Problem A. Indian Summer

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

Indian summer is such a beautiful time of the year! A girl named Alyona is walking in the forest and picking a bouquet from fallen leaves. Alyona is very choosy — she doesn't take a leaf if it matches the color and the species of the tree of one of the leaves she already has. Find out how many leaves Alyona has picked.

Input

The first line contains an integer n ($1 \leq n \leq 100$) — the number of leaves Alyona has found. The next n lines contain the leaves' descriptions. Each leaf is characterized by the species of the tree it has fallen from and by the color. The species of the trees and colors are given in names, consisting of no more than 10 lowercase Latin letters. A name can not be an empty string. The species of a tree and the color are given in each line separated by a space.

Output

Output the single number — the number of Alyona's leaves.

Examples

stdin	stdout
5 birch yellow maple red birch yellow maple yellow maple green	4
3 oak yellow oak yellow oak yellow	1

Problem B. Cola

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

To celebrate the opening of the Winter Computer School the organizers decided to buy in n liters of cola. However, an unexpected difficulty occurred in the shop: it turned out that cola is sold in bottles 0.5, 1 and 2 liters in volume. At that, there are exactly a bottles 0.5 in volume, b one-liter bottles and c of two-liter ones. The organizers have enough money to buy any amount of cola. What did cause the heated arguments was how many bottles of every kind to buy, as this question is pivotal for the distribution of cola among the participants (and organizers as well).

Thus, while the organizers are having the argument, discussing different variants of buying cola, the Winter School can't start. Your task is to count the number of all the possible ways to buy exactly n liters of cola and persuade the organizers that this number is too large, and if they keep on arguing, then the Winter Computer School will have to be organized in summer.

All the bottles of cola are considered indistinguishable, i.e. two variants of buying are different from each other only if they differ in the number of bottles of at least one kind.

Input

The first line contains four integers — n, a, b, c ($1 \le n \le 10000$, $0 \le a, b, c \le 5000$).

Output

Print the unique number — the solution to the problem. If it is impossible to buy exactly n liters of cola, print 0.

Examples

stdin	stdout
10 5 5 5	9
3 0 0 2	0

Problem C. Holidays

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

School holidays come in Berland. The holidays are going to continue for n days. The students of school №N are having the time of their lives and the IT teacher Marina Sergeyevna, who has spent all the summer busy checking the BSE (Berland State Examination) results, has finally taken a vacation break! Some people are in charge of the daily watering of flowers in shifts according to the schedule. However when Marina Sergeyevna was making the schedule, she was so tired from work and so lost in dreams of the oncoming vacation that she perhaps made several mistakes. In fact, it is possible that according to the schedule, on some days during the holidays the flowers will not be watered or will be watered multiple times. Help Marina Sergeyevna to find a mistake.

Input

The first input line contains two numbers n and m ($1 \le n, m \le 100$) — the number of days in Berland holidays and the number of people in charge of the watering respectively. The next m lines contain the description of the duty schedule. Each line contains two integers a_i and b_i ($1 \le a_i \le b_i \le n$), meaning that the i-th person in charge should water the flowers from the a_i-th to the b_i-th day inclusively, once a day. The duty shifts are described sequentially, i.e. $b_i \le a_{i+1}$ for all i from 1 to $n - 1$ inclusively.

Output

Print "OK" (without quotes), if the schedule does not contain mistakes. Otherwise you have to find the minimal number of a day when the flowers will not be watered or will be watered multiple times, and output two integers — the day number and the number of times the flowers will be watered that day.

Examples

stdin	stdout
10 5 1 2 3 3 4 6 7 7 8 10	OK
10 5 1 2 2 3 4 5 7 8 9 10	2 2
10 5 1 2 3 3 5 7 7 7 7 10	4 0

Note

Keep in mind that in the second sample the mistake occurs not only on the second day, but also on the

sixth day, when nobody waters the flowers. However, you have to print the second day, i.e. the day with the minimal number.

Problem D. Hyperdrive

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

In a far away galaxy there are n inhabited planets, numbered with numbers from 1 to n. They are located at large distances from each other, that's why the communication between them was very difficult until on the planet number 1 a hyperdrive was invented. As soon as this significant event took place, $n - 1$ spaceships were built on the planet number 1, and those ships were sent to other planets to inform about the revolutionary invention.

Paradoxical thought it may be, but the hyperspace is represented as simple three-dimensional Euclidean space. The inhabited planets may be considered fixed points in it, and no two points coincide and no three points lie on the same straight line. The movement of a ship with a hyperdrive between two planets is performed along a straight line at the constant speed, the same for all the ships. That's why the distance in the hyperspace are measured in hyperyears (a ship with a hyperdrive covers a distance of s hyperyears in s years).

