3 60 2 75 1 10 4	3

Problem C : Simplification

Amin records the price of his stock every now and then as a data point (t_i, p_i) in his notebook, where p_i is the price of his stock at day t_i . The sequence of these data points represents a piecewise-linear function F displaying the history of prices over a period of time. Indeed, F connects every pair of consecutive points by a straight line segment. If the price is not recorded for some day t , $F(t)$ can be used as an estimate instead.

His collected data is getting larger and larger as he has been tracking the price of his stock over a long period of time. Therefore, he has decided to reduce his data by throwing away some of his recorded data points and constructing a new piecewise-linear function F' with the remaining points. F' is a so-called “simplification” of F . Amin wants to create the simplification in such a way that F' is a good approximation for F . To this end, he has defined an error measure as follows.

Let F be defined over a strictly increasing sequence $L = \langle t_1, \dots, t_n \rangle$ of days, and F' be defined over a subsequence $L' = \langle t'_1, \dots, t'_m \rangle$ of L , where $t'_1 = t_1$, $t'_m = t_n$, and $F'(t'_i) = F(t'_i)$ for $1 \leq i \leq m$. (We call m the size of F' .) The error of F' is defined as the maximum of $|F'(t_k) - F(t_k)|$ for all $1 \leq k \leq n$. For example, in the following figure, we have 6 data points, labeled 1 through 6, whose coordinates are the same as those presented in the second sample input, and F' is a simplification of F of size 3 with data points 1, 4 and 6. In this figure, F is depicted by solid lines, and F' by dashed lines. The error measure for F' is realized by the vertical distance of point 2 to F' .



Amin’s goal is to minimize the size of F' , while the error of F' is bounded by a given value δ .

Input

The first line of input contains a positive integer n ($2 \leq n \leq 2000$) that shows the size of F . Each of the next n lines contains two integers t_i, p_i ($1 \leq t_i, p_i \leq 10^6$), where p_i is the price of the stock at day t_i . The last line contains the error limit δ which is a non-negative integer at most 10^6 .

Output

In the output, print the minimum possible size of F' .

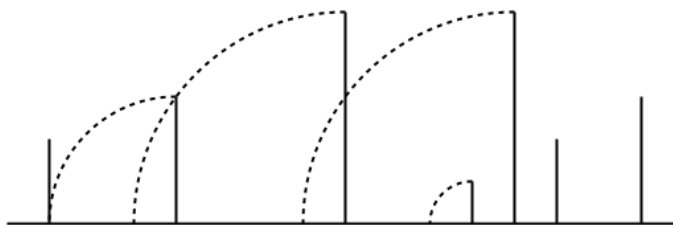
Example

Standard Input	Standard Output
3 1 10 3 100 10 20 90	2

Standard Input	Standard Output
6 10 10 20 20 35 14 50 20 60 10 70 10 8	3

Problem D : Dominoes

Pixar is gearing up to introduce its next animated film, Elemental, at the 2023 Cannes Film Festival. But one of the scenes needs re-rendering. In this scene, there are n dominoes in a straight line, and one of them is supposed to fall and drop a number of subsequent dominoes in a domino-like manner. Considering it is less than 1 month away from the 2023 Cannes Film Festival, Pixar asked you to write a program that calculates the cost of re-rendering the scene for n initial domino choices.



To simplify the calculations, we assume that the dominoes are displayed from the side like line segments with no width on the coordinate axis, and they fall to the left. They are numbered from left to right and domino i has height l_i and is located at x_i . The cost of re-rendering this scene is equal to the area of the moving part of the scene, i.e. the union area of quarter circles of dropped dominoes. Note that domino falling areas may overlap, and the overlapped area should be calculated only once. The picture illustrates the falling of dominoes in the first sample test, when the initial domino choice for falling is the domino number 5.

Input

The first line of input contains a positive integer n , indicating the number of dominoes. The next n lines, each consists of a pair of integers, x_i and l_i , indicating the location and the height of the domino number i . It is guaranteed that $n \leq 200,000$ and $1 \leq x_i, l_i \leq 10^9$. Dominoes are given from left to right; i.e. $x_i < x_{i+1}$.

