1200. Horns and Hoofs

Time limit: 0.25 second
Memory limit: 64 MB

The famous venturer Ostap B. decided to organize a firm "Horns & Hoofs" which will produce horns and hoofs. First of all, Ostap has studied the market, the manufacturing process, and the local conditions.

The calculations showed that each horn would give a profit of A roubles and each hoof would bring B roubles. It must be taken into account that there are similar products in the market already and that horns and hoofs are in a sense interchangeable. Therefore the total amount of produced goods must not exceed K pieces each month, otherwise the excess will not be sold.

Besides, Ostap B. knows that the local racketeers fight monopolism and collect each month for each type of goods a "tax" which is equal (in roubles) to the square of the produced amount. For example, if the firm has produced two horns and three hoofs, then they must pay 4+9=13 roubles.

Having heard about the success of students of the Department of Mathematics and Mechanics in the All-Russian finals of business game Nixdorff Delta, Ostap appealed to the dean's office. He asked to calculate the optimal production volumes for his new firm. The dean is sure that his students will cope with this task.

Input

The first line contains real numbers A and B ($-10000 \le A,B \le 10000$) with a two fractional digits precision. These numbers are the profits (in roubles) given by each horn and by each hoof respectively.

The next line contains an integer K which is the maximal number of goods that could be sold each month ($1 \le K \le 10000$).

Output

You should output at the first line the maximal possible profit with a two fractional digits precision. The next line should contain the optimal production volumes.

If there are several possible answers, you should output the one with the least amount of horns, and if there is still a tie with the least amount of hoofs.

Sample

input	output
34.20 61.70 45	1239.50 16 29

Problem Author: Magaz Asanov
Problem Source: USU Internal Contest, March 2002

1201. Which Day Is It?

Time limit: 1.0 second
Memory limit: 64 MB

Sometimes it is of great importance to know which day of the week a given date will be. And we start searching for the nearest calendar. Being lucky we may find one. And find out that this one does not contain the date you need. What a pity!

Thus you are asked to create a calendar that will be able to process any given date in the years range from 1600 till 2400. Given a date, your program should print (see the examples below) a correct calendar for the month containing the date. Do not forget about the leap years. A year is considered to be leap if it is multiple of 4 except it is multiple of 100 except it is multiple of 400. For example 1996 is a leap year, 1900 is not a leap year (it is a multiple of 4 and multiple of 100) and 2000 is a leap year (it is a multiple of 4, multiple of 100 and multiple of 400 as well).

Input

The first line of input contains a date, i.e. three integer numbers: day (1–31), month (1–12) and year (1600–2400) separated by spaces.

Output

The output should contain exactly 7 lines with the correct calendar for the month containing the given date. Format of a calendar is given by the examples below (for a reading convenience spaces in output example are replaced with dots, real output should contain spaces instead). And do not forget to highlight the given date by square brackets.

Samples

input	output
16 3 2002	mon.....4...11...18...25 tue.....5...12...19...26 wed.....6...13...20...27 thu.....7...14...21...28 fri...1....8...15...22...29 sat...2....9..[16]..23...30 sun...3...10...17...24...31
1 3 2002	mon.....4...11...18...25 tue.....5...12...19...26 wed.....6...13...20...27 thu.....7...14...21...28 fri.[.1]...8...15...22...29 sat...2....9...16...23...30 sun...3...10...17...24...31

Problem Author: Alexander Klepinin
Problem Source: USU Internal Contest, March 2002

1202. Rectangles Travel

Time limit: 1.0 second
Memory limit: 64 MB

On an endless sheet of checked paper there are axes of coordinates. A unit for measurement in this coordinates system is the length of the square's edge. Also there are N rectangles on this sheet of paper. Their edges are parallel to the coordinate axis, and go through the borders between the squares. If we denote the coordinates of the lower left corner of the i-th rectangle with (x_i, y_i), and the coordinates of its upper right corner with (x^i, y^i), we will see, that:

$$x_1 = 0, y_1 = 0$$
$$x^i = x_{i+1}$$
$$2 \le x^i - x_i \le 100$$
$$2 \le y^i - y_i \le 100$$

If two rectangles with numbers i and $i + 1$ overlap, their common border disappears:

A traveler starts his way from the point with the coordinates (1, 1), which, as it follows from the rules above, certainly lays in the first rectangle. The traveler walks strictly along the edges of the squares. He is not allowed to walk on the borders of the rectangles. Thus, he can leave one rectangle for another only through their disappeared common border. There is an example of some beginning of his walk on the picture.

The traveler's goal is the point $(x^n - 1, y^n - 1)$, which is obviously situated inside the last rectangle.

Input

In the first line there is a positive integer n, $0 < n < 100000$ — the number of rectangles on the plane. Then n lines follow, each one of them containing four integer numbers x_i, y_i, x^i, y^i, separated with spaces, satisfying the above rules.

Output

You should output the length (the measurement unit is the edge of one square) of the shortest possible route for the traveler to go from the point (1, 1) to the point $(x^n - 1, y^n - 1)$, or the number −1, if there exists no such route (the latter possibility is realized, for instance, on the picture).

Sample

input	output
2 0 0 3 5 3 1 5 7	8

Problem Author: Leonid Volkov
Problem Source: USU Internal Contest, March 2002

1203. Scientific Conference

Time limit: 1.0 second
Memory limit: 64 MB

Functioning of a scientific conference is usually divided into several simultaneous sections. For example, there may be a section on parallel computing, a section on visualization, a section on data compression, and so on.

Obviously, simultaneous work of several sections is necessary in order to reduce the time for scientific program of the conference and to have more time for the banquet, tea-drinking, and informal discussions. However, it is possible that interesting reports are given simultaneously at different sections.

A participant has written out the time-table of all the reports which are interesting for him. He asks you to determine the maximal number of reports he will be able to attend.

Input

The first line contains the number $1 \le N \le 100000$ of interesting reports. Each of the next N lines contains two integers T_s and T_e separated with a space ($1 \le T_s < T_e \le 30000$). These numbers are the times a corresponding report starts and ends. Time is measured in minutes from the beginning of the conference.

Output

You should output the maximal number of reports which the participant can attend. The participant can attend no two reports simultaneously and any two reports he attends must be separated by at least one minute. For example, if a report ends at 15, the next report which can be attended must begin at 16 or later.

Sample

input	output
5 3 4 1 5 6 7 4 5 1 3	3

Problem Author: Magaz Asanov
Problem Source: USU Internal Contest, March 2002

1204. Idempotents

Time limit: 1.0 second
Memory limit: 64 MB

The number x is called an idempotent modulo n if

$$x*x = x \pmod{n}$$

Write the program to find all idempotents modulo n, where n is a product of two distinct primes p and q.

Input

First line contains the number k of test cases to consider ($1 \le k \le 1000$). Each of the following k lines contains one number $n < 10^9$.

Output

Write on the i-th line all idempotents of i-th test case in increasing order. Only nonnegative solutions bounded by n should be printed.

Sample

input	output
3 6 15 910186311	0 1 3 4 0 1 6 10 0 1 303395437 606790875

Problem Author: Pavel Atnashev
Problem Source: USU Internal Contest, March 2002

1205. By the Underground or by Foot?

Time limit: 1.0 second
Memory limit: 64 MB

Imagine yourself in a big city. You want to get from point A to point B. To do that you may move by foot or use the underground. Moving by the underground is faster but you may enter and exit it only at the stations. To save your time you decided to write a program to find the fastest route.

Input

The first line contains two floating point numbers. First of them is the speed of traveling by foot. The second one is the speed of traveling by the underground. Both speeds lie in limits from 0.5 to 10000. The second speed is always greater than or equal to the first one.

Then description of the underground follows. It starts with an integer N in the first line that is the number of the underground stations ($2 \le N \le 200$). The following N lines contain two floating point numbers each (i-th line contains the coordinates of i-th station). Then the description of the connections between stations follows. Each connection is determined by the pair of integers, i.e. by the numbers of connected stations. All connections in the list are different. There are no more than 400 connections. The list of connections is terminates with a pair of zeroes.

We assume that all underground connections are straight. So the time we need to travel between stations is equal to the distance between these stations divided by the speed of traveling by the underground. You may travel between connected stations to both directions. The time of traveling by foot between two points is equal to the distance between them divided by the speed of traveling by foot.

It should be mentioned also that entering and exiting the underground and changing trains are possible at the stations only and takes no time.

At last the coordinates of the points A and B are given, tha pair of coordinates in a line.

Output

The first line should contain the minimal time needed to get from the point A to the point B. Time should be given with the precision of 10^{-6}. The second line describes the use of the underground while traveling. It starts with the number of visited stations with tha following list of visited stations in the order they should be visited.

Sample

input	output
1 100 4 0 0 1 0 9 0 9 9 1 2 1 3 2 4 0 0 10 10 10 0	2.6346295 4 4 2 1 3

Problem Author: Alexander Klepinin
Problem Source: USU Internal Contest, March 2002

1206. Sum of Digits of the Sum of Numbers

Time limit: 1.0 second
Memory limit: 64 MB

Let us denote the sum of digits of the number N by S(N). In this problem, we want you to determine, how often the following equality holds:

$$S(A + B) = S(A) + S(B)$$

Input

The input contains a single integer K, $2 \le K \le 50$.

Output

You should output the number of pairs of K-digit numbers A and B to satisfy the above equality. Counting that number you should be aware of the following:

1. numbers A and B should not have leading zeroes;
2. while counting the total number of pairs, the order of numbers A and B matters, thus, for instance, (12, 26) and (26, 12) are different pairs that satisfy the conditions of the problem.

Sample

input	output
2	1980

Problem Author: Leonid Volkov
Problem Source: USU Internal Contest, March 2002

1207. Median on the Plane

Time limit: 0.5 second
Memory limit: 64 MB

The are N points on the plane (N is even). No three points lie on the same straight line. Your task is to select two points in such a way, that straight line they belong to divides the set of points into two equal-sized parts.

Input

First line contains one integer N ($4 \le N \le 10000$). Each of next N lines contains pair of integers x_i, y_i ($-10^6 \le x_i, y_i \le 10^6$), the coordinates of i-th point.

Output

Print the numbers of selected points.

Sample

input	output
4 0 0 1 0 0 1 1 1	1 4

Problem Author: Pavel Atnashev
Problem Source: USU Internal Contest, March 2002

1208. Legendary Teams Contest

Time limit: 0.5 second
Memory limit: 64 MB

Nothing makes as old as years. A lot of cool contests are gone, a lot of programmers are not students anymore and are not allowed to take part at the contests. Though their spirit is fresh and young as it was years ago! And so once they decided to make a contest at the Ural State University among the veteran teams…

To make the contest interesting, they decided to invite as much "legendary" teams as possible. The jury has made a short list of teams, which have shown the best results in the old good times, thus being worthy to hold the name of "legendary". All those teams were invited to take part of the contest, and all of them accepted the invitations. But they have forgotten one important thing at the jury: during the long history of the contests at the university, the teams happened to change and some programmers managed to contest in different "legendary" teams. Though, the jury decided not to give up the initial idea and to form as much legendary teams as possible to participate at the contest — and your program should help the jury!

Input

The first line contains a positive integer K, $1 \le K \le 18$. It is the number of all the legendary teams. There follow the descriptions of the teams in K lines. Each of those lines contains three different names of the team members of the respective team. All names are written with not more than 20 small Latin letters.

Output

You should output the maximal possible number of legendary teams of veterans, that could simultaneously participate at the contest.

Sample

input	output
7 gerostratos scorpio shamgshamg zaitsev silverberg cousteau zaitsev petersen shamgshamg clipper petersen shamgshamg clipper bakirelli vasiliadi silverberg atn dolly knuth dijkstra bellman	4

Problem Author: Leonid Volkov
Problem Source: USU Internal Contest, March 2002

1209. 1, 10, 100, 1000...

Time limit: 1.0 second
Memory limit: 64 MB

Let's consider an infinite sequence of digits constructed of ascending powers of 10 written one after another. Here is the beginning of the sequence: 110100100010000… You are to find out what digit is located at the definite position of the sequence.

Input

There is the only integer N in the first line ($1 \le N \le 65535$). The i-th of N left lines contains the integer K_i — the number of position in the sequence ($1 \le K_i \le 2^{31} - 1$).

Output

You are to output N digits 0 or 1 separated with a space. More precisely, the i-th digit of output is to be equal to the K_i-th digit of described above sequence.

Sample

input	output
4 3 14 7 6	0 0 1 0

Problem Author: Alexey Lakhtin
Problem Source: USU Open Collegiate Programming Contest October'2002 Junior Session

1210. Kind Spirits

Time limit: 1.0 second
Memory limit: 64 MB

Ivanushka the Fool lives at the planet of 0-level. It's very unpleasant to live there. An awful climate, 80 hours working week, ugly girls… He, as well as every inhabitant of his planet, dreams to get to a planet of *N*-th level. To the paradise.

At each of the *i*-th level planets there are several hyperspace transfers to some of the (*i*+1)-st level planets (but there are no reverse ways). Every transfer is guarded by a spirit. The spirits are usually evil: they demand many galactic bank-notes for each transfer. You know, everyone wants to go to a higher level planet. And one has to pay for the pleasure. More than Ivanushka can even imagine. However, extraordinary situations like a lack of a labor-force at one of the higher level planets sometimes happen, and then the spirits - the guards of the transfers — become kind. Sometimes they give galactic bank-notes themselves if only someone goes to their planets.

In order to embody his dream of heavenly planet Ivanushka has done two things. First of all, he has borrowed a complete map of the Universe. It's written on the map how much the spirits demand or give for a transfer from this or that planet to another one of the next higher level. Secondly, he has hired a staff of young talanted programmers in order that they will help him to draw the way on the map from his planet to the one of *N*th level so that he would spend for the spirits as little money or even earn as much as it is possible.

Input

The first line contains an integer N ($0 < N < 30$) — an amount of levels of the planets on Ivanushka's map. Then follow N blocks of information that describe interlevel transfers. More precisely, the *i*th informative block describes the scheme of transfers from (*i*−1)-st level planets to the ones of *i*th level. Those blocks are separated with a line that contains the only symbol "*". Planets of each level are numbered with sequential positive integers starting from 1. Each level contains not more than 30 planets. There is the only planet of 0-level: the one that Ivanushka lives at. The first line of a block contains a number K_i — an amount of planets of the *i*th level. THen follow K_i lines — one for each planet of the *i*th level. Every line consists of numbers of planets separated with a space of the previous (*i*−1)st level that one can get from them to the current planet, and the corresponding fees. A fee for each transfer is an integer number from −32768 to 32767; a negative fee means that the kind spirit is ready to pay for such a transfer. Each description line is ended by zero.

Output

should contain the only number — the minimal fee that Ivanushka might pay for a transfer to some planet of the *N*th level. The answer may be negative: it means that Ivanushka will not only get to a heavenly planet, but will earn some galactic bank-notes. It's known that there exists if only one way from Ivanushka's planet to the one of *N*th level.

Sample

input	output
3 2 1 15 0 1 5 0 * 3 1 -5 2 10 0 1 3 0 2 40 0 * 2 1 1 2 5 3 -5 0 2 -19 3 -20 0	-1

Problem Author: Leonid Volkov
Problem Source: USU Open Collegiate Programming Contest October'2002 Junior Session

1211. Collective Guarantee

Time limit: 2.0 second
Memory limit: 64 MB

Somebody of N boys and girls broke mummy's favourite cup. Mummy became angry and numbered the children with positive integers from 1 to N. Then she approached the child number 1 and asked: "Who has broken the cup?" "Me," — he (or she) answered and was punished.

You, of course, understand that the story is idealized. Practically (we don't know if it was true or not) the boy or the girl number 1 said: "It wasn't me! It was the child number K_1!" Then mummy approached number 2 and asked him (or her) the same question…

Some children tried to answer the truth. Others replied in order to say something. But some children had agreed not to give away a juvenile to mummy: each one of them pointed someone else from the group — in a circle. As a result — mummy was racked. She was despaired to remember what every child had told her about the cup and she wrote down all thier "evidences" on a sheet of paper. Now she's willing to investigate the cause. First of all she decided to find out if there is a "collective guarantee" between some of the children so that it was described above. You are to write a program that would help mummy in such a case — to all appearences it was not the first and not the lsat cup.

Input

The first lines contains a positive integer T ($1 \le T \le 16$) — the number of input tests. Each test consists of two lines: the first one contains a number N ($1 \le N \le 25000$) — and amount of children. The second line contains N numbers separated with a space — that are the evidences of the children. Mummy has written down at ith position of the line the number of a child that the ith child pointed to, or 0 if the ith child suddenly confessed.

Output

You should write one line for each test. It should contain "YES", if the evidences of the children at least seem to be noncontradictory: exactly one child has confessed the he (or she) had broken the cup, and there is no group of children that point to each other in a circle. Otherwise you are to output "NO".

Sample

input	output
4 4 2 0 2 2 4 2 0 2 1 5 2 3 4 1 3 3 0 3 2	YES YES NO NO

Problem Author: Leonid Volkov
Problem Source: USU Open Collegiate Programming Contest October'2002 Junior Session

1212. Battleship

Time limit: 1.0 second
Memory limit: 64 MB

Once two friends Petya and Vasya decided to play “Battleship” at the lesson of computer science at school. Finishing place his ships on the field Petya fell to thinking how many ways of placing his last K-deck ship exist. He tried to calculate it quickly but soon lost a count. Then Petya looked around and suddenly saw computers (there's no surprise: the children played at the lesson of computer science, but by the moment Patya was carried away by the game so much that he didn't notice the computers). He thaught a bit and decided to write a program that would solve his problem. But so far as he was backward (it wasn't the first time that he played “Battleship” during the lesson) he didn't succeed. Please, help Petya with his problem.

Input

The first line contains three numbers separated with a space — the vertical size of the playing field N ($1 \le N \le 30000$), the horizontal size of the field M ($1 \le M \le 30000$) and a number of already placed ships on the field L ($0 \le L \le 30$). Then follow L lines describing the ships location. Each description consists of three numbers and a letter separated with a space. The numbers are the coordinates of upper-left cell of a ship (the coordinates of upper-left cell of the playing field are (1,1)) and a number of ship decks. The letter defines the ship orientation (“V” — if it stands vertically, “H” — if horizontally). The last line contains the only positive integer K — the number of decks of the last ship that Petya wants to place.

We'll explain to those who has never played the “Battleship” that a i-deck ship is the rectangular of $i \times 1$ cells. Ships may have from one to four decks. According to the standard rules of the game, no two ships may contact each other neither by their edges nor by the vertices.

Output

You should output the only number — an amount of different ways of placing the Petya's last K-deck ship.

Sample

input	output
4 4 2 1 2 2 V 3 1 2 H 2	4

Problem Author: Anton Botov and Anatoly Uglov
Problem Source: USU Open Collegiate Programming Contest October'2002 Junior Session

1213. Cockroaches!

Time limit: 1.0 second
Memory limit: 64 MB

It's well-known that the most tenacious of life species on the Earth are cockroaches. They live everywhere if only there in food. And as far as they are unpretentious in food you can find them absolutely everywhere.

A little Lyosha studies at school on a Space station. During one of the school competitions his class has reached the final. A task of the final contest is to exterminate all the cockroaches in the cargo module within minimal time.

Within the long history of the competitions a unified tactics was worked out. The tactics is as follows: a poison gas is let in one of the module compartments and after that the baffle that separates the compartment from one of the adjacent ones is opened. Cockroaches can't stand the smell of the gas and run to the other compartment. When there's no cockroaches in the treated compartment the baffle is closed. Afterwards analogously the next compartment is treated, and so on. The goal is to move all the cockroaches to the floodgate of the cargo module. Then the outward door is opened and all the cockroaches are engulfed by an open Space.

Lyosha is responsible for programming the control board of the baffles in his team. The baffles are opened slowly, so it's very important to make do with minimal number of baffle openings in order to win in the contest. Your task is to help Lyosha to compute this number.

Input

The first line contains a name of the floodgate compartment. Each of the next lines contains description of one of the baffles — the names of two compartments separated with a dash (-). The last line contains the only symbol "#". There are cockroaches in all the compartments of the module at first. It's possible to get to the floodgate from every compartment of the module passing several baffles. The total number of compartments doesn't exceed 30. The name of a compartment consists of no more than 20 Latin letters and digits. The large and the small letters should be distinguished.

Output

Your program is to output the only number — the minimal amount of baffles that should be opened (and then closed) in order to move all the cockroaches to the floodgate.

Sample

input	output
Gateway Machinery-Gateway Machinery-Control Control-Central Control-Engine Central-Engine Storage-Gateway Storage-Waste Central-Waste #	6

Problem Author: Eugeny Krokhalev
Problem Source: USU Open Collegiate Programming Contest October'2002 Junior Session

1214. Strange Procedure

Time limit: 1.0 second
Memory limit: 64 MB

It's often happens in programming that one has to test and debug an existing program code. Imagine that your colleague has passed you fragment of program code of his because he is to work at another program.

Here follows the fragment of the program code — a procedure with two parameters:

Pascal

```
procedure P(x, y: integer);
var
    i, j: integer;
begin
    if (x > 0) and (y > 0) then
    begin
        for i := 1 to x + y do
        begin
            y := x * x + y;
            x := x * x + y;
            y := round(sqrt(x + (y / abs(y)) * (-abs(y))));
            for j := 1 to 2 * y do
                x := x - y;
        end;
    end;
    writeln(x, ' ', y);
end;
```

C++

```
void P(int x, int y)
{
    if (x > 0 && y > 0)
    {
        for (int i = 0; i < x + y; i++)
        {
            y = x * x + y;
            x = x * x + y;
            y = sqrt(x + y / abs(y) * -abs(y));
            for (int j = 0; j < 2 * y; j++)
                x -= y;
        }
    }
    printf("%d %d\n", x, y);
}
```

Python

```
def P(x: int, y: int):
    if x > 0 and y > 0:
        for i in range(x + y):
            y = x * x + y
            x = x * x + y
            y = math.floor(math.sqrt(x + y / abs(y) * -abs(y)))
            for j in range(2 * y):
                x -= y
    print(x, y)
```

Your task is unusual: in order to debug the procedure it's necessary to work out a program that would restore input parameters given output data from console. It's guaranteed that no variable has left its type during the processing of the procedure.

Input

The only line contains integers x and y, those are printed to console at the end of the procedure ($-32000 \le x, y \le 32000$).

Output

Output integers x and y that were given to the procedure as input parameters.

Sample

input	output
1 1	1 1

Problem Author: Anatoly Uglov
Problem Source: USU Open Collegiate Programming Contest October'2002 Junior Session

1215. Exactness of Projectile Hit

Time limit: 1.0 second
Memory limit: 64 MB

Inexactness of projectile hit may be compensated
by increasing of the projectile diameter.
Sergey Sizy

In the problem you are to determine the minimal diameter that may compensate inexactness of projectile hit in each concrete case. Assume that all the targets are convex polygons. A hit is the situation when the circle crater that the projectile leaves (the crater diameter equals to the one of the projectile) covers at least one point of the target.

Input

The first line contains three numbers — coordinates of the hit point of the projectile center and the number of polygon sides N ($3 \le N \le 100$). The next N lines contain the vertices coordinates in counter-clockwise order. All the coordinates are integers from [−2000,2000].

Output

You are to output the only number which is the minimal diameter of a projectile that will cover the target rounded with three digits after the decimal point.

Sample

input	output
2 -1 8 0 1 1 0 2 0 3 1 3 2 2 3 1 3 0 2	2.000

Problem Author: Anton Botov and Anatoly Uglov
Problem Source: USU Open Collegiate Programming Contest October'2002 Junior Session

1216. Two Pawns and One King

Time limit: 1.0 second
Memory limit: 64 MB

Somewhere on the $N \times N$ chessboard, there are one white pawn, one black pawn, and the black King as well. A game is being played according to the usual chess rules* (except the absence of the white King). White is supposed to win, when it manages to promote its pawn to the Queen (even in the case when this queen would be immediately beaten by the next black's move). Otherwise, Black is considered to be the winner (even if in some position White has no possibilities to move its pawn). Given an initial position, your program is to determine the winner.

* Usual chess rules mean exactly the following:

- White moves first.
- In the initial position, pawns are not on the first or the N-th horizontal row.
- White pawn is allowed to move to the fourth horizontal from the second, and black pawn is allowed to move to the (N−3)-rd horizontal row from the (N−1)-st. Pawns can't jump over other pieces.
- Just after the white pawn makes a move from $x2$ to $x4$, the black pawn may beat the white pawn *en passant*, moving from $y4$ to $x3$, if y is a vertical line, neighboring to the x-vertical.
- In all other cases, pawns move one field along the vertical and beat one field along the diagonal. King moves one field in any direction.
- If it is not Black's turn to move, the black King should not be under check, i.e. in the position, when the white pawn could beat it.
- When a black pawn reaches the first horizontal, it should be promoted to the Queen, Rook, Knight or Bishop, according to the wish of the Black player. (But when a white pawn reaches the N-th horizontal, it should be promoted to the Queen and our game immediately finishes).

Input

The first line contains a single integer N, $6 \le N \le 26$. In the second line, the positions of the white pawn, black pawn and black King (in this very order!) are given, separated by one or several white spaces. A description of a position consists of a small Latin letter, which denotes the vertical, and (without a space) an integer number (from 1 to N) denoting the horizontal.

Output

Output a message "WHITE WINS", when White can win the game according to the above described rules, and "BLACK WINS" otherwise.

Samples

input	output
10 h5 i5 b3	WHITE WINS
8 d5 h6 b6	BLACK WINS

Problem Author: Leonid Volkov
Problem Source: The Seventh Ural State University collegiate programming contest

1217. Unlucky Tickets

Time limit: 1.0 second
Memory limit: 64 MB

Strange people live in Moscow! Each time in the bus, getting a ticket with a 6-digit number, they try to sum up the first half of digits and the last half of digits. If these two sums are equal, they suppose such a ticket to be a lucky one. A person, who owns the lucky ticket, should dream about something, eat the ticket (no, it's not made of chocolate, it's made of paper!) and the dream will come true… At least, they believe it!

Strange people live in St.Petersburg! Each time in the bus, getting a ticket with a 6-digit number, they try to sum up the digits on the odd positions and the digits on the even positions. If these two sums are equal, they suppose such a ticket to be a lucky one. A person, who owns the lucky ticket, should dream about something, eat the ticket (no, even in St. Petersburg lucky tickets are not made of chocolate, they're made of paper!) and the dream will come true… At least, they believe it!

In the "third Russian capital" — Yekaterinburg — we laugh about such strange ideas. We are practical. We are not superstitious, even a little bit. But we understand that too much luck cannot be good. Thus we consider every ticket, which is lucky both in "Moscow sense" and "St. Petersburg sense" to be unlucky. If we get an unlucky ticket in the bus, we throw it away and leave the bus immediately! Two examples of unlucky tickets are 472175 and 810513.

You are to write a program, which calculates the total number of unlucky N-digit tickets.

Input

The input contains a single even positive integer N ($2 \le N \le 20$) — the number of digits in the ticket. Please note, that, for example 00742544 is a valid 8-digit ticket (by the way, it is a St.Petersburg-style lucky ticket).

Output

Your program should output a single integer number — the total number of unlucky N-digit tickets.

Sample

input	output
4	100

Problem Author: Leonid Volkov
Problem Source: The Seventh Ural State University collegiate programming contest

1218. Episode N-th: The Jedi Tournament

Time limit: 1.0 second
Memory limit: 64 MB

Decided several Jedi Knights to organize a tournament once. To know, accumulates who the largest amount of Force. Brought each Jedi his lightsaber with him to the tournament. Are different the lightsaber, and Jedi different are. Three parameters there are: length of the saber, Force of the Jedi and how good the Light side of the Force the Jedi can use. If in at least two parameters one Jedi than the other one stronger is, wins he. Is not possible a draw, because no Jedi any equal parameter may have. If looses a Jedi, must leave the tournament he.

To determine, which Jedi the tournament can win, your program is. Can win the tournament a Jedi, if at least one schedule for the tournament possible is, when the last one remains he on the tournament, not looses any match. For example, if Anakin stronger than Luke by some two parameters is, and Luke stronger than Yoda by some two parameters is, and Yoda stronger than Anakin, exists in this case a schedule for every Jedi to win the tournament.

Input

In the first line there is a positive integer $N \leq 200$, the total number of Jedi. After that follow N lines, each line containing the name of the Jedi and three parameters (length of the lightsaber, Force, Light side in this order) separated with a space. The parameters are different integers, not greater than 100000 by the absolute value. All names are sequences of not more than 30 small and capital letters.

Output

Your program is to output the names of those Jedi, which have a possibility to win the tournament. Each name of the possible winner should be written in a separate line. The order of the names in the output should correspond to the order of their appearance in the input data.

Sample

input	output
5 Solo 0 0 0 Anakin 20 18 30 Luke 40 12 25 Kenobi 15 3 2 Yoda 35 9 125	Anakin Luke Yoda

Problem Author: Leonid Volkov
Problem Source: The Seventh Ural State University collegiate programming contest

1219. Symbolic Sequence

Time limit: 1.0 second
Memory limit: 64 MB

Your program is to output a sequence of 1 000 000 lowercase Latin letters. This sequence should satisfy the following restrictions:

- Every letter occurs not more than 40 000 times in the sequence;
- Every possible subsequence with two letters length occurs not more than 2 000 times;
- Every possible subsequence with three letters length occurs not more than 100 times;

Input

For this problem no input is provided.

Output

In a single line of the output write some sequence, which satisfies the properties described above.

Problem Author: Pavel Atnashev, Leonid Volkov, text by Pavel Atnashev
Problem Source: The Seventh Ural State University collegiate programming contest

1220. Stacks

Time limit: 0.5 second
Memory limit: 0.75 MB
Language limit: C, C++, Pascal

Imagine, that you are employed by a software development company. You work now on the famous "D++ project", which is devoted to the creation of a new generation programming language. Your particular task is quite prosaic, though. You are to develop the memory manager being able to work with a large number of stacks.

Input

The first line of the input contains the total number of stack operations N, $0 < N \le 100000$. Each of the next N lines contains a description of a stack operation, either in the form PUSH A B (meaning to push B into stack A), or in the form POP A (meaning to pop an element from stack A), where A is the number of stack ($1 \le A \le 1000$), and B is an integer ($0 \le B \le 10^9$). You may assume, that every operation is correct (i.e., before each POP operation, the respective stack is not empty).

Output

For each POP operation, described in the input, output the value, which this POP operation gets from the top of that stack, to which it is applied. Numbers should appear according to the order of the POP operations in the input. Each number should be output in a separate line.

Sample

input	output
7 PUSH 1 100 PUSH 1 200 PUSH 2 300 PUSH 2 400 POP 2 POP 1 POP 2	400 200 300

Notes

In C++, it is recommended to use stdio instead of iostream to save a reasonable amount of memory.

Problem Author: Pavel Atnashev
Problem Source: The Seventh Ural State University collegiate programming contest

1221. Malevich Strikes Back!

Time limit: 1.0 second
Memory limit: 64 MB

After the greatest success of Malevich's "Black Square" the famous artist decided to create a new masterpiece. He took a large sheet of checked paper and filled some cells with black. After that he realized the picture to be too complicated. He was afraid, that people would not understand the sense of the painting. Thus, Malevich decided to cut out a smaller picture of the special form. It should be a black square with its sides parallel to the sides of the list. A white square rotated by 45 degrees should be placed inside the black square. The corners of the white square should lay on the sides of the black square. You can see an example of such picture on the figure.

The original paper size is $N \times N$, $0 < N \leq 100$. Your program should help Malevich to find the largest figure corresponding to the pattern described above.

Input

The input contains several test cases. Each test case starts with the size of paper N. The following N lines of the test case describe the original painting: "1" denotes a black and "0" denotes a white cell. End of the input is marked by a zero value for N.

Output

Your program should output the size (i.e. the maximum width or height) of the largest figure, which Malevich would like to cut out. If no such figure exists, output "No solution".

Sample

input	output
6 1 1 0 1 1 0 1 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0 1 1 1 0 1 1 1 1 1 4 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0	5 No solution

Problem Author: Nikita Shamgunov
Problem Source: The Seventh Ural State University collegiate programming contest

1222. Chernobyl' Eagles

Time limit: 1.0 second
Memory limit: 64 MB

A Chernobyl' eagle has several heads (for example, the eagle on the Russian National Emblem is a very typical one, having two heads; there exist Chernobyl' eagles having twenty-six, one and even zero heads). As all eagles, Chernobyl' eagles are very intelligent. Moreover, IQ of a Chernobyl' eagle is exactly equal to the number of its heads. These eagles can also enormously enlarge their IQ, when they form a group for a brainstorm. IQ of a group of Chernobyl' eagles equals to the product of IQ's of eagles in the group. So for example, the IQ of a group, consisting of two 4-headed eagles and one 7-headed is 4*4*7=112. The question is, how large can be an IQ of a group of eagles with a given total amount of heads.

Input

There is one positive integer N in the input, $N \leq 3000$ — the total number of heads of Chernobyl' eagles in a group.

Output

Your program should output a single number — a maximal IQ, which could have a group of Chernobyl' eagles, with the total amount of heads equal to N.