When the ship reaches an inhabited planet, the inhabitants of the planet dissemble it, make $n-2$ identical to it ships with a hyperdrive and send them to other $n - 2$ planets (except for the one from which the ship arrived). The time to make a new ship compared to the time in which they move from one planet to another is so small that it can be disregarded. New ships are absolutely identical to the ones sent initially: they move at the same constant speed along a straight line trajectory and, having reached a planet, perform the very same mission, i.e. are dissembled to build new $n - 2$ ships and send them to all the planets except for the one from which the ship arrived. Thus, the process of spreading the important news around the galaxy continues.

However the hyperdrive creators hurried to spread the news about their invention so much that they didn't study completely what goes on when two ships collide in the hyperspace. If two moving ships find themselves at one point, they provoke an explosion of colossal power, leading to the destruction of the galaxy!

Your task is to find the time the galaxy will continue to exist from the moment of the ships' launch from the first planet.

Input

The first line contains a number n ($3 \le n \le 5000$) — the number of inhabited planets in the galaxy. The next n lines contain integer coordinates of the planets in format "x_i y_i z_i" ($-10^4 \le x_i, y_i, z_i \le 10^4$).

Output

Print the single number — the solution to the task with an absolute or relative error not exceeding 10^{-6}.

Examples

stdin	stdout
4 0 0 0 0 0 1 0 1 0 1 0 0	1.7071067812

Problem E. Anfisa the Monkey

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

Anfisa the monkey learns to type. She is yet unfamiliar with the "space" key and can only type in lowercase Latin letters. Having typed for a fairly long line, Anfisa understood that it would be great to divide what she has written into k lines not shorter than a and not longer than b, for the text to resemble human speech more. Help Anfisa.

Input

The first line contains three integers k, a and b ($1 \le k \le 200$, $1 \le a \le b \le 200$). The second line contains a sequence of lowercase Latin letters — the text typed by Anfisa. It is guaranteed that the given line is not empty and its length does not exceed 200 symbols.

Output

Print k lines, each of which contains no less than a and no more than b symbols — Anfisa's text divided into lines. It is not allowed to perform any changes in the text, such as: deleting or adding symbols, changing their order, etc. If the solution is not unique, print any of them. If there is no solution, print "No solution" (without quotes).

Examples

stdin	stdout
3 2 5 abrakadabra	ab rakad abra
4 1 2 abrakadabra	No solution

Problem F. BerPaint

Input file:	`stdin`
Output file:	`stdout`
Time limit:	5 seconds
Memory limit:	256 megabytes

Anfisa the monkey got disappointed in word processors as they aren't good enough at reflecting all the range of her emotions, that's why she decided to switch to graphics editors. Having opened the BerPaint, she saw a white rectangle $W \times H$ in size which can be painted on. First Anfisa learnt to navigate the drawing tool which is used to paint segments and quickly painted on that rectangle a certain number of black-colored segments. The resulting picture didn't seem bright enough to Anfisa, that's why she turned her attention to the "fill" tool which is used to find a point on the rectangle to paint and choose a color, after which all the area which is the same color as the point it contains, is completely painted the chosen color. Having applied the fill several times, Anfisa expressed her emotions completely and stopped painting. Your task is by the information on the painted segments and applied fills to find out for every color the total area of the areas painted this color after all the fills.

Input

The first input line has two integers W and H ($3 \le W, H \le 10^4$) — the sizes of the initially white rectangular painting area. The second line contains integer n — the number of black segments ($0 \le n \le 100$). On the next n lines are described the segments themselves, each of which is given by coordinates of their endpoints x_1, y_1, x_2, y_2 ($0 < x_1, x_2 < W, 0 < y_1, y_2 < H$). All segments have non-zero length. The next line contains preset number of fills m ($0 \le m \le 100$). Each of the following m lines defines the fill operation in the form of "*x y color*", where (x, y) are the coordinates of the chosen point ($0 < x < W, 0 < y < H$), and *color* — a line of lowercase Latin letters from 1 to 15 symbols in length, determining the color. All coordinates given in the input are integers. Initially the rectangle is "white" in color, whereas the segments are drawn "black" in color.

Output

For every color present in the final picture print on the single line the name of the color and the total area of areas painted that color with an accuracy of 10^{-6}. Print the colors in any order.

