TASK	MARKO	GEPPETTO	ARTUR	SAVEZ	VUDU	DRZAVA
input	standard input (*stdio*)					
output	standard output (*stdout*)					
time limit	2 seconds	1 second	1 second	1 second	1 second	1 second
memory limit	64 MB	64 MB	64 MB	64 MB	64 MB	64 MB
score	**50**	**80**	**100**	**120**	**140**	**160**
	total 650					

Good old Marko came across a new feature on his mobile phone – T9 input! His phone has a keyboard consisting of numbers looking like this:

1	2 abc	3 def
4 ghi	5 jkl	6 mno
7 pqrs	8 tuv	9 wxyz

In order to input a word using this keyboard, one needs to press a key multiple times for the required letter. More specifically, if the required letter is the first letter mapped to the key, one key press is needed, if it's the second, two key presses are needed and so on. For instance, if we want to input the word "giht", we will press the following keys: g-4 i-444 h-44 t-8. The new possibility Marko discovered enables you to input text more easily because you **don't need several presses** per letter anymore, just one. The software will try to figure out what word from the dictionary you are trying to input.

Marko is quite sceptical of new technologies (at least new for him) and he is afraid that errors will be frequent. That is the reason why he decided to test his hypothesis that the errors are frequent. Marko knows by heart the whole dictionary in the mobile phone. The dictionary consists of N words consisting of lowercase letters from the English alphabet, the total length of the word not exceeding1 000 000 characters. He will give an array of key presses S, of total length at most 1 000, and wants to know how many words from the dictionary can be mapped to the given array of key presses if the T9 input feature is used.

INPUT

The first line of input contains the integer N, the number of words in the dictionary. ($1 \leqslant N \leqslant 1\,000$). Each of the following N lines contains a single word. The last line of input contains the string S ($1 \leqslant |S| \leqslant 1000$) consisting of digits 2-9.

OUTPUT

The first and only line of output must contain the number of words from the dictionary possible to construct from the letters on the keys determined by the string S.

SAMPLE TESTS

input	input	input
3 tomo mono dak 6666	2 ja la 52	3 dom fon tom 366
output	**output**	**output**
1	2	2

Clarification of the first example: "mono" is the only word that has all the letters located on key 6..
Clarification of the second example: The first letter of both words is located on key 5 and the second letter of both words is located on key 2.

Everyone's favorite character and puppet-maker Geppetto has opened a new pizza place, the best in town. Geppetto is trying to make the best pizza possible, but at the same time he doesn't want to have a small selection of pizzas.

He makes his pizzas out of N ingredients marked with numbers from 1 to N. All that would be simple if he could mix any ingredient with every ingredient on the pizza, but unfortunately, that is not the case. Sometimes some ingredients cannot mix and that creates additional complications for our pizza master.

There are M pairs of ingredients that cannot be on the same pizza at the same time. Given these restrictions, Geppetto wants to know how many different pizzas he can make. Help him answer this question. Two pizzas are considered different if there is an ingredient of index i that is on one pizza, but not on the other.

INPUT

The first line of input contains two integers N and M ($1 \leqslant N \leqslant 20$, $1 \leqslant M \leqslant 400$). Each of the following M lines contains two **different numbers** a i b, they represent the prohibition of mixing ingredients marked with a and b on the pizza. ($1 \leqslant a, b \leqslant N$). All pairs of ingredients are not necessarily distinct, some pair could occur multiple times.

OUTPUT

The first and only line of output must contain the number of different pizzas given the restrictions in the task.

SAMPLE TESTS

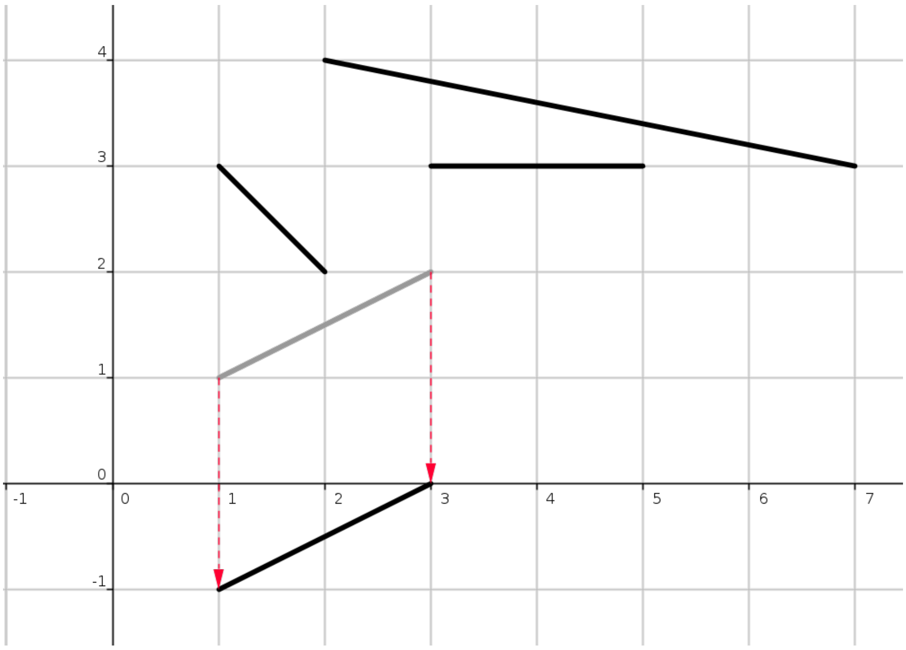
input	input	input
3 2 1 2 2 3	3 0	3 3 1 2 1 3 2 3
output	**output**	**output**
5	8	4