Sample

input	output
5	6

Problem Author: folklore, proposed by Leonid Volkov
Problem Source: The Seventh Ural State University collegiate programming contest

1223. Chernobyl' Eagle on a Roof

Time limit: 1.0 second
Memory limit: 64 MB

Once upon a time an Eagle made a nest on the roof of a very large building. Time went by and some eggs appeared in the nest. There was a sunny day, and Niels Bohr was walking on the roof. He suddenly said: "Oops! All eggs surely have the same solidity, thus there is such non-negative number E that if one drops an egg from the floor number E, it will not be broken (and so for all the floors below the E-th), but if one drops it from the floor number E+1, the egg will be broken (and the same for every floor higher, than the E-th)." Now Professor Bohr is going to organize a series of experiments (i.e. drops). The goal of the experiments is to determine the constant E. It is evident that number E may be found by dropping eggs sequentially floor by floor from the lowest one. But there are other strategies to find E for sure with much less amount of experiments. You are to find the least number of eggs droppings, which is sufficient to find number E for sure, even in the worst case. Note that dropped eggs that are not broken can be used again in following experiments.

The floors are numbered with positive integers starting from 1. If an egg has been broken being dropped from the first floor, you should consider that E is equal to zero. If an egg hasn't been broken even being dropped from the highest floor, consider that E is also determined and equal to the total number of floors.

Input

Input contains multiple (up to 1000) test cases. Each line is a test case. Each test case consists of two numbers separated with a space: the number of eggs, and the number of floors. Both numbers are positive and do not exceed 1000. Tests will end with the line containing two zeroes.

Output

For each test case output in a separate line the minimal number of experiments, which Niels Bohr will have to make even in the worst case.

Sample

input	output
1 10 2 5 0 0	10 3

Problem Author: folklore, proposed by Alexander Klepinin, text by Nikita Shamgunov
Problem Source: The Seventh Ural State University collegiate programming contest

1224. Spiral

Time limit: 1.0 second
Memory limit: 64 MB

A brand new sapper robot is able to neutralize mines in a rectangular region having integer height and width (N and M respectively). Before the robot begins its work it is placed near the top leftmost cell of the rectangle heading right. Then the robot starts moving and neutralizing mines making a clockwise spiral way (see picture). The spiral twists towards the inside of the region, covering all the cells. The region is considered safe when all the cells are visited and checked by the robot.

Your task is to determine the number of the turns the robot has to make during its work.

Input

The input contains two integers in the following order: N, M ($1 \le N, M \le 2^{31} - 1$).

Output

The output consists of a single integer value — the number of the turns.

Sample

input	output
3 5	4

Problem Source: 2002-2003 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk, October 2002

1225. Flags

Time limit: 1.0 second
Memory limit: 64 MB

On the Day of the Flag of Russia a shop-owner decided to decorate the show-window of his shop with textile stripes of white, blue and red colors. He wants to satisfy the following conditions:

1. Stripes of the same color cannot be placed next to each other.
2. A blue stripe must always be placed between a white and a red or between a red and a white one.

Determine the number of the ways to fulfill his wish.

Example. For $N = 3$ result is following:

Input

N, the number of the stripes, $1 \le N \le 45$.

Output

M, the number of the ways to decorate the shop-window.

Samples

input	output
1	2
2	2
3	4

Problem Source: 2002-2003 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk, October 2002

1226. esreveR redrO

Time limit: 1.0 second
Memory limit: 64 MB

This task is quite unconventional. We do not describe the way you must transform the input, which is a plain text containing Latin letters and other possible characters (numeric digits, punctuation marks etc.) We only say that the word being a subject to transformations is considered to be a sequence of Latin letters (both uppercase and lowercase) that is followed by the end of line, or the end of file, or a character which is not a Latin letter.

So, write a program to convert this "scrambled" text into a readable form.

Input

The input contains some ciphered text. The input contains no more than 1000 lines. The length of each line does not exceed 255 characters. The text may contain any printable characters.

Output

The output must contain the deciphered text from the input.

Sample

input
This is an example of a simple test. If you did not understand the ciphering algorithm yet, then write the letters of each word in the reverse order. By the way, "reversing" the text twice restores the original text.
output
sihT si na elpmaxe fo a elpmis tset. fI uoy did ton dnatsrednu eht gnirehpic mhtirogla tey, neht etirw eht srettel fo hcae drow ni eht esrever redro. yB eht yaw, "gnisrever" eht txet eciwt serotser eht lanigiro txet.

Problem Source: 2002-2003 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk, October 2002

1227. Rally Championship

Time limit: 1.0 second
Memory limit: 64 MB

A high-level international rally championship is about to be held. The rules of the race state that the race is held on ordinary roads and the route has a fixed length. You are given a map of the cities and two-way roads connecting it. To make the race safer it is held on one-way roads. The race may start and finish anyplace on the road. Determine if it is possible to make a route having a given length *S*.

Input

The first line of the input contains integers *M*, *N* and *S* that are the number of cities, the number of roads the length of the route ($1 \le M \le 100$; $1 \le N \le 10\,000$; $1 \le S \le 2 \cdot 10^6$).

The following *N* lines describe the roads as triples of integers: *P, Q, R*. Here *P* and *Q* are cities connected with a road, and *R* is the length of this road. All numbers satisfy the following restrictions: $1 \le P, Q \le M$; $1 \le R \le 32000$.

Output

Write YES to the output if it is possible to make a required route and NO otherwise. Note that answer must be written in capital Latin letters.

Samples

input	output
3 2 20 1 2 10 2 3 5	NO
3 3 1000 1 2 1 2 3 1 1 3 1	YES

Problem Source: 2002-2003 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk, October 2002

1228. Array

Time limit: 1.0 second
Memory limit: 64 MB

Imperative programming languages allow the use of both linear and multi-dimensional arrays. E.g. in Pascal for an array named X the expression **array**[0..2, 0..1, 0..3] declares a three-dimensional array having the following boundaries for each dimension: 0..2, 0..1, 0..3. (We consider only zero-based arrays here though other values are possible in Pascal for lower bounds of each dimension.)

It is always possible to determine the order in which the items of the array are enumerated. Consider that this order is determined by the principle "right-hand indices change faster". This means that the last (rightmost) index iterates through all possible values, then the index that is next to it (second from the right) changes its value by 1, and the last index iterates between the lower and upper boundaries again, and so on.

Example. The items of the array mentioned above are enumerated in the following order: X[0,0,0], X[0,0,1], X[0,0,2], X[0,0,3], X[0,1,0], X[0,1,1], X[0,1,2], X[0,1,3], X[1,0,0], X[1,0,1], X[1,0,2], X[1,0,3], X[1,1,0], X[1,1,1], X[1,1,2], X[1,1,3], X[2,0,0], X[2,0,1], X[2,0,2], X[2,0,3], X[2,1,0], X[2,1,1], X[2,1,2], X[2,1,3].

Let an n-ary array X is declared as **array**[0..k_1, 0..k_2, …, 0..k_n]. The theory says that the order P of any item $X[i_1, i_2, \ldots, i_n]$ is calculated as $P(i_1, i_2, \ldots, i_n) = 1 + D_1*i_1+D_2*i_2+\ldots D_n*i_n$, if we use the enumeration described above. Here $D_1, D_2, \ldots, D_n$ are so-called *index multipliers*.

Example. For the array in discussion the index multipliers are $D_1 = 8$, $D_2 = 4$, $D_3 = 1$. Then, for example, the order of X[1,0,3] will be $P(1,0,3) = 1+8*1+4*0+1*3 = 12$.

Your task is to calculate the unknown upper boundaries ($k_1, k_2, \ldots, k_n$) for given index multipliers $D_1, D_2, \ldots, D_n$ and total number of items s in the array.

Input

The first line of the input contains n — the number of dimensions ($1 \le n \le 20$) and s — the total number of items in the array ($1 \le s < 2^{31}-1$). The following n lines contain the index multipliers $D_1, D_2, \ldots, D_n$.

Output

Determine the upper boundaries for each dimension of the array in order: $k_1, k_2, \ldots, k_n$ ($0 < k_i \le 1000$). The numbers in the output may be delimited with spaces and/or line breaks.

Sample

input	output
3 24 8 4 1	2 1 3

Problem Source: 2002-2003 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk, October 2002

1229. Strong Brickwork

Time limit: 1.0 second
Memory limit: 64 MB

The builders must cover a rectangular area of size $M \times N$ (M and N are even numbers) with two layers of bricks that are rectangles of size 1×2. The first layer of the bricks has been already completed. The second layer (in an effort to make the brickwork really strong) must be done so, that no brick in it rests on a brick from the first layer.

Given the layout of the bricks in the first layer, determine the possible layout for the second one, or show that it is impossible to make the second layer.

Example. The two pictures show the layout of the two layers respectively. The size of the area is 2×4. Each brick is marked with its number on both halves.

Input

N, M — dimensions of the area. Then N lines follow having M numbers each that describe the layout of the first layer. Each brick is marked with two equal numbers written in the squares of the area that are covered by this brick. All bricks are marked with whole numbers ranging from 1 to the total number of the bricks. M and N are even numbers not exceeding 100.

Output

If there is no solution, then write −1. If the solution exists, write N lines with M numbers each that describe the layout of the second layer in the way shown above.

Sample

input	output
2 4 1 1 2 2 3 3 4 4	2 1 1 4 2 3 3 4

Problem Source: 2002-2003 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk, October 2002

1230. Introspective Program

Time limit: 1.0 second
Memory limit: 64 MB

There is probably no programmer that had not heard of a famous problem set by Norbert Wiener. The problem is to write a program that, being run, sends its source code to the output device. Note, that it is not allowed to use the files containing the source code, or system-dependent features (e.g. knowing the address of a memory area that could possibly contain the source code). Such a program is called an *introspective* one. A theorem is well known, which states that it is possible to write an introspective program in almost any programming language.

It would be unfair to demand writing an introspective program in a language known to a different degree by different participants. Therefore we describe a formal syntax for a new language called *PIBAS*. And your task is to write a program that writes an introspective *PIBAS* program to the output.

Testing this *PIBAS* program includes a syntax check and a test run. A program is considered correct if it outputs a string equal to its source code.

PIBAS description:

1. A *PIBAS* program consists of one or more operators separated with ";" (semicolon). A program is written in one line that is no longer than 32000 characters.
2. Two types of operators exist: string assignment operator and output operator.
3. Assignment operator is used as follows: <String variable>=<String expression>
4. String variable is denoted with a single capital Latin letter.
5. String expression is either a string variable, or a string constant, or a substring function, or a concatenation of string expressions with the use of "+" (plus) character.
6. String constant is a sequence of any printable characters enclosed in double (") or single (') quotes. This sequence cannot contain the quote character it is enclosed in. Examples: 'Rybinsk', "O key!", "I don't know solution."
7. A substring function is used in the following way: $(<string variable>,<unsigned integer>,<unsigned integer>). The second parameter sets the starting character of the substring and the third one sets its length. Character numeration is one-based.
8. Output operator: ?<string expression>.

The total length of all strings sent to output may not exceed 32000 characters.

Examples:

PIBAS program	Output
`?"Hello, "+'World!'`	`Hello, World!`
`A='World, Hello!';?$(A,8,5);?", ";B=$(A,1,5)+'!';?B`	`Hello, World!`

Input

Input is not used.

Output

The output must contain an introspective *PIBAS* program.

Problem Source: 2002-2003 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk, October 2002

1231. Turing: One, Two, Three, …

Time limit: 1.0 second
Memory limit: 64 MB

A Turing machine used for computability research is well known to computer scientists. We will give a brief description of this abstraction. Turing machine is an automatic device that works on a *tape* (1) of potentially unlimited length. The tape is divided into cells each containing a character. One of the cells is called a *viewed*, or *current*, one (2). At any point of time the Turing machine has a *condition* that is stored in the *control unit* (4). Besides, the *read/write* head (3) of the control unit is pointing to the current cell.

The control unit can execute one action per time interval (*step*). The action includes a state change, a possible change of the character in the current cell, and a possible movement to the adjacent cell. These actions are defined in a special table, called a *control table*. We will denote the movements along the tape with the following symbols: "<" — to the left, ">" — to the right, "=" — no movement.

The control table is actually a program for the Turing machine. The work of the Turing machine is considered to be done if no line in the control table contains the combination of the current character and condition.

Control table example:

Current condition	Current character	New condition	New character	Movement
1	-	2	-	>
2	-	3	+	>
3	#	4	#	<
4	+	4	+	<
4	-	5	-	=

Notice. This example only illustrates the definition of the table.

The input data for the Turing Machine are placed beforehand in the cells of the tape. The result is written to the same tape. Assume that the initial condition for the Turing machine is equal to 1 and the input data on the tape are bounded on both ends with '#' characters. (All tape cells except those that filled with minuses are filled with '#' character.) The control block is placed at the leftmost '-' character of the input data. The input tape contains '-' (minus) character repeated n times ($1 \le n \le 200$), and the input contains an integer k.

Imagine that the minuses are placed in circle. Starting with the first one each k-th uncrossed minus is crossed, i.e. it turns into a '+' (plus). The execution stops when there is only one uncrossed minus is left. Your task is to describe the control table for the Turing machine that will cross all minuses except one (it's position defined according to the above algorithm, but you may use any method) for the given k. For example, for $n = 10$ and $k = 3$ the fourth minus will remain uncrossed.

You may place the following characters on the tape: '+', '#', 'A'..'Z'. The cells initially filled with minuses may only contain '-' and '+' characters. After the execution the read/write head must point to the uncrossed minus. The number of the steps s must not exceed 1 000 000. The number of the line in the control table p must not exceed 10000. Tape size limited with 10001 cells (5000 both side from the initial read/write head position).

Input

The input contains an integer number k ($1 \le k \le 200$).

Output

The output describes the control table for the Turing machine for the given k. The first line of the output contains the number of rows p in the table ($1 < p < 10000$). Then p lines follow describing the table itself. Each row of the table contains five items: current condition (an integer number), current character (a character), new condition (an integer number), new character (a character), moving direction (a character). The items are separated with a single space characters. The condition numbers may range from 1 to 30000.

Sample

input	output
2	5 1 - 2 - > 2 - 3 + > 3 # 4 # < 4 + 4 + < 4 - 5 - =

Notes

Note, that this example is correct only for $n = 2$. It just shows output format.

Problem Source: 2002-2003 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk, October 2002

1232. Asteroid Landing

Time limit: 1.0 second
Memory limit: 64 MB

A guided probe is launched from the space station located at the distance of h from the surface of a large asteroid. The probe must land at the asteroid. The probe moves straight forward for a fixed distance d, after that it receives a new command from the station. The command defines the new direction for the movement. Each movement of the probe must help it to get closer to the surface. The control signals from the station are transmitted only within a cone having a vertex angle of α.

So, the trajectory of the probe is a broken line with segments of equal length, which is lying inside the cone described above. The last segment of the trajectory must also be of length d, lie inside the transmission cone and end at the surface of the asteroid.

Your task is to determine if it is possible to perform the landing of the probe taking into consideration the above conditions. If the landing is possible, then find the trajectory of minimal length including the coordinates of the ends of each segment. The landing point must be found, too.

The coordinates of the points are Cartesian. Ox and Oy lie on the surface of the asteroid, and Oz passes through the space station.

Input

h ($0.5000 \le h \le 99.9999$), d ($1.0000 \le d \le 5.0000$), α (the angle is in radians, $0.1000 \le \alpha \le 2.0000$). All numbers are float with no more than 4 digits after decimal point.

Output

n — the number of segments in the trajectory, or −1, if landing is impossible

$x_1\ y_1\ z_1$

$x_2\ y_2\ z_2$

…

$x_n\ y_n\ z_n$ — the coordinates of the points where the probe receives control signals, and the landing point. All coordinates must be calculated to within 0.0001.

Sample

input	output
11 5 2	3 0 3 7 3 3 3 3 −1 0

Problem Source: 2002-2003 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk, October 2002

1233. Amusing Numbers

Time limit: 1.0 second
Memory limit: 64 MB

Let us consider the set of integer numbers between 1 and N inclusive. Let us order them lexicographically (i. e. like in the vocabulary), for example, for N = 11 the order would be: 1, 10, 11, 2, 3, 4, 5, 6, 7, 8, 9.

Let us denote the position of the number K in this ordering as $Q_{N,K}$. For example, $Q_{11,2} = 4$. Given numbers K and M find the smallest N such that $Q_{N,K} = M$.

Input

Input contains two integer numbers K and M ($1 \le K, M \le 10^9$) separated by a space.

Output

If such N that $Q_{N,K} = M$ exists then write the smallest such N, otherwise write 0.

Samples

input	output
2 4	11
2 1	0
100000001 1000000000	100000000888888879
1000000000 11	0

Problem Author: Andrew Stankevich
Problem Source: 2002-2003 ACM Northeastern European Regional Programming Contest

1234. Bricks

Time limit: 1.0 second
Memory limit: 64 MB

The prisoner of the "IF" castle has decided to run away by disassembling the brick wall in his prison cell. To hide his work from his jailors he shall get rid of the bricks that he removes from the wall. All bricks have a shape of rectangular parallelepiped with the size of A × B × C inches and are so strong that they are impossible to break. However, there's a small rectangular sewer hole in the cell's floor with the size of D × E inches that goes deep down as a rectangular well of the same size (so deep it is, that its depth could not be measured and can be neglected). The prisoner have precisely (up to a tenth of an inch!) measured all the sizes A, B, C, D, E and wants to know if it is possible to dispose of the castle's bricks through the hole in the floor. Please, answer this question for him.

Input

The only line contains numbers A, B, C, D, and E separated by spaces. A, B, C are the lengths of brick's sides, and D, E are the lengths of hole's sides. All lengths are at least 1 and at most 10 inches and have at most 1 digit after decimal point.

Output

Write a single word `YES` if it is possible to dispose of the bricks through the hole or `NO` otherwise.

Samples

input	output
1.0 2.0 1.5 1.4 1.0	NO
1.0 2.0 1.5 1.5 1.0	YES

Problem Author: Elena Andreeva
Problem Source: 2002-2003 ACM Northeastern European Regional Programming Contest

1235. Cricket Field

Time limit: 1.0 second
Memory limit: 64 MB

Once upon a time there was a greedy King who ordered his chief Architect to build a field for royal cricket inside his park. The King was so greedy, that he would not listen to his Architect's proposals to build a field right in the park center with pleasant patterns of trees specially planted around and beautiful walks inside tree alleys for spectators. Instead, he ordered neither to cut nor to plant even a single tree in his park, but demanded to build the largest possible cricket field for his pleasure. If the Kind finds that the Architect has dared to touch even a single tree in his park or designed a smaller field that it was possible, then the Architect will loose his head. Moreover, he demanded his Architect to introduce at once a plan of the field with its exact location and size.

Your task is to help poor Architect to save his head, by writing a program that will find the maximum possible size of the cricket field and its location inside the park to satisfy King's requirements.

The task is somewhat simplified by the fact, that King's park has a rectangular shape and is situated on a flat ground. Moreover, park's borders are perfectly aligned with North-South and East-West lines. At the same time, royal cricket is always played on a square field that is also aligned with North-South and East-West lines. Architect has already established a Cartesian coordinate system and has precisely measured the coordinates of every tree. This coordinate system is, of course, aligned with North-South and East-West lines. Southwestern corner of the park has coordinates (0, 0) and Northeastern corner of the part has coordinates (W, H), where W and H are the park width and height in feet respectively.

For this task, you may neglect the diameter of the trees. Trees cannot be inside the cricket field, but may be situated on its side. The cricket field may also touch park's border, but shall not lie outside the park.

Input

The first line contains three integer numbers N, W, and H, separated by spaces. N ($0 \le N \le 100$) is the number of trees in the park. W and H ($1 \le W, H \le 10000$) are the park width and height in feet respectively.

Next N lines describe coordinates of trees in the park. Each line contains two integer numbers X_i and Y_i separated by a space ($0 \le X_i \le W$, $0 \le Y_i \le H$) that represent coordinates of i^{th} tree. All trees are located at different coordinates.

Output

Write a single line with three integer numbers P, Q, and L separated by spaces, where (P, Q) are coordinates of the cricket field Southwestern corner, and L is a length of its sides. If there are multiple possible field locations with a maximum size, then output any one.

Sample

input	output
7 10 7 3 2 4 2 7 0 7 3 4 5 2 4 1 7	4 3 4

Problem Author: Andrey Demidov
Problem Source: 2002-2003 ACM Northeastern European Regional Programming Contest

1236. Decoding Task

Time limit: 1.0 second
Memory limit: 64 MB

In the near future any research and publications about cryptography are outlawed throughout the world on the grounds of national security concerns. The reasoning for this is clear and widely accepted by all governments - if cryptography literature is public like in the old times, then everybody (even criminals and terrorists) could easily use it to hide their malicious plans from the national and international security forces. Consequently, public cryptographic algorithms and systems have ceased to exist, and everybody who needs strong protection for their secrets is forced to invent proprietary algorithms.

The ACM Corporation has lots of competitors who are eager to learn its trade secrets. Moreover, the job to protect their secrets is complicated by the fact, that they are forced to use intercontinental communication lines which are easy to eavesdrop on, unlike internal lines of the ACM Corporation which are well guarded. Therefore, the ACM Corporation have invented the Intercontinental Cryptographic Protection Code (ICPC) which they are very proud of, and which is considered unbreakable - nobody has even tried to break it yet, but that is about to change.

The group of hackers was hired by the rival company, which does not disclose its name to them, to break ICPC. As the first step, they have bribed one of the programmers who implemented the software for ICPC and have learned how ICPC works. It turns out, the ICPC uses very long key which is a sequence of bytes generated by some sophisticated and random physical process. This key is changed weekly and is used to encrypt all messages that are sent over intercontinental communication lines during the week. This programmer has also proudly told them, that ICPC is the fastest code in the world, because (having the benefit of highly sophisticated code generation) they simply perform bitwise exclusive OR (XOR) operation between the bytes of the message and the key. That is, the i^{th} byte of the encrypted message $E_i = K_i$ XOR C_i, where K_i is the i^{th} byte of the key and C_i is the i^{th} byte of the original clear-text message.

Having learned how ICPC works, they have started to look for the way to reliably obtain the key every week, which is the only thing that is still missing to listen for all intercontinental communications of the ACM Corporation (eavesdropping on the intercontinental lines themselves has indeed turned out to be an easy task). An attempt to bribe the security officers who guard and distribute the key has failed, because the security officers (having the profession with one the highest salaries of that time) have turned out to be too expensive to bribe.

During the search for alternative solutions, they have stumbled upon a clerk, who sends weekly newsletters to various employees and departments. Fortunately, these newsletters are being sent just after the change of the key and the messages are usually long enough to recover sufficient portions of the key by studying original newsletters and their encoded forms. However, they could not covertly find anyone who will disclose the newsletter contents on a weekly basis, because all the employees were bound by a Non-Disclosure Agreement (NDA) and the penalty for the disclosure of any corporate message according to this NDA is death.

Yet they were able to convince this clerk (for a small reward) to do a seemingly innocent thing. That is, while sending the copies of newsletter throughout the corporation, he was instructed to insert an extra space character in the beginning of some messages but send other copies in their original form. Now the task to recover the key is straightforward and it is you, who shall create a program for this. The program is given two ICPCed messages where the first message is N bytes, and the second one is N+1 bytes and is the result of encoding the same clear-text messages as the first one, but with one extra space character (represented by the byte with the decimal value of 32) in the beginning. The program shall find the first N+1 bytes of the key that was used to encode the messages.

Input

The input consists of two lines. The first line consists of 2N characters and represents the encoded message N bytes long. The second line consists of 2N+2 characters and represents the encoded message N+1 bytes long. Here $1 \le N \le 10000$. Each message is written on a single line in a hexadecimal form byte by byte without spaces. Each byte of the message is represented by two characters '0'-'9', 'A'-'F' that represent the hexadecimal value of the corresponding byte.

Output

Write a single line that represents N+1 bytes of the recovered key in the same hexadecimal format as in the input.

Sample

input	output
05262C5269143F314C2A69651A264B 610728413B63072C52222169720B425E	41434D2049435043204E454552432732

Problem Author: Elena Kruchkova
Problem Source: 2002-2003 ACM Northeastern European Regional Programming Contest

1237. Evacuation Plan

Time limit: 1.0 second
Memory limit: 64 MB

The City has a number of municipal buildings and a number of fallout shelters that were build specially to hide municipal workers in case of a nuclear war. Each fallout shelter has a limited capacity in terms of a number of people it can accommodate, and there's almost no excess capacity in The City's fallout shelters. Ideally, all workers from a given municipal building shall run to the nearest fallout shelter. However, this will lead to overcrowding of some fallout shelters, while others will be half-empty at the same time.

To address this problem, The City Council has developed a special evacuation plan. Instead of assigning every worker to a fallout shelter individually (which will be a huge amount of information to keep), they allocated fallout shelters to municipal buildings, listing the number of workers from every building that shall use a given fallout shelter, and left the task of individual assignments to the buildings' management. The plan takes into account a number of workers in every building - all of them are assigned to fallout shelters, and a limited capacity of each fallout shelter - every fallout shelter is assigned to no more workers then it can accommodate, though some fallout shelters may be not used completely.

The City Council claims that their evacuation plan is optimal, in the sense that it minimizes *the total time to reach fallout shelters* for all workers in The City, which is the sum for all workers of the time to go from the worker's municipal building to the fallout shelter assigned to this worker.

The City Mayor, well known for his constant confrontation with The City Council, does not buy their claim and hires you as an independent consultant to verify the evacuation plan. Your task is to either ensure that the evacuation plan is indeed optimal, or to prove otherwise by presenting another evacuation plan with the smaller total time to reach fallout shelters, thus clearly exposing The City Council's incompetence.

During initial requirements gathering phase of your project, you have found that The City is represented by a rectangular grid. The location of municipal buildings and fallout shelters is specified by two integer numbers and the time to go between municipal building at the location (X_i, Y_i) and the fallout shelter at the location (P_j, Q_j) is $D_{i,j} = |X_i - P_j| + |Y_i - Q_j| + 1$ minutes.

Input

The input consists of The City description and the evacuation plan description. The first line consists of two numbers N and M separated by a space. N ($1 \le N \le 100$) is a number of municipal buildings in The City (all municipal buildings are numbered from 1 to N). M ($1 \le M \le 100$) is a number of fallout shelters in The City (all fallout shelters are numbered from 1 to M).

The following N lines describe municipal buildings. Each line contains there integer numbers X_i, Y_i, and B_i separated by spaces, where X_i, Y_i ($-1000 \le X_i, Y_i \le 1000$) are the coordinates of the building, and B_i ($1 \le B_i \le 1000$) is the number of workers in this building.

The description of municipal buildings is followed by M lines that describe fallout shelters. Each line contains three integer numbers P_j, Q_j, and C_j separated by spaces, where P_i, Q_i ($-1000 \le P_j, Q_j \le 1000$) are the coordinates of the fallout shelter, and C_j ($1 \le C_j \le 1000$) is the capacity of this shelter.

The description of The City Council's evacuation plan follows on the next N lines. Each line represents an evacuation plan for a single building (in the order they are given in The City description). The evacuation plan of i^{th} municipal building consists of M integer numbers $E_{i,j}$ separated by spaces. $E_{i,j}$ ($0 \le E_{i,j} \le 1000$) is a number of workers that shall evacuate from the i^{th} municipal building to the j^{th} fallout shelter.

The plan is guaranteed to be *valid*. Namely, it calls for an evacuation of the exact number of workers that are actually working in any given municipal building according to The City description and does not exceed the capacity of any given fallout shelter.

Output

If The City Council's plan is optimal, then write the single word `OPTIMAL`. Otherwise, write the word `SUBOPTIMAL` on the first line, followed by N lines that describe your plan in the same format as in the input. Your plan need not be optimal itself, but must be valid and better than The City Council's one.

Samples

input	output

3 4 -3 3 5 -2 -2 6 2 2 5 -1 1 3 1 1 4 -2 -2 7 0 -1 3 3 1 1 0 0 0 6 0 0 3 0 2	SUBOPTIMAL 3 0 1 1 0 0 6 0 0 4 0 1
3 4 -3 3 5 -2 -2 6 2 2 5 -1 1 3 1 1 4 -2 -2 7 0 -1 3 3 0 1 1 0 0 6 0 0 4 0 1	OPTIMAL

Problem Author: Vladimir Kotov
Problem Source: 2002-2003 ACM Northeastern European Regional Programming Contest

1238. Folding

Time limit: 1.0 second
Memory limit: 64 MB

Bill is trying to compactly represent sequences of capital alphabetic characters from 'A' to 'Z' by folding repeating subsequences inside them. For example, one way to represent a sequence AAAAAAAAAABABABCCD is 10(A)2(BA)B2(C)D. He formally defines folded sequences of characters along with the unfolding transformation for them in the following way:

- A sequence that contains a single character from 'A' to 'Z' is considered to be a folded sequence. Unfolding of this sequence produces the same sequence of a single character itself.
- If S and Q are folded sequences, then SQ is also a folded sequence. If S unfolds to S' and Q unfolds to Q', then SQ unfolds to S'Q'.
- If S is a folded sequence, then X(S) is also a folded sequence, where X is a decimal representation of an integer number greater than 1. If S unfolds to S', then X(S) unfolds to S' repeated X times.

According to this definition it is easy to unfold any given folded sequence. However, Bill is much more interested in the reverse transformation. He wants to fold the given sequence in such a way that the resulting folded sequence contains the least possible number of characters.

Input

The input contains a single line of characters from 'A' to 'Z' with at least 1 and at most 100 characters.

Output

Write a single line that contains the shortest possible folded sequence that unfolds to the sequence that is given in the input. If there are many such sequences then write any one of them.

Samples

input	output
AAAAAAAAAABABABCCD	9(A)3(AB)CCD
NEERCYESYESYESNEERCYESYESYES	2(NEERC3(YES))

Problem Author: Roman Elizarov
Problem Source: 2002-2003 ACM Northeastern European Regional Programming Contest

1239. Ghost Busters

Time limit: 1.0 second
Memory limit: 64 MB

The famous Ghost Busters team has decided to upgrade their Ectomobile (aka Ecto-1) with a powerful proton gun and an advanced targeting system. Egon has designed and built all the hardware which consists of ectoplasmic scanner and a proton gun that has two degrees of freedom and can automatically rotate and fire in a 90 degrees trihedral angle. You have been hired to write a prototype for the targeting software.

Ghosts are detected by ectoplasmic scanner and are represented as floating spheres. The coordinates of their centers and radii are delivered from the ectoplasmic scanner to the targeting software. The coordinate system is aligned is such a way, that the proton gun fires from the point (0, 0, 0) anywhere into $X \geq 0$, $Y \geq 0$, $Z \geq 0$ trihedral angle. The gun fires a proton ray in a straight line and is so powerful, that even a touch of its ray is enough to kill a ghost. The ray of the proton gun is able to kill a virtually unlimited number of ghosts on its way.

For the first prototype for the targeting software, you are asked to write a program that determines the maximal number of ghosts that can be killed with a single shot of the proton gun.

Input

The first line contains a single integer N ($0 \leq N \leq 100$) - the number of ghosts detected by the ectoplasmic scanner of Ecto-1. The following N lines describe detected ghosts - one ghost per line. The description of i^{th} ghost (ghosts are numbered from 1 to N) consists of 4 integer numbers X_i, Y_i, Z_i, and R_i, separated by spaces. X_i, Y_i, Z_i ($1 \leq X_i, Y_i, Z_i \leq 10000$) are the coordinates of the ghost's center, and R_i ($1 \leq R_i \leq \min(X_i, Y_i, Z_i)$) is the ghost's radius. Because ghosts are ectoplasmic, they can be arbitrarily placed in respect to each others. They can intersect, fit inside each other, coincide with each other, etc.

Output

On the first line write a single integer number - the maximal number of ghosts that can be killed with a single shot of the proton gun. On the second line write the identifying numbers of the ghosts to be killed in an arbitrary order separated by spaces. If there are multiple ways to kill this number of ghosts then write any one.

Samples

input	output
2 1200 1200 3900 300 160 160 820 60	2 1 2
13 1200 1200 3900 300 160 160 820 60 100 10 10 10 10 100 10 10 10 10 100 10 10 10 10 10 50 50 50 10 100 100 75 20 100 75 100 20 75 100 100 20 3000 4000 7000 2600 100 1000 1000 50 1000 100 1000 100	5 6 7 9 10 11

Notes

It is possible to kill both ghosts in sample #1 only because the proton gun is so powerful that it kills ghosts by mere touch.
Sample #2 has 11 more ghosts (for a total of 13). Picture shows the view of these ghosts through the ectoplasmic scope sight that Egon has attached to proton gun to calibrate it and to debug your targeting software. Outlines show ghosts as seen through the scope.

Problem Author: Roman Elizarov
Problem Source: 2002-2003 ACM Northeastern European Regional Programming Contest

1240. Heroes of Might and Magic

Time limit: 1.0 second
Memory limit: 64 MB

In the new version of the famous game “Heroes of Might and Magic” heroes themselves take active part in battles. More of that, hero can defeat some monsters alone, without any supporting army. In this problem you are asked to develop the program which would find the strategy for a mage hero fighting face to face with a pack of monsters.