Examples

stdin	stdout
4 5 6 1 1 1 3 1 3 3 3 3 3 3 1 3 1 1 1 1 3 3 1 1 1 3 3 2 2 1 red 2 2 blue	blue 0.00000000 white 20.00000000
5 5 5 1 1 2 2 2 2 4 2 4 2 4 4 4 4 2 4 2 4 2 2 2 3 3 black 3 3 green	green 4.00000000 white 21.00000000
7 4 9 1 2 2 3 2 3 3 2 3 2 2 1 2 1 1 2 3 2 4 2 4 2 5 3 5 3 6 2 6 2 5 1 5 1 4 2 2 2 2 black 2 2 red	red 2.00000000 white 26.00000000

Note

Initially the black segments painted by Anfisa can also be painted a color if any of the chosen points lays on the segment. The segments have areas equal to 0. That is why if in the final picture only parts of segments is painted some color, then the area, painted the color is equal to 0.

Problem G. Shooting Gallery

Input file:	stdin
Output file:	stdout
Time limit:	10 seconds
Memory limit:	256 megabytes

Berland amusement park shooting gallery is rightly acknowledged as one of the best in the world. Every day the country's best shooters master their skills there and the many visitors compete in clay pigeon shooting to win decent prizes. And the head of the park has recently decided to make an online version of the shooting gallery. During the elaboration process it turned out that the program that imitates the process of shooting effectively, is needed. To formulate the requirements to the program, the shooting gallery was formally described. A 3D Cartesian system of coordinates was introduced, where the X axis ran across the gallery floor along the line, along which the shooters are located, the Y axis ran vertically along the gallery wall and the positive direction of the Z axis matched the shooting direction. Let's call the XOY plane a shooting plane and let's assume that all the bullets are out of the muzzles at the points of this area and fly parallel to the Z axis. Every clay pigeon can be represented as a rectangle whose sides are parallel to X and Y axes, and it has a positive z-coordinate. The distance between a clay pigeon and the shooting plane is always different for every target. The bullet hits the target if it goes through the inner area or border of the rectangle corresponding to it. When the bullet hits the target, the target falls down vertically into the crawl-space of the shooting gallery and cannot be shot at any more. The targets are tough enough, that's why a bullet can not pierce a target all the way through and if a bullet hits a target it can't fly on. In input the simulator program is given the arrangement of all the targets and also of all the shots in the order of their appearance. The program should determine which target was hit by which shot. If you haven't guessed it yet, you are the one who is to write such a program.

Input

The first line contains an integer n ($1 \le n \le 10^5$) — the number of targets. Each of the subsequent n lines contains the description of a target. The target is described by five integers x_l, x_r, y_l, y_r, z, that determine it's location in space ($0 \le x_l < x_r \le 10^7, 0 \le y_l < y_r \le 10^7, 0 < z \le 10^7$). The next line contains an integer m ($1 \le m \le 10^5$), determining the number of shots. Then in m lines shots are described. Every shot is determined by the coordinates of a bullet on the shooting plane (x, y) ($0 \le x, y \le 10^7$, the coordinates of bullets are integers). The shots are given in the order of their firing. The intervals between shots are large enough, and a target falls very quickly, that's why assume that a falling target can not be an obstruction for all the shots following the one that hit it.

Output

For every shot in the single line print the number of the target which the shot has hit, or 0, if the bullet did not hit any target. The targets are numbered starting from 1 in the order in which they were given in the input data.

Examples

stdin	stdout
2 1 4 1 4 1 2 5 2 6 2 4 0 0 3 3 4 5 3 5	0 1 2 0

Problem H. Phone Number

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

Alas, finding one's true love is not easy. Masha has been unsuccessful in that yet. Her friend Dasha told Masha about a way to determine the phone number of one's Prince Charming through arithmancy.

The phone number is divined like that. First one needs to write down one's own phone numbers. For example, let's suppose that Masha's phone number is 12345. After that one should write her favorite digit from 0 to 9 under the first digit of her number. That will be the first digit of the needed number. For example, Masha's favorite digit is 9. The second digit is determined as a half sum of the second digit of Masha's number and the already written down first digit from her beloved one's number. In this case the arithmetic average equals to $(2+9)/2 = 5.5$. Masha can round the number up or down, depending on her wishes. For example, she chooses the digit 5. Having written down the resulting digit under the second digit of her number, Masha moves to finding the third digit in the same way, i.e. finding the half sum the the third digit of her number and the second digit of the new number. The result is $(5+3)/2 = 4$. In this case the answer is unique. Thus, every i-th digit is determined as an arithmetic average of the i-th digit of Masha's number and the $i-1$-th digit of her true love's number. If needed, the digit can be rounded up or down. For example, Masha can get:

12345

95444

Unfortunately, when Masha tried dialing the number, she got disappointed: as it turned out, the number was unavailable or outside the coverage area. But Masha won't give up. Perhaps, she rounded to a wrong digit or chose the first digit badly. That's why she keeps finding more and more new numbers and calling them. Count the number of numbers Masha calls. Masha calls all the possible numbers that can be found by the described means of arithmancy, except for, perhaps, her own one.