Clarification of the first example: Geppetto can make pizzas consisting of the following ingredients: , 1, 1, 3, 2, 3. Notice that a pizza can be without ingredients.
Clarification of the second example: Geppetto can make a pizza using any combination of the ingredients.
Clarification of the third example: Geppetto can make a pizza that either doesn't contain any ingredients or contains only one ingredient.

You have most definitely heard the legend of King Arthur and the Knights of the Round Table. Almost all versions of this story proudly point out that the **roundness** of the Round Table is closely related to Arthur's belief of equality among the Knights. That is a lie! In fact, Arthur's choice of table is conditioned by his childhood traumas.

In fact, Arthur was forced to clean up **quadratic** tables from a young age after a tournament in pick-up sticks[1] had been played on them. After the tournament, typically there would be a bunch of sticks on the table that do **not touch** each other. In the spirit of the game, the organizers issued strict regulations for the table cleaners. More precisely, the sticks on the table need to be removed **one by one** in a way that the cleaners pull them in the **shortest way** towards **the edge of the table closest** to where they are currently sitting. They also mustn't rotate or touch the other sticks while doing this (not even in the edge points).

In this task, we will represent the table in the coordinate system with a square that has opposite points in the coordinates (0,0) and (10 000, 10 000), whereas the sticks will be represented with straight line segments that lie within that square. We will assume that Arthur is sitting at the edge of the table lying on the x-axis. Then the movement of the stick comes down to **translating** the line segment along the shortest path towards the x-axis until the stick falls off the table (as shown in the image). It is your task to help Arthur determine the order of stick movements that meets the requirements from the previous paragraph.



INPUT

The first line of input contains the integer N ($1 \leqslant N \leqslant 5\,000$), the number of sticks on the table. Each of the following N lines contains four integers x_1, y_1, x_2, y_2 ($0 \leqslant x_1, y_1, x_2, y_2 \leqslant 10\,000$) that denote the edge points of a stick.

OUTPUT

The first and only line of output must contain space-separated stick labels in the order which they need to be taken off the table. A stick's label corresponds to its position in the input sequence.

If there are multiple possible solutions, output **any of them**.

[1] A game that involves carefully moving sticks.

SCORING

In test cases worth 40% of total points, it will hold $1 \leqslant N \leqslant 10$. In test cases worth 60% of total points, it will hold $1 \leqslant N \leqslant 300$.

SAMPLE TESTS

input	input	input
4 1 3 2 2 1 1 3 2 2 4 7 3 3 3 5 3	4 0 0 1 1 1 2 0 3 2 2 3 3 4 0 3 1	3 4 6 5 5 2 1 15 1 3 2 8 7
output	**output**	**output**
2 4 1 3	4 3 1 2	2 3 1

Clarification of the first example: The example corresponds to the image from the task. Another possible solution is 2 1 4 3.

There are eight planets and one planetoid in the Solar system. It is not a well known fact that there is a secret planet S4 inhabited by small creatures similar to bears, their codename being Lodas. Although this fact is well hidden from the public, the association Savez sent a team lead by general Henrik to study the Lodas. It has been discovered that Lodas have the ability of teleportation and he wants to hire them in his army.

One Lod consists of N strings where the ith string is denoted by x_i. Research has shown that the number of teleportations a Loda can make depends on one special **subsequence** (not necessarily consecutive) of these strings. Strings x_i and x_j ($i < j$) can both be in that sequence if and only if string x_j both **starts** with and **ends** with string x_i. The number of teleportations a Loda can make is the length of the longest described subsequence.

Determine the number of teleportations.

INPUT

The first line of input contains of the integer N, the number of strings. Each of the following N lines contains one string consisting of uppercase letters of the English alphabet. The input data will be such that there will be less than **two million** characters in total.

OUTPUT

The first and only line of output must contain the number of teleportations a Loda can make.