Output

Output n lines, where the i -th contains the cost of re-rendering the scene with i -th domino as the initial falling domino. All numbers must have an absolute or relative error of at most 10^{-6} .

Example

Standard Input	Standard Output
7	3.141592654
2 2	7.068583471
5 3	24.659773696
9 5	0.785398163
12 1	42.250963921
13 5	44.164186876
14 2	49.714124052
16 3	

Problem E : Parking Party

Peyman has decided to host a dinner party in his house at Zabol. His house has a rectangular parking lot which can be represented by a grid with n rows and m columns. A car can park in one of the $n \times m$ cells in this grid. However, some cells are occupied by immovable pillars and thus, no car can park in or pass through them. Each row or column in the parking lot has an entrance on both its ends. When a car enters the parking lot through an entrance, it can just move forward straightly in the corresponding row or column; it stops and parks in a grid cell when it reaches the opposite entrance or when the next cell is occupied by a pillar or another parked car. Additionally, a car cannot enter a row or column if its first cell is already occupied.

Peyman wants to maximize the number of cars that can be parked in his parking lot. In order to do that, he can instruct the guests on which entrance to take upon arrival. Your task is to help Peyman achieve this task.

Input

The first line of input contains two space-separated integers n and m ($1 \leq n, m \leq 1000$), the number of rows and columns in the parking lot, respectively. Each of the following n lines contain a string of length m made of “.” and “o” characters. the j -th character of the $(i + 1)$ -th line is an “o” if the cell in row i and column j contains a pillar, and it is a “.” if it is empty.

Output

Print a single integer k , the maximum number of cars Peyman can park in his parking lot.

Example

Standard Input	Standard Output
3 3 .o. o.o .o.	4

Standard Input	Standard Output
3 4 ooooo	7

Problem F : Ammunition Storage

The Do-Barareh military area is like an $n \times m$ grid, each cell of which has a specific height. The commander of this military area is looking for a rectangular sub-area of this area, with width and height least 2, whose its four corner cells are higher than the rest of its cells. He plans to install watchtowers in the corners of this sub-area to monitor the entire sub-area and use it for ammunition storage. Your job is to help the commander to find out how many valid sub-areas there are to choose as the ammunition storage. You can assume cell heights are distinct.

Input

The first line of input contains two space-separated integers n and m ($2 \leq n, m \leq 750$). Each of the next n lines contains m space-separated integers showing the cell heights. It is guaranteed cell heights are distinct numbers between 1 and nm (inclusive).

Output

Print the number of valid sub-areas to be used as an ammunition storage.

Example

Standard Input	Standard Output
3 3 9 4 8 2 1 3 7 5 6	7

Problem G : Laboratory Report

You are assigned to write a software for a medical laboratory. A portion of the software is about generating reports on the conducted medical tests. A medical test consists of a series of measurements on specimens (like blood or urine) sampled from a patient. Each measurement has a name, a unit, and a result (value). For example, the “Cholesterol” in a blood sample might be 180 mg/dl. Additionally, each measurement is attached with a *reference table*. A reference table is a general set of rules classifying a (status) flag for the patients based on their measurement results. For example, if “Triglycerides” has a result in the range [10, 190], then it is classified as “normal”; and it is flagged as “low” or “high” if it has a result below or above this range, respectively.

Given all the measurement reference tables and the results for several patients, you have to generate the report of their medical tests.

Input

A *text* in the input specification of this problem is a non-empty string of arbitrary printable ASCII characters, including the space, punctuation, and alphanumeric characters, but not the newline character. It is guaranteed that a text does not start or end with the white space characters. In addition, all real numbers in the input are in decimal format with at most 7 characters, and their absolute value will not exceed 10^7 .

The input consists of multiple (at least 2 and at most 100) sections. Each section ends with a single line containing 80 consecutive “=” characters.