Each hero initially has HP_H hit points and MP_H mana points. Heroes can use different spells. Your hero knows three spells: Lighting Bolt, Teleport and Heal. Each spell costs one mana point.

Each monster has HP_M hit points. Pack of monsters is a single group of several monsters who act as one. Therefore if initially the pack consists of N_M monsters, they have $N_M \times HP_M$ hit points. As the battle proceeds, monsters' number of hit points decreases. If monsters have H hit points, that means that the group consists of ceiling(H / HP_M) monsters (ceiling is a function that returns the smallest integer number not less its argument).

The battle runs on a one-dimensional battlefield consisting of N + 1 squares, numbered starting from 0. Your hero resides on the square number 0 and does not move. Monsters initially reside on N^{th} square and can move. Monsters can move at most V squares a turn.

The battle consists of turns. First your hero makes a turn, then the monsters, and so on. Monsters' strategy is very easy - they move in the direction of your hero min(V, P - 1) squares where P is the square number where they were in the beginning of their turn. If the monsters are on the square number 1 in the end of the movement, then they strike your hero. If there are K monsters left in a pack, their strike decreases hit points of the hero by K. If your hero has non-positive hit points, then the hero is defeated.

Your hero's turn is always the casting of some spell. Lighting Bolt spell removes L_P hit points from a pack of monsters, where P is the square number on which the monsters reside. Teleport spell moves monsters to any desired square (except 0 where your hero resides). Heal spell adds dH hit points to hero. However, his hit points never exceed HP_H, so if after using Heal spell his hit points are greater then HP_H, they are decreased to HP_H. If your hero has zero mana points and there is at least one monster left in the pack, then the hero is defeated.

Find the strategy which would allow your hero to defeat monsters. Monsters are defeated if their hit points are non-positive.

Input

The first line contains positive integers separated by spaces in the following order: N, HP_H, MP_H, HP_M, N_M, V, dH. ($1 \le N \le 10$, $2 \le HP_H \le 100$, $1 \le MP_H \le 50$, $1 \le HP_M \le 10$, $1 \le N_M \le 10$, $1 \le V \le N$, $1 \le dH < HP_H$). The second line contains N integer numbers L_1, L_2, …, L_N ($1 \le L_P \le 10$), separated by spaces.

Output

If the hero cannot win the battle, write the word `DEFEATED`. In the other case write the word `VICTORIOUS` on the first line and then write any sequence of hero's actions that leads to victory, where each line starting from the second one must correspond to one hero's turn. The first character of the line must be one of the following:

- `L` - Cast Lighting Bolt spell.
- `T` - Cast Teleport spell.
- `H` - Cast Heal spell.

If the hero casts Teleport spell then `T` character must be followed by a space and an integer number from 1 to N - the square number where the monsters should be teleported to.

Samples

input	output
2 3 2 1 2 1 1 1 1	VICTORIOUS L L
2 3 2 3 1 1 1 1 1	DEFEATED
4 4 3 1 4 1 1 3 1 1 1	VICTORIOUS L T 2 L
1 6 5 1 4 1 3 1	VICTORIOUS L L H L L

Problem Author: Andrew Stankevich
Problem Source: 2002-2003 ACM Northeastern European Regional Programming Contest

1241. Inlay Cutters

Time limit: 1.0 second
Memory limit: 64 MB

The factory cuts rectangular M × N granite plates into pieces using a special machine that is able to perform cuts in 4 different directions: vertically, horizontally, and diagonally at the angle of 45 degrees to the sides of the plate. Every cut is a straight line that starts and ends on the side of the plate.

The factory has been ordered to produce tiles for the inlay, each tile of which is a 45 degrees right triangle. To reduce the time to deliver the tiles it was decided to take all triangles from the already cut plates. Information about all performed cuts is available and your task is to compute the number of triangles of any size that were produced.

Input

The input describes the cuts that were performed on a single rectangular plate. The first line contains three integers M, N, and K, separated by spaces. M and N ($1 \le M, N \le 50$) are the dimensions of the plate, and K ($0 \le K \le 296$) is the number of cuts. Next K lines describe the cuts. i^{th} cut is described by four integer numbers $X_{i,1}$, $Y_{i,1}$, $X_{i,2}$, and $Y_{i,2}$, separated by spaces, that represent the starting and ending point of the cut. Both starting ($X_{i,1}$, $Y_{i,1}$) and ending ($X_{i,2}$, $Y_{i,2}$) points of the cut are situated on the plate's border. Both points of the cut are different and the cut goes through the plate. Here, the coordinates by the X axis run from 0 to M, and the coordinates by the Y axis run from 0 to N. All cuts are different.

Output

Write the number of triangles that were produced by the cuts.

Samples

input	output
4 4 4 1 4 1 0 0 4 4 0 0 0 4 4 0 2 4 2	6
7 4 6 6 0 7 1 1 4 1 0 0 4 4 0 0 0 4 4 0 2 7 2 7 0 3 4	8

Problem Author: Elena Kruchkova
Problem Source: 2002-2003 ACM Northeastern European Regional Programming Contest

1242. Werewolf

Time limit: 1.0 second
Memory limit: 64 MB

Knife. Moonlit night. Rotten stump with a short black-handled knife in it. Those who know will understand. Disaster in the village. Werewolf.

There are no so many residents in the village. Many of them are each other's relatives. Only this may help to find the werewolf. The werewolf is merciless, but his descendants never become his victims. The werewolf can drown the village in blood, but he never kills his ancestors.

It is known about all the villagers who is the child of whom. Also, the sad list of the werewolf's victims is known. Your program should help to determine the suspects. It would be a hard task, if a very special condition would not hold. Namely, citizens of the village are not used to leave it. If some ancestor of some citizen lives in the village, then also his immediate ancestor does. It means, that, for example, if the father of the mother of some citizen still lives in the village, than also his mother still lives.

Input

The first line contains an integer N, $1 < N \leq 1000$, which is the number of the villagers. The villagers are assigned numbers from 1 to N. Further is the description of the relation "child-parent": a sequence of lines, each of which contains two numbers separated with a space; the first number in each line is the number of a child and the second number is the number of the child's parent. The data is correct: for each of the residents there are no more than two parents, and there are no cycles. The list is followed by the word "BLOOD" written with capital letters in a separate line. After this word there is the list of the werewolf's victims, one number in each line.

Output

The output should contain the numbers of the residents who may be the werewolf. The numbers must be in the ascending order and separated with a space. If there are no suspects, the output should contain the only number 0.

Samples

input	output
8 1 3 3 6 4 5 6 2 4 6 8 1 BLOOD 3 8	4 5 7
6 1 2 3 2 1 4 3 4 2 6 5 2 5 4 BLOOD 2 5	0

Problem Author: Leonid Volkov
Problem Source: Ural State University Personal Programming Contest, March 1, 2003

1243. Divorce of the Seven Dwarfs

Time limit: 1.0 second
Memory limit: 64 MB

After the Snow White with her bridegroom had left the house of the seven dwarfs, their peaceful and prosperous life has come to an end. Each dwarf blames others to be the reason of the Snow White's leave. To stop everlasting quarrels, the dwarfs decided to part. According to an ancient law, their common possessions should be divided in the most fair way, which means that all the dwarfs should get equal parts. Everything that the dwarfs cannot divide in a fair way they give to the Snow White. For example, after dividing 26 old boots, each dwarf got 3 old boots, and the Snow White got the remaining 5 old boots. Some of the numbers are very large, for example, the dwarfs have 123456123456 poppy seeds, so it is not easy to calculate that the Snow White gets only one seed. To speed up the divorce, help the dwarfs to determine quickly the Snow White's part.

Input

The only line contains an integer N that represents the number of similar items that the dwarfs want to divide ($1 \le N \le 10^{50}$).

Output

You should output the number of items that pass into the possession of the Snow White.

Sample

input	output
123456123456	1

Problem Author: Stanislav Vasilyev
Problem Source: Ural State University Personal Programming Contest, March 1, 2003

1244. Gentlemen

Time limit: 0.5 second
Memory limit: 64 MB

Let's remember one old joke:

Once a gentleman said to another gentleman:
— What if we play cards?
— You know, I haven't played cards for ten years…
— And I haven't played for fifteen years…
So, little by little, they decided to resurrect their youth. The first gentleman asked a servant to bring a pack of cards, and before starting playing out weighed in his hand the pack.
— It seems to me, one card is missing from the pack… — he said and gave the pack to the other gentleman.
— Yes, the nine of spades, — the man agreed.

An incomplete pack of cards is given. The program should determine which cards are missing.

Input

The first line contains a positive integer, which is the weight in milligrams of the given incomplete pack. The second line contains an integer N, $2 \le N \le 100$ — the number of cards in the complete pack. In the next N lines there are integers from 1 to 1000, which are the weights of the cards in milligrams. It's guaranteed that the total weight of all cards in the complete pack is strictly greater than the weight of the incomplete pack.

Output

If there is no solution, then output the single number 0. If there are more than one solutions, then you should write −1. Finally, if it is possible to determine unambiguously which cards are missing in the incomplete pack as compared to the complete one, then output the numbers of the missing cards separated with a space in ascending order.

Samples

input	output
270 4 100 110 170 200	2 4
270 4 100 110 160 170	-1
270 4 100 120 160 180	0

Problem Author: Alexander Petrov
Problem Source: Ural State University Personal Programming Contest, March 1, 2003

1245. Pictures

Time limit: 1.0 second
Memory limit: 64 MB

Artist Ivanov (not the famous Ivanov who painted "Christ's apparition to people", but one of the many namesakes) once managed to rent inexpensively an excellent studio. Alas, as he soon discovered, the inexpensiveness was caused by objective reasons. A murder happened long ago in the house where he rented the room, and now the ghost living in the house each night renews blood spots on the walls of all the rooms. Ivanov's studio did not escape this damnation.

Nevertheless, being a creative person, Ivanov quickly found a simple solution to the problem. He decided to paint one or two pictures and hang them on the (single) wall where the spots appear each night so that the spots would be covered by the pictures. Of course, he does not want to spend too much time doing this work. That is why he plans to use not more than two pictures and wants the total area of the pictures to be minimal.

All the blood spots are circles. Each picture has a rectangular form with sides parallel to the axes, and the minimally possible size of a picture in each of the dimensions is 100 millimeters. If it is necessary to paint two pictures, then they should be hanged to the wall without overlaying. Each spot must be covered by exactly one picture.

Input

The first line contains the number of the spots N, $0 < N \le 1000$. Each of the next N lines contains the description of the corresponding spot. A spot is described by three positive integers; they are the radius of the spot and the Cartesian coordinates of the center of the spot. Everything is measured in millimeters and all these numbers do not exceed 10000.

Output

Output the minimal total area (in square millimeters) of the pictures (not more than two) necessary to cover all the spots.

Sample

input	output
3 50 50 50 50 250 50 10 150 250	40000

Problem Author: Alexander Petrov (text — Leonid Volkov)
Problem Source: Ural State University Personal Programming Contest, March 1, 2003

1246. Tethered Dog

Time limit: 1.0 second
Memory limit: 64 MB

A dog is tethered to a pole with a rope. The pole is located inside a fenced polygon (not necessarily convex) with nonzero area. The fence has no self-crosses. The Olympian runs along the fence bypassing the vertices of the polygon in a certain order which is not broken during the jog. A dog pursues him inside the fenced territory and barks. Your program is to determine how (clockwise or counter-clockwise) the rope will wind after several rounds of the Olympian's jog.

Input

The first input line contains a number N that is the number of the polygon vertices. It's known that $3 \le N \le 200000$. The next N lines consist of the vertices plane coordinates, given in an order of Olympian's dog. The coordinates are a pair of integers separated with a space. The absolute value of each coordinate doesn't exceed 50000.

Output

You are to output "cw", if the rope is winded in a clockwise order and "ccw" otherwise.

Sample

input	output
4 0 0 0 1 1 1 1 0	cw

Problem Author: Evgeny Kobzev
Problem Source: Ural State University Personal Programming Contest, March 1, 2003

1247. Check a Sequence

Time limit: 0.5 second
Memory limit: 64 MB

There is a sequence of integer numbers $A_1, A_2, \ldots, A_S$, and a positive integer N. It's known that all elements of the sequence $\{A_i\}$ satisfy the restriction $0 \le A_i \le 100$. Moreover, it's known that the sum of all elements of the sequence is equal to $S + N$. You are to write a program that given a sequence $\{A_i\}$ and a number N will answer the question: is it true that for all $1 \le i \le j \le S$ the following inequality holds:

$$A_i + A_{i+1} + \ldots + A_j \le (j - i + 1) + N\,?$$

Input

The first input line contains two separated with a space positive numbers S and N that do not exceed 30000. Then follow S lines with one number in a line that are elements of the sequence $\{A_i\}$.

Output

Output "YES", if the mentioned above inequality holds for all the values of the parameters i and j, and "NO" otherwise.

Samples

input	output
4 3 2 3 0 2	YES
4 5 1 0 5 3	NO

Problem Author: Alexander Mironenko
Problem Source: Ural State University Personal Programming Contest, March 1, 2003

1248. Sequence Sum

Time limit: 1.0 second
Memory limit: 64 MB

You are given a sequence of real numbers. And you ought to sum them all. And don't forget to output the result.

Input

The first input line contains a number of the sequence elements N ($N \leq 1000$). The next N lines contain the elements in exponential notation with 19 significant digits in mantissa. All the numbers are in a range from 10^{-100} to 10^{100}.

Output

You are to output the only number which is the sum of all the input numbers. This number is to be presented in an exponential notation with 19 correct digits in mantissa.

Sample

input	output
2 1.5e3 1.5e-3	1.5000150000000000e3

Problem Author: Alexander Klepinin
Problem Source: Ural State University Personal Programming Contest, March 1, 2003

1249. Ancient Necropolis

Time limit: 2.5 second
Memory limit: 4 MB

Aerophotography data provide a bitmap picture of a hard-to-reach region. According to the suggestions of scientists, this region is a cemetery of an extinct civilization. Indeed, the picture, having been converted to a binary form, shows distinctly visible areas, dark (marked with symbols 1) and light (marked with 0). It seems that the dark areas are tombstones. It's easy to either confirm or reject the hypothesis since the race that lived in the region knew astronomy, so tombstones were always oriented along the Earth's parallels and meridians. That is why the dark areas in the picture should have the form of rectangles with the sides parallel to the axes. If it is so, then we indeed have a picture of a cemetery of an extinct race. Otherwise, new hypotheses should be suggested.

Input

The first input line contains two integers *N* and *M*, which are the dimensions of the picture provided by the aerophotography. Each of the next *N* lines contains *M* zeros or ones separated with a space. The numbers *N* and *M* do not exceed 3000.

Output

Output "Yes" if all connected dark areas in the picture are rectangles and "No" otherwise.

Samples

input	output
2 2 0 1 1 1	No
3 3 0 0 1 1 1 0 1 1 0	Yes

Problem Author: Nikita Shamgunov and Leonid Volkov
Problem Source: Open collegiate programming contest for student teams, Ural State University, March 15, 2003

1250. Sea Burial

Time limit: 1.0 second
Memory limit: 64 MB

There is Archipelago in the middle of a shoreless ocean. An ancient tribe of cannibals lives there. Shamans of this race have been communicating with gods and admonishing people for ages. They could generate a rain during a drought and clear the sky in a raining season. A long time ago the first shaman of the tribe jumped into one of the seas and drowned while being in a sacred trance. Since then all the land inside this sea is regarded as sacred. According to an ancient law, all shamans must be buried on a sacred land. However, souls of dead shamans cannot get on with each other, so each shaman must be buried on a separate island. An old prophecy says that if two shamans are buried on the same land, then a dreadful time will come and the tribe will perish.

How many shamans will the tribe outlive? This problem bothered all the chiefs of the tribe who were coming into power. So one of the chiefs ordered to compile a map of Archipelago. The cannibals toiled for a whole year and coped with the task. But the map was too large and complicated to count all the sacred islands. So the tribe's shaman appealed to gods and asked them to help with counting the islands. And the tribe was blessed with a Programmer and a Computer, which came down to earth in a cloud of light and fire. Yes, you are this Programmer, and you are destined to live with these cannibals until you have counted the islands; then you'll be transferred back home. Remember that there may be seas inside islands, and islands inside those seas.

Input

The first input line contains four integers W, H, X and Y, separated with one or several spaces. $1 \le W, H \le 500$ are respectively the width and the height of the map. $1 \le X \le W$ and $1 \le Y \le H$ are the coordinates of the point where the first shaman drowned. The next H lines contain description of the map. Each line contains W symbols describing correspondent fragments of the map. Symbol "." stands for a sea fragment and symbol "#" stands for a land fragment. Two fragments belong to the same sea if they are adjacent horizontally, vertically, or diagonally. Two fragments belong to the same island if they are adjacent horizontally or vertically (but not diagonally). Land fragments that are adjacent to the map's border are not considered as islands. Coordinates are counted from the left upper corner.

Output

The output should contain a single integer, which is the number of the islands inside the sacred sea.

Samples

input	output
9 7 1 1 .....# .#######. .#.....#. .#.#.#.#. .#.....#. .#######. #.....	3
9 7 3 3 .....# .#######. .#.....#. .#.#.#.#. .#.....#. .#######. #.....	2

Problem Author: Stanislav Skorb (prepared by Ivan Dashkevich)
Problem Source: Open collegiate programming contest for student teams, Ural State University, March 15, 2003

1251. Cemetery Manager

Time limit: 1.0 second
Memory limit: 64 MB

There is a tradition at the USU championships to call the most hard-to-solve problems coffins. But to distribute coffins is also a very difficult problem. Consider a cemetery with places arranged in the form of a rectangle having N rows and M columns ($1 \le N, M \le 100$). At the initial moment of time the cemetery is empty. Incoming coffins are put in the row with empty places that has a minimal number; if there are several empty spaces in this row, then the column with the minimal number is chosen. From time to time the cemetery's clients are visited by their living relatives; it is considered to be a pleasure for the clients. But it's only a headache for the cemetery manager, since because of these visitors he cannot give to new clients places that have been used. Happily, visitors are not perfect, so after some time they forget where their relatives have been lying. That is why if a client was not visited for more than successive 1000 days, then on the 1001st day the manager regards the grave as empty. However, relatives of the adjacent clients (of those for whom the differences in the numbers of rows and columns are not greater than 1) may notice strange changes, so the manager puts a new client on a used place only if all the neighboring graves have not been visited for the last 100 days (this is a period of time sufficient for a neighbor's relatives to forget who was lying next to him or her). If, notwithstanding all the efforts of the manager, there is no place where he can put a new client, then the client is sent to a crematorium.

We have a complete list of arriving clients and coming visitors for some period starting from the foundation of the cemetery. Basing on this information, you should determine how many clients have been sent to a crematorium.

Input

The first input line contains numbers N and M that describe the size of the cemetery. Each of the next lines describes an event. A description starts with the time of the event measured in days (it can be from 0 to 7600). Then the type of the event is given: either *d* (arrival of a new client) or *v* (a visit of relatives) followed with the number of the client who has visitors. It is guaranteed that in the case of visit the client with such number was described in the input earlier.

The events are ordered according to their time. The input contains not more than 15000 events, and not more than 10000 of them describe arrivals of new clients.

Output

Output the number of clients that have been sent to a crematorium.

Sample

input	output
2 2 1 d 1 d 1 d 1 d 300 d 500 v 2 1001 d 1002 d 1002 d 1003 v 3 1003 d 1003 d 1236 v 2 2032 v 2 2033 d	3

Notes

1. Each tomb has up to 8 neighbors.
2. If a client was buried on day T then the tomb may be dug over on day T+1001 and may not be dug over on day T+1000.
3. If a tomb was visited on day T then its neighbors may be dug over on day T+101 and may not be dug over on day T+100.
4. A tomb is dug over as soon as there is an opportunity (see items 2 and 3).
5. During a funeral relatives notice nothing including the neighbors.
6. The clients are numbered in the the order that they arrive (including those who was sent to crematorium).
7. If there is already no tomb or the client has been sent to the crematorium immediately then a visit affects nothing.
8. The next in turn client may be always burried in an empty tomb inspite of the neighbor tombs visits (the neighbors' relatives wouldn't be surprised having found out that the adjacent empty tomb is already occupied).

Problem Author: Stanislav Vasilyev
Problem Source: Open collegiate programming contest for student teams, Ural State University, March 15, 2003

1252. Sorting the Tombstones

Time limit: 1.0 second
Memory limit: 64 MB

There is time to throw stones and there is time to sort stones…

An old desolate cemetery is a long dismal row of nameless tombstones There are N tombstones of various shapes. The weights of all the stones are different. People have decided to make the cemetery look more presentable, sorting the tombstone according to their weight (in increasing or decreasing order). The local custom allows to transpose stones if there are exactly K other stones between them.

Input

The first input line contains an integer N ($1 \le N \le 130000$). Each of the next N lines contains an integer X, the weight of a stone in grams ($1 \le X \le 130000$).

Output

The output should contain the single integer — the maximal value of K ($0 \le K < N$), that makes possible the sorting of the stones according to their weights.

Sample

input	output
5 30 21 56 40 17	1

Problem Author: Alexey Lakhtin
Problem Source: Open collegiate programming contest for student teams, Ural State University, March 15, 2003

1253. Necrologues

Time limit: 1.0 second
Memory limit: 64 MB

You know that many necrologues even the most heartfelt are very similar. Our partners from the ACM Company (Advanced Cemetery Management), which is a sponsor of today's thematic problem set, decided to disclose some secrets of skill. As soon as we got into the workshop of the word-painters we found out that:

1. there are N ($1 \le N \le 9$) sample necrologues;
2. each sample necrologue has not more than 1000 symbols (capital and small Latin letters, digits, punctuation marks, spaces and carriage return marks);
3. each sample may contain not more than 10 references to other samples (the reference is marked by *, which is followed by a number of a sample that is referred to; the *-mark is used in the samples in no other way but as a reference mark).

A reference may be *activated* at a client's pleasure: the sequence *M is substituted with the sample necrologue number M (with all its references).

This system worked properly up to the moment that a very rich client entered the workshop. He wanted to add a stone to his mother's-in-law cairn and wished a necrologue according to the sample number 1 with *all* the references activated (including the ones that can appear after activating the references in the first sample, and so on).

It's necessary to write a program in order to understand if it's possible to fulfill the wishes of the client and what will appear in this necrologue.

Input

The first input line contains a number N, an amount of the necrologues samples. Than follow the samples texts. Each samples starts from a new line and ends with the # symbol.

Output

Consider the necrologue forming procedure that starts from the first sample, activates all the references than activates all the references in the result of the previous step and so on ad infinitum. If such a procedure leads to the text not longer than 10^6 symbols (spaces and line feeds are considered as symbols) you should output the necrologue text. You should output # otherwise.

Sample

input	output
7 She w*7s *7 *2 wom*7n. *3# wonderful# Everyone loved her *5. We will miss her *5.# Some text *6# very much# Another text *4# a#	She was a wonderful woman. Everyone loved her very much. We will miss her very much.

Problem Author: Leonid Volkov
Problem Source: Open collegiate programming contest for student teams, Ural State University, March 15, 2003

1254. Die Hard

Time limit: 5.0 second
Memory limit: 64 MB

There is a city with a grid of square blocks of the $N \times M$ size. There are buildings in some blocks, some blocks are blank. John is in the block (x_0, y_0). He may move from a block to an adjacent one in horizontal, vertical or diagonal direction with velocity V. He is told over the radio the list of points where bombs are located. John is to disarm them in the same order that they follow in the list or he will die hard with a vengeance. If he can't reach some bomb he moves to the next one. All the bombs are located outside the buildings.

What minimal time will John need to finish his job if he disarms a bomb immediately?

Input

The first line contains numbers N, M, K (an amount of bombs) and V, separated with a space, satisfying the restrictions $1 \le N, M \le 75$; $1 \le K \le 1000$; $0.01 < V < 10.00$. Then a city map follows: M lines of N symbols. The symbol '.' means a blank block, '#' stands for a building. Then follow the line that contains coordinates (x_0, y_0). The input is ended by K lines with bombs coordinates in that very order that John passed them.

Output

You should output the single number which is the minimal time necessary to do the job. The time should be printed with two digits after a decimal point.

Sample

input	output
4 3 3 1.23 .... ##.. .... 1 1 1 3 4 1 4 3	8.66

Problem Author: Pavel Atnashev
Problem Source: Open collegiate programming contest for student teams, Ural State University, March 15, 2003

1255. Graveyard of the Cosa Nostra

Time limit: 1.0 second
Memory limit: 64 MB

There is a custom among the Ural Mafiosi — a big Mafioso's coffin is to be carried by all his subordinates. The length of the coffin (in meters) equals to the number of the Mafioso's subordinates in order not to let the carriers to encumber each other. As it happens, according to the ancient custom the width of a coffin is equal to 1 meter. So, the length of a coffin shows a dead man's authority. By the way, the Ural Mafiosi are very scrupulous in matters of authority and will not bear neighborhood with less authoritative Mafioso. So, at one cemetery it's possible to bury Mafiosi with equal authority. According to the Mafiosi's custom a cemetery must be square. A cemetery length must be an integer number of meters.

You are to count how many Mafiosi can be buried on the cemetery of the given size. Coffins must be parallel to cemetery borders, coffins mustn't overlap each other and get off the cemetery.

Input

Contains two numbers — a length of the cemetery N ($1 < N < 10000$) and a length of a coffin K ($1 < K < 10000$).

Output

The single integer number — the most amount of the coffins of the size $1 \times K$ that may be buried at the cemetery of the size $N \times N$.

Sample

input	output
5 3	8

Problem Author: Stanislav Vasilyev, Alexey Lakhtin
Problem Source: Open collegiate programming contest for student teams, Ural State University, March 15, 2003

1256. Cemetery Guard

Time limit: 1.0 second
Memory limit: 64 MB

In some black-black wood at the black-black cemetery there was a golden gravestone. This gravestone was guarded by two black-black dogs. Each dog sits enchained near a black-black pole and nearby in the wood there is a black-black guard's house. Every morning the guard leaves the house to bring the dogs plates with food. He places the plates so that the dogs may eat remaining enchained to their poles.

Compute the shortest way that the guard is to walk in order to feed both dogs (the guard may easily carry food to both dogs at the same time and may feed them in an arbitrary order).

Input

The first line contains three numbers: the distance in meters from the guard's house to the first pole R_1, from the guard's house to the second pole R_2 and the distance between the poles R_3. The second line consists of one integer which is the length of each dog's chain R_4 (the chains of the dogs are identical). The numbers R_i ($i = 1, 2, 3$) satisfy the restriction $0 \le R_i \le 20000$; $1 \le R_4 \le 20000$.

Output

Output the single number which is the length of the shortest guard's way in meters within three digits after a decimal point.

Sample

input	output
1000 2000 1000 250	3500.000

Problem Author: Alexander Petrov (prepared by Alexander Mironenko)
Problem Source: Open collegiate programming contest for student teams, Ural State University, March 15, 2003

1257. Hyphenation

Time limit: 1.0 second
Memory limit: 64 MB

Programmers are strange people. They don't want to rejoice in life's pleasures, to read books, to listen to the music, to write letters to each other. The code, debug, rebuild something all the time. So our acquaintance programmer Vasechkin is not pleased with the existing software. This time he rages against the text editor. This editor can't carry words over to the next line. What can a man do? Artemy Sidorovich, seemingly, is to teach this editor do this.

Input

The first line contains integer N ($0 \le N \le 100$) — an amount of hyphenation rules. Then there are N lines with the rules description. Each rule looks like "$string_1$-$string_2$", where $string_1$ and $string_2$ are not empty and contain only small and capital Latin letters. Summary length of $string_1$ and $string_2$ does not exceed 40 symbols. This record means that if a word is ended with a substring $string_1 string_2$, then $string_2$ may be carried to the next line and after $string_1$ appears a hyphen-mark '-'. Then there is a text typed by Artemy Sidorovich. A *word* is a sequence of small and capital Latin letters. Any other symbols are word separators. Each line is not longer than 250 symbols. The text consists of not more than 450 lines. Each word is not longer than 40 symbols.

Output

Output a text formatted according to the hyphenation rules with respect to the fact that the paper width is 40 symbols. Each output line can't be wider than 40 symbols (including the hyphen '-' and excluding the line feed). If there are several rules that can be used the the word is to be carried to the next line according to the rule that provides the minimal length of the carried part of the word. If a word can't be carried to the next line so that the length of the line wouldn't exceed 40 the word should be carried to the next line wholly. The hyphenation rules are not case sensitive (the rule "m-Ing" can be used in the word "SwImMiNg"). The existing line feeds, spaces and stops must be saved.

Sample

input
12 un-usual for-ced re-al sum-mer N-ED home-work sec-ret ho-lidays Wi-zard th-Er L-so Holi-days Harry Potter was a highly unusual boy in many ways. For one thing, he hated the summer holidays more than any other time of year. For another, he really wanted to do his homework but was forced to do it in secret, in the dead of night. And he also happened to be a wizard.
output
Harry Potter was a highly unusual boy in many ways. For one thing, he hated the summer holi- days more than any other time of year. For another, he really wanted to do his homework but was for- ced to do it in secret, in the dead of night. And he al- so happened to be a wizard.

Problem Author: Den Raskovalov
Problem Source: Open collegiate programming contest for high school children of the Sverdlovsk region, October 11, 2003

1258. Pool

Time limit: 1.0 second
Memory limit: 64 MB

During the lunch-break the programmer Vasechkin likes to hack around his rectangular room. His way starts at his workplace and ends at the point where the thought to continue the work comes to his mind. We know that when Vasechkin runs against a wall he bounds according to the law “the angle of incidence equals to the angle of reflection”. Besides being a very straight person Vasechkin moves from wall to wall in the direct line segment. The baleful front office decided to find out how much time Vasechkin wastes during his promenades. It’s very easy to find out the time by dividing the length of the way that Vasechkin has passed over his average speed (that was measured by the front office beforehand). So it’s necessary to find out the way length! And since Vasechkin’s bumps are well-heard it’s sufficient to know the order of Vasechin’s collisions with walls. May be there is a more simple way to calculate the time wasted by the programmer but he front office believes that its solution of the problem is the most correct.

Input

The first line consists of two integers W and D — they are the width and the depth of Vasechkin’s room in meters ($2 \le W, D \le 1000$). The second line contains the initial Vasechkin’s position coordinates ($0 < x_0 < W$; $0 < y_0 < D$), measuring from the front-left corner of the room. The third line — the final position coordinates ($0 < x_1 < W$; $0 < y_1 < D$), measuring from the front-left corner of the room. All coordinates are integers. The fourth line contains the sequence of letters L, R, F, B, denoting the sequential Vasechkin collisions with walls — the left, right, front and back one respectively. An amount of collisions doesn’t exceed 1000. The programmer never collides a corner of his room and his initial and final positions don’t lie on walls.

Output

Output the length of Vasechkin’s way from the initial point to the final one rounded to four decimals after the point.

Sample

input	output
10 20 9 1 1 19 FLRLRB	52.8015

Problem Author: Pavel Egorov
Problem Source: Open collegiate programming contest for high school children of the Sverdlovsk region, October 11, 2003

1259. How to Become Star?

Time limit: 1.0 second
Memory limit: 64 MB

The people think about this problem for several centuries. The programmer Vasechkin's front office decided to find it out. But the front office is the front office, so the task to find out the answer to the question "How to become a star?" was given to its subordinate Vasechkin.

It's often happens that account's and programmer's notion of the problem much differ. So this time it came off not exactly how the front office has thought. Vasechkin formalized the problem as follows.

Definition. A *star* is the closed broken line built by the final amount of steps of the following algorithm:

1. Fix and arbitrary angle α ($0 < \alpha < \pi$).
2. The first link is (0, 0) — (1, 0).
3. The second link is the resultant of the turn by the angle α counter-clockwise with respect to the point (1, 0) of the first one.
4. The (i + 2)-nd link is the resultant of the turn by the angle α counter-clockwise of the (i + 1)-st one with respect to the free end (the opposite to the one that is connected to the i-th link) of the (i + 1)-st link.
5. The algorithm stops immediately when the broken line is closed.

Definition. A number of the vertices of the star is the number of the broken line's links.

Input

The only integer N ($3 \le N \le 100000$).

Output

Output an amount of different stars with N vertices.

Samples

input	output
5	2
9	3

Problem Author: Pavel Egorov
Problem Source: Open collegiate programming contest for high school children of the Sverdlovsk region, October 11, 2003

1260. Nudnik Photographer

Time limit: 1.0 second
Memory limit: 64 MB

> *If two people were born one after another with one second difference and one of them is a child, then the other one is a child too. We get by induction that all the people are children.*

Everyone knows that the mathematical department of the Ural State University is a big family of N persons, 1, 2, 3, …, N years old respectively.

Once the dean of the department ordered a photo if his big family. There were to be present all the students of the department arranged in one row. At first the dean wanted to arrange them by their age starting from the youngest student, but than he decided that it would look unnatural. Than he advised to arrange the students as follows:

1. The 1 year old student is to sit at the left end of the row.
2. The difference in ages of every two neighbors mustn't exceed 2 years.