SCORING

In test cases worth 40% of total points, it will hold ($1 \leqslant N \leqslant 500$).

SAMPLE TESTS

input	input	input
5 A B AA BBB AAA	5 A ABA BBB ABABA AAAAAB	6 A B A B A B
output	**output**	**output**
3	3	3

Clarification of the first example: Prefix and suffix can intersect so subsequence is $A \rightarrow AA \rightarrow AAA$
Clarification of the third example: Strings in the subsequence are allowed to be equal so subsequence is $A \rightarrow A \rightarrow A$ or $B \rightarrow B \rightarrow B$

Young Mirko has been buying voodoo dolls lately. Considering that he is very interested in the cheapest purchase possible, he has been tracking the prices of voodoo dolls each day. His price list consists of doll prices in the last N days, where doll price a_i represents the price of a doll i days ago.

Mirko thinks he has noticed a connection between the average doll price in a sequence of **consecutive** days and the price on the following day. He wants to test his hunch and is puzzled by a very interesting question: "For a given P, how many different consecutive subsequences in the last N days are there, when the **average** doll price was greater than or equal to P?"

Two consecutive subsequences are considered different if their beginnings or ends are different.

INPUT

The first line of input contains the integer N, the sequence length ($1 \leqslant N \leqslant 1\,000\,000$).
The second line of input contains N prices a_i ($0 \leqslant a_i \leqslant 1\,000\,000\,000$).
The third line of input contains a number P. ($0 \leqslant P \leqslant 1\,000\,000\,000$).

OUTPUT

The first and only line of output must contain the answer to Mirko's question for a given P.

SCORING

In test cases worth 30% of points the sequence length N will be less than or equal to 10 000.

SAMPLE TESTS

input	input	input
3 1 2 3 3	3 1 3 2 2	3 1 3 2 3
output	**output**	**output**
1	5	1

Clarification of the first example: The only subsequence that has an average greater than or equal to 3 is {3}.
Clarification of the second example: The subsequences that have an average greater than or equal to 2 are {1, 3}, {1, 3, 2}, {3}, {3, 2}, {2}.

A distant country has N cities in it. The elections were just held in this country, so the new prime minister was elected. Currently, there is not a single road in this country, so the prime minister decided to modernize the country by connecting some cities with two-way highways and form counties. Two cities will be located in the **same** county if it is possible to get to one city from the other using the newly built roads. Each city will be located in **exactly one** county. Each conty consists of one or more cities.

The cities are represented as points in a two-dimensional coordinate system. The road between two cities is represented as a line segment connecting two points where the cities are located. The length of the road is equal to the length of the line segment in kilometers.

The country is currently suffering from recession, so the prime minister decided that, because of the lack of budget, they will not build roads **longer than** D kilometers. Additionally, the prime minister is happy about the small things, so he will be happy if, in at least one county, there exists a **nonempty subset** of cities (it can include all cities in the county) where the total sum of residents is divisible by K. For instance, if $K = 4$ and there is a county with cities that have 3, 5, 7 residents respectively, the prime minister will be happy because the sum of residents in the first two cities is equal to 8.

Help the prime minister in cutting the costs by determining the minimal D such that the prime minister can build roads and be happy about the small things at the same time.

INPUT

The first line of input contains the integers N and K ($1 \leqslant N \leqslant 50\,000, 1 \leqslant K \leqslant 30$). Each of the following N lines contains three integers $x_i, y_i, k_i (0 \leqslant x_i, y_i, k_i \leqslant 100\,000\,000)$, that represent the x coordinate of the city, the y coordinate and the number of residents in that city, respectively. There will not be two cities with the same coordinates in the input data. Additionally, there **will not** be a single city with the number of residents divisible by K.

OUTPUT

The first and only line of output must contain the minimal D such that it is possible to build roads with the condition that the prime minister is happy. Output D rounded to 3 decimal places. The input data will be such that there is always a solution.

SCORING

In test cases worth 40% of points the number of cities N will be less than or equal to 1000.

SAMPLE TESTS

input	input	input
3 3 0 4 4 1 5 1 2 6 1	5 11 0 0 1 0 1 2 1 0 3 1 1 4 5 5 1 20 20 10	6 5 20 20 9 0 0 3 0 1 1 10 0 1 10 1 6 12 0 3
output	**output**	**output**
1.414	5.657	2.000

Clarification of the first example: The only way to keep the prime minister happy is if all the cities are in the same county. The minimal D for which that is possible is 1.414.

Clarification of the second example: The prime minister will be happy if the first 5 cities are in the same county. If D = 5.657, the prime minister can connect cities 1, 2, 3, 5 with city 4. In that case, the sum of residents in cities 1, 2, 3, 4, 5 will be 11, which is divisible by 11, so the prime minister will be happy.