The reference tables are described in the first section. There are at least 1 and at most 100 reference tables, and the lines of two adjacent tables are separated with a single line containing 75 consecutive “-” characters. The first line of a reference table contains the measurement name, a text of length at most 25, which is unique among all the reference tables. The second line contains a text of length at most 15 as the measurement unit. The remaining of the reference table describes its flag classification rules and is either a single line, or a set of pairs of lines. In the former case, the single line specifies the criteria for classifying the result as “Normal”. In this case, if the measurement result does not meet this criteria, it is classified as “Abnormal”. In the latter case where the remaining of the reference table is in the form of a non-empty set of (at most 10) pairs of lines, the classification rules are more complex. In this case, the first line of each pair specifies the classification criteria, and the second line contains a text of length at most 25 as the flag name preceded by two space characters. In any of the cases mentioned above, the classification criteria is a text of length at most 50 in one of the following forms (A and B are real numbers, and x is a measurement result):

Criteria format	Meaning
$< A$	$x < A$
$\leq A$	$x \leq A$
$> A$	$x > A$
$\geq A$	$x \geq A$
$[A, B]$ or $A \sim B$	$A \leq x \leq B$
$[A, B)$	$A \leq x < B$
$(A, B]$	$A < x \leq B$
(A, B)	$A < x < B$

2022 ICPC Asia Tehran Regional Contest

There might be arbitrary number of spaces between the different parts of a text specifying a classification criteria. So for example, $[2, 5]$, $[2 , 5]$, and $2 \sim 5$ are valid and equivalent texts. When the flag classification rules of a reference table are described with a set of pairs of lines, it is guaranteed that the measurement value space is correctly partitioned by the given set of classification criteria; i.e., each possible measurement result matches exactly one of the provided classification criteria.

The next sections in the input specify the measurement results for the patients. The first line of each section contains a text of length at most 60 as a patient's name. Each of the next lines describes a measurement result for that patient with the measurement name and value (a real number) separated with an arbitrary number (between 1 and 20) of space characters. There are at least 1 and at most 100 measurement results for a single patient. Each measurement name appears at most once in the measurement results of a patient.

Output

For each section in the input except the first section (which describes the reference tables), you shall write the laboratory test report of its corresponding patient followed by a single line containing 80 consecutive "=" characters. A report consists of a line containing the patient's name and a table depicting the test results. The table format is illustrative in the sample output. If there were k measurements in the corresponding input section, this table should have $k + 4$ lines of exact length 75. All the empty areas shall be filled with the space character. The table has 4 columns:

- **Test**: The measurement name; starting at the beginning of the line (the 1-st character).
- **Result**: The measurement result value, printed in the exact format as the input; starting at the 27-th character of the line.
- **Unit**: The measurement unit; starting at the 35-th character of the line.
- **Flag**: The classified flag based on the measurement result and the reference table; starting at the 51-st character of the line. Nothing (but space characters) should be written in this cell if the flag is "Normal".

The measurement rows in the output table should be printed in the same order of appearance as in the corresponding input section. In addition to the measurement rows, the table should have a heading line and three separator lines (containing 75 consecutive "-" characters) as depicted in the sample output.

Notice: The solutions for this problem are judged very strictly. In order for your output to be considered correct, all its characters, including the white space or upper/lower case letters, must exactly match the expected output.

Example

```

Standard Input

Cholesterol
mg/dl
130~200
-----
HDL-chol
mg/dl
>= 55
-----
Fast Blood Sugar
mg/dl
<70
  Hypoglycemia
70 ~ 99
  Normal
(99, 126)
  Impaired (Prediabetic)
>= 126
  Diabetic
-----
Secret Poison
mg/L
< 0.1
  Noneffective
0.1 ~0.5
  Mild
( 0.5 , 1.5  ]
  Moderate
(1.5,5)
  Severe
[5, 10]
  Lethal (non-instant)
> 10
  Instant Killer
=====
Mr. Fat
Fast Blood Sugar 130
Cholesterol      230
HDL-chol        55
=====
413%3! 4n@+01!3\#!/(# /\@\/@1n`/
Secret Poison    3.8
Cholesterol      150
=====
Y@1d@ 49#@|=@21!
Fast Blood Sugar 90
Secret Poison    7.9
=====