The dean decided that thereby the students would seem look as they were arranged by their ages (one can hardly see the difference in ages of 25 and 27 years old people). There exist several arrangements satisfying to the requirements. Photographer didn't want to thwart dean's desire and made the photos of all the possible mathematical department students' arrangements.

Input

There is the integer number N, $1 \le N \le 55$.

Output

the number of photos made by the photographer.

Sample

input	output
4	4

Notes

If $N = 4$ then there are following possible arrangements: (1,2,3,4), (1,2,4,3), (1,3,2,4) and (1,3,4,2).

Problem Author: Alexander Ipatov
Problem Source: Open collegiate programming contest for high school children of the Sverdlovsk region, October 11, 2003

1261. Tips

Time limit: 1.0 second
Memory limit: 64 MB

The favorite resting place of the Ural programmers is Threenland island. There is only one tribulation: dollars and euro don't go about here. So the tourists are to exchange their money into threets (the currency of Threenland). There go about 1 threet, 3 threets, 9 threets, 27 threets, …, 3^k threets, … Once programmer Vasechkin, after the N-threets bill was given to him found out, that he'd got one paper of each denomination. Waiters in Threenland have a custom to keep the change. Waiters like to get the tip that can be presented by a set of papers in which paper of each denomination appears not more than once. Otherwise their feelings are hurt. They have a peeve on a client if they don't get tips at all. Help please Vasechkin to pay for the dinner and not to hurt the waiter.

Input

consists of an integer N. $1 \le N \le 10^7$.

Output

Output two integers separated with a space – that is the sum that Vasechkin is to pay and an amount of tips. If there are several solutions choose any of them. If there is no solution output 0. Remember that Ural programmers are not rich, so Vasechkin can't pay more than 4294967291 threets.

Sample

input	output
5	9 4

Problem Author: Alexander Ipatov
Problem Source: Open collegiate programming contest for high school children of the Sverdlovsk region, October 11, 2003

1262. Pseudo-Roman Number

Time limit: 1.0 second
Memory limit: 64 MB

We can’t say that all the programmers are absent-minded people but it is usual to some of them… Once Artemy Sidorovich tested his program. Particularly it was to be able to define a day of the week by the date. Artemy Sidorovich inputted "October 11, 2003" and got the answer "Saturday". "Aha!" — thought Artmy Sidorovich and started to search for a mistake (the calendar that was hung in his room said that October 11th, 2003 is Monday).

After an hour Artemy Sidorovich glances up and saw big digits in the calendar: 1999. Swearing under his breath and promising to through away the old calendar he looked at the clock. The hour hand was at the mark IIII. The day was almost finished.

— It’s interesting, — thought Artermy Sidorovich, — I’ve seen many times that the number 4 is written down by the Roman numerals as IV. It turns out that a decimal number can’t be represented by the Roman number unambiguously. He looked again at the calendar and thought so:

— Let the numbers 1, 5, 10, 50, 100, 500, 1000 be denoted by the Roman numerals I, V, X, L, C, D, M. Then the number 1999 may be represented as MDCCCCLXXXXVIIII or simply MIM. Or may be MCMXCIX. It’s evident that the record MIM is the shortest. But which one is correct?

In order to adjust differences Artemy Sidorovich decided:

We'll call a pseoudo-roman number the sequence of numerals: $A_1A_2\ldots A_n$, where:

1. Every numeral denotes on of the numbers 1, 5, 10, 50, 100, 500, 1000, … The different digits correspond to the different numbers. Let’s denote the number according to the numeral A as $[A]$.
2. There may be not more than 3 identical numerals one after another, if the numerals denote a power of 10 and not more than 2 identical numerals otherwise.
3. In the number $A_1A_2\ldots A_n$ the following two statements are correct:
   - $[A_i] \geq [A_{i+1}]$ or
   - ($[A_i] < [A_{i+1}] \leq 10[A_i]$ and $[A_i] = 10^k$), where $i < n$.
4. Before a numeral there may not be more than one lower numeral.
5. $[A_i] \geq [A_{i+1}] \geq [A_{i+2}]$, or $[A_{i+2}] < [A_i] < [A_{i+1}]$, or $[A_{i+1}] < [A_{i+2}] \leq [A_i]$, where $i < n - 1$
6. $A_1 = [A_1]$.
   $A_1A_2\ldots A_n = A_2\ldots A_n - [A_1]$, if $[A_1] < [A_2]$.
   $A_1A_2\ldots A_n = A_2\ldots A_n + [A_1]$, if $[A_1] > [A_2]$.

Then the number 4 will be written down as IV, and not as IIII (according to the rule 2). The number 1999 will be written down as MCMXCIX. It’s not the shortest way but every number is represented unambiguously.

Input

There is a decimal integer number N, $1 \leq N \leq 10^{2003}$.

Output

Output the integer K — that is an amount of numerals in pseudo-roman notation of the number N.

Sample

input	output
1939	8

Notes

1939 = MCMXXXIX

Problem Author: Alexander Ipatov
Problem Source: Open collegiate programming contest for high school children of the Sverdlovsk region, October 11, 2003

1263. Elections

Time limit: 1.0 second
Memory limit: 64 MB

The next in turn elections are to come soon. All the fences are postered with leaflets and the mail boxes are full of throwaways. Cheeky guys are looking at us from TV's and promise to make our life better… And programmer Vasechkin is knee-deep in work. He is to write a program that would calculate the results of voting.

Input

The first line contains a number of candidates N ($1 \le N \le 10000$) and a number of electors M ($1 \le M \le 10000$). Then M lines follow, each one contains a number of candidate that the elector voted for. The candidates are numbered with integers from 1 to N.

Output

Output N lines. The i-th line should contain the percent of electors that voted for the i-th candidate (to within 2 decimals).

Sample

input	output
3 6 1 2 3 2 1 1	50.00% 33.33% 16.67%

Problem Author: Den Raskovalov
Problem Source: Open collegiate programming contest for high school children of the Sverdlovsk region, October 11, 2003

1264. Workdays

Time limit: 1.0 second
Memory limit: 64 MB

After a success of the previous Vasechkin's program that allowed to calculate the results of the elections in cause of two days Artemy Sidorovich was placed at the head of the department. At the moment Artemy Sidorovich prepares a task for his subordinate — programmer Petechkin. The task is to write a very useful function that would ease the life of all the department programmers. For each integer from 0 to M the function would calculate how many times this number appears in the N-element array. Artemy Sidorovich deems that the function should work as follows (the sample code for $N = 3$, $M = 1$):

C	Pascal

`if (arr[0]==0) ++count[0];`
`if (arr[0]==1) ++count[1];`
`if (arr[1]==0) ++count[0];`
`if (arr[1]==1) ++count[1];`
`if (arr[2]==0) ++count[0];`
`if (arr[2]==1) ++count[1];`
	`if arr[0]=0 then count[0] := count[0] + 1;`
	`if arr[0]=1 then count[1] := count[1] + 1;`
	`if arr[1]=0 then count[0] := count[0] + 1;`
	`if arr[1]=1 then count[1] := count[1] + 1;`
	`if arr[2]=0 then count[0] := count[0] + 1;`
	`if arr[2]=1 then count[1] := count[1] + 1;`

Artemy Sidorovich wants to estimate the time that Petechkin will need to execute the task. We know that Petechkin needs one second to write a line of the code (he's fast, isn't he?). Artemy Sidorovich doesn't know exactly bounds for M and N. Your task is to write program that would calculate a number of seconds that Petechkin will write the code.

Input

The only line contains integers N ($0 \le N \le 40000$) and M ($0 \le M \le 40000$).

Output

Output an amount of seconds that Petechkin needs to write the program.

Sample

input	output
3 1	6

Problem Author: Den Raskovalov
Problem Source: Open collegiate programming contest for high school children of the Sverdlovsk region, October 11, 2003

1265. Mirror

Time limit: 1.0 second
Memory limit: 64 MB

Ri-i-i-i-ing… Ri-i-i-i-ing…

— W-what the hell!!!??? — cursed major Pronin, hardly opening his eyes and looking at the watch on the bed-side table. — Who wants me on Sunday at 5 a.m.?

Major took his pistol, pointed it at the telephone and squinted his left eye. The image blurred and he couldn't aim. The yesterday party told on him.

Ri-i-i-i-ing… Ri-i-i-i-ing… — crackled the telephone-set busily.

Had overcome a great temptation, Pronin put his pistol away, reached the telephone and and took a receiver. "Whoever it is, he won't be alive by the next morning," — he thought at that minute. Major imagined the caller lying in the mass of blood and himself, standing next to the corpse with the fuming pistol with a smile on his face.

— Major Pronin? — he heard from the receiver.

Had listened a familiar voice, major stood at attention.

— Yes sir! Mister general, sir!

— You have an assignment. A car is waiting for you near your doorway. See you in the office in 20 minutes.

The line disconnected.

Pronin entered the general's office after 19 minutes and 50 seconds after the call.

— A very important investigation, — said the general after a short salutation. — Possibly, a murder. Yesterday's night a famous banker Kozlov A. E. knocked down a pedestrian. It seemed an ordinary case but the victim appeared to be his opponent at the legal process that our… "client" has won. The detained suspect claims that he didn't notice the pedestrian in the rear-view mirror while driving out of his garage. He lays stress that it was an accident. We'd better verify it. Here are the documents. You are to give a report to me personally immediately after your investigation ends. As you know our Kozlov A. E. is a mayor contender. There mustn't be a mistake!

— Yes, sir! — bawled the major. — May I go?

— Go!

Pronin looked through the on-site review protocol. Meticulous officers of the criminal investigation department worked out a very exact description of the situation. In particular, he found the metering results of the car mirror and the location of the body.

— Stop! — said the major. I can determine with the help of this data if our banker lies or not saying that he didn't see the pedestrian!

He took all the data and left to the analytical department to his acquainted programmer. He reached the door and…

Knock, knock, Contestant!
Major has you.

Input

There are 8 real numbers within 6 digits after a decimal point, separated with one or several spaces and/or linefeeds — ($-1000 \le x_1, y_1, x_2, y_2, x_{m1}, y_{m1}, x_{m2}, y_{m2} \le 1000$). Those are the banker's eyes coordinates (major calculated them by the deductive method according to the situation description worked out by the criminal investigation department officers), the victim body at the moment of blow (calculated the same way) and both ends of the mirror coordinates, respectively. It's guaranteed that that the points (x_1, y_1) and (x_2, y_2) don't lie on one straight line, passing through the points (x_{m1}, y_{m1}) and (x_{m2}, y_{m2}). It's also guaranteed that the points (x_{m1}, y_{m1}) and (x_{m2}, y_{m2}) are not the same.

Output

You are to output "VISIBLE" , if one can see the point (x_2, y_2) from the point (x1, y1) in the rear-view mirror with the ends coordinates (x_{m1}, y_{m1}) and (x_{m2}, y_{m2}), or "INVISIBLE", otherwise.

Samples

input	output
1 0 0 -1 0 0 1 1	VISIBLE
0.000001 0 0 0.000001 -0.000001 -0.000001 0.000001 0.000001	INVISIBLE

Notes

We must say that the rear-view mirror in the banker's car is double-sided but it reflects the light as all the mirrors (the angle of incidence equals to the angle of reflection).

Problem Author: Alexander Klepinin (prepared Ivan Dashkevich)
Problem Source: Ural State University championship, October 25, 2003

1266. Kirchhoff's Law

Time limit: 1.0 second
Memory limit: 64 MB

Vasya's dad, as you know, is good in math but the son instead of following his father's steps, studies physics at school. Once Vasya asked his father to help him to solve a simple problem — to find out the resistance of the resistors system. Dad answered him: "Here is nothing to think about. You are to numerate the conductors nodes, to write the I=U/R law for each conductor. Then remember that the sum of currents at each of the nodes except the first and the last equals to zero, you may assume potential in the first node equal to one and in the last node — zero. Then you get a simple system of linear equations. Hence you find potentials in the intermediate nodes and currents between all the nodes. It's left only to divide the voltage by the total current from the first node and…

But Vasya is not good in math, so his dad was to write the system of equations himself and to solve it. Vasya looks at the end of the book of problems and says that there is another answer. Dad tried to solve the problem again and got another answer. Vasya looked at the answer and said again: "Wrong". Dad resolves the problem for the third time and Vasya holds his own. Dad got tired to solve the problem manually and he decided to use a computer seeing that the students of mathematical department of the Ural State university are ready to write the required program.

Input

The first line contains integers N and M; N is a number of nodes in the circuit ($2 \le N \le 20$), M is the number of resistors ($1 \le M < 1000$). Each of the next M lines consists of three integers A_i, B_i и R_i — description of a resistor that has resistance R_i connecting the nodes A_i and B_i ($1 \le A_i < B_i \le N$; $1 \le R_i \le 1000$). There may be many resistors between two nodes.

Output

Your task is to output the total resistance between the nodes 1 and N rounded within two digits after a decimal points.

Sample

input	output
4 5 1 2 15 2 4 5 1 3 10 3 4 10 2 3 1	9.40

Problem Author: Alexander Klepinin (prepared Stanislav Vasilyev)
Problem Source: Ural State University championship, October 25, 2003

1267. Yekaterinburg Subway

Time limit: 1.0 second
Memory limit: 64 MB

There is only one branch-line of Yekaterinburg subway. But what a branch-line! Each station is an architectural masterpiece. It is no wonder that all the guests of our city try to have time to observe our subway in detail. And, as usual, the tourists try to manage in the minimal possible time – they have much to see! In an effort to make Yekaterinburg more attractive the city administration decided to work out a program that would calculate the optimal, in respect to the time, route passing all the stations. Of course, the turn of speech "the city administration decided to work out a program" doesn't represent the facts exactly – the administration decided and this is your team who is to work it out.

The subway contains one branch-line. The trains run here with a definite interval that is evaluated by the integer number of minutes, intervals between the stations are known as well (they are evaluated by the integer number of minutes, too). You may assume that train stops at each station are instant. A tourist leaves a train instantly and it takes one second to board a train. A tourist needs 58 seconds to admire at a station. At the end of the sight seeing a tourist is to return to the starting station of his trip. He has already seen the station in the very beginning and so you are not to add the time of seeing it to the total trip time. A tourist may start his observation any time he likes, so the total trip time is to be counted out from the moment that he enters the train at the departure station.

Input

The first line contains the number of stations N ($1 \le N \le 16$). The stations are numbered from 1 to N. The second line consists of $N - 1$ nonnegative number of minutes that doesn't exceed 10^5—the time that a train runs from the first station to the second. From the second to the third, and so on (surely a train runs from the i-th to (i+1)st station and from the (i+1)st to the ith the same time). The third line contains the number of station that is the initial point of the trip. The fourth consists of three numbers: an interval between the trains, the departure time of the first train from the first station and the departure time of the first train from the N-th station;. all the numbers are nonnegative integers and measure minutes. The interval between the trains is nonzero and doesn't exceed 10^5, and the departure time from the terminal stations doesn't exceed the interval.

Output

should contain the duration of the shortest possible observation of all the stations in minutes.

Sample

input	output
3 5 7 2 4 0 1	28

Problem Author: Leonid Volkov (prepared Leonid Volkov and Alexander Somov)
Problem Source: Ural State University championship, October 25, 2003

1268. Little Chu

Time limit: 0.25 second
Memory limit: 64 MB

The favorite occupation of Little Chu is to sleep. Strictly speaking, he is busy with nothing but sleeping. Sometimes he wakes up and than the mankind makes some Great Discovery. For the first time Little Chu woke up K days after his birth. For the second time he woke up K^2 after his birth. For the third time — K^3 days after his birth. This rule still holds true.

Each time whem Little Chu wakes up he looks at the calendar and remembers what day of week is today. They say that if the day of week will be repeated, than Litle Chu will start crying and his tears will flood the world.

Your task is to make the largest number of the Great Discoveries and maximally to delay the doomsday. Determine when should Little Chu be awaken for the first time if it is known that he can't sleep more than one week after his birth.

Input

The first line contains integer T ($1 \le T \le 6553$) — the number of tests. Each of the next T lines contains integer N ($2 < N < 65536$) — the number of days in the week. N is prime.

Output

K for each input test.

Sample

input	output
4 3 5 7 11	2 3 5 8

Problem Author: Pavel Atnashev
Problem Source: Ural State University championship, October 25, 2003

1269. Obscene Words Filter

Time limit: 1.0 second
Memory limit: 8 MB

There is a problem to check messages of web-board visitors for the obscene words. Your elder colleagues commit this problem to you. You are to write a program, which check if there is at least one obscene word from the given list in the given text as a substring.

Input

The first line consists of integer n ($1 \le n \le 10000$) — an amount of words. The next n lines contain the list of words that we can't allow to use in our well-educated society. A word may contain any symbol but the ones with codes 0, 10 and 13. The length of each word doesn't exceed 10000 symbols. The total list of words doesn't exceed 100 KB. Then there is an integer m — the number of lines of the text. A size of the text doesn't exceed 900 KB.

Output

the number of line and the number of position separated with a space, where an obscene word occurs for the first time. If there are no obscene words, output "Passed".

Sample

input	output
5 dear sweetie angel dream baby 8 Had I the heavens' embroidered cloths, Enwrought with golden and silver light, The blue and the dim and the dark cloths Of night and light and the half-light, I would spread the cloths under your feet: But I, being poor, have only my dreams; I have spread my dreams under your feet; Tread softly because you tread on my dreams.	6 33

Problem Author: Pavel Atnashev
Problem Source: Ural State University championship, October 25, 2003

1270. Unicube

Time limit: 1.0 second
Memory limit: 64 MB

Unicube is a remarkable set of toy-bricks developed by B. P. Nikitin especially for the children. Vasya's dad made such a set to his son. Nothing special — a set of 27 colored toy-bricks — each edge is colored red, blue or yellow. But the coloring allows to build the cube 3×3×3 with outer surface of any of the colors. Moreover, it allows (but not easily) to build a 3×3×3 cube with chess coloring not only of the outer edges, but of all the 6 (and taking into account two sides of each section, all the 12) sections of the cube. In short this toy is for the children up to 10 years and Vasya played it with his father.

When they had built all the figures suggested in the book (which could be built) Vasya began to bother his father with the question if it was possible to build such-and-such a figure with such-and-such a coloring. At first father tried to build Vasya's figures. Sometimes he succeeded and sometimes not. At the end father made a request to the students of the Ural State University to automate his distressing toil. Of course, the students equipped with computers would easily implement his request.

Your task is to write a program that given n information about the set of toy-bricks and the figure that is to be built, would suggest how to arrange the toy-bricks in order to build the figure. All the figures in the problem are the located in the three-dimensional space cells cubes such that some of the edges must be of a definite color.

Input

The first line consists of one integer N that is the number of toy-bricks in the set ($1 \le N \le 30$). Then there are N lines with the toy-bricks description: for each toy-brick in the order "front, upper, right, lower, left, back" pointed out the color of the edge number (integer from 1 to 30). Then in a new line there is an integer M that is the total number of toy-bricks in the figure to be built ($1 \le M \le 30$). At last there are M lines with the figure description in the following format: <X> <Y> <Z> <F> <U> <R> <D> <L> <B>, where X, Y, Z are coordinates of the cell (all of them are from 0 to 30), F, U, R, D, L, B are demands to the colors of the toy-brick in this cell in the same order as in description of the available toy-bricks colors (from 1 to 30; if there is 0 then there might be an arbitrary color). Here we use the left coordinate system (e.g. OX axis looks right, OY looks forward and OZ looks up) which is natural for Vasya building the figure on the floor.

Output

If there is at least one solution of the problem then your program is to output an arbitrary solution in the form of M lines. The i-th line is to contain the number of toy-brick (in the order given in the input starting from 1) for a cell number i (in the same order as in the input starting from 1). If there is no solution, then you are to output 0.

Sample

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

Problem Author: Alexander Mironenko
Problem Source: Ural State University championship, October 25, 2003

1271. Sailing Directions

Time limit: 3.0 second
Memory limit: 64 MB

Attention, attention! Moscow is speaking! All the radio stations are on air. We broadcast the urgent announcement.

Today the sea trials of the new supertanker “Oil industry worker” will take place. The new development of our scientists — the system “Sailing Directions” — will be world-wide demonstrated. This system provides an optimal control of the ship without a human in a square harbor. A ship (and all the ships have a triangle form) may move from the given start position to the final one among the other ships anchored in the harbor along the shortest distance. Everything necessary for the safe test was stipulated on the “Oil industry worker”: a thick steel sheeting; electronic control system; satellite communications with the complex of coordinates determination; and sensitive radar. But as usual meddled in a human element. Vovochka, the captain’s son, secretly stole to the ship just before the presentation, set down at the computer and decided to while away the time playing his favorite computer game. As a result a computer virus penetrated into the computer and it spoilt several functions of the program “Sailing Directions”. Now the ship can’t turn around its axis. You are to write a program that provides the shortest tanker route length to the given point.

Input

The first line consists of two integer numbers DX and DY — the dimensions of the harbor ($0 < DX, DY < 10^8$). Each of the lines 2-4 contain two integers — the tanker coordinates (remember that the “Oil industry worker” is triangle as all the ships). The fifth line contains a point where the tanker is to come in the end (namely its vertex described in the second input line). The sixth line consists of the integer N (the number of other ships in the harbor, $0 \le N \le 40$). The next $3N$ lines contain those ships coordinates. All the coordinates are within the harbor. The harbor corner is the point of origin. The ships don’t intersect.

Output

the minimal length of the route rounded to three digits after a decimal point, or −1, if the “Oil industry worker” won’t be able to reach the final point because of the injuries in the navigation program.

Sample

input	output
2003 2003 20 50 10 30 30 30 140 60 1 80 1000 100 20 60 20	146.569

Notes

- Harbor is rectangle with coordinates of corners (0,0), (*DX*,0), (*DX*,*DY*), (0,*DY*).
- Ship may not sail through harbor borders.
- Ships may touch each other and harbor borders.

Problem Author: Alexey Lakhtin
Problem Source: Ural State University championship, October 25, 2003

1272. Non-Yekaterinburg Subway

Time limit: 1.0 second
Memory limit: 64 MB

A little town started to construct a subway. The peculiarity of the town is that it is located on small islands, some of them are connected with tunnels or bridges. The mayor is sure that the subway is to be under the ground, that's why the project must use the less bridges the better. The only request for the subway is that the townsmen could get by metro (may be with changes) from every island to every island. Fortunately, we know that there is enough tunnels and bridges for it. It was decided to construct as less passages from island to island as possible to save money.

Your task given a town plan to determine the minimal possible number of bridges that is necessary to use in the subway construction.

Input

The first line contains three integers separated with a space: N (the number of islands, $1 \le N \le 10000$), K (the number of tunnels, $0 \le K \le 12000$) and M (the number of bridges, $0 \le M \le 12000$). Then there are K lines; each line consists of two integers — the numbers of islands, connected with the corresponding tunnel. The last M lines define bridges in the same format.

Output

the minimal number of bridges necessary for the subway construction.

Sample

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

Problem Author: Magaz Asanov (prepared Igor Goldberg)
Problem Source: Ural State University championship, October 25, 2003

1273. Tie

Time limit: 1.0 second
Memory limit: 64 MB

The subway constructors are not angels. The work under the ground and… Well, they are not angels. And where have you seen angels? It is all in a lifetime! Show me first somebody who has never… and then… all of us are people. And Vasya and me, too. May be we've overdrunked ourselves. But a little. And the ties lie crookedly… At that time they seemed to lie straight. No, we can't say that it must be so — criss-cross, but not all of them criss-cross! Some of the ties lie almost properly… Crookedly you say? And I'd say normally… After the yesterday's party? May be, may be… The ties that lie criss-cross we'll take away and it'll be OK, the train will pass on term, not by this New Year but by the next one. There's not much to disjoint. We'll pull out this tie and may be that one. Next to nothing! One, two, three…

Rails are two parallel straight lines that are for the users' accommodation parallel to the Y axis and have the coordinates X=0 and X=1. The "pell-mell" ties are arbitrary segments with the vertices on the rails in the integer points of the coordinate scale. At the first elimination of defects step you are to remove several ties that would disappear all the crossings. And, of course, after the yesterday's party the less you work the better, so you are to remove the minimal possible number of ties.

Input

The first line contains integer K ($0 \le K \le 100$) — the number of laid ties. Then there are K lines, each of them contains two integers Y_1 and Y_2 that describe the location of the next in turn tie — the tie described by the pair Y_1 and Y_2 connects the points (0, Y_1) и (1, Y_2). The absolute values of the numbers Y_1 and Y_2 don't exceed 1000. There are no identical among the numbers Y_1 and among the numbers Y_2.

Output

the minimal number of ties that are to be removed in order to eliminate crossings.

Sample

input	output
3 0 1 3 0 1 2	1

Problem Author: Magaz Asanov (prepared Leonid Volkov)
Problem Source: Ural State University championship, October 25, 2003

1274. Fractional Arithmetic

Time limit: 1.0 second
Memory limit: 64 MB

Your task is to write a program, which implements addition, subtraction, multiplication and division operations on fractions. Input and output fraction format is as follows:

- the sign of a number (must be written only if its absence may lead to misrepresentation of the number),
- the integer part of a number (the integer part that equals zero must be omitted if the numerator and the denominator are present),
- a space character (must be omitted if the integer or fractional part is absent),
- the numerator (if it is not equal to zero),
- the "/" sign (written only if there is a numerator),
- the denominator (must be omitted if there is no numerator),
- the denominator cannot be equal to zero.

Samples of fractional number representation: "-7 3/4", "8 1/2", "-7/11", "0", "11".

Limitations
All limitations apply for both input and output data. The sign of a fraction: if the fraction is positive, the sign is not written; if the fraction is negative, the "-" sign must be present. The integer part and the numerator may have values ranging from 0 to 30000, the denominator may have values from 1 to 30000. The operation sign may be one of the four possible symbols: "+" (addition), "-" (subtraction), "*" (multiplication), "/" (division). The expression is always correct, so the second operand in the division operation cannot be zero.

Input

The first line – a fraction (the first operand);
The second line – the sign of an operation;
The third line – a fraction (the second operand).
Both fractions may be reducible. The numerator is always less than the denominator.

Output

A single line that contains an irreducible proper fraction (result) in the format described above.

Sample

input	output
-3 1/6 + 2/4	-2 2/3

Problem Author: © Sergey G. Volchenkov, 2003(volchenkov@yandex.ru); Vladimir N. Pinaev, 2003(vpinaev@mail.ru; http://www.pic200x.chat.ru); Michael Y. Kopachev, 2003 (mkopachev@krista.ru).
Problem Source: 2003-2004 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk, October 15-16, 2003

1275. Knights of the Round Table

Time limit: 0.1 second
Memory limit: 0.5 MB
Language limit: C, C++, Pascal

N knights gathered at the King Arthur's round table. Each of them has several goblets near him. It is possible that knights have different number of goblets. The goblets are brought (and also carried away) by a servant who can carry only two goblets at a time (one for each hand). When the servant comes he can either bring two goblets, or carry them away. Note that he can serve exactly two knights that sit at a fixed distance K from each other.

For example, if K=1 then the knights who sit side by side are served.

By the end of the feast each of the knights should have an equal predefined number of goblets near him. The number of the times the servant has to come must be minimized.

Your task is to write a program, which plans the servant's work during the feast.

Limitations
2 <= N <= 1000
1 <= K <= N-1
Initial and final number of goblets near each knight is not greater than 1000 and it is always non-negative.
The total number of servant's visits is not greater than 30000.

Input

The first number contains three numbers separated with white-space.
N – the number of knights,
K – "arm-span" of the servant,
F – the final number of goblets each of the knights must have by the end of the feast.
The following N numbers separated with spaces or EOL characters describe the initial number of goblets near each knight. It is assumed that the knights are numbered in a cyclic manner, i.e. the first knight sits after the N-th one.

Output

If it is impossible to reach the goal, write "–1" (without quotes) to the output. If the solution exists then the first line must contain a single integer M – the number of the servant's visits. The following M lines must carry triples: two numbers (the numbers of knights being served) and "+" (plus) character if the goblets are brought or "–" (minus) if they are carried away. The data on each of these lines must be separated with a white-space character.

Sample

input	output
3 1 4 1 2 3	3 1 2 + 1 2 + 3 1 +

Problem Author: © Sergey G. Volchenkov, 2003(volchenkov@yandex.ru); Vladimir N. Pinaev, 2003(vpinaev@mail.ru; http://www.pic200x.chat.ru); Michael Y. Kopachev, 2003 (mkopachev@krista.ru).
Problem Source: 2003-2004 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk, October 15-16, 2003

1276. Train

Time limit: 1.0 second
Memory limit: 4 MB

Train Ltd., a company that is in for railroad transportation received an order to form a train having a certain number of carriages. The problem is that Train Ltd. has carriages built in different years, therefore each of the carriages may have one of the two kinds of coupling at each side. The company also has one locomotive at its disposal.

The coupling systems for both the locomotive and the carriages are labeled as “A” or “B”. It is impossible to turn either the locomotive or a carriage in the opposite direction.

You are given information about the carriages and the locomotive. Your task is to determine the number of ways to form different trains using the given carriages. The additional requirement is that the coupling systems at each of the ends of the train must correspond to those of the locomotive.

The trains are considered different if there is at least one mismatch when they are compared from one end to another.

Example 1. Let the company possess the following carriages: “AA”, “AA”, “AB”, “BA”, “BA” and the locomotive “AB”. The train must have four carriages. Then it is possible to form only two different trains having these carriages: “BAAAABBA” and “BAABBAAA”. It is possible to connect the locomotive at the left end of this train (using coupling “B”) or at the right one (using coupling “A”).

Example 2. Let the company now have only one carriage of each type: “AA”, “AB”, “BA”, “BB”, and the locomotive is “AA”. The train must have three carriages now. There are three ways to form a train: “AAABBA”, “ABBAAA” and “ABBBBA”.

Input

The first line of the input contains two integers separated with white-space character: N ($0 < N \le 40$) — the number of carriages the company has at its disposal, and K ($0 < K \le N$) — the required length of the train (measured in carriages). The following $N + 1$ lines describe the coupling systems for the locomotive (line 2) and the carriages. These descriptions are given as “AB”, “AA”, “BB” or “BA” (without quotes).

Output

Write the word “YES” if it is possible to form one (or more) trains according to the given parameters, or “NO” otherwise. If it is possible to form a train then write the number of different ways of doing so to the second line.

Samples

input	output
4 4 AB AA AB BA BA	YES 2
4 4 BA AA AB BA BA	NO

Problem Author: © Sergey G. Volchenkov, 2003(volchenkov@yandex.ru); Vladimir N. Pinaev, 2003(vpinaev@mail.ru; http://www.pic200x.chat.ru); Michael Y. Kopachev, 2003 (mkopachev@krista.ru).
Problem Source: 2003-2004 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk, October 15-16, 2003

1277. Cops and Thieves

Time limit: 1.0 second
Memory limit: 64 MB

The Galaxy Police (Galaxpol) found out that a notorious gang of thieves has plans for stealing an extremely valuable exhibit from the Earth Planetary Museum — an ancient microprocessor. The police chiefs decided to intercept the criminals on the way from their refuge to the museum. A problem arose while planning the police operation: would it be possible for the Galaxpol staff to control all the possible routes of the criminals?

The galaxy transport system is designed as follows. Each planet has a transport station that is connected to some of the other stations via two-way teleportation channels. Transport stations vary in their sizes, so different numbers of policemen may be required to take control over different stations. In order not to upset the operation, it was decided to leave the planets that are next to the museum or the refuge without any police control.

Help the Galaxpol to place their staff at the stations in order to block all possible routes of the thieves.

Input

The first line of the input contains a single integer $0 < K \le 10000$ — the number of policemen engaged to control the stations.

The second line has four integers: N, M, S and F delimited with white-space character.

N is the number of stations in the galaxy (the stations are numbered from 1 to N); $2 < N \le 100$.

M is the number of teleportation channels; $1 < M \le 10000$.

S is the number of the planet (and the station) where the museum is; $1 \le S \le N$.

F is the number of the planet (and the station) where the thieves' refuge is; $1 \le F \le N$.

The next line contains N integers (x_1, …, x_N) separated with white-space character — the number of policemen required to control each of the stations ($\sum_{i=1}^{N} x_i \le 10000$).

Then M lines follow that describe the teleportation channels. Each of these lines contains a pair of space-delimited integers — the numbers of stations being connected by a channel. The channel system is designed so that it is possible to reach any station from any other one (probably it would require several channel transitions).

Output

Write "YES" if it is possible to block all the possible routes within given limitations, and "NO" otherwise.

Samples

input	output
10 5 5 1 5 1 6 6 11 1 1 2 1 3 2 4 3 4 4 5	NO
10 5 5 1 5 1 4 4 11 1 1 2 1 3 2 4 3 4 4 5	YES

Problem Author: © Sergey G. Volchenkov, 2003(volchenkov@yandex.ru); Vladimir N. Pinaev, 2003(vpinaev@mail.ru; http://www.pic200x.chat.ru); Michael Y. Kopachev, 2003 (mkopachev@krista.ru).
Problem Source: 2003-2004 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk, October 15-16, 2003

1278. “… Connecting People”

Time limit: 1.0 second
Memory limit: 64 MB

A new model of a mobile phone is powered by a special sound generation processor, developed expressly for this purpose.