Input

The first line contains nonempty sequence consisting of digits from 0 to 9 — Masha's phone number. The sequence length does not exceed 50.

Output

Output the single number — the number of phone numbers Masha will dial.

Examples

stdin	stdout
12345	48
09	15

Problem I. Toys

Input file:	stdin
Output file:	stdout
Time limit:	5 seconds
Memory limit:	256 megabytes

Little Masha loves arranging her toys into piles on the floor. And she also hates it when somebody touches her toys. One day Masha arranged all her n toys into several piles and then her elder brother Sasha came and gathered all the piles into one. Having seen it, Masha got very upset and started crying. Sasha still can't calm Masha down and mom is going to come home soon and punish Sasha for having made Masha crying. That's why he decides to restore the piles' arrangement. However, he doesn't remember at all the way the toys used to lie. Of course, Masha remembers it, but she can't talk yet and can only help Sasha by shouting happily when he arranges the toys in the way they used to lie. That means that Sasha will have to arrange the toys in every possible way until Masha recognizes the needed arrangement. The relative position of the piles and toys in every pile is irrelevant, that's why the two ways of arranging the toys are considered different if can be found two such toys that when arranged in the first way lie in one and the same pile and do not if arranged in the second way. Sasha is looking for the fastest way of trying all the ways because mom will come soon. With every action Sasha can take a toy from any pile and move it to any other pile (as a result a new pile may appear or the old one may disappear). Sasha wants to find the sequence of actions as a result of which all the pile arrangement variants will be tried exactly one time each. Help Sasha. As we remember, initially all the toys are located in one pile.

Input

The first line contains an integer n ($1 \le n \le 10$) — the number of toys.

Output

In the first line print the number of different variants of arrangement of toys into piles. Then print all the ways of arranging toys into piles in the order in which Sasha should try them (i.e. every next way must result from the previous one through the operation described in the statement). Every way should be printed in the following format. In every pile the toys should be arranged in ascending order of the numbers. Then the piles should be sorted in ascending order of the numbers of the first toys there. Output every way on a single line. Cf. the example to specify the output data format. If the solution is not unique, output any of them.

Examples

stdin	stdout
3	5 {1,2,3} {1,2},{3} {1},{2,3} {1},{2},{3} {1,3},{2}

Problem J. Triminoes

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

There are many interesting tasks on domino tilings. For example, an interesting fact is known. Let us take a standard chessboard (8×8) and cut exactly two squares out of it. It turns out that the resulting board can always be tiled using dominoes 1×2, if the two cut out squares are of the same color, otherwise it is impossible.

Petya grew bored with dominoes, that's why he took a chessboard (not necessarily 8×8), cut some squares out of it and tries to tile it using triminoes. Triminoes are reactangles 1×3 (or 3×1, because triminoes can be rotated freely), also the two extreme squares of a trimino are necessarily white and the square in the middle is black. The triminoes are allowed to put on the chessboard so that their squares matched the colors of the uncut squares of the chessboard, and also the colors must match: the black squares must be matched with the black ones only and the white ones — with the white squares. The triminoes must not protrude above the chessboard or overlap each other. All the uncut squares of the board must be covered with triminoes.

Help Petya find out if it is possible to tile his board using triminos in the described way and print one of the variants of tiling.

Input

The first line contains two integers n and m ($1 \le n, m \le 1000$) — the board size. Next n lines contain m symbols each and represent the board description. If some position contains ".", then the square in this position has been cut out. Symbol "w" stands for a white square, "b" stands for a black square. It is guaranteed that through adding the cut squares can result in a correct chessboard (i.e. with alternating black and white squares), thought, perhaps, of a non-standard size.

Output

If at least one correct tiling exists, in the first line print "YES" (without quotes), and then — the tiling description. The description must contain n lines, m symbols in each. The cut out squares, as well as in the input data, are marked by ".". To denote triminoes symbols "a", "b", "c", "d" can be used, and all the three squares of each trimino must be denoted by the same symbol. If two triminoes share a side, than they must be denoted by different symbols. Two triminoes not sharing a common side can be denoted by one and the same symbol (c.f. sample).

If there are multiple correct ways of tiling, it is allowed to print any. If it is impossible to tile the board using triminoes or the correct tiling, for which four symbols "a", "b", "c", "d" would be enough, doesn't exist, print "NO" (without quotes) in the first line.

Examples

stdin	stdout
6 10 .w.wbw.wbw wbwbw.w.w. bw.wbwbwbw w.wbw.wbwb ...wbw.w.w ..wbw.wbw.	YES .a.aaa.ccc baccc.c.a. ba.dddcbab b.aaa.cbab ...bbb.b.b ..ccc.ddd.
2 2 wb bw	NO
1 3 wbw	YES bbb
1 3 ...	YES ...