```

2022 ICPC Asia Tehran Regional Contest

Standard Output

Mr. Fat

```
-----
Test                Result  Unit      Flag
-----
Fast Blood Sugar    130     mg/dl     Diabetic
Cholesterol         230     mg/dl     Abnormal
HDL-chol           55      mg/dl
```

=====
413%3! 4n@+01!3\#!/!(# /\/@\@1n`/

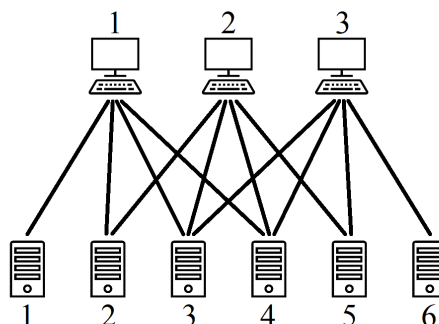
```
-----
Test                Result  Unit      Flag
-----
Secret Poison       3.8     mg/L      Severe
Cholesterol         150     mg/dl
```

=====
Y@1d@ 49#@|=@21!

```
-----
Test                Result  Unit      Flag
-----
Fast Blood Sugar    90      mg/dl
Secret Poison       7.9     mg/L      Lethal (non-instant)
```

Problem H : Network Topology in Hezardastan

Hezardastan, a leading information technology group in Iran, has a huge data center containing n servers and m terminals (where $m \leq n$). A terminal is a pair of keyboard and monitor that can be connected to a server for administrative purposes. The servers are numbered 1 through n and the terminals are numbered 1 through m . This data center has a network topology in which not every terminal is necessarily able to connect to every server. For example, the figure below depicts 3 terminals and 6 servers where a terminal can connect to a server if a line is drawn between them.



A subset S of the servers with size m is called *manageable* if its members are allowed by the network topology to be simultaneously managed by the terminals, i.e. each terminal can be connected to a distinct server in S . For example, the subset $\{2, 3, 6\}$ in the example above is manageable as its members can be respectively managed by the terminals $\{1, 2, 3\}$. A subset of the servers is called *unmanageable* if it has size m and is not manageable. A network topology is called *totally manageable* when it causes no unmanageable subset of servers. For example, the network topology shown in the example above is totally manageable, but if the connection link between terminal 2 and server 5 is removed, then it will not be totally manageable anymore since the subsets $\{1, 5, 6\}$, $\{2, 5, 6\}$, $\{3, 5, 6\}$, and $\{4, 5, 6\}$ will become unmanageable. Given a network topology for the data center, you have to find if it is totally manageable or it makes an unmanageable subset.

Input

The first line of input contains two integers m and n separated with a single space ($1 \leq m \leq 150$, $1 \leq n \leq 400$, $m \leq n$). The next m lines describe the network topology by an $m \times n$ matrix. Each of these lines contains n space-separated integers which are either 0 or 1. The j -th number (for $1 \leq j \leq n$) in the $(1+i)$ -th line of input (for $1 \leq i \leq m$) is 1 if terminal i can connect to server j , and it is 0 otherwise.

Output

If the given network topology is totally manageable, you only have to print 1 in the first line of output. Otherwise, you should print 0 in the first line of output and an unmanageable subset of servers in the second line in the form of m space-separated integers (indicating the server numbers, in any arbitrary order). If there are multiple unmanageable subsets, you can print any one of them.

Example

Standard Input	Standard Output
<pre>3 6 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1</pre>	<pre>1</pre>