The processor has a sound generator, an instruction pointer (IP), command memory (100 cells) and a stack (100 cells).

The instruction set of this processor consists of the two commands only:

CALL X – a subroutine call command – pushes the incremented current value of the IP (the return address) on the top of the stack and sets the IP to X;

BELL&RET – the combined command “bell-return” – emits a sound of a fixed (unit) duration and performs control return (pops a value from the top of the stack and assigns it to the IP) or stops the execution of the program (if the stack is empty).

Any command (together with its operand) occupies exactly one memory cell.

The return address occupies only one cell on the stack, too.

The work cycle of the processor starts when it is necessary to emit a sound. At this moment the IP points to the zero memory cell. The processor stops its work if the stack is empty after emitting a sound (when BELL&RET command has been executed).

The manufacturers of the processor affirm that by using of this processor may be emitted sound of rather long duration.

Your task is to write a program, which accepts K as input and produces a program for this processor to emit a sound of duration K.

Limitations
The resulting program may have no more than 100 commands including exactly one BELL&RET command, being always the last one. Unused memory cells following the BELL&RET command are considered free.

Input

There will be one integer number K in the input ($0 < K \le 10^9$) – sound duration.

Output

Output a program for the processor described above. The execution of this program leads to emission of the sound of duration K. The first line should correspond to the zero cell of the memory, the second line – to the first cell, the third – to the second one etc. All lines, with exception for the last one, may contain only CALL commands. The operand of the CALL command is an integer (from 1 to 99) and must be separated from the command by a white-space character.

The last line must contain the BELL&RET command.

Samples

input	output
1	BELL&RET
4	CALL 3 CALL 3 CALL 3 BELL&RET

Problem Author: © Sergey G. Volchenkov, 2003(volchenkov@yandex.ru); Vladimir N. Pinaev, 2003(vpinaev@mail.ru; http://www.pic200x.chat.ru); Michael Y. Kopachev, 2003 (mkopachev@krista.ru).
Problem Source: 2003-2004 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk, October 15-16, 2003

1279. Warehouse

Time limit: 1.0 second
Memory limit: 64 MB

A warehouse is an N×M meter sized rectangle, which is divided into sections of 1×1 meter. The warehouse is served by a roof-mounted crane. 1×1 meter sized containers may be stacked one atop another in each section.

A new lot of K containers of the same kind arrived to the warehouse. It was decided to place the new containers so that the sections having less containers would be filled first.

For example, let N=3, M=3, K=10 and the number of containers in each section is represented in the table below.

	1	2	3
X	1	2	3
Y	4	5	6
Z	7	8	9

In this case the new containers will be sequentially placed in sections: x1, x1, x2, x1, x2, x3, x1, x2, x3, y1. After that the heights of the sections will be as follows:

	1	2	3
x	5	5	5
y	5	5	6
z	7	8	9

Your task is to write a program, which determines the minimum height of the sections after placing new containers.

Input

The first line contains three integer numbers N, M, K ($0 < N, M \le 100$, $0 < K \le 10^7$), where N and M are dimensions of the warehouse, and K is the number of new containers.

Each of the following N lines contains M space-separated integer numbers ranging from 1 to 1000. These numbers are the heights of the corresponding sections.

Output

Output the minimum height of the warehouse sections after the placement of new containers.

Samples

input	output
2 2 3 1 3 2 4	3
3 3 10 1 2 3 4 5 6 7 8 9	5

Problem Author: © Sergey G. Volchenkov, 2003(volchenkov@yandex.ru); Vladimir N. Pinaev, 2003(vpinaev@mail.ru; http://www.pic200x.chat.ru); Michael Y. Kopachev, 2003 (mkopachev@krista.ru).
Problem Source: 2003-2004 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk, October 15-16, 2003

1280. Topological Sorting

Time limit: 1.0 second
Memory limit: 64 MB

Michael wants to win the world championship in programming and decided to study N subjects (for convenience we will number these subjects from 1 to N). Michael has worked out a study plan for this purpose. But it turned out that certain subjects may be studied only after others. So, Michael's coach analyzed all subjects and prepared a list of M limitations in the form "s_i u_i" ($1 \le s_i, u_i \le N$; $i = 1, 2, \ldots, M$), which means that subject s_i must be studied before subject u_i.

Your task is to verify if the order of subjects being studied is correct.

Remark. It may appear that it's impossible to find the correct order of subjects within the given limitations. In this case any subject order worked out by Michael is incorrect.

Limitations

$1 \le N \le 1000$; $0 \le M \le 100000$.

Input

The first line contains two integers N and M (N is the number of the subjects, M is the number of the limitations). The next M lines contain pairs s_i, u_i, which describe the order of subjects: subject s_i must be studied before u_i. Further there is a sequence of N unique numbers ranging from 1 to N — the proposed study plan.

Output

Output a single word "YES" or "NO". "YES" means that the proposed order is correct and has no contradictions with the given limitations. "NO" means that the order is incorrect.

Samples

input	output
5 6 1 3 1 4 3 5 5 2 4 2 1 2 1 3 4 5 2	YES
5 6 1 3 1 4 3 5 5 2 4 2 1 2 1 2 4 5 3	NO

Problem Author: © Sergey G. Volchenkov, 2003(volchenkov@yandex.ru); Vladimir N. Pinaev, 2003(vpinaev@mail.ru; http://www.pic200x.chat.ru); Michael Y. Kopachev, 2003 (mkopachev@krista.ru).
Problem Source: 2003-2004 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk, October 15-16, 2003

1281. River Basin

Time limit: 1.0 second
Memory limit: 64 MB

You are given a map of rivers of some continent. Every river is shown as a broken line, which begins with a river head and ends either at the point where the river flows into another one, or on the river mouth. The vertexes of the broken line are the turning points of the river-bed, or the points of tributary flow.

We will consider the river basin as a convex polygon of minimum area that contains the river and all its tributaries.

Remark. According to the definition of river basin the same territory may belong to the basins of different rivers.

Sample. A continent with three rivers is shown. The coordinates of the rivers and areas of the basins are given in the table.

River name	X	Y	Basin area
river1	6	9	12
	5	11	
	3	12	
	1	7	
river2	7	9	1.5
	5	7	
	5	5.5	
river3	3	10	9.5
	5	8	
	4	6	
	5	5.5	
	6	5	
	3	5	

Your task is to calculate the maximum among all river basin areas.

Input

The first line contains the number of the rivers N. The rest of the input contains N blocks describing the rivers.

Each block *i* consists of:

- One line which contains *ki* – the number of the points in river description;
- *ki* lines containing pairs of real numbers *xj* and *yj* (1 <= *j* <= *ki*), separated by space characters – the coordinates of the points that describe the river-bed.

Limitations.
$0 < N <= 10$, $\sum ki <= 1000$, $-1000 <= xj, yj <= 1000$.

Output

contains one number – the area of the largest basin calculated with two digit precision.

Samples

input	output
3 4 6 9 5 11 3 12 1 7 3 7 9 5 7 5 5.5 6 3 10 5 8 4 6 5 5.5 6 5 3 5	16.00

2 4 6 9 5 11 3 12 1 7 6 3 10 5 8 4 6 5 5.5 6 5 3 5	12.00

Problem Author: © Sergey G. Volchenkov, 2003(volchenkov@yandex.ru); Vladimir N. Pinaev, 2003(vpinaev@mail.ru; http://www.pic200x.chat.ru); Michael Y. Kopachev, 2003 (mkopachev@krista.ru).
Problem Source: 2003-2004 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk, October 15-16, 2003

1282. Game Tree

Time limit: 1.0 second
Memory limit: 64 MB

A game for two players is determined by its tree. The competitors make moves in turn. The first competitor starts the game. The game ends up with either a draw, or a victory of one of the players. The leaf nodes of the tree of this game may have values equal to one of three numbers: “+1” – victory of the first competitor, “–1” – victory of the second competitor, “0” – draw.

You have to find out who will win if both competitors follow the right strategy.

Input

The nodes of the tree are numbered with successive integer numbers. The root of the tree always has number 1.

The first line contains an integer N ($1 < N \leq 1000$) – the number of the nodes in the game tree. Next N-1 lines describe the nodes – one line for each node (with exception for the first node). The second line will contain the description of the second node of the tree, the third line – the description of the third node, and so on.

If the node is a leaf of the tree, the first symbol of the line is “L”, followed by a space, then the number of the ancestor of this node goes, another space, and the result of the game (+1: victory of the first player, –1: victory of the second one, 0: draw).

If the node is an inner one then the line contains the first symbol “N”, a space character and the number of the ancestor of this node.

Output

contains “–1” if the second competitor wins, “+1” if so does the first and “0” if the result is a draw.

Samples

input	output
7 N 1 N 1 L 2 -1 L 2 +1 L 3 +1 L 3 +1	+1
7 N 1 N 1 L 2 -1 L 2 +1 L 3 +1 L 3 0	0
18 N 1 N 1 N 2 L 2 +1 N 3 L 3 +1 L 3 +1 L 4 -1 L 4 +1 N 4 N 6 L 6 -1 L 6 -1 L 11 -1 L 11 +1 L 12 +1 L 12 -1	+1

Problem Author: © Sergey G. Volchenkov, 2003(volchenkov@yandex.ru); Vladimir N. Pinaev, 2003(vpinaev@mail.ru; http://www.pic200x.chat.ru); Michael Y. Kopachev, 2003 (mkopachev@krista.ru).
Problem Source: 2003-2004 ACM Central Region of Russia Quarterfinal Programming Contest, Rybinsk, October 15-16, 2003

1283. Dwarf

Time limit: 1.0 second
Memory limit: 64 MB

Venus dwarfs are rather unpleasant creatures: ugly, malicious, and mean-spirited. Maybe it’s because of hard living conditions on their planet… but the fact remains: each of them is ready to sell his own mother in order to save up his pot of gold and to preserve it to the end of his days.

The dwarfs are especially nervous about the Mercury leprechauns who are always glad to empty a dwarf’s pot and to fill it with solar dust instead of the gold. The dwarfs are weak-sighted and can’t distinguish dust from gold. That is why each dwarf once a year visits the Central Galaxy Bank (CGB), where experienced specialists authenticate the content of the pot taking a small commission for the job.

When the amount of gold in a pot becomes less than or equal to a certain level, the life of a dwarf has no sense anymore, so he clears the world of his wretched soul: with the remaining gold he buys in a zoo the largest Jupiter toad and creeps under it which results in crushing his chest.

Input

The input contains three integers separated with spaces. The first number is the amount of gold in a dwarf’s pot at the initial moment. The second number is the amount of gold at which the dwarf’s life becomes senseless. Both values are measured in grams and don't exceed $2^{31} - 1$. The third number is the CGB commission (from 1 to 100); this is the percentage of gold that is taken from the pot as a way of payment for the verification.

Output

The output should contain the number of years that is left to the dwarf.

Samples

input	output
19 10 50	1
1000 1 1	688

Problem Author: Leonid Volkov (prepared by Ivan Dashkevich)
Problem Source: USU Personal Contest 2004

1284. Space Poker

Time limit: 1.0 second
Memory limit: 64 MB

The rules of playing space poker are so complicated that there's no sense in trying to explain them here. We'll say only that at the beginning of a game each player is given N ($N \le 20$) cards. Each card has a value (an integer from 1 to 100) and a suit (an integer from 1 to 10). All the cards are different. Suits with odd numbers are called blue and suits with even numbers are called yellow. The cards are dealt by a special card machine, which guarantees that the Steinpuper rule is satisfied. This rule says that if at the beginning of a game a player has X different blue suits and Y different yellow suits, then $|X-Y| \le 1$. The cards are given to a player one by one and the player puts them into his hand from left to right.

In order to become acquainted with his cards before the game starts, the player *arranges the cards*. This means that by means of *atomic operations* he attains such a disposition of his cards that

1. All the cards of one suit lie one after another.
2. Blue and yellow suits alternate (this is always possible due to the Steinpuper rule).
3. Inside a suit the cards are ordered either in the ascending or in descending order and all the suits are ordered in the same way.

An atomic operation consists in taking one of the cards and moving it to any place (i.e., to the leftmost position, to the rightmost position, or between any two cards). Obviously, arranging the cards can be performed in many ways, but the aim is to do it with the minimal number of operations.

Input

The first line contains the number of cards N. The following N lines describe the cards in the order in which they are given by the machine. Each card is described by its value and suit separated with a space.

Output

The output should contain the minimal number of operations necessary for arranging the cards.

Sample

input	output
5 10 1 12 2 8 2 4 4 7 4	2

Problem Author: Leonid Volkov
Problem Source: USU Personal Contest 2004

1285. Thread in a Hyperspace

Time limit: 1.0 second
Memory limit: 64 MB

There are two starcrafts and a drop of water (nobody knows where it comes from) in a hyperspace (it's well-known that hyperspace has 8 dimensions). Whereas there're anisotropic distortions because of the hyperspace transfer you may assume the ships as points (A and B) and the drop as a sphere with radius R and center in the point C. Coordinates of all the points are integer and their absolute values don't exceed 1000. The drop is motionless. The radius R is a positive integer not exceeding 3000. The distance from the point C to the points A and B is greater than R.

The starcraft B is distressed (and motionless as well), and the starcraft A hurries to succor. You are to find out the length of the short cut from the point A to the point B not crossing the sphere (the starcraft may move along the edge of the sphere).

Input

There are three lines in succession containing coordinates of the points A, B and C respectively. Each of the lines consists of 8 integers. The fourth line contains positive integer R, that is the sphere radius.

Output

Should contain the length of the short cut within 2 digits after a decimal point. There must be exactly 2 digits after a decimal point. The result is to be rounded according to the standard mathematical rules.

Sample

input	output
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 10 10 10 10 5 5 5 5 3	1.00

Problem Author: Alexander Mironenko and Alexey Lakhtin
Problem Source: USU Personal Contest 2004

1286. Starship Travel

Time limit: 1.0 second
Memory limit: 64 MB

It is well known that a starship equipped with class B hyperengine is able to travel from any planet to any other planet. But your starship got severe damage in the last travel and now its movement ability is limited. The starship's technician determined that with the damaged hyperengine the vehicle can move from a point with coordinates (i,j) only to a point from the following list: $(i+q, j+p)$, $(i-q, j+p)$, $(i+q, j-p)$, $(i-q, j-p)$, $(i+p, j+q)$, $(i-p, j+q)$, $(i+p, j-q)$, $(i-p, j-q)$ (all the coordinates here are integers and are given in the standard intergalaxy system). Help the captain of your ship to find out if the ship is able to reach the destination planet on its own or a repair ship must be called.

Input

The first line contains two integers p and q (the two remaining discrete power rates of the damaged hyperengine) separated with a space. The second line contains the coordinates of the point where the spaceship is now. The third line contains the coordinates of the destination planet. The numbers in the second and third lines are also separated with spaces. All the numbers are integers and do not exceed $2 \cdot 10^9$ in absolute value.

Output

If the commander can move the damaged starship to the destination planet, write 'YES'. Write 'NO' if a repair ship must be called.

Samples

input	output
4 6 0 0 10 10	YES
4 6 0 0 9 9	NO

Problem Author: Alexander Klepinin
Problem Source: USU Personal Contest 2004

1287. Mars Canals

Time limit: 1.0 second
Memory limit: 64 MB

There is a quadrate area on the Mars surface wreathed in ideal net of canals. They plot the surface into equal squares (we neglect here the surface curvature). Each side of the quadrate area plotted into N square regions.

Archeological investigations showed that there was an ancient country Yatik in this area. The inhabitants cultivated a special grain — sir — that was a basis of their food ration. There is sir of two kinds: coarse-grained and small-grained. As a matter of fact, the end of Yatik empire started after the civil war between the fanciers of the sir sorts. But until recently noone new which of the parties won that time. The scientists look forward to guess the riddle on the grounds of the last voyage to Mars results. They found out which kind of sir was sowed the last in each square of Yatik. According to the ancient tradition sir was sowed in the sequence of squares (parallel to the north-south or east-west directions or at the angle 45° to them), one may suppose that the supporters of the party-winner made the longest sowings.

Input

The first input line contains a size of the square area — N ($1 \le N \le 1400$). Then there follow N lines. Each of them consists of N symbols. A letter “s” in the i-th line and j-th row means that in the according square region small-grained sir was sowed the last, a letter “S” means that coarse-grained sir was sowed the last. You may assume that the inhabitants of the area sowed nothing but sir. Each square region was sowed with only one sort of sir.

Output

The first line should contain a symbol “s”, if the party of small-grained sir fanciers won in the civil war. And symbol “S”, if the winners were the fanciers of the coarse-grained sir. If it’s impossible to define a winner then the first line should contain one symbol “?”. The second line should contain integer number — the maximal length of the one sort of sir sowing.

Samples

input	output
3 SsS sSs SsS	S 3
2 sS Ss	? 2

Problem Author: Alexander Mironenko (prepared by Igor Goldberg)
Problem Source: USU Personal Contest 2004

1288. Astrolocation

Time limit: 1.0 second
Memory limit: 64 MB

— Did you call for me, general?

— Come in. You were recommended to me as a young promising programmer. As you probably know, our new … though how can you know, it's all secret. Well, our department needs a program for the analysis of … let's say signals from … let's call it a locator. This locator is designed for searching for … well, objects, which move in a layer of … no, I can't tell you in a layer of what. The locator is placed on our new … well, it also can move. You'll get the signal through the interface … though it's not important, for you it will be a sequence of numbers, each from 0 to 100. First you find in this sequence a signal from the front wall of the layer, this is a number 95 or greater. If there are several such numbers, the first of them will be regarded as the front wall. And you should take into account that someplace before this number in the sequence there must be a number not exceeding 35, otherwise it's not a wall, it's … well, you don't have to know what. Further you have to check two chan… two places where these objects can be. You'll be given for each of the places two numbers, which are the beginning and the end of the search zone counting from the front wall of the layer. You have to search in these zones for an object, that means you have to find maximal numbers in these zones of numbers. If a maximal number is equal to or greater than a certain level, which you'll be given, then you report that you have found an object and give us its location. If there are several maximal numbers in a search zone, we regard only the first of them as an object. There's another complication: the width of the layer is not constant though it has a lower bound, and you'll have to watch for a signal from the back wall of the layer. This signal is also a number 95 or greater and it must be not earlier than after a certain amount of numbers from the front wall. Note that somewhere between the front and the back wall there must be a number not exceeding 35. Of course, if you don't find a signal from the front wall, you don't have to search for objects or the back wall at all. And you don't have to search for anything behind the back wall. I suppose you understand that a signal from the back wall may fall into a search zone, then you don't regard it as an object. Is everything clear to you?

— And if…

— No ifs. You're free, you may go now. Wait, one more thing: according to our recent investigations, the signal decays inside the layer, so after you have found the front layer you have to multiply each subsequent number by a decay coefficient and only then you may analyze it. The decay coefficient for the m-th number is calculated by the formula $1 + (m - f) \cdot A$, where f is the location of the front wall, and the value A will be given to you later. That's all, now go.

Input

The input begins with 9 numbers separated with spaces or ends of line. These numbers are:

1. the total amount of numbers in the sequence N ($4 \le N \le 10000$);
2. the minimal layer width D ($1 \le D \le N - 1$), which is the minimal possible difference between the positions of the numbers corresponding to the back wall and the front wall;
3. the beginning of the first search zone S1 ($1 \le S1 \le N - 1$) (if S is the position of the front layer, then S1 + S is the first number of the search zone);
4. the end of the first search zone E1 ($S1 \le E1 \le N - 1$) (the point S + E1 must also be checked);
5. the level at which an object is reported in the first zone L1 ($1 \le L1 \le 100$; if a signal is equal to or greater than L1, then there is an object, whose location is the first maximal number in the search zone);
6. S_2,
7. E_2,
8. L_2 are similar numbers for the second search zone;
9. the parameter for calculating the decay coefficient A ($0 \le A \le 0.1$; A has no more than 3 digits after decimal point).

The following N lines contain a record of the signal; in each line there is one number from 0 to 100.

Output

If the front wall of the layer was not found, you should output “No surface.”. If the front wall was found, you should output “Surface found at *.”, where there is the position of the front wall instead of * (the very first number of the sequence has position 1). In this case the next line of the output should contain your report concerning the back wall: either “Bottom found at *.”, where instead of * there is the difference between the positions of the back wall and the front wall, or “No bottom.” The next line should contain information on the first search zone. If no numbers were analyzed in this zone, you should output “Channel 1: No search.”, otherwise there should be one of the two messages: either “Channel 1: Object at *.”, if an object was found, where instead of * there is the difference between the positions of the object and of the front wall, or “Channel 1: No object.”, if no object was found. The last line should contain an analogous information on the second search zone (note that there must be “Channel 2” in this line).

Sample

input	output

10 4 1 1 50 2 2 56 0.09 9 26 96 56 21 73 73 93 96 97	Surface found at 3. Bottom found at 4. Channel 1: Object at 1. Channel 2: No object.

Problem Author: Alexander Mironenko
Problem Source: USU Personal Contest 2004

1289. One Way Ticket

Time limit: 1.0 second
Memory limit: 64 MB

A crowed of volunteers dressed in the star striped overalls have filled the starport. There starcraft leaves to the thorium mines of Haron. Their job will be hard and dangerous. Will many of them return home?

This question has an exact answer. Each of the volunteers has a ticket to the starcraft. There signed a compartment number where he is to fly. The starcraft is not passenger, it departs to Haron once a month full of the volunteers and returns back loaded with thorium. This precious metal can be loaded not in all of the compartments. In such a compartments volunteers can return home. All the others are to stay at Haron... Forever.

The compartments are numbered. The numbers of the unfit compartments (and only their numbers) are so that the digit root of the first half of the number equals to the digit root of the second half. To calculate the digit root of a number one is to summarize the digits of the number; if the result is greater or equal than 10, then the digits of the result are to be summarized again; and so on until one gets a number from 0 to 9.

How many volunteers will return home?

Input

Even integer number $N \leq 200000$ – an amount of digits used to number the compartments. There are exactly 10^N compartments numbered from 0 to 10^N-1 with the leading zeros.

Output

The number of compartments free of thorium on the way home.

Sample

input	output
2	10

Problem Author: Leonid Volkov
Problem Source: USU Personal Contest 2004

1290. Sabotage

Time limit: 1.0 second
Memory limit: 64 MB

It is the seventh year of the terrible harmful Galaxy War... Leo Hao is one of the first defenders of his planet. He is lucky! He has gone through many troubles. For example, he stayed alive after the close combat with a meklon warrior – a perfect killing machine. He lost his left leg, right eye and spent five long months in hospital. After that incident, he had to leave the army and return to the Earth.

But Leo is lucky twice! He was able to find a good job after all these terrible incidents. Now he is a leading programmer in “U.S. Robots”. He was involved into the creation of software for zero level defense system. However, even there he was faced with interplanetary intervention! Just a few days ago it was found out that one of his co-workers is not a human! No! Physically he was a human of course, but parasitical Darloxian – agent of the most odious race in the Galaxy, captured his mind.

Obviously, mind corrupted by Darlok agent was not able to write high-quality code. That why Leo is now reviewing his code. It’s terrible!!! It is not effective, slow, dirty and tangled. It must be rewritten!

However, Leo faced trouble during the exploration of the following function: input is the array of positive integer numbers. First, function prints quantity of numbers in the array onto a sheet of paper. Then quantity of numbers in the array greater than 1 is printed. Then quantity of numbers greater than 2 and so on, until the function encounters zero (zero is never printed out). After that, special mechanical manipulator puts this sheet of paper into scanner, which reads this set of numbers into memory and the described operation repeats again. After that the new paper with numbers comes out from the printer. The scanner reads these new numbers, and stores them into the array. This array is the result of the function.

Example. Input: 4 1 6.
After first stage printer prints 3 2 2 2 1 1
After second stage the result of the function will be 6 4 1

Leo feels that it can be done more effectively. Your goal is to write a program, which will be able to replace the function written by Darlok agent, and will be much faster.

Input

First line of input contains the number N ($0 \le N \le 25000$). The next N lines contain integers p_i ($1 \le p_i \le 25000$) one per line. It is the input for the described function.

Output

Output should contain the result of the function, written by Darlok Agent.

Sample

input	output
3 4 1 6	6 4 1

Problem Author: Folklore, prepared by Den Raskovalov
Problem Source: IX Open Collegiate Programming Contest of the High School Pupils (13.03.2004)

1291. Gear-wheels

Time limit: 1.0 second
Memory limit: 64 MB

— Arny! What happened with coordinator?

Bad working coordinator was the everlasting trouble of their spaceship. Arny already had been working under this trouble while his not very attentive and responsible mate just noticed the breakage. Judging by schematics the broken module of coordinator consists of the set of special gears connected with each other in order to transfer the traction from the kinetic generator to the lot of antenna driving engines.

Despite the extraterrestrial origin of these gears, they are connected by usual terrestrial method: the cogs of one gear-wheel get into the slots between cogs of another gear-wheel. So the rotation of the first gear-wheel is transmitted to the second gear-wheel.

After the multiple Arny's revisions, no unnecessary gears stayed in the coordinator. It means that there is no cycles in gears connection graph. The only problem now is to check that all the gears have right directions and speeds of rotation.

Input

First line of input contains the number of gear-wheels in mechanism N ($1 \le N \le 1000$). The next N lines contain the information about the gear-wheels. i-th line contains K ($1 \le K \le 1000$) — the number of cogs on the i-th gear-wheel followed by the list of gears, that are connected to the i-th one. Zero ends the list.

The last line of input contains the number of gear-wheel, that is connected to the kinetic-generator and the speed of its rotation V ($1 \le V \le 1000$). This gear-wheel rotates in counter-clockwise direction.

Output

Output should contain N lines. In the i-th line there is a speed of rotation of i-th gear-wheel in the form of irreducible fraction. Numerator and denominator of this fraction should be separated by the sign '/'. If speed is negative, it is assumed that the gear-wheel rotates in clockwise direction (in this case the minus sign should be displayed before numerator). Otherwise the gear-wheel rotates in counter-clockwise direction. If speed equals zero than numerator should be equal 0 and denominator should be equal to 1. It is guaranteed that neither numerator nor denominator of all speeds will be greater than 10^6.

Sample

input	output
4 10 2 3 0 20 1 0 40 1 4 0 100 3 0 1 6	6/1 -3/1 -3/2 3/5

Problem Author: Pavel Egorov
Problem Source: IX Open Collegiate Programming Contest of the High School Pupils (13.03.2004)

1292. Mars Space Stations

Time limit: 1.0 second
Memory limit: 64 MB

BANG! Loud sound of warning sirens had filled the living and service rooms of the space cruiser “Admiral Brisko” — flagship of 3rd earth fleet. It was on the way to Sirius, when light Zerg scout-ship appeared from nowhere.

— Capta-a-ain!
— Quiet! Damage report!
— Captain! Before the enemy ship was destroyed, it had damaged our left reactor, left engine and the onboard computer. We are doomed! They will intercept and destroy us!

It was absolutely useless to even try to get to Sirius without left engine. The same was about returning to the Earth: enemy ship certainly reported their coordinates to the rest of Zerg fleet, and light Zerg interceptors could be already on the way to their feeble cruiser with only one goal. And this goal was not to help them to repair the left engine…
There was only one solution — try to reach the nearest Mars space station. Uh! You don’t know what is a Mars space station?! It is a new defensive weapon of Martians — mobile, well defended and almost invulnerable space fort for defense of strategically important routes of Union of Five (union of the Earth, Mars, Venus, Andromeda and Sirius). There are many space stations located in special way along such important routes. Every space-week, in order to maintain secrecy the positions of space stations change according to weekly updates of the Secret Mars Key (SMK)!

— Call the programmer here! — bark out the captain.

Let’s guess what is your role here? Right! Here is your play! You task is to write program for calculation of the distance to the next space station — “Andromeda-Sirius-4”. It’s good that the secret algorithm of the station positioning is not a secret at all! Mmm… May be except Martians themselves… It’s quite simple: first base is located at the distance equal to SMK from the beginning of the path (Sirius in our case). The next station is located at the distance of F(SMK) from the first. Third station — at the distance of F(F(SMK)). And so on. Here F — is the Top Secret Mars Function (TSMF) It’s value is the sum of the cubes of digits its argument in decimal notation (for example F(12) = 1^3 + 2^3 = 9). So if the distance from the (I − 1)-th to I-th stations is X, then the distance between I-th and (I + 1)th stations is F(X). “What a nonsense?!” — you’ll say. And you will be absolutely right, but you should not be so strict — the authors of that idea were just small pretty downy rabbits — Martians.

Your cruiser is located between (N − 1)-th and N-th space stations at the distance of L from (N − 1)-th station. Taking N, K (Secret Mars Key) and L as input your program should output the distance M between your cruiser and N-th station. Oh, by the way! The value of SMK is always divisible by 3. It’s normal for Martians — all their numbers are divisible by 3.

Input

Number T ($2 \le T \le 33333$) is placed in the first line of input — it is the number of tests for your program. It followed by the next T lines. Each of these T lines contains 3 integer numbers: N ($2 \le N \le 33333$), K ($3 \le K \le 33333$) and L ($L \ge 1$).

Output

T lines. I-th line contains the calculated value of M for I-th test case.

Sample

input	output
2 4 6 123 7 93 49	18 104

Problem Author: Denis Musin
Problem Source: IX Open Collegiate Programming Contest of the High School Pupils (13.03.2004)

1293. Eniya

Time limit: 1.0 second
Memory limit: 64 MB

It’s restless now on the slips of the intergalactic port’s sixth dock of planet of Torn. No longer then in a month the reconstruction of the small ironclad corvette “Eniya” will be finished. And again this battle ship and its brave team would have to struggle for the control over plutonium mines of Sibelius. The work cannot be stopped even for a second, self-powered laser welders work round the clock. Joints of robots-repairers fuse because of this permanent work. Nevertheless, they can’t stop not for a single moment.

Now in all this turmoil it is discovered that corvette’s thermopanels again need an urgent processing with thorium sulphide. It is known that the processing of the one square meter of the panel needs 1 nanogramm of sulphide. In general, it is needed to process N rectangular panels, which dimensions are A by B meters. It is necessary to calculate as fast as possible, how much sulphide is needed in general for the processing of all panels of “Eniya”. Moreover, do not forget, that the panels need processing of both sides.

Input

The only line contains integers N ($1 \le N \le 100$), A ($1 \le A \le 100$), B ($1 \le B \le 100$).

Output

Output the weight of thorium sulphide in nanogramms needed for the processing.

Sample

input	output
5 2 3	60

Problem Author: Den Raskovalov
Problem Source: IX Open Collegiate Programming Contest of the High School Pupils (13.03.2004)

1294. Mars Satellites

Time limit: 1.0 second
Memory limit: 64 MB

Four artificial satellites travel in one plane along the areostationary orbit around Mars. They have code names A, B, C and D and travel exactly in this order. Venus's scouts for military purposes (for what particular purpose they did not say) decided to find a distance between satellites C and D. All Mars satellites could measure distances to the other satellites, that is why all what is needed to do is to penetrate in the computer system of satellite C and measure the distance to satellite D (or vice versa). Nevertheless, Martians are not so stupid and have not very bad defense. That is why all what could Venus's scouts do is to break the defense of satellites A and B (that were older models). They measured distances from satellites A and B to satellites C and D, but now they do not know how to find the distance from C to D using these measurements. You can help them.

Input

There are 4 numbers: distances from A to D, from A to C, from B to D and from B to C in thousands kilometers (integers from 1 to 10000). Satellites can measure distance even through the planet and you may assume that orbit is a circle. Do not assume the radius of the orbit equal to 20392 km as it should be for the real areostationary orbit.

Output

If it is impossible to find out the distance from C to D with these data, you should print "Impossible.", otherwise you are to print "Distance is X km.", where X is the required distance in kilometers (rounded to the integer number).

Sample

input	output
4 7 5 7	Distance is 5385 km.

Problem Author: Vladimir Yakovlev
Problem Source: IX Open Collegiate Programming Contest of the High School Pupils (13.03.2004)

1295. Crazy Notions

Time limit: 0.5 second
Memory limit: 64 MB

For five days robot-loader JK546L54p has been buried under the thick layer of the Sibelian plutonium slag. The terrible strike of the atmospheric electricity has led to the depressurization of the robot's fuel elements. Who will examine this heap of fused, broken metal here, where there is no any robot technician even at distance of a hundred parsecs? Robot-commissar even did not try to investigate what happened with JK546L54p. He ordered to throw him out into dumps and that is all. Nobody noticed that positron brains of JK546L54p were still working. If only the robopsychologist was here with JK546L54p! Of course, he would be killed with the hard gamma radiation in a moment, but… If he attached the visualizer of thoughts to the fused connectors of JK546L54p! He would see the strange performance. Robot was creating! No, I am not joking. He was investigating. Semi casual objects arose in his mind, and he examined them. Crazy properties, crazy theorems.

Besides, here is an example. Let's take an expression $1^n+2^n+3^n+4^n$. How much zeros does its decimal notation end with? JK546L54p solved this problem, and you, student, could you?

Input

The only line contains an integer n ($1 \le n \le 300000$).

Output

Output the number of zeroes the decimal notation of $1^n+2^n+3^n+4^n$ ends with.

Samples

input	output
1	1
3	2

Problem Author: Den Raskovalov
Problem Source: IX Open Collegiate Programming Contest of the High School Pupils (13.03.2004)

1296. Hyperjump

Time limit: 1.0 second
Memory limit: 64 MB

Developed in the beginning of XXI century, hyperjump remains the primary method of transportation for distances up to thousands parsecs. But physicists have recently discovered an amazing phenomenon. They believe the duration of the hyperjump alpha phase can be easily controlled. Alpha phase is the period when hyper-spacecraft accumulates its gravity potential. The larger is the gravity potential accumulated, the less energy is required to complete the hyperjump. Your task is to write a program, which would help pilots decide when to enter and when to leave the alpha-phase, in order for the hyperspacecraft to accumulate the largest possible gravity potential.

The most crude gravity field model (which you will have to use) yields the sequence of integers p_i, which represent field intensities at different moments in time. According to this model, if the alpha-phase begins at moment i and ends at moment j, then the value of gravity potential accumulated will be equal to the sum of sequence elements at places from i-th to j-th inclusive.

Input

The first line of the input contains an integer N being the number of elements in the intensity values sequence ($0 \le N \le 60000$). Next N lines specify sequence elements, each line containing a single integer p_i ($-30000 \le p_i \le 30000$).

Output

The only line of output contains the largest possible value of the gravity potential that can be accumulated by a hyperspacecraft during the alpha phase. You should assume that the initial gravity potential of a hyperspacecraft is equal to zero.

Samples

input	output
10 31 -41 59 26 -53 58 97 -93 -23 84	187
3 -1 -5 -6	0

Problem Author: Den Raskovalov
Problem Source: IX Open Collegiate Programming Contest of the High School Pupils (13.03.2004)

1297. Palindrome

Time limit: 1.0 second
Memory limit: 64 MB

The “U.S. Robots” HQ has just received a rather alarming anonymous letter. It states that the agent from the competing «Robots Unlimited» has infiltrated into “U.S. Robotics”. «U.S. Robots» security service would have already started an undercover operation to establish the agent’s identity, but, fortunately, the letter describes communication channel the agent uses. He will publish articles containing stolen data to the “Solaris” almanac. Obviously, he will obfuscate the data, so “Robots Unlimited” will have to use a special descrambler (“Robots Unlimited” part number NPRx8086, specifications are kept secret).

Having read the letter, the “U.S. Robots” president recalled having hired the “Robots Unlimited” ex-employee John Pupkin. President knows he can trust John, because John is still angry at being mistreated by “Robots Unlimited”. Unfortunately, he was fired just before his team has finished work on the NPRx8086 design.

So, the president has assigned the task of agent’s message interception to John. At first, John felt rather embarrassed, because revealing the hidden message isn’t any easier than finding a needle in a haystack. However, after he struggled the problem for a while, he remembered that the design of NPRx8086 was still incomplete. “Robots Unlimited” fired John when he was working on a specific module, the text direction detector. Nobody else could finish that module, so the descrambler will choose the text scanning direction at random. To ensure the correct descrambling of the message by NPRx8086, agent must encode the information in such a way that the resulting secret message reads the same both forwards and backwards.
In addition, it is reasonable to assume that the agent will be sending a very long message, so John has simply to find the longest message satisfying the mentioned property.

Your task is to help John Pupkin by writing a program to find the secret message in the text of a given article. As NPRx8086 ignores white spaces and punctuation marks, John will remove them from the text before feeding it into the program.

Input

The input consists of a single line, which contains a string of Latin alphabet letters (no other characters will appear in the string). String length will not exceed 1000 characters.

Output

The longest substring with mentioned property. If there are several such strings you should output the first of them.

Sample

input	output
Kazak	aza

Problem Author: Eugene Krokhalev
Problem Source: IX Open Collegiate Programming Contest of the High School Pupils (13.03.2004)

1298. Knight

Time limit: 1.0 second
Memory limit: 64 MB

Even paratroopers have vacations. The flight to Sirius in the depths of “The Admiral Brisco” Leo Hao whiled away with chessboard. No, he did not like usual chess game, and in addition, he did not have likely rival. The whole day Leo amused himself with an interesting thing: he tried to travel over all cells of the chessboard with the knight so that the knight visited each cell only one time. Leo attempted one time, then second, but always something was wrong. Leo became a little angry. Then he attempted board 4*4 instead of 8*8. Again failure after failure. A little angry, with the chessboard under his arm, Leo went to look for a local programmer. They two together indeed will solve this problem.

Input

The only line contains an integer N ($1 \le N \le 8$).

Output

If it is possible to travel with the knight over the square field $N \times N$ cells, then output should contain N^2 lines with tour over the chessboard with mentioned property, otherwise the only word “IMPOSSIBLE”.

Samples

input	output
3	IMPOSSIBLE
5	a1 c2 e1 d3 e5 c4 d2 e4 c5 a4 b2 d1 e3 d5 b4 a2 c1 e2 c3 b1 a3 b5 d4 b3 a5

Notes

If you have solved this problem, pay attention to the problem "Knight Mare".

Problem Author: Folklore. Prepared by Den Raskovalov
Problem Source: IX Open Collegiate Programming Contest of the High School Pupils (13.03.2004)

1299. Psylonians

Time limit: 1.0 second
Memory limit: 64 MB

The Psylonian race is known to wield by far the most advanced military battle-robots. During the fight for Terra, it took four of these battle-robots only three hours to completely wipe out one of the ancient cities with adjacent territories. You'd better watch your back with them. The Earth Defense Forces command is utterly preoccupied. Personnel is being trained for military operations against Psylonians in case the armed conflict bursts out again. But working out good tactics requires modeling of the battle-robot's artificial intelligence unit. And you are to help us!

The robot's behavior turned out to be rather simple. He can use just four basic patterns: "Defense", "Attack", "Guard" and "Patrol". Moreover, the behavior control unit you have to model implements a rather simple scheme: it takes some parameters as input and outputs one of the following commands for the robot to execute:
LEFT X – spend X energy units for turning left;
RIGHT X – spend X energy units for turning right;
FRONT X – spend X energy units for moving forward;
BACKWARD X – spend X energy units for moving backward;
FIRE P – spend P fire units for attacking the enemy;
STOP – nothing to do.

In all these cases X is calculated as follows: If the robot's power supply has more than 100 energy units available, then X = 100; otherwise, X is the amount of power units remaining. Similarly, P is calculated as the smallest of two values: 20 and the actual amount of ammunition units remaining.

The most simple behavioral pattern is "Guard". When it's active, robot does not walk, it can only turn. It picks the most dangerous target, and determines angle between robot's current line of sight and direction to the target. If the angle is less than 5 degrees (by absolute value), robot shoots. Otherwise, if the angle is greater than or equals to 5, the robot turns left; if the angle is less than or equals to -5, the command is to turn right. If there are no targets around, robot does nothing.

According to the "Defense" pattern, robot's actions are defined as follows: calculate the number of enemies and multiply it by 20. If the result is not less than the amount of ammunition units left, the robot will "Retreat and Return Fire". Otherwise, it will act as if following the "Guard" pattern.

The "Attack" pattern works as follows: If N*NP > M*MP*3, then the robot will "Advance and Return Fire"; else it will "Retreat and Return Fire". Here: N is the total number of allied battle-robots; NP is the average amount of energy units they have remaining; M is the total number of enemy robots; MP is the average amount of energy units the enemy robots have remaining.

When the battle-robot "Retreats and Returns Fire", it checks the angle to the most dangerous enemy. If the angle is 5 degrees or more, or there are no enemies, the robot just moves backward. Otherwise, it fires its guns.

Similarly, when the battle-robot "Advances and Returns Fire", it checks the angle to the most dangerous enemy. If the angle is 10 degrees or more, or there are no enemies, the robot just moves forward or shoots otherwise.

When "Patrolling", the robot follows its waypoint route, if there are no enemies. If the robot encounters enemy when patrolling, then it acts as if the "Defense" pattern was active. When following a route, robot uses this algorithm to move to the next waypoint:

If the angle to the next waypoint exceeds 20 degrees by absolute value, the robot turns into direction to waypoint. Note that it does not matter whether the robot is moving forward or backward while following the route. You should consider this when calculating the turn direction - the turn angle should be minimal.

Otherwise, if the angle to the next waypoint does not exceed 20, then the robot just moves towards the next waypoint.

Input

The first line contains two integers separated by space; the first integer is the amount of energy units available, the second one is the amount of ammunition available. The second line contains a single character, which determines the operating pattern: G stands for "Guard", D for "Defense", A for "Attack", and P for "Patrol". On the third line there are three integers: amount of enemies, average amount of energy units enemy bots possess, and angle to the most dangerous enemy target.

If the robot follows the "Attack" pattern, then the fourth line contains two integers: amount of allied bots and average amount of power units remaining in their power supplies, respectively. If the robot follows the "Patrol" pattern, then the fourth line contains only one integer: angle to the next waypoint.

Angle values range from -179 to 180 inclusive, all other integers are non-negative and none of them exceeds 1000.

Output

The command for the robot using the format described above.

Samples

input	output
100 100 P 0 0 0 -100	LEFT 100
10 1000 A 1 10 30 100 1000	FRONT 10
10 1000 G 1 10 4	FIRE 20

Problem Author: Pavel Egorov
Problem Source: IX Open Collegiate Programming Contest of the High School Pupils (13.03.2004)

1300. Taxes

Time limit: 1.0 second
Memory limit: 64 MB

There is a taxation system based on the known parameters $N_1 \ldots N_m$, $S_1 \ldots S_{m+1}$. in some country.

Let the total annual income of a citizen is equal to K bibriks (*bibrik* is a local currency). A citizen pays S_1 percents from the sum N_1 bibriks and less, he pays S_1 percents of N_1 plus S_2 percents of $(K-N_1)$ from the sum from N_1 to N_2. If the income is $N_2 \le K \le N_3$ then the citizen must pay S_1 percents of N_1 plus S_2 percents of (N_2-N_1) plus S_3 percents of $(K-N_2)$ and so on.

If the total annual income exceeds N_m then he is to pay S_1 percents of N_1 plus S_2 percents of (N_2-N_1) plus … plus S_{m+1} percents of $(K - N_m)$.

Let the function $T(K)$ be the total tax from the annual income K. Moreover, in the assumed country the government pays to the citizens additional L percents as the regional coefficient. This coeficient is taxed independently from the salary.

I.e. if the salary of a citizen is R bibriks he may legally spend $R-T(R) + L\%*R-T(L\%*R)$. The taxes to pay are $T(R) + T(L\%*R)$.

The calculation of the regional coefficient and all kinds the taxes is made with rounding off to two digits after a decimal point. The initial income is given within two digits after a decimal point too.

Each employer automatically retains the taxes in favor of the government from the annual income of his employee taking into account the regional coefficient.

If a citizen works at several places, in the end of a year he is to make a recalculation of the payed taxes because the sum of retained taxes in different places may be not equal to the tax from the total income.

You are to write a program that calculates and outputs the difference between the tax from the total income and the sum that is retained by all his employers.

Input

The first line contains the regional coefficient L. Then there are pairs $N_1\ S_1$, $N_2\ S_2$, …, $N_m\ S_m$, $0\ S_{m+1}$. Each pair is located in a separate line. The number in pairs are separated by one or several spaces. The values of L and coefficients S_i are given in percents (i.e. they are integers from 0 to 99).

Each of the next lines contains the total annual net profit of the citizen on some enterprise, where he worked, taking into account the regional coefficient and taxes.

The input is ended by −1. All the parameters are integer not greater than 10^9 within two digits after a decimal point. The number of different workplaces of the citizen doesn't exceed 30. An amount of taxation coefficients (m) is not greater than 20.

Output

The difference between the tax from the total income and the sum that is retained by all his employers with exactly two digits after a decimal point.

Sample

input	output
15 12000000 12 24000000 20 36000000 25 48000000 30 0 35 12000000 12000000 -1	937233.19

Problem Source: II Collegiate Students Urals Programming Contest. Yekaterinburg, April 3-4, 1998

1301. Cube in Labyrinth

Time limit: 1.0 second
Memory limit: 64 MB

There is a cube on the rectangular X × Y board. The cube side is equal to the side of a cell of the board. During one turn the cube may roll over its edge moving to the vertically or horizontally neighboring cell. There may be walls between some cells that are obstacles. The cube may not roll over the obstacles. The cube may not leave the board.

You are to find the minimal number of turns necessary to move the cube from the initial point with coordinates A and B to the given final point with coordinates C and D. Moreover, in the final position the upper side must be the same as it was in the initial position.

Input

The first input line contains two numbers X and Y separated with one or more spaces. The second line consists of the numbers A and B, and the third line consists of the numbers C and D presented in the same way. Then an information about the walls may follow. All the numbers are integers; $2 \le X,Y \le 10$.

After a symbol ‘v’, located in a separate line, there are pairs of integers describing the walls. Here the pair of numbers M and N define a wall between the cells N, M and N+1, M. Each pair of numbers is located in a separate line.

After a symbol ‘h’, located in a separate line, there are pairs of integers describing the horizontal walls in the same way. The pair M, N define a wall between the cells N, M and N, M+1.

Output

The only line containing the minimal number of moves. If such a displacement is impossible, you should output “No solution”.

Sample

input	output
10 2 1 1 10 1 v 2 1 6 2 h 4 1	11

Problem Source: II Collegiate Students Urals Programming Contest. Yekaterinburg, April 3-4, 1998

1302. Delta-wave

Time limit: 1.0 second
Memory limit: 64 MB

A triangle field is numbered with successive integers in the way shown on the picture below.

The traveller needs to go from the cell with number m to the cell with number n. The traveller is able to enter the cell through cell edges only, he can not travel from cell to cell through vertices. The number of edges the traveller passes makes the length of the traveller's route.

Write the program to determine the length of the shortest route connecting cells with numbers m and n.

Input

Input contains integers m and n separated with space ($1 \le m, n \le 10^9$).

Output

Output the length of the shortest route.

Sample

input	output
6 12	3

Problem Source: II Collegiate Students Urals Programming Contest. Yekaterinburg, April 3-4, 1998

1303. Minimal Coverage

Time limit: 1.0 second
Memory limit: 64 MB

Given set of line segments $[L_i, R_i]$ with integer coordinates of their end points. Your task is to find the minimal subset of the given set which covers segment [0, M] completely (M is a positive integer).

Input

First line of the input contains an integer M ($1 \le M \le 5000$). Subsequent lines of input contain pairs of integers L_i and R_i ($-50000 \le L_i < R_i \le 50000$). Each pair of coordinates is placed on separate line. Numbers in the pair are separated with space. Last line of input data contains a pair of zeroes. The set contains at least one and at most 99999 segments.

Output

Your program should print in the first line of output the power of minimal subset of segments which covers segment [0, M]. The list of segments of covering subset must follow. Format of the list must be the same as described in input with exception that ending pair of zeroes should not be printed. Segments should be printed in increasing order of their left end point coordinate.

If there is no covering subset then print “No solution” to output.

Samples

input	output
1 -1 0 -5 -3 2 5 0 0	No solution
1 -1 0 0 1 0 0	1 0 1

Problem Source: II Collegiate Students Urals Programming Contest. Yekaterinburg, April 3-4, 1998

1304. Parallelepiped

Time limit: 0.5 second
Memory limit: 64 MB

Two opposite vertices of the parallelepiped A with the edges parallel to the datume lines, have coordinates (0, 0, 0) and (u, v, w) correspondingly ($0 < u < 1000$, $0 < ;v < 1000$, $0 < w < 1000$).

Each of the n points of the set S is defined by its coordinates (x(i), y(i), z(i)), $1 \le i \le n \le 50$. No pair of points of the set S lies on the straight line parallel to some side of the parallelepiped A.

You are to find a parallelepiped G of the maximal volume such that all its sides are parallel to the edges of A, G completely lies in A (G and A may have common boundary points) and no point of S lies in G (but may lie on its side).

Input

The first line consists of the numbers u, v, w separated with a space. The second line contains an integer n. The third, …, (n+2)-nd line – the numbers x(i), y(i), z(i)separated with a space.

All coordinates are non-negative, not greater than 1000 and written with not more than two digits after a decimal point.

Output

One number – the volume of G with two digits after a decimal point. If the true volume has more than two digitrs after a decimal point you should round off the result to two digits after a decimal opint according to the common mathematical rules.

Sample

input	output
1.0 1.0 1.0 1 0.5 0.5 0.5	0.50

Problem Source: II Collegiate Students Urals Programming Contest. Yekaterinburg, April 3-4, 1998

1305. Convex Hull

Time limit: 1.0 second
Memory limit: 64 MB

Let a finite set of points M be defined on plane. The plane has a usual Cartesian coordinates. Well-formed convex hull of set M is minimal (relative to inclusion) convex set, containing M, and bounded by closed broken line. All sections of this broken line should be parrallel to axes or inclined by 45°.

Your task is to find a well-formed convex hull for a given set M.

Input

In the first line an number N (1<=N<=100 000) of following lines is written. In the second and all next lines coordinates of set's points is written. In every line there are coordinates (two numbers separated several spaces, each number is greater or equal to 0 and less or equal to 1000) of only one point. Some points of set can overlap, thus the same coordinates can be found in different lines.

Output

Your program should print the sequence of broken line's vertices. Vertices should be enumretated in the counter clock-wise order. As a first vertex any of them can be taken. In every line exactly one vertex's coordinates (two numbers, separated by spaces) should be put out. Every vertex of broken line should be mentioned only once.

No three consecutive vertices of broken line should lie on a straight line.

Sample

input	output
4 3 3 3 1 2 2 4 2	3 1 4 2 3 3 2 2

Problem Source: II Collegiate Students Urals Programming Contest. Yekaterinburg, April 3-4, 1998

1306. Sequence Median

Time limit: 1.0 second
Memory limit: 1 MB
Language limit: C, C++, Pascal

Given a sequence of N nonnegative integers. Let's define the median of such sequence. If N is odd the median is the element with stands in the middle of the sequence after it is sorted. One may notice that in this case the median has position $(N+1)/2$ in sorted sequence if sequence elements are numbered starting with 1. If N is even then the median is the semi-sum of the two "middle" elements of sorted sequence. I.e. semi-sum of the elements in positions $N/2$ and $(N/2)+1$ of sorted sequence. But original sequence might be unsorted.

Your task is to write program to find the median of given sequence.

Input

The first line of input contains the only integer number N — the length of the sequence. Sequence itself follows in subsequent lines, one number in a line. The length of the sequence lies in the range from 1 to 250000. Each element of the sequence is a positive integer not greater than $2^{31}-1$ inclusive.

Output

You should print the value of the median with exactly one digit after decimal point.

Sample

input	output
4 3 6 4 5	4.5

Notes

In C++, it is recommended to use stdio instead of iostream to save a reasonable amount of memory.

Problem Source: II Collegiate Students Urals Programming Contest. Yekaterinburg, April 3-4, 1998

1307. Archiver

Time limit: 1.0 second
Memory limit: 4 MB

It is custom to start each problem given at a programming contest with a 'tale', in order to link the problem to the real world as well as to fog its essence, especially if the essence seems too easy to understand. But this problem has no tale, because, first, it is quite unusual, and, second, the problem itself is about brevity.

Suppose that we are given a text. An archive of this text is a text satisfying the following requirements:

1. An archive is a program in one of the programming languages allowed by the rules of the contest.
2. The first line of the archive is “{PAS}”, or “/*C*/”, or “//CPP”.
3. After compiling and executing an archive, we obtain the original text.
4. The length of an archive is strictly less than the length of the original text.

You should write a program that outputs an archive for a given text. The archive is compiled and executed with the same parameters and restrictions that are used for compiling and executing the submitted program containing the solution of the problem. The archive might not to be in the same language as a generating it solution. Checking the problem the judges determines the archive language according to the first line («{PAS}» — Pascal/Delphi, «/*C*/» — C, «//CPP» — C++).

Input

The input contains a text of length not less than 20000 and not more than 200000 symbols. The text may contain capital and lower-case English letters, digits, punctuation signs, line breaks, and quotation marks. It is guaranteed that all the texts used as tests for this problem are literary texts in English.

Output

You should output an archive of the text given in the input.

Sample

input
12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455565 7
output
//CPP #include<iostream.h> int main() {for(int i=1;i<58;i++)cout<<i;return 0;}

Notes

The sample input is just an illustration, it does not satisfy the requirements since it is too short and not a literary text.

Problem Author: Idea - Leonid Volkov, prepared by Pavel Egorov and Leonid Volkov
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1308. Dean's Pyramid

Time limit: 1.0 second
Memory limit: 64 MB

I guess everyone who was in the cabinet of dean of USU math-mech faculty remembered the glass pyramid lying on the deans table. There is a legend that several students remembered this pyramid on all treir lifes as a part of a hard test. The everlasting question of every dean – to sent down a weak student or to give him the last chance.

The legend says that to get this last chance some weak students was to bring this pyramid by 70 rolls from one point of the table as closer as possible to given one another. Usually the destination point was the point of the deans table where the list of dismissed students lied. At the end of its path pyramid should stand on its base. And ideally pyramid should cover the locus of deans signature on the dismissing list.

After the student spent all his rolls or after he confessed that he couldn't bring it closer, the dean measured the distance between the center of pyramids base and the destination point. The legend says that the student was given the desired last chance, if this distance was record-breaking small.

May be it's just a beautiful legend but you can go to the deans cabinet right now and make sure that the pyramid lays on the table and the golden fog charming swirls in its depth… So, just in case, you'd better to train a little in rolling the pyramid from one point to another.

You can assume that the pyramids base is square and its lateral faces – regular triangles. You can roll the pyramid by turning it from one face to the adjacent one around some edge. During this turning the edge should not slide on surface of the table. Moreover to make the test harder the dean demands you to obey such a rule: if after the turn around some edge the pyramid stands on its base, the next turn can be performed either around the same edge or around the opposite edge of the pyramids base only. There are no any restrictions on the rolling from the triangle faces.

Input

Input contains two real numbers – coordinates of the destination point. The pyramids edge length is concidered to be equal to 1 in the coordinate system. The origin coincides with the center of the pyramids base at the initial moment. The edges of the pyramids base at the initial moment are parallel to the coordinate axes.

Output

Output should contain only one real number – the minimal possible distance between the center of the pyramids base after rolling and the destination point within 4 digits after a decimal point. The base edges may be not parallel to the coordinate axes at the final moment of time, but the pyramid should stand on its base. You can perform not more than 70 turns of pyramid during its rolling.

Sample

input	output
2.3660254037 1.3660254038	0.0000

Problem Author: Idea - Stanislav Vasilyev, prepared by Pavel Egorov, Alexander Mironenko
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1309. Dispute

Time limit: 0.5 second
Memory limit: 64 MB

Dispute is a great thing! It is known that the truth is born in a dispute. Two organizers of the Ural Championship have an argument. The first of them says that computing the value of a function is a very stupid and useless problem for a programming contest. His reasoning is that when the definition of a function is known and there is enough time for the necessary preparations, it is possible to calculate the value of the function at any point very fast. The second organizer asserts that not any function can be calculated fast enough. To resolve this dispute, they decided to make an experiment. So you are to prove that you are really able to calculate the value of a function at any point fast enough.

The function f(n), where n is an integer, is defined recursively by the following expressions:

- f(0) = 0,
- f(n) = g(n, f(n-1)),

where $g(x,y) = ((y-1)x^5 + x^3 - xy + 3x + 7y) \% 9973$, the symbol % denotes taking the residue of division.

Input

The only line contains an integer n ($0 \le n \le 10^8$).

Output

You are to write a program that outputs the value f(n). And it should perform the necessary computations very fast!

Sample

input	output
50	6300

Problem Author: Idea - Alexander Klepinin, prepared by Alexander Klepinin, Stanislav Vasilyev
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1310. ACM Diagnostics

Time limit: 0.5 second
Memory limit: 64 MB

Have you ever heard about ACM? May be, may be… It is known to be a new project of computing machinery lab. ACM (Abstract Computation Machine) — it is a new device for quite complex computations. There is only one problem: the pre-production model doesn't work properly. So, the diagnostics is necessary to find out the reasons of the device faults.

There are some necessary facts known about the internal structure of the device. It consists of L independent registers. Each of these registers is able to store the number in the range from 1 to M. Not all registers are used for computations — some of them are used for the proposes of error control. The values of these registers are chosen in such way, that the sum of the values of all registers is divisible by given number K.

The values of all registers of ACM completely define the state of device. So for proposes of diagnostics it is enough to know the values of all these registers. The problem is that developers who created the diagnostics procedure had decided to optimize the presentation of diagnostics information. So instead of simple list of register values the procedure returns the single integer — code of the state.

This code should completely describe the state of device. That's why developers decided to calculate it such way. State of device — is an arbitrary vector of length L, satisfying the demands defined above. And the code of the state is just an index of this vector in the lexicographically increasing list of all possible states. (Note that the first state has index 0). It's easy to understand that such vector completely defines a state of the ACM.

Now the device faults and outputs code of the state. The error code has become useful! Now it is necessary to reconstruct the values of device registers from this code. As you guessed — it is your entry!

Input

The first line of input contains three numbers L, M and K ($1 \le L \le 100$; $2 \le M \le 50$; $1 \le K \le 50$) The second line contains an integer number N — error code, returned by the ACM device.

Output

You are to write the program that outputs the values of device registers, corresponding to the code N. I.e. L integer numbers, separated by spaces.

Sample

input	output
3 10 4 213	9 6 1

Problem Author: Idea - Alexander Klepinin, prepared by Alexander Klepinin, Pavel Egorov
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1311. Stable Construction

Time limit: 0.5 second
Memory limit: 64 MB

As they say, every man must build a house, plant a tree, and grow a son during his life. So a programmer Vasechkin decided not to stay behind and to start building his own house. Vasechkin knew that his taste was not bad and thus he himself decorated the facade with brickwork. He carefully attached each brick with cement solution to a vertical wall. The result was very nice. Each brick was placed horizontally and lay on no more than one brick from the lower row. Vasechkin finished his work and thought. The time would pass and the cement solution between the bricks would crack. And one would like this beauty to remain for ages. So would it be stable even without the cement? It is necessary to perform complicated calculations, taking into consideration that all bricks have the same height, density, and width, but different lenghts. Also, a brick (or a system of bricks) with center of gravity on the edge of its support or outside it is considered to be unstable.

Input

The first line contains an integer H, which is the number of brick rows, i.e., the height of the wall ($1 \le H \le 10000$). Then the disposition of bricks in the rows is given, row by row, beginning from the topmost one. For each row, there is a line containing an integer K, which is the number of bricks in this row ($0 \le K \le 1000$); each of the next K lines contains two integers L_i and R_i, which are the coordinates of the left and right edges of the i-th brick. It is known that $L_i < R_i < L_{i+1}$ and $0 \le L_i, R_i \le 10000$. The number of all bricks is between 1 and 100 000.

Output

You should write a program that outputs “Yes” if all the bricks are stable and “No” otherwise.

Sample

input	output
3 1 10 20 3 1 7 13 17 100 200 2 0 20 60 160	Yes

Problem Author: Idea - Alexey Lakhtin, prepared by Alexey Lakhtin
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1312. Tray

Time limit: 0.5 second
Memory limit: 64 MB

Once I dropped into a cafe to have a snack. I took a first course (a big round plate), a salad (a smaller round plate), and a cup of tea (on a small saucer). I tried to arrange all this on a rectangular tray, which was quite a problem. “Yes, a problem! A nice problem for the Urals Championship!”, I thought.

Now you are to solve this problem. I will give you the size of the tray and of the three plates, and you should try to arrange the plates on the tray.

Input

The input contains the sides of a rectangular H and W and the radii of circles R_1, R_2, R_3. All the numbers are positive integers and do not exceed 10^6. The numbers are separated with spaces or line breaks.

Output

You should output the only number 0 if the required arrangement is impossible. Otherwise, you should output three pairs of numbers, which are the coordinates of the centers of the plates. The coordinates must be given in the following coordinate system: the origin is the corner of the tray, the X axis is directed along the side of the tray whose length is given first, the Y axis is directed along the other side of the tray. The order of the pairs of coordinates should correspond to the order of the radii given in the input data.

Samples

input	output
800 400 200 200 50	200.0000 200.0000 600.0000 200.0000 400.0000 350.0000
800 400 200 200 51	0

Problem Author: Idea - Alexander Petrov, prepared by Alexander Petrov and Alexander Mironenko
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1313. Some Words about Sport

Time limit: 0.5 second
Memory limit: 64 MB

Ural doctors worry about the health of their youth very much. Special investigations showed that a lot of clever students instead of playing football, skating or bicycling had participated in something like Programming Olympiads. Moreover, they call it sports programming! To sit near the monitor and think during 5 hours a day – is it a sport? To do it two times per year during the contests – it is more or less normal, but during the preparations to the nearest contest they spend several hours a week sitting at their computers! It would be possible to understand if they were some blockheads and dunces, but they are ones of the best students all over the world!

To save students from the harmful habit to sit at the computer for hours, Ural doctors has invented a fundamentally new monitor with diagonal trace of a beam in its electron-beam tube. Soon the winners of Ural Programming Championship would be awarded with such monitors. In the specially designed square monitor the electronic beam would scan the screen not horizontally but diagonally. The difference of the lengths of different diagonals causes such effects as non-uniform brightness, flashing and non-linear distortions. The terrible properties of such monitors would break of the habit of looking at the monitor for hours. There is a little problem: the majority of computer video cards generates the normal “rectangle” signal for monitor. So it is necessary to develop special adapter-program, which should transform the usual “rectangle” signal to the signal necessary for this kind of monitors. Program should be fast and reliable. That’s why the development of this program is entrusted to the participants of the Ural Championship for Sports Programming.

Input

The first input line contains the single integer N ($1 \le N \le 100$) – the number of pixels on the side of new square monitor. It is followed by N lines, each containing N positive integers not exceeding 100 divided by spaces. It is the image outputting by the usual video card (as you can see the color depth of new monitor is not so large – anyway usual programmer does not need more than 100 colors).

Output

You are to write the program that outputs the sequence for input into the new monitor. Pixels are numbered from the upper-left corner of the screen diagonally from left ot right and bottom-up. There is no need to explain details – look at the sample and you'll understand everything.

Sample

input	output
4 1 3 6 10 2 5 9 13 4 8 12 15 7 11 14 16	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Problem Author: Idea - Stanislav Vasilyev, prepared by Stanislav Vasilyev and Alexander Klepinin
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1314. Chase in Subway

Time limit: 0.5 second
Memory limit: 64 MB

The police has let a criminal slip. He has disappeared in the complicated grid of subway lines, where a chase is senseless. But the criminal does not know that there is a radio beacon attached to his clothes. The beacon sends a signal to the police from each of the stations visited or passed through by the criminal (it is not possible to detect a signal from a tunnel between stations, because the signal is too faint). Having the information about the sequence of stations passed by the criminal, the police wants to determine the stations where the criminal might be going to, in order to set watch posts at these stations.

The police knows that the criminal behaves quite logically; he has a goal (the subway station where his shelter is located), and he is moving there using one of the shortest routes. For the criminal, the length of a route is determined by the number of spans only (in the subway, a span is a tunnel between two adjacent stations). The length of a route does not depend on the lengths of spans or the number of line changes.

Input

The first line of input contains the number N of lines in the subway ($1 \le N \le 50$). Each of the next N lines contains a description of the corresponding line. A description starts with an integer K ($2 \le K \le 50$), which is the number of stations of the line. Then there are the numerical indices of the stations of the line, in the order in which the stations are on the line. The indices are integers from 1 to 32767. All the numbers are separated with spaces. If there is the same station index in the descriptions of two different lines, then these lines have an intersection at this station, where a change can be made. The last line of input contains surveillance data: an integer $M \ge 1$, which is the number of stations where the criminal was registered, and M numbers, which are the indices of these stations in the order in which the criminal visited them.

Output

You should output the indices of all stations that can be the goal of the criminal, in the ascending order, one per line.

Sample

input	output
3 2 61 62 5 75 20 85 50 61 3 10 20 30 3 30 20 85	50 61 62 85

Problem Author: Idea - Leonid Volkov, prepared by Alexander Somov, Leonid Volkov, Igor Goldberg
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1315. MDPAR and MIIAR

Time limit: 1.0 second
Memory limit: 64 MB

Speleology is a very interesting occupation. But it is also quite risky. According to the statistics, accidents are most often in spring, and many of them are caused by unexpected cave floods. In the last few years, there were several rescue operations, and each of them required quite a lot of financial resources and manpower. For the sake of economy, the Ministry for Extreme Situations issued the following order.

Order № 321/1.
For the sake of economy of resources at spring rescue operations, it is ordered to:

1. Create a database of all caves and all speleologists of the Russian Federation.
2. Put into geostationary orbits 12 satellites S-349857 to make possible the exact determination of the location of speleologists in caves.
3. Employ programmers to develop systems of satellite control.
4. Create a device interacting with the satellites for automatically issuing rescue instructions to a speleologist. The device specification is given in Appendices A and B.
5. Oblige speleologists to have special equipment for urgent communication with a rescue center, including the device described in Article 4.