2022 ICPC Asia Tehran Regional Contest

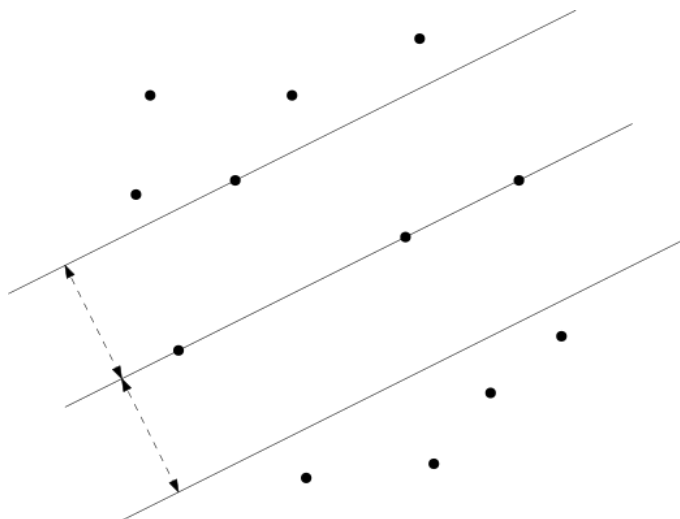
Standard Input	Standard Output
3 6	0
1 1 1 1 0 0	1 5 6
0 1 1 1 0 0	
0 0 1 1 1 1	

Problem I : Windcatchers

Yazd is the city of windcatchers. There are several famous and historical windcatchers in Yazd, including the windcatcher of Dowlatabad Garden, which is the tallest one in the world.

The new mayor of Yazd is going to celebrate the history of windcatchers. To this end, he has decided to construct a new highway in the city with the following properties:

- The highway is composed of two ways of the same width.
- The two ways are adjacent, with a straight line in common (which we call the middle line).
- The highway must not pass through any windcatcher. However, windcatchers may lie on the boundary of the ways, including on the middle line.
- To celebrate the windcatchers, the highway must have at least two windcatchers exactly on its middle line.



The mayor wants to construct a highway of maximum possible width satisfying all the above conditions. However, due to a large number of windcatchers in the city, finding the best place for such a highway is not an easy task. As such, the mayor has decided to hire you to find the best location for constructing the highway. To simplify things, each windcatcher is represented by a single point in the plane. Moreover, we may assume that the constructed highway has infinite length. An example of a highway of maximum width among a set of points (windcatchers) is illustrated in the above figure.

Input

The first line of input contains a positive integer n , indicating the number of windcatchers. The next n lines, each consists of a pair of integers, x and y , indicating the location of a windcatcher in the city. Note that line $i + 1$ contains the location of windcatcher i . For simplicity, we assume that each windcatcher is a point in two dimensions, and no two windcatchers lie on the same point. Moreover, we assume that there are at least three noncollinear windcatchers in the city. It is guaranteed that $3 \leq n \leq 4,000$ and $0 \leq x, y \leq 10^9$.

Output

Print a float number indicating the width of a maximum-width highway satisfying all the required conditions. Your output will be considered correct if its absolute or relative error is at most 10^{-9} .

Example

Standard Input	Standard Output
3 0 0 0 1 1 0	1

Standard Input	Standard Output
4 15 18 11 20 20 9 7 8	9.356972863938

Problem J : Magic with Cards

Mahsa has been practicing shuffling cards for a few months now. Tonight, she finally decided to invite her friends over and show off her new skills. So she picks up a deck with $2n$ cards, shows her friends the face of the cards without changing the deck order and asks someone to pick two positions i and j in the deck. Then, she tells everyone that she is going to move the card in the i -th position to the j -th position by applying only two types of shuffles.

Assume the cards in the deck are $\langle c_1, c_2, \dots, c_{2n} \rangle$. Mahsa can perform these two shuffles as many times as she wants:

Riffle: Divide the cards into two parts $\langle c_1, \dots, c_n \rangle$ and $\langle c_{n+1}, \dots, c_{2n} \rangle$ and produce $\langle c_1, c_{n+1}, c_2, c_{n+2}, \dots, c_n, c_{2n} \rangle$.

Scuffle: From $\langle c_1, c_2, \dots, c_{2n} \rangle$, produce $\langle c_2, c_1, c_4, c_3, \dots, c_{2n}, c_{2n-1} \rangle$.

Help Mahsa find out the minimum number of shuffles she needs, and she'll figure out the rest.

Input

The input consists of a single line containing three space-separated integers n , i and j ($1 \leq n \leq 10^5$ and $1 \leq i, j \leq 2n$).

Output

Print a single integer, the minimum number of shuffles required to bring the i -th card to j -th position. If it is not possible to do so, print -1 instead.

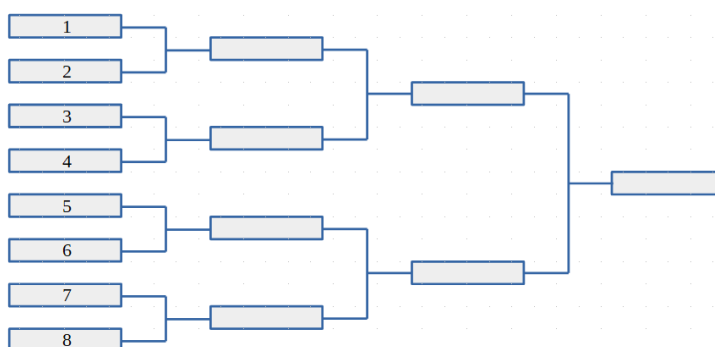
Example

Standard Input	Standard Output
4 3 8	3
Standard Input	Standard Output
5 4 1	5
Standard Input	Standard Output
1 1 1	0

Problem K : Iranian Hazfi Cup

The Iranian Hazfi Cup is a football tournament organized every year in a knockout format; i.e. the loser of each match is immediately eliminated from the tournament, and the winner gets to play in the next round. Every year, 2^k teams participate in this tournament (for some positive integer k). All teams start the tournament in the first round and after each round, half of the teams that are still in the tournament are eliminated. The k -th round is the final round, where two teams compete for the championship. In total, $2^k - 1$ matches are held.

The *tournament bracket* of the Hazfi Cup is determined ahead of time in the drawing ceremony in the presence of special guests. It determines which teams are facing each other in the first round, and which other teams they might encounter if they advance to the next rounds. Precisely, in the drawing ceremony, all 2^k teams are randomly mapped to the positions $1, 2, \dots, 2^k$ in the first round as depicted in the figure for $k = 3$.



The Iranian football federation must start organizing the Hazfi Cup 2023. As many of the special guests might refuse to attend the drawing ceremony this year, the federation has decided to use the same tournament bracket as Hazfi Cup 2022. Unfortunately, last year's tournament bracket is not available, but all match results of last year's tournament are available in an arbitrary order. It can be shown that the tournament bracket can be uniquely determined from these match results. Your task is to recover the tournament bracket from the match results of Hazfi Cup 2022 in order to answer the following fans' common questions for this year:

- In which round two teams, A and B, may face each other?

Input

The input starts with a line containing two space-separated integers k , the number of rounds in the tournament ($1 \leq k \leq 10$), and n , the number of fans' questions ($1 \leq n \leq 1000$). The match results of the Hazfi Cup 2022 come in the next $2^k - 1$ lines; one line for each result. Each match result is of the form:

- $teamA \ g_A \ - \ g_B \ teamB$

where $teamA$ and $teamB$ are different non-empty strings of lowercase English letters of length at most 100, and g_A and g_B denote the number of goals scored by $teamA$ and $teamB$, respectively ($g_A \neq g_B$). In the case of a draw, the winner is determined by penalty shootouts, and the match result is of the form:

- $teamA \ g \ (p_A) \ - \ g \ (p_B) \ teamB$

where g is the number of goals scored by each team during the main game, and p_A and p_B are the number of goals scored by $teamA$ and $teamB$ during the penalty shootouts, respectively ($p_A \neq p_B$). The number of scored goals (i.e. g_A, g_B, g, p_A , and p_B) are all non-negative integers less than 100. Note that each line denoting a match result in the input contains exactly 4 space characters.

The input ends with n queries. Each query is given in a separate line containing two different team names delimited by a space character.

Output

For each query in the input, print as the answer, a single integer in a separate line in the output.

Example

Standard Input	Standard Output
2 3	2
perspolis 1(4) - 1(3) esteghlal	2
sepahan 0 - 2 perspolis	1
esteghlal 4(1) - 4(0) shamoshak	
shamoshak sepahan	
shamoshak perspolis	
esteghlal shamoshak	