Minister.

Appendix A. The device specification.
The device has two modules.

1. The module for detecting the possibility of automated rescue (MDPAR):
   - determines which part of a cave is filled with water for the known configuration of the cave;
   - determines whether automated rescue is possible if the location of a speleologist and the maximal duration of his underwater stay are known.
2. The module for issuing instructions for automated rescue (MIIAR):
   - given the location of a speleologist, determines the direction of movement guaranteeing reaching the surface.

Appendix B. The principles of filling caves with water.
This document is a result of investigations of the Institute for Cave Studies. A cave is filled with water according to the following rules:

1. The cave is regarded as a collection of cubicles.
2. A cubicle is filled with water if there is a path from this cubicle to the surface.
3. A path is a sequence of cubicles that have common side.

Here only the paths having no downward segments are considered.

You are to implement the MDPAR.

Input

The first line contains 5 integers W, H, X, Y, D, which are respectively the width and depth of a cave (in cubicles), the X and Y coordinates of a speleologist, and the number of cubicles through which he can swim without air. The following H lines describe the configuration of the cave. Each of these lines contains W characters: "#" denotes a wall, i.e., a cubicle inaccessible both for the speleologist and for water, and a dot "." denotes air, i.e., a cubicle that is possibly accessible for the speleologist and can be filled with water. The module should be able to operate in the following ranges of the parameters: $1 \le W, H \le 500$; $1 \le X \le W$; $1 \le Y \le H$. $1 \le D \le 1000$. Cubicles are numbered from left to right, from bottom to top. It is known that a speleologist can reach the surface while cave is not filled with water.

Output

You should output "Can be rescued by himself" if the speleologist can reach the surface following the instructions issued by the MIIAR. Otherwise, you should output "Rescue operation required".

Samples

input	output

9 4 8 2 5 ##..##..# ##.#.#### ##......# #########	Rescue operation required
9 4 8 2 6 ##..##..# ##.#.#### ##......# #########	Can be rescued by himself

Problem Author: Idea - Alexander Klepinin, prepared by Alexander Klepinin, Ivan Dashkevich
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1316. Electronic Auction

Time limit: 0.5 second
Memory limit: 64 MB

There is a deficit in cast-iron pigs in the country. They are sold at an electronic auction. Customers make their bids: announce a price at which they are ready to buy a pig. From time to time a seller puts up for sale K pigs at a price of X bibriks each. The first K customers who offered the same or higher price get one pig each.

Customers may cancel their bids (after a purchase a bid remains valid until it is canceled). Only bids made in a current month are valid, so each month a customer should renew his bid. If a seller did not sell all the pigs offered for sale, then the unsold pigs remain at his storehouse and don't participate in the auction any more.

Each sold cast-iron pig makes a profit of 0.01 bibriks for the auction. Having a month's log of auction operations, you are to calculate the profit of the auction in this month.

Input

The input contains a month's operations log, one operation per line. There are three types of operations:

- "BID X" — a customer announces that he is ready to buy a pig at a price of X bibriks;
- "DEL X" — a customer cancels his bid for a pig at a price of X bibriks;
- "SALE X K" — a seller puts up for sale K pigs at a price of X bibriks.

X is between 0.01 and 10000.00 bibriks and has at most 2 digits after the decimal point. K is an integer between 1 and 100000. The number of operations does not exceed 100000. All operations are correct. The last line contains the word "QUIT".

Output

Output the profit of the auction in the current month with 2 digits after the decimal point.

Sample

input	output
BID 0.01 BID 10000 BID 5000 BID 5000 SALE 7000 3 DEL 5000 SALE 3000 3 SALE 0.01 3 QUIT	0.06

Problem Author: Idea - Pavel Atnashev, prepared by Pavel Atnashev, Pavel Egorov
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1317. Hail

Time limit: 1.0 second
Memory limit: 64 MB

A special device was mounted to defend a car parking from a hail. Large hailstones are detected by a special radar and evaporated by a powerful laser. The parking lot has the shape of a convex polygon and is surrounded by a high fence. The height of the fence is h meters. The laser is located somewhere strictly inside this polygon at the ground level. The laser can evaporate hailstones, which pass at a distance of no more than d meters from it. The fence is impenetrable for laser beams. Hailstones fall vertically downward. It is known that the location and power of the laser are such that it can destroy all hailstones falling to the territory of the parking lot before they reach the level of the upper edge of the fence. The laser destroys all hailstones it can, regardless of whether they would fall inside or outside the fence. The laser can perform all the necessary operations instantly, so it can destroy several hailstones in different places practically simultaneously. Knowing the coordinates of the places where hailstones would fall, you should determine how many of them will be destroyed by the laser.

Input

The first line contains an integer n ($3 \le n \le 10$), which is the number of polygon vertices, and a real number h ($1.00 \le h \le 100.00$), which is the height of the fence. The following n lines contain pairs of real numbers, which are coordinates (in meters) of the polygon vertices. The vertices are given in the order of bypass. The next line contains the number d ($h < d < 1000.00$) and the coordinates of the laser. It is followed by a line containing the number of hailstones k ($1 \le k \le 100$). And the following k lines contain pairs of real numbers, which are x and y coordinates of hailstones. Absolute values of all coordinates do not exceed 1000.

Output

You are to write a program that outputs exactly one integer, which is the number of destroyed hailstones.

Sample

input	output
4 10.00 1.00 0.00 0.00 1.00 -1.00 0.00 0.00 -1.00 50.00 0.00 0.00 5 0.00 0.00 1.00 1.00 2.00 2.00 3.00 3.00 4.00 4.00	3

Problem Author: Idea: Pavel Atnashev, prepared by Pavel Atnashev, Alexey Lakhtin
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1318. Logarithm

Time limit: 1.0 second
Memory limit: 64 MB

Given a set A of N unordered 128-bit numbers. You are to compute a value of the function

where A_k is the k^{th} element of A, $\log_{10}X$ — the integer part of the decimal logarithm of X. We'll assume that $\log_{10}0 = 0$.

Input

The first input line contains a number $N \le 5000$. In the following N lines there are 128-bit numbers A_k presented by sets of numbers (a_{1k}, a_{2k}, a_{3k}, a_{4k}), each of them lies in range from 0 to 2^{32}-1. The number A_k can be obtained from this set according to the formula

$A_k = 2^{96}a_{1k} + 2^{64}a_{2k} + 2^{32}a_{3k} + a_{4k}$.

Output

You are to output the value of the function for the given set.

Sample

input	output
2 0 0 0 2324 0 2332 0 0	44

Problem Author: Idea: Nikita Shamgunov, prepared by Nikita Shamgunov, Anton Botov
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1319. Hotel

Time limit: 1.0 second
Memory limit: 64 MB

— You programmers are lucky! You don't have to deal with these terrible people – designers… This story happened with me not so long ago. We had an order from a company building a new hotel. One day they brought a sketch to our workshop. They said that THIS was invented by a very cool designer. They said they had paid heaps of money for THIS. So, THIS had to be built. In general, THIS was not a very complex thing. It was just a square set of shelves where a porter puts guests' mail. Usual hotels have usual stands with shelves for this purpose. But this cool designer had turned everything upside down! To be more precise, not exactly upside down, but upon a corner. Moreover, the cells should be numbered from the right to the left, from the top to the bottom, looking at THIS, staying on its corner, of course. Tell me please, how can the master attach the labels with numbers to THIS? He will look on the shelves, staying normally on its side, you know. He will get tangled on the fourth label already! I will get tangled on the seventh, myself… Actually one should make such designers to label the shelves themselves.

— Oh! You are the cool programmer, I know. Couldn't you help me? I need just a printout of the table with an arrangement of the labels in the cells. But not in such way as THIS will hang on the wall, but as THIS stands on the table of my workshop. Yes, I understand that you are busy, but you are busy every time! Preparations to the Ural Championship, tests, solutions… So what? If you can't do it yourself – entrust your competitors with this task. They are the best programmers all over the world, aren't they? I don't believe that they couldn't print the desired table having the size of the square! I would never believe it! So… Excellent! I will take the desired printout away after the contest.

Input

The input consists of the only one integer N ($1 \le N \le 100$), which is the size of the square.

Output

You are to write a program that outputs the table of numbers, as they would be arranged when THIS would stand in the workshop. The label with number 1 should be in the upper right corner and other numbers should be arranged along the diagonals from the top to the bottom. The label with the last number (N*N) should be in the lower left corner.

Sample

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

Problem Author: Stanislav Vasilyev
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1320. Graph Decomposition

Time limit: 0.5 second
Memory limit: 64 MB

There is a simple graph with an even number of edges. You are to define if it is possible to present it by the set of pairs of adjacent edges (having a common vertex).

Input

contains a non-empty sequence of the integers pairs. The length of the sequence is even and is from 2 to 1050. Each pair of integers denotes vertices identifiers of one edge. All the identifiers are from 1 to 1000. You may assume that there are no loops and multiple edges in the graph defined by the input data.

Output

“1” (without quotation marks), if the decomposition is possible and “0” otherwise.

Samples

input	output
1 2 2 3 3 1 1 10	1
1 2 2 3 3 1 4 10	0

Problem Author: Idea: Alexander Petrov, prepared by Alexander Petrov, Leonid Volkov
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1321. Floor Indicator

Time limit: 0.5 second
Memory limit: 64 MB

"Let's go!", thought a hotel's manager entering an elevator. He pressed the tenth floor button and meditated. The day was not easy. The manager looked at the floor indicator, saw the number 9, and prepared to get out. But the elevator did not stop. The nine gave place to eight, then to seven. The manager became amazed. He remembered precisely that he had entered the elevator at the first floor. He was sure that the elevator goes up. Yes, it was not an easy day, but not to such an extent! Then he saw eight instead of seven, then there was nine again, then ten, and the elevator stopped.

The strange behavior of the elevator worried the manager. The next morning he decided that the problem was with the floor indicator, and so a repairman should be called for.

The repairman comes by a helicopter, enters the building through a window at one of the floors, gets into the elevator, and goes several floors up or down comparing the numbers on the indicator with the numbers of floors. The indicator can show several digits, and each digit place has 7 short linear indicating lamps shown here:

These lamps allow to show any digit:

The indicator does not show leading zeros and has no "extra" lamps, that is lamps which will never light up in this building. A properly working lamp switches on or off when it is needed; a defective lamp is always on or always off. During his journey in the elevator, the repairman must find all the defective lamps. For the sake of economy, it is necessary to minimize the number of passages between floors needed for this work. The floors are numbered with successive integers starting with 1.

Input

The only line contains the number of floors in the building N ($4 < N < 10^{1000}$).

Output

You should output the minimal number of passages between adjacent floors that the repairman should go in the elevator in order to find all the defective lamps.

Sample

input	output
10	8

Problem Author: Idea Leonid Volkov, prepared by Pavel Egorov, Igor Goldberg
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1322. Spy

Time limit: 0.25 second
Memory limit: 64 MB

The secret service detected an acting foreign agent. Frankly speaking — a spy. A surveillance showed that each week the spy sends strange unreadable texts to somebody via the Internet. In order to find out which information became available to the spy, it is necessary to decipher the texts. Secret service agents got into the spy's apartment, discovered a cipher machine, and found out the principle of its operation.

An input of the machine is a text line $S_1 = s_1s_2...s_N$. The machine constructs all cyclic permutations of this line, i.e., $S_2 = s_2s_3...s_Ns_1$, ..., $S_N = s_Ns_1s_2...s_{N-1}$. Then the set S_1, S_2, ..., S_N is sorted lexicographically in the ascending order, and the lines are written out in this order in a column, one under another. Thus an array N × N is obtained. One of the rows of this array contains the initial word. The number of this row and the last column of the array are the output of the machine.

For example, if the initial word S_1 = abracadabra, then the following array is formed:

1. aabracadabr = S_{11}
2. abraabracad = S_8
3. abracadabra = S_1
4. acadabraabr = S_4
5. adabraabrac = S_6
6. braabracada = S_9
7. bracadabraa = S_2
8. cadabraabra = S_5
9. dabraabraca = S_7
10. raabracadab = S_{10}
11. racadabraab = S_3

In this case, the output of the machine is the number 3 and the line rdarcaaaabb.

So it is clear how the cipher machine operates. However, no deciphering machine was found. But as the information can certainly be deciphered (otherwise there is no sense in sending it), you have to invent a deciphering algorithm.

Input

The first and the second lines contain an integer and a string respectively. This is the output of the cipher machine. Both the number and the length of the string do not exceed 100000. The string may contain only the letters a-z, A-Z and the underlining character. The lexicographic order on the set of words is determined by the following order of characters:

ABCDEFGHIJKLMNOPQRSTUVWXYZ_abcdefghijklmnopqrstuvwxyz

The characters here are given in the ascending order.

Output

The only line should contain the initial message.

Sample

input	output
3 rdarcaaaabb	abracadabra

Problem Author: Idea: Alexander Klepinin, prepared by Alexander Klepinin, Stanislav Vasilyev
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1323. Classmates

Time limit: 1.5 second
Memory limit: 64 MB

Tanya almost left for school when the telephone rang. It was the director of studies. She said the first three lessons that day had been cancelled because of an electricity failure. Tanya was the head girl of the class and the director of studies asked her to pass this news to her classmates.

— What shall I do? — thought Tanya, — there is almost no time! OK, now I'm going to call Lena, then Katya, then Masha. Lena will meantime call Vitya, she knows his telephone number, Vitya will call Masha. No, I'll call Masha myself: let him better call Misha. Katya will call Natasha... No, it won't work. They quarreled yesterday. Thus there is no time to think. I must immediately call Lena. Hit-or-miss everyone will know the news.

Tanya managed to send this message to all her classmates. But someone knew it very late and someone heard this news from several people. In the evening, Tanya decided to work out a plan of calls and not to let it ride the next time. After all, she is the head girl of the class!.. But the problem turned out to be not so easy.

Help Tanya to work out a plan of calls such that a news might be delivered to all the pupils as soon as possible. All the pupils of the class must receive the message but not more than once. It takes exactly one minute to pass the news over the telephone. At the beginning only the head girl knows the news.

To solve the problem, Tanya wrote down the list of her classmates and for each classmate the list of those whom he or she might call. You may assume that if Masha can call Katya, then Katya can call Masha, too (even if only one connection is mentioned in the list). It is known that a message can be delivered to everyone in the class through a sequence of calls.

Input

The first line contains the number of pupils N in Tanya's class ($1 \le N \le 10$). The second line contains the integer number M ($0 \le M \le 45$). Each of the following M lines contains a pair of pupil's names who can call each other separated by space. The last line contains the name of the head girl. All the names in the class differ and consist of no more than 20 capital and small Latin letters.

Output

The first line of the output should contain the time in minutes necessary to spread the news to all the class according to the suggested plan. Then there is a description of the plan. The calls that should be made simultaneously must be arranged in groups. Groups should be ordered according to the time. Each group should start with a line containing the amount of calls in the group. Each call must be described in a separate line. The description of call consists of a pair of names (who calls and whom) separated by a space.

Sample

input	output
6 7 Tanya Lena Tanya Katya Tanya Masha Lena Natasha Lena Vitya Natasha Vitya Masha Vitya Tanya	3 1 Tanya Lena 2 Tanya Masha Lena Vitya 2 Vitya Natasha Tanya Katya

Problem Author: Idea: Evgeny Kobzev, prepared by Pavel Atnashev, Alexander Mironenko
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1324. Extra Spaces

Time limit: 0.5 second
Memory limit: 64 MB

A programmer Petrov took part in a programming contest outside his home university for the first time. There he suddenly understood that using an alien computer was not so nice. The computer he was working on even didn't have his favorite text editor. And, unfortunately, the program committee had given a text formatted by a nasty DOS text editor in such a way that all right ends of lines were at the same level. Of course, it had been performed by inserting extra spaces in many places. To read such a text was a torture for Petrov. It was his luck he found the FAR Manager, which could help to delete all these disgusting spaces replacing a combination of two spaces by one space. However, there were too many spaces, so such operation had to be performed several times, because after a replacement FAR did not search for the specified combination in the processed text. For example, if there are six successive spaces, then after one round of replacement the first two spaces are replaced by one space, the middle two spaces are replaced by one space, and the last two spaces are replaced by one space. As a result, we have three successive spaces. The second round of replacement deletes the first two of the three spaces and puts one space instead of them. So we need one more round of replacement, which replaces the remaining two spaces by one space. On the whole, three rounds of replacement are needed.

Petrov had already pressed Ctrl+F7, but than had a sudden thought: what if he first replaced each three spaces by one, and then each two spaces by one? So six successive spaces would be processed by two operations only! But which sequence of operations would be optimal if a text contained rows of spaces of a length not exceeding N?

Your task is to determine the minimal number of replacement rounds (each of which replaces rows of spaces of a certain length by one space) necessary for processing a text containing sequences of spaces of any length from 1 to L. You should also offer a scheme of the replacements. If there are many such schemes, then you should choose an optimal scheme among them, i.e., a scheme that also reduces any sequence of up to K spaces ($K \geq L$) for a maximal possible K. If there are several optimal schemes, you may give any one of them.

Input

The input contains an integer L, $L < 2000000$, which is the maximal length of a row of spaces in the text.

Output

The first line of the output should contain an integer R, which is the minimal number of replacement rounds. The next R lines should describe an optimal scheme of replacements. Each of these lines must contain the length of row which are replaced by one space during the correspondent round.

Sample

input	output
22	4 6 3 2 2

Problem Author: Idea: Stanislav Vasilyev, prepared by Stanislav Vasilyev, Igor Goldberg
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1325. Dirt

Time limit: 0.5 second
Memory limit: 64 MB

— Hello, may I speak to Petrov, please? Hello, my darling… You know, there was a little accident at our home… No, no, don't worry, your computer was not damaged. It is only a bit dirty there now. Well, I should say it's very dirty there and I'm at my Mom's now. Of course, I'll clean it… When? Well, maybe when I have my vacation. What? Well, when we are back from Turkey… the next vacation then. I'll stay at Mother's until then, and you may live here also. No, no, I don't insist, sure, you may stay at home if you wish so. I prepared boots for you, they are at the door. But please, don't make it worse, before you step on a clean floor, change your boots, put on your slippers, they are at the door also. Take them with you when you walk through the dirt. And when you walk on a clean floor, take the boots with you. You see, the dirt is in different places. OK, my love? Thank you!

It is not a great pleasure to change boots each time you get from a clean floor to a dirty floor and vice versa, it's easier to walk extra several meters. So it is necessary to find a way of getting from one place in the apartment to another with the minimal possible number of boots changes; and among these paths the shortest one must be found.

To begin with, it is natural to determine an optimal way of passing the Most Important Route: from the computer to the refrigerator.

Input

The first line of the input contains two integers M and N, which are dimensions of the apartment (in meters), 1 <= N, M <= 500. The two integers in the second line are the coordinates of the computer, and the third line contains the coordinates of the refrigerator. Each of the following M lines contains N symbols; this is the plan of the apartment. On the plan, 1 denotes a clean square, 2 denotes a dirty square, and 0 is either a wall or a square of impassable dirt. It is possible to get from one square to another if they have a common vertex. When you pass from a clean square to a dirty one or vice versa, you must change shoes. The computer and the refrigerator are not on the squares marked with 0.

The upper left square of the plan has coordinates (1, 1).

Output

You should output two integers in one line separated with a space. The first integer is the length of the shortest path (the number of squares on this path including the first and the last squares) with the minimal possible number of boots changes. The second number is the number of boots changes. If it is impossible to get from the computer to the refrigerator, you should output 0 0.

Sample

input	output
3 7 1 1 3 7 1200121 1212020 1112021	8 4

Problem Author: Idea: Stanislav Vasilyev, prepared by Stanislav Vasilyev, Pavel Egorov
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1326. Bottle Taps

Time limit: 3.0 second
Memory limit: 64 MB

Programmer Petrov has a hobby to collect beer-bottle taps. There's nothing unusual — he knows hundreds of programmers that like beer. And they collect taps, too. Not everyone, but some of them.

Frankly speaking, he has bought a part of his collection. But unfortunately he hasn't got some rare taps to complete his collection. He has found some programmers over the Internet that are ready to sell him these taps. Some of the programmers sell the taps in sets with big discounts.

It's left to find an optimal offer. Petrov can explain to his wife why he is to store the taps but he won't be able to prove why he is to spend money for the collection. So he is to buy the taps as cheap as possible.

Petrov has written down all the variants and has started thinking. There's no way to find out the solution of the problem without a program!

Input

The first line contains an integer N, an amount of available taps ($1 \le N \le 20$). The following N lines contain prices of bottles with the taps if one buys them in stores. The next line contains an integer M ($0 \le M \le 100$) — an amount of offers to sell the taps. The following M lines describe the sets. The first number of each line is the price of the set and the second one is the amount of taps in the set. Then there are numbers of the taps in the set (each number lies in the range from 1 to N). The numbers in a set are unique. All the prices are positive integers and do not exceed 1000. The last line begins with the amount of taps that Petrov plans to buy. Then their numbers follow separated by spaces. These numbers are unique, too.

Output

Output the minimal sum of money that Petrov should spend on obtaining the necessary taps.

Sample

input	output
4 10 11 12 13 3 17 2 1 3 25 3 2 3 4 15 2 3 4 3 1 3 4	25

Problem Author: Idea: Evgeny Kobzev, prepared by Pavel Atnashev, Alexander Mironenko
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

1327. Fuses

Time limit: 1.0 second
Memory limit: 64 MB

"Janus Poluektovich (I don't remember anymore whether -A or -U) used the machine only once. He brought with him a small semitransparent box, which he connected to the Aldan. In approximately ten seconds of operation with this device, all the circuit breakers blew, and Janus Poluektovich apologized, took his box, and departed."

Sasha Privalov, a young programmer working in the SRITS (Scientific Research Institute for Thaumaturgy and Spellcraft), finds his job rather enjoyable. Indeed, he is the only programmer of such a wonderful machine as Aldan-3 - that's a refreshing shift from a dull job in Leningrad. There is just a single problem, and the problem's name is Janus Poluektovich.

On Privalov's first workday, Janus burdened Aldan with the task of four-dimensional convolution in the conjuration space. Aldan worked for a while, flashing its lights and rewinding tapes, then a fuse blew and the machine shut down. Well, replacing fuses is something even a programmer can do. But Janus is rather absent-minded, and he, being lost in thoughts about his convolution problem, forgot about the weak fuse next day. So, on a third day Janus launched his program again, blowing another fuse. The fourth day went calmly, but on a fifth day one more fuse had to be replaced. And Janus is still not going to give up…

Nevertheless, these accidents don't bother Sasha, as long as he has enough spare fuses.

Your task is to help Sasha in making the requisition for spare parts. The requsition is made for a specific period - from the A-th workday to the B-th workday inclusive. You should calculate, how many fuses Janus is going to blow with his programs in the specified period of time.

Input

The first line contains an integer A. The second line contains an integer B. $1 \le A \le B \le 10000$.

Output

The output should contain one number - the amount of fuses that will be blown by Janus in the interval from day A until day B.

Samples

input	output
1 5	3
100 200	50

Problem Author: Den Raskovalov
Problem Source: The 10th Collegiate Programming Contest of the High School Pupils of the Sverdlovsk Region (October 16, 2004)

1328. Fireball

Time limit: 1.0 second
Memory limit: 64 MB

Today almost everybody knows about the scientific research department of the Night Watch, located in Solovetz city. Due to the artful actions of Zavulon (the boss of Day Watch, you know) this Scientific Research Institute for Thaumaturgy and Spellcraft (SCITS) was absolutely declassified and removed from secret list already in 60s.

However this fact had not made any harm to its ability of research work. For example right now 3^{rd}-level wizard Vitka Korneev tests a new battle-spell of fireball in his lab in SCITS. Oh… fireball is such a ball of fire that is used for… m-m-m… for neutralization of undesirable consequences.

New fireball appeared to be just an ingenious invention! First of all due to the incongruence of transgression inside the incub-transformation's psy-field it has a zero radius. But its greatest characteristic is the ability of remaining stable during the predefined number of collisions with obstacles. This characteristic is called *N*-stability: fireball is *N*-stable if it stays stable after *N* collisions but explodes after $(N + 1)^{th}$ collision. So, you may consider, that *N*-stable fireball loses one level of stability and becomes $(N - 1)$-stable after each collision with a wall. For example ordinary fireball is 0-stable. So with this invention it became possible to strike an enemy with fireball after several ricochets from the walls. So the military value of this invention is beyond questions. In addition, new *N*-stable fireball ($N > 0$) has quite unusual behavior: After collisions it rebounds only from concrete walls! So, it easily flies through any other obstacles. (The theory ties this fact with the accumulation of bio-emotional energy by all static constructions of living quarters). This fact, as you can guess, causes additional military value of new invention: now it is not necessary to provide clear trajectory for the thrown fireball — it will fly thwough any obstacles before it damages the target.

But it is long way from the first prototype to the mass usage. First of all it is necessary to investigate the trajectory of the fireball flight. The following experiment is prepared for this purpose: in the rectangle room two points *A* and *B* are being chosen at random. One wizard stands at the point *A* and the target is placed at the point *B*. Wizard creates *N*-stable fireball while his assistant calculates the direction of throw with the help of special program. The direction of throw is selected such way that thrown fireball rebounds from the walls exactly *N* times and then hits the target. At the same time it should do this with the shortest trajectory (i.e. as quickly as possible).

So, you are to write this special program for direction calculation. The scientists of SCITS tell that all fireballs rebound from the walls according to the law: "angle of incidence equals angle of reflection". And after collision with room's corner it rebounds exactly in the opposite direction. Moreover, the theory of fireballs says that, due to continuity, one collision with corner equals two collisions with walls. So, 2-stable fireball explodes after the second collision if the first was with room's corner. And finally you may assume that the fireball is a point moving in straight lines with constant velocity.

Input

The first line contains two numbers — width and length of the room. The second line contains the number *N* — *N*-stability of fireball. The third line contains four numbers — coordinates of points *A* and *B*.

All numbers are integers and are separated by one or more spaces. Points *A* and *B* lie inside the room but not on its border. The room's width and length do not exceed 1000 and are greater than 1. *N* is between 0 and 10 inclusive.

Output

Angle in degrees (with 2 digits after decimal point), that gives the desired direction of fireball. If there are several such angles your program should output the minimal one.

Angle and coordinates are measured as shown on the figure.

Sample

input	output
1000 10 3 101 5 128 8	45.00

Notes

Characters and background are taken from books "Monday Begins on Saturday" (Arkady and Boris Strugatsky) and tetralogy "Night Watch", "Day Watch", "Twilight Watch" and "Final Watch" (Sergey Lukyanenko).

Problem Author: Pavel Egorov
Problem Source: The 10th Collegiate Programming Contest of the High School Pupils of the Sverdlovsk Region (October 16, 2004)

1329. Galactic History

Time limit: 1.0 second
Memory limit: 64 MB

It is very hard for one person to learn all galactic history. But, on the other hand, every diplomat who wants to hold a more important post in a galactic empire must know the subject well. For example, letting a spoon fall among high-rankers of the star system Arcturus means offending them awfully. (Didn't you hear that the last conflict between systems Arcturus and Alpha flamed up because of the only shattered glass?)

Fortunately, the solution was found in the Galactic Academy. For diplomats of the lower rank it is enough to learn just a single branch of history – the one that concerns only the cluster of star systems, in which he is going to work. (Diplomats of the lower rank negotiate only with planets that are located in one star cluster. How come we didn't guess this earlier?)

Taking this very important observation into consideration, it was decided to replace a single intergalactic course with several separate courses, each covering only the part of history that refers to only one star cluster. Of course, it is necessary to learn history in chronological order, beginning from the origin of humanity. That's why the history of the Earth needs to be included in all collections of separate histories. Then things become complicated: for example, emigrants from Centaurus system colonized the star system of Herdsman, so the textbook on the history of Herdsman system has to contain the early history of Centaurus system. In order to decide, in which textbooks which phases of history should be included, historians of Galactic Academy divided general intergalactic history into many small milestones. Then all milestones were combined into one big tree (omnipresent biologists helped historians in this work, as they had always been using these trees). The milestone referring to early history of the Earth (before the space colonization) was declared the root. Milestones referring to history of star systems close to solar system appear to be its sons (because these systems were colonized by emigrants from Earth) and so on. That's all! To determine milestones that have to be included in a particular textbook it is only required to determine quickly, whether the milestone A is located in a subtree with the root in milestone B.

Input

In the first line there is a number N ($N \le 40000$), which defines the total number of milestones. In the next N lines there are descriptions of each milestone.

Each milestone is defined by two numbers: ID – an unique numerical identifier of a milestone and ParentID – identifier of the milestone which is its father in a tree. ParentID for the root equals to -1.

$(N+2)^{th}$ line contains number L ($L \le 40000$) – amount of queries. The next L lines contain descriptions of queries: on each line there are two different numbers A and B. All identifiers lie between 0 and 40000.

Output

For each query it is necessary to write in separate line:

- 1, if milestone A is a root of subtree which contains milesone B.
- 2, if milestone B is a root of subtree which contains milesone A.
- 0, if no one of the first two conditions is true.

Sample

input	output
10 234 -1 12 234 13 234 14 234 15 234 16 234 17 234 18 234 19 234 233 19 5 234 233 233 12 233 13 233 15 233 19	1 0 0 0 2

Problem Author: Evgeny Krokhalev
Problem Source: The 10th Collegiate Programming Contest of the High School Pupils of the Sverdlovsk Region (October 16, 2004)

1330. Intervals

Time limit: 0.5 second
Memory limit: 64 MB

— Superstitions, prejudices… — said the stranger absent-mindedly. — Indolence of mind and envy, envy, shaggy envy… — He interrupted himself. — I beg your pardon, Aleksandr Ivanovich, I would dare to ask your permission to take away this ladle. Unfortunately, iron is almost not transparent for hyperfield, and growth of the hyperfield intensity in a small volume…

After the Vybegallo's "ideal consumer" incident in the Scientific Research Institute for Thaumaturgy and Spellcraft, an automatic security system is being put into operation urgently. It is to guarantee that in any case the total hyperfield intensity won't exceed a critical value. They pin hopes on Sasha Privalov and his Aldan machine to automagically process readings of sensors that are located all over the Institute.

All the sensors are numbered with integers ranging from 1 to N ($1 \le N \le 10000$). The reading of ith hyperfield intensity sensor is integer k_i ($-10000 \le k_i \le 10000$). Aldan is to process quickly queries like "What is the sum of intensities read from the sensors with numbers from i to j ($i \le j$)? The number of queries Q is expected to be rather large ($0 \le Q \le 100000$).

Input

The first input line contains integer N. The following N lines contain k_i numbers (one at a line). Then there are the integer Q and Q pairs of numbers i, j (each pair is in a separate line).

Output

should contain Q lines with the sums of the corresponding intensity sensors readings.

Sample

input	output
5 1 2 3 -1 4 3 1 5 4 4 1 4	9 -1 5

Problem Author: Den Raskovalov
Problem Source: The 10th Collegiate Programming Contest of the High School Pupils of the Sverdlovsk Region (October 16, 2004)

1331. Vladislava

Time limit: 1.0 second
Memory limit: 64 MB

— We call this satellite Vladya. As you know Vladislava has one more satellite, artificial as well, and also extra-terrestrial. It is smaller. We call it Slava, you get it? The planet is called Vladislava and it is natural to call its satellites Vladya and Slava, isn't it?

— Yes, of course,— said Gorbovsky. He was familiar with this beautiful argumentation. He heard it for the third time. — You suggested it very wittily, August. Vladya and Slava — Vladislava. Splendid!

— You call these satellites respectively Y-one and Y-two on the Earth,— continued Bader,— Vladya and Slava. But we, we call them other way. We call them Vladya and Slava.

He looked at Valkenstein strictly. Valkenstein worked his jaws. As far as he knew "we" ment Bader himself and only Bader.

N artifacts of extra-terrestrial civilizations were found on the planet Vladislava.

The scientific spacecraft settled on the planet's orbit and launched automatic probes, which found out that a storm-boat will be able to touch down only in one of M points on the planet's surface because of the rough relief. Inasmuch as the transportation on the planet's surface takes much time, it is reasonable to find the nearest landing place for each artifact.

Input

The first line contains integers N and M ($1 \le N, M \le 5000$) — amounts of artifacts and landing grounds respectively. Then there is positive real $R \le 1000$ — that is the planet radius. Each of the following M lines consists of latitude W_i and longitude L_i of one of the landing grounds, W_i, L_i are real numbers, $|W_i| \le 90$, $|L_i| \le 180$. The next N lines contain artifacts coordinates — latitude w_i and longitude l_i — real numbers, $|w_i| \le 90$, $|l_i| \le 180$. All real numbers are rounded to 6 digits after decimal point.

Output

You are to output exactly N lines. ith line should contain the distance between the i^{th} artifact and the nearest landing ground within two digits after a decimal point.

Samples

input	output
1 2 1 0 0 1 1 0 0	0.00
2 1 1 0 0 0 90 0 45	1.57 0.79

Problem Author: Alexander Bikbaev
Problem Source: The 10th Collegiate Programming Contest of the High School Pupils of the Sverdlovsk Region (October 16, 2004)

1332. Genie Bomber

Time limit: 1.0 second
Memory limit: 64 MB

The last survivor was a certain Pitirim Schwartz, an erstwhile monk and inventor of the forked musket rest, who was selflessly laboring on the genie-bomber project. The essence of the project was to drop on the enemy cities bottles with genies who had been held imprisoned no less than three thousand years. It is well known that genies in their free state are capable only of destroying cities or constructing palaces. A thoroughly aged genie, reasoned Schwartz, was not about to start building palaces, and therefore things would go badly for the enemy. A definite obstacle to the realization of this concept was an insufficient supply of bottled genies, but Schwartz counted on overcoming this through the deep dragging of the Red and Mediterranean Seas.

The genie-bomber project has eventually entered the experimentation stage. Research fellows' doubles have constructed N cities in the testing area. Each of these cities is a circle with a radius of r. As M. M. Kamnoedov provided only one bottled genie for the experiment, experimenters decided to demolish as many cities as possible for the sake of science. It is generally known that a genie demolishes everything in the range of R around the bottle's touchdown point. Any city contained completely within this demolition area is ruined. Before the experiment is conducted, you are required to find the maximum possible number of cities that one genie can ruin.

Input

The first line contains the number N of cities ($1 \le N \le 100$). The following N lines contain the coordinates x_i, y_i of city centers (x_i, y_i are integers, $|x_i|, |y_i| \le 10000$). City centers don't coincide with each other.

The last line contains radius R of the genie's area of destruction and the city radius r ($1 \le R, r \le 10000$). R and r are integers.

Output

Output the maximum number of cities that can be destroyed by one bottled genie.

Samples

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

Problem Author: Alexander Bikbaev
Problem Source: The 10th Collegiate Programming Contest of the High School Pupils of the Sverdlovsk Region (October 16, 2004)

1333. Genie Bomber 2

Time limit: 1.0 second
Memory limit: 64 MB

Thanks to your effortful work, the genie-bomber has successfully passed all the tests. But Modest Matveevich Kamnoedov has some more problems to solve. He is concerned with efficiency of the bombing. Basically, he needs to calculate the area of a testing ground that a genie bombardment can cover.

We will assume that the testing ground is the portion of the plane defined by two inequalities: $0 \le x \le 1$, $0 \le y \le 1$. One genie blast zone covers a circle on the plane. Point of the testing ground is covered with the bombardment if it lies within one or more blast zones. You have to calculate the coverage efficiency – that is, the portion of the testing ground area that is covered with the bombardment.

Input

In the first input line there will be one number N, which is the number of bombs dropped ($N \le 10$). The following N lines contain three numbers x_i, y_i, r_i each, where x_i and y_i define the center of the i^{th} genie bomb blast zone ($0 \le x_i, y_i, r_i \le 1$), and ri defines its radius.

Output

should contain one real number - coverage efficiency percentage (that is, percentage of the testing ground area covered with the bombardment in relation to the total testing ground area, where the testing ground is the square defined by inequalities $0 \le x \le 1$, $0 \le y \le 1$). Precision of the result is required to be 1% or better.

Samples

input	output
1 0.5 0.5 0.5	78.539816
2 0.2 0.2 0.5 0.8 0.8 0.5	80.74292

Problem Author: Den Raskovalov
Problem Source: The 10th Collegiate Programming Contest of the High School Pupils of the Sverdlovsk Region (October 16, 2004)

1334. Checkers

Time limit: 1.0 second
Memory limit: 64 MB

Taking the front staircase, which to my memory was used only once when the most august personage from Africa came to visit, I descended into the limitless vestibule decorated with a multi-century accumulation of layers of architectural excesses, and peered into the gatehouse window. Two Maxwell macro-demons were oscillating about in its phosphorescent gloom. They were playing at the most stochastic of all games — pitch-and-toss. They occupied all their free time with this diversion. Looking more like poliomyelitis virus colonies under an electron microscope than anything else, they were huge, indescribably inept, lethargic, and dressed in worn liveries. As befit Maxwell demons, they opened and closed doors throughout all their life. They were experienced, well-trained exemplars, but one of them, the one in charge of the exit door, had reached retirement age, which was comparable to the age of the galaxy, and now and then reverted into second childhood, malfunctioning ignominiously. Thereupon, someone from Technical Maintenance would put on a driving suit, enter the gatehouse with its argon atmosphere, and bring the oldster back to reality. Following instructions, I cast a spell on both of them, that is, I crossed the information channels and locked the input-output peripherals to myself. The demons did not react, being otherwise absorbed. One was winning, and, correspondingly, the other was losing, which greatly disturbed them, since it upset the statistical equilibrium.

The fact is that pennies matching is not stochastic enough. Sasha is lost in thoughts what game may be better. Maybe it is stochastic checkers? This is a very simple game. Demons in turns put checkers of two colors on the black cells of the chess field. Loses that player, after whose move one of the opponents gets the opportunity to fell the checker.

Checker A may be felled by checker B if checkers A and B are of different color and a cell, on which B comes after "jumping over" checker A, is free. You need to determine what move the given set of stochastic checkers was lost on.

Input

There is a description of 32 stochastic checkers' moves in traditional chess notation in the input. Moves of white pieces are in lines with odd numbers, and moves of black pieces are in lines with even numbers.

Output

Output the number of move on which a set was lost, or the word "Draw", if a set was not lost by either demon.

Sample

input	output
a1 a3 a5 a7 b2 b4 b6 b8 c1 c3 c5 c7 d2 d4 d6 d8 e1 e3 e5 e7 f2 f4 f6 f8 g1 g3 g5 g7 h2 h4 h6 h8	5

Problem Author: Den Raskovalov
Problem Source: The 10th Collegiate Programming Contest of the High School Pupils of the Sverdlovsk Region (October 16, 2004)

1335. White Thesis

Time limit: 0.5 second
Memory limit: 64 MB

"At this point, the baccalaureate of black magic, Magnus Feodorovich Redkin, brought in his keys, looking obese, customarily preoccupied, and hurt. He obtained his baccalaureate three hundred years ago for inventing the invisibility socks. Since then, he has been improving them over and over. The socks became culottes, and then pants, and now they are referred to as trousers. Still, he remained unable to make them work properly. At the last session of the seminar on black magic, when he made his serial presentation "On Certain Novel Aspects of the Redkin Invisibility Trousers," he was once more overtaken by disaster. During the demonstration of the updated model, something in its inner workings stuck, and the trousers, with a bell-like click, became invisible themselves, instead of their wearer. It was most embarrassing. However, Magnus Feodorovich worked mostly on a dissertation whose subject sounded something like "The Materialization and Linear Naturalization of the White Thesis, as an Argument of the Sufficiently Stochastic Function E Representing the Not Quite Imaginable Human Happiness."

Here he had achieved significant and important results, from which it followed that humanity would be literally swimming in not quite imaginable happiness, if only the White Thesis itself could be found, and most importantly if we could understand what it is and where it could be found."

According to Redkin's last hypothesis, the White Thesis is a positive integer triplet (A, B, C) satisfying the following properties: $A^2 + B^2$ is divisible by C and A, B, C are pairwise distinct. The hypothesis also states that all three White Thesis components lie between the squares of two consecutive integers N and N+1.

Input

contains one integer N ($2 \le N \le 30000$).

Output

Output three different integers A, B and C, that ($A^2 + B^2$) is a multiple of C and $N^2 \le A, B, C \le (N+1)^2$. If two or more such triplets exist, output any one. If there are no such triplets, then output "No solution".

Samples

input	output
2	8 6 4
1000	1000000 1000756 1000976

Problem Author: Den Raskovalov
Problem Source: The 10th Collegiate Programming Contest of the High School Pupils of the Sverdlovsk Region (October 16, 2004)

1336. Problem of Ben Betsalel

Time limit: 1.0 second
Memory limit: 64 MB

"B-but, my dear f-fellows," said Feodor Simeonovich, having diligently deciphered the handwriting. "This is B-Ben B-Beczalel's problem! Didn't C-Cagliostro prove ththat it had no s-solution?"

"We know that it has no solution, too," said Junta, bristling immediately. "But we wish to learn how to solve it"

"H-how strangely you r-reason, C-Cristo… H-how can you look for a solution, where it d-does not exist? It's s-some sort of n-nonsense.

"Excuse me, Feodor, but it's you who are reasoning strangely. It's nonsense to look for a solution if it already exists. We are talking about how to deal with a problem that has no solution. This is a question of profound principle, which, I can see, is not within your scope, since you are an applications type. Apparently I started this conversation with you for nothing."

Problems that do not have solution — that's cool, of course. However, sometimes you want to solve something in solution of which nobody doubts. For example, to present an integer in the form of ratio of square and cube of some integers. But why does this problem always have a solution?… Ok, you will see :)

Input

The only line contains an integer n ($1 \le n \le 10^9$).

Output

In the first line output an integer m. In the second — an integer k. m^2 should be equal to $k^3 \cdot n$; $1 \le m, k \le 10^{100}$.

Samples

input	output
18	12 2
1	1 1

Problem Author: Den Raskovalov
Problem Source: The 10th Collegiate Programming Contest of the High School Pupils of the Sverdlovsk Region (October 16, 2004)

1337. Bureaucracy

Time limit: 1.0 second
Memory limit: 64 MB

After the wonderful properties of the Field of Wonder in the Fools Country were discovered, the huge bureaucratic mechanism appeared. So, to plant his coin in the field, Pinocchio has to gather a lot of documents. Moreover it is impossible to get some documents without getting some set of another documents first.

The issue of one type of documents is a prerogative of exactly one bureaucrat. And all of these bureaucrats are so lazy, that they agree to work only one day a week. So there are incredibly long queues to the bureaucrat's offices during the visiting days and it really takes a whole day to get only one document.

Pinocchio wants to realize a profit on his investments as soon as possible. The only way to do it is to gather all necessary documents as soon as possible. He found out what bureaucrats he needs to visit and numbered them from 1 to N inclusive. He found out the visiting day of week and the set of documents he should have on his hands during the visit for each of the bureaucrat.

After a short time of thinking, Pinocchio understood that he can't find the optimal solution of his problem. And then he promised to pay a half of his future profits to one, who will help him.

Input

The first line of input contains amount of bureaucrats in the Fools Country N ($1 \le N \le 100$) and amount of days in a week according to calendar of Fools Country L ($1 \le L \le 100$). The next line contains the numbers of bureaucrat's visiting days A_i ($1 \le A_i \le L$). The next N lines describe the sets of documents, necessary for receiving the corresponding document. Set of documents consists of numbers delimited by spaces. It is known, that i^{th} line doesn't contain document with number i. Each line is ended by 0 which means the end of the set. If the set is empty (line contains single 0), the document can be gathered without any other documents. After these lines there is one more number — current day of week K ($1 \le K \le L$). The next line contains the list of documents Pinocchio already has. This list consists of numbers delimited by spaces and it ends with 0. And the last line of input contains the list of documents necessary for Pinocchio in the same format.

Though the Fools Country seems to have an ideal income source but because of the official circumlocution the state can't get a huge part of the taxes. As a result there is enough money to keep only one office. So there may work not more than one official at the same time.

Output

If it is impossible to get the necessary set of documents then output "No Solution". Otherwise it should output the minimal amount of days (excluding the current day) he will spend gathering the necessary set of documents, and in the next line it should output all gathered documents delimited by spaces in chronological order. If there are several such answers then the program may output any of them.

Sample

input	output
2 7 1 2 0 1 0 1 1 0 2 0	1 2

Problem Author: Anatoly Uglov, Evgeny Krokhalev
Problem Source: The 10th Collegiate Programming Contest of the High School Pupils of the Sverdlovsk Region (October 16, 2004)

1338. Automobiles

Time limit: 3.0 second
Memory limit: 64 MB

Everyone knows what traffic congestion is. Even supercomputer won't be able to lay an optimal passage of the city center in the rush-hours. But it is possible to simulate the traffic flows.

For that the following experiment is carried out. We choose several check points on the streets net of the city and one goal T from the checkpoints. We send a car from each checkpoint (except T) to the point T along the shortest way. In the point T we fix how many cars came from the North, how many from the South, from the East and from the West. So we can judge about the functioning capacity of the approaches to the point T.

You are to carry out such an experiment. No, you are not given a car! You are no to go anywhere. You are only to write a program simulating the experiment.

Input

Input contains the city plan description in the following format. The first line contains two integers W and H ($1 \le W, H \le 500$) – the width and the height of the plan. The next H lines consists of the streets net and checkpoints description. A symbol "." means a place where a building is. A symbol "#" means a road fragment. A symbol "o" (small Latin letter) means a checkpoint. A road fragment always occupy a cell completely. Two road fragments belong to one road if and only if they have a common side.

Then there is a series of assignments of the experiment. First of all there is a number of assignments M ($0 \le M \le 20$). Each of the next M lines contains the number of the goal point T for the corresponding experiment. Assume that the checkpoints are numbered bottom-up and from left to right.

If some car is to choose a way from several shortest ones the next scheme of priorities acts: South, North, West, East.

Output

You are to output the results of each experiment in the following format:

```
Experiment #N: North: Rn, South: Rs, East: Re, West: Rw
```

where Rn, Rs, Re and Rw - an amount of cars that came in the experiment number N to the goal point from the North, South, East and West respectively.

Sample

input	output
10 5 ..####.... ..o..o.... ..####.#o. ......##.. .o#####... 1 4	Experiment #1: North: 0, South: 1, East: 0, West: 0

Problem Author: Alexander Klepinin
Problem Source: USU Championship 2004

1339. Babies

Time limit: 1.0 second
Memory limit: 64 MB

O tempora! O mores!

Present-day babies progress quickly. There are exactly k boys and k girls in the kindergarten. Some boys like some girls. But in this age the boys are still knights, so, if some boy like some girl then he likes the only girl and moreover one and the same girl can't be liked by more than one boy. And the girls in this age are true ladies. So, if a girl likes a boy she likes the only one, and different girls like different boys.

The children are ingenuous. Their secret amorousness is well-known to the nurse. Once the group decided to go for a walk and the nurse made up her mind to fall the children in pairs so that if there is a boy or a girl in love in a pair then the boy likes his pair-mate or the girl likes the boy. Help the nurse to arrange the described pairs. You may assume that either the boys or the girls enumerated with positive integers from 1 to k.

Input

The first line contains the integer k — the number of boys ($1 \le k \le 250\ 000$). The second line consists of the numbers of girls that are liked by boys: if the i'th boy likes some girls, her number is at the i'th position; if the i'th boy likes nobody, there is 0 at the i'th position. The numbers are separated with a space. The third line consists of the analogous information about the girls.

Output

You should output the sequence of k integers. The i'th element of the sequence is the number of a girl that is a pair-mate of the i'th boy. The numbers are separated with a space.

Sample

input	output
3 3 0 0 0 2 0	3 2 1

Problem Author: Magaz Asanov
Problem Source: USU Championship 2004

1340. Cucaracha

Time limit: 1.0 second
Memory limit: 64 MB

Once upon a time a cockroach Vasya running along the kitchen noticed a wonderful crumb. Vasya's internal voice whispered him that he should reach the crumb using minimum of his vital energy. It meant that he should make the minimal number of his cockroach steps. And even Vasya's little brain understood that he was to count the motion path neatly.

Input

The first line contains four numbers X, Y, A, R — the cockroach's initial coordinates, the initial angle between the axis Ox and his motion path and the minimal turning radius of the cockroach (have you thought that cockroaches can turn around at one point?). The second line consists of two numbers KX and KY — the desired crumb coordinates.

Cockroaches count coordinates off the kitchen center and measure in cockroach steps. Inasmuch as a cockroach has six legs, this measure is rather relative, so you aren't to think that a cockroach can run only integer number of steps. For example, 314.15 steps is the length of Vasya's night walk.

The absolute values of all the coordinates X, Y, KX, KY and the radius R do not exceed 10000. The angle is counted off the axis Ox counter-clockwise. It's measured in degrees (cockroaches don't know radians) and lies in the diapason from 0 to 360.

Output

You are to find the length of the shortest Vasya's motion path towards the crumb. The result is to be outputted within 4 digits after a decimal point.

Sample

input	output
1 2 90 10 21 2	31.4159

Problem Author: Pavel Egorov
Problem Source: USU Championship 2004

1341. Device

Time limit: 1.0 second
Memory limit: 64 MB

Major (M): You claimed that your device would be able to fly round the Earth several times and to miss not more than a couple of centimeters?

Designer (D); Yes! Our gravitational fields system of navigation absolutely...

M: Furthermore it can't be fixed by detectors and doesn't have a receiver or transmitter.

Engineer (E): It was your demand that nobody could detect the device...

M: We gave it a simple task to fly round the square. It didn't return to the initial point.

D: Was that square large?

M: It's none of your business! This is the State secret! You are to find the device!

Programmer (P): How did you programme the mission profile?

M: The device was to fly one conditional length unit to the North, the same distance to the East, the same distance to the South and then to the West. It passed more than 40 minutes since the device was to return. If they find it before us!.. In short, you are to find it!

D: It's understood. Where was the initial point?

(The major flags and in two seconds the designer lies on the floor with his hands tied and two gunpoints look at his nape).

M: Why do you need this information?

E: You misunderstood! We don't need information! But if we knew the initial point coordinates we could say where the device was...

(In two seconds two gunpoints look at the engineer' nape, too).

M: Who interests this information? Where is the device? One, two, ...

P: You can't understand! If the device reached the North Pole it can't continue not to the North. Not to the East. Only to the South! Where the device is depends on where it started.

(Major aims at the programmer.)

M: No, it didn't reach the Pole. It was taken into account.

P: Let me write a program that would count the final coordinates of the device. You'll input the latitude, the longitude and the value of your conditional length unit yourself! The program would give you the answer keeping the absolute secrecy.

M: I'll give you a chance. You three have got a computer and five hours... Less than five hours already. If we do not get the coordinates... You'll suffer first.

Input

The first line contains the initial latitude W. $-90 < W < 90$. The second line – the initial longitude L, $-180 < L \leq 180$. The third line contains the length of the square side, which the device was to fly round. The length is given in kilometers. The device keeps the fixed distance 6400 km from the Earth center of mass. The South Pole has latitude –90, the North Pole – latitude 90. The East direction is counted off the 0^{th} meridian in the positive direction.

Output

You should output the final latitude and longitude of the device within three digits after a decimal point.

Sample

input	output
56.846841 53.36673 1124.427	56.847 60.631

Problem Author: Stanislav Vasilyev
Problem Source: USU Championship 2004

1342. Enterprise

Time limit: 2.5 second
Memory limit: 64 MB

To bind a broom it's a hard work. As there is a very big demand for this high-tech product an brooms binding enterprise is to have a big amount of production workshops. You are to help such an enterprise to allocate the work among the workshops. Each workshop can bind from 0 to K brooms a day. Economists of the enterprise found out that each bound broom has a different prime cost: in most cases the more brooms were bound a day the less prime cost has the last broom bound that day. However, there may be more complicated situations. As a first approximation you may assume every dependence linear. So decided the economists when they determined a dependence of the next in turn broom's prime cost on the industrial output of the workshop. You are to find out the optimal work load of the workshops.

Input

The first line contains two integers N and M ($1 \le N, M \le 1000$) — an amount of workshops and the required industrial output of brooms, respectively.

Then workshops description follows. The (i+1)-st line describes the i-th workshops with three numbers K_i, P_i, and Q_i ($1 \le K_i \le 100$; $0 \le P_i, Q_i \le 1000$) — they are the maximal number of brooms that can be bound at the i-th workshop a day, the prime cost of the first broom and the prime cost of K_i-th broom at the i-th workshop. As it was mentioned above the cost of j-th broom's production is the linear with respect to j function.

Output

If the enterprise can't produce the required number of brooms your program is to output the maximal number of brooms V that can be bound at the enterprise.

Besides, you are to output the total costs on production of M (or V if the enterprise can't bind M) brooms with optimal allocation of industrial outputs within two digits after a decimal point.

The output format is to be as in sample outputs below.

Samples

input	output
2 10 6 20 15 100 100 100	Minimum possible cost: 505.00
2 10 5 30 14 1 20 20	Maximum possible amount: 6 Minimum possible cost: 130.00

Problem Author: Magaz Asanov and Pavel Egorov
Problem Source: USU Championship 2004

1343. Fairy Tale

Time limit: 1.0 second
Memory limit: 64 MB

12 months to sing and dance in a ring their celestial dance. One after another they hold a throne. The first is young and fierce January and the last is elderly and wise December. Leaving the throne, every month cry out a digit. During a year a 12-digit number is formed. The Old Year uses this number as a shield on his way to the Abyss of Time. He defend himself with this shield from the dreadful creatures of Eternity. Because of hard blows the shield breaks to pieces corresponding to the divisors of the number.

Your task is to help the months to forge the shield for the Old Year such that it couldn't be broken to pieces.

Input

The first line contains a number of months that already left the throne. The second line contains the digits already cried out.

Output

Output an arbitrary 12-digits integer that starts with the given digits and that has no nontrivial divisors. It's guaranteed that the solution exists.

Sample

input	output
5 64631	646310554187

Problem Author: Pavel Atnashev
Problem Source: USU Championship 2004

1344. Gigantic Turnip

Time limit: 1.0 second
Memory limit: 64 MB

As usual old man has planted a turnip. Every year he plants a turnip and every year happens one and the same thing — the turnip grows gigantic. May be the old man has good artificial fertilizers, may be his surname is Michurin, may be the atomic station that is not far from that place tells on the yield. And this year the turnip has grown very big and the old man is to take it out.

The old man tried to call in an old wife aid but she suffers from sciatica since the previous year when they tried to take off a turnip. Granddaughter after that event received medical treatment in Switzerland (that turnip turned very expensive), and a dog, a cat and a mouse have got idiosyncrasy to root crops upon the whole.

The old man has grieved. Que faire? (*What is to be done? — Fr*) The turnip is to be taken off!

Suddenly the old man recalls that he's got a precious book at the garret — a course of physics by Pyoryshkin for the secondary school. He remembered that very interesting things are described in this book – pulleys. The construction of such pulleys may help the old man to get along without the old wife, granddaughter, the dog, the cat and the mouse.

The old man takes the book, reads it, — and assembles a construction. Now he is sitting at the porch of his house and thinks how much will this construction help him to save on the force?

Pulley is a wheel with a rope thrown over it and another rope that is bound to the axle. If the rope that is bound to the axle looks up and the ends of the other rope that is thrown over the wheel look down then it is a pulley of the *first type*. If the rope that is bound to the axle looks down and the ends of the thrown over rope look up then it is a pulley of the *second type*.

Help the old man, given a configuration of blocks of the first and the second types, to determine the win in the force while the crop hasn't rotted. Assume that two pulleys coupled with a rope are on different heights.

Input

The first line contains integer N($1 \le N \le 32$) — that is the number of pulleys. The pulleys are numbered with integers from 1 to N. The next N lines consist of the blocks description in the following format: T X1 X2 Y, where T — the pulley type (1 or 2), the ends of the rope thrown over the current pulley look towards the pulleys number X1 and X2, the rope that is bound to the axle looks towards the pulley number Y. However there are some exclusions.

1. Some going up ropes are bound to the ceiling of the hothouse (the old man planted turnips in hothouses). In this case there is 0 instead of the number of pulley to which this ripe was to go.
2. Exactly one rope that goes down from a pulley is bound to the turnip with mass 1 centner. In this case in the line with description of the i-th pulley number -1 instead of the pulley number is written.
3. Exactly one rope going down from a pulley is free. Then there is -2 instead of the number of pulley in the description of the i-th pulley.

All the ropes are vertical.

Output

You are to output the mass of a load in centners (within 4 digits after a decimal point) that is to be hung on the free rope in order to balance the system of pulleys. If there is no such a mass output "`No solution`". If any mass suits output the string "`Any`".

Samples

input	output
2 1 -1 2 0 2 1 0 -2	2.0000
4 1 3 2 0 2 1 0 4 1 -1 4 1 2 3 2 -2	No solution
4 1 2 2 0 2 1 1 -2 1 4 4 0 2 3 3 -1	Any

Problem Author: Pavel Egorov
Problem Source: USU Championship 2004

1345. HTML

Time limit: 1.0 second
Memory limit: 64 MB

Once a veteran of the ACM contests, thinking about the younger generation, decided to help them to master with the cobwebs of the sports programming. He decided to make an Internet site that would contain articles on programming, different interesting problems, solutions and the sources of those solutions. But his marvelous sources look faded and poor, not as they look in his favorite development framework: the key words are not emphasized, comments don't differ from the other text... Nothing appeals to the eye.

He knows the basis of the HTML and he understands that it is a very unpleasant work to add coloring his sources tags manually.

- To write a program that adds tags is a duck soup! – he thought. – Or not a duck soup… May be it' easier to do it manually… Or, may be… Eureka! If I can't solve this problem, I'll give it at the next ACM contest – some will surely solve the problem!… And if they make mistakes… Let them try!…

Input

The correct source of a program in Pascal is given The length of the input text is not longer than 100 000 symbols.

Output

You are to add formatting HTML tags so that the source text would look as it is required. The requirements are as follows:

1. All the comments must be enclosed in the pair of tags "" and ""
2. All the key words must be enclosed in the pair of tags "" and ""
3. All the strings must be enclosed in the pair of tags "" and ""
4. All the numbers must be enclosed in the pair of tags "" and ""
5. If key words, strings or numbers come upon the comments then they are assumed as a part of the comment and not as key words, strings or numbers.
6. If key words, comments or numbers come upon the strings then they are assumed as a part of the string and not as key words, comments or numbers.

A **string** is the sequence of symbols enclosed in a pair of quotation marks " ' " that does not contain other quotation marks. Or a symbol "#" which is followed by nonempty sequence of digits. In the second case it's necessary to take the maximal sequence. E. g. in the sequence "#123" a subsequence "#1" is not a string and the entire sequence "#123" is a string.

A **number** begins with a digit and contains only digits and possibly one point ".", followed by one or more digits. As in the case of a string a number is the maximal by inclusion sequence of symbols that satisfies the given above requirement.

An **identifier** may start with a letter or underscore ("_") and contains letters, digits and underscores. As in the cases of strings and numbers, identifier is a maximal by inclusion sequence satisfying the given requirements.

There are **comments** of two types: comment of 1st type begins with '{' and ends with '}', both braces are the part of the comment; comment of 2nd type begins with '//' and ends with line feed, slashes are the part of the comment while line feed is not. Nested comments are the part of the most outer comment.

Key words are the following identifiers: 'and', 'array', 'begin', 'case', 'class', 'const', 'div', 'do', 'else', 'end', 'for', 'function', 'if', 'implementation', 'interface', 'mod', 'not', 'of', 'or', 'procedure', 'program', 'record', 'repeat', 'shl', 'shr', 'string', 'then', 'to', 'type', 'unit', 'until', 'uses', 'var', 'with', 'while'.

Here every key word is enclosed in the pair quotation marks. The key words are given in the lower case but occur in any case in the text. E. g., ImPlEmentAtioN is a key word.

Sample

input	output

Begin
  writeln('Hello world!');
end.
	Begin
	  writeln('Hello world!');
	end.

Notes

Many of you have guessed that if you save the result in a file output.html and to add in the beginning of the file the following lines:

```
<STYLE>
    span.string {color: fuchsia;}
    span.number {color: darkblue;}
    span.keyword {font-weight: bold; color: black;}
    span.comment {font-style: italic; color: gray;}
</STYLE>
<PRE>
```

and a line

```
</PRE>
```

in the end, then having opened this file in a browser you'll see the input text with the colored syntax.

Problem Author: Pavel Egorov
Problem Source: USU Championship 2004

1346. Intervals of Monotonicity

Time limit: 1.0 second
Memory limit: 64 MB

It's well known that a domain of any continuous function may be divided into intervals where the function would increase monotonically or decrease monotonically. A number of intervals of such a partition we will call a *complexity* of the partition. A *complexity of a continuous function* is the minimal possible complexity of partition in the domain into the monotonicity intervals.

The notion of complexity may be defined not only for continuous functions. In particular, it is applicable to the functions specified on a grid.

Input

The input contains a description of a function F, specified on a grid. The first line contains two numbers A and B — the first and the last point of the integer grid with step 1 ($0 \le A < B \le 100\,000$). The second line contains the values table of the function F. The table consists of the integers F(A), F(A+1), …, F(B) separated with a space and/or linefeeds. All the values of the function F are in diapason from –100 000 to 100 000.

Output

Output the only number — the complexity of the function F.

Sample

input	output
1 10 1 2 3 4 2 1 -1 3 6 7	3

Problem Author: Alexander Klepinin
Problem Source: USU Championship 2004

1347. Blog

Time limit: 0.4 second
Memory limit: 64 MB

The last time many users keep internet diaries — blogs. The users keep their diaries and occasionally read other's. Those users like to count their virtual friends. A friend for them is anyone who reads their twaddle. Assume that a user reads another's blog if he mentions the one in his diary.

You are to output for each user a list of his friends, the people who regard him as a friend and a list of mutual friends, i.e. those who are friends of the assumed user and regard him as a friend.

You are given the blogs of those users. For each user you are to output the mentioned above three lists. The users mark their friends with a tag <friend>. A *tag* is the line that consist of small Latin letters between the marks *lower than* (<) and *greater than* (>). A *closing tag* is the string of small Latin between the symbols *lower than* and *slash* (</) in the beginning and *greater than* (>) in the end.

Input

The first line contains a integer N ($1 \le N \le 100$) — a number of blogs in the input. The first line of each blog consists a name of the blogger. All the bloggers' names are different. The next line starts with a tag <blog>. The last blog line ends with a closing tag </blog>. There is no other tags in the text of blog. Each opening and closing tag has it's pair tag in the text. Symbols <, > and / occur only in tags. The input contains only small and capital Latin letters, digits, symbols greater than, lower than, slash, full stops, commas and underlining symbols. The friends' names are case sensitive. A tag can't be torn by a line feed. A friend's name may consist of small and capital Latin letters and an underlining symbol. A friend's name can't be empty. Blogs line doesn't exceed 255 symbols. A blog contains not more than 1000 lines. An amount of friends mentioned in a blog doesn't exceed 100.

Output

For each blogger from the input you are to output a list of those who's blogs he mentioned (friends), a list of those who mentioned him in his blog (is a friend of) and a list of those who is present in both previous lists. Each list is to be outputted in a separate line and sorted in the alphabetic order. Names in a list are to be separated with a comma and space. Before the "friends" list you are to output s string "1: ". Before an "is a friend of" list you are to output a string "2: ". Before a list of "mutual friends" you are to output a string "3: ". Before those three lists you are to output a name of the blogger and an empty line beforehand (if it is not the first input blogger). A blogger is not a friend of himself even if he mentioned himself in his blog.

Sample

input	output
3 xoposhiy <blog> Tomorrow I found <friend>_denplusplus_</friend> to be smartest blogger in the net. Also I received interesting link from <friend>strange_human</friend> </blog> _denplusplus_ <blog> Some shit about my work. </blog> strange_human <blog> <friend>xoposhiy</friend> <friend>_denplusplus_</friend> </blog>	xoposhiy 1: _denplusplus_, strange_human 2: strange_human 3: strange_human _denplusplus_ 1: 2: strange_human, xoposhiy 3: strange_human 1: _denplusplus_, xoposhiy 2: xoposhiy 3: xoposhiy

Problem Author: Den Raskovalov
Problem Source: USU Junior Championship March'2005

1348. Goat in the Garden 2

Time limit: 1.0 second
Memory limit: 64 MB

A goat is tied to a peg (in a point C) in a garden with a strong rope of the length L (i.e. a goat may eat a grass that is not farther than L meters from the peg). There is a bed of pineapples that he loves very much. The bed is a line segment with the ends A and B.

Humph… We wonder, how much the goat is to stretch the roap in order to reach at least one pine apple? And all the pineapples?

Input

There are points' A, B and C coordinates and a length of the rope L in the input. All the numbers are integer, $L \geq 0$, all the coordinates don't exceed 10000 by the absolute value. The numbers are separated with spaces or line feeds.

Output

The first line should contain the minimal length that the goat is to elongate the rope in order to reach the pineapples bed. The second line should contain the minimal length that the goat is to elongate the rope in order to eat all the pineapples from the bed. All the numbers are to be outputted within two digits after a decimal point.

Sample

input	output
8 -6 8 6 0 0 7	1.00 3.00

Problem Author: Pavel Egorov
Problem Source: USU Junior Championship March'2005

1349. Farm

Time limit: 1.0 second
Memory limit: 64 MB

Here is a farm. Here is a farmer that counts how many animal live in his farm: a camels, b sheep, c green cockroaches. Occurs that $a^n + b^n = c^n$. n is given. You are to find all the rest.

Input

n ($0 \le n \le 100$)

Output

Three different integers (a, b and c) such that $a^n + b^n = c^n$, $1 \le a, b, c \le 100$. If there are several solutions you should output the one where a is minimal. If there are several solutions with the minimal a you should output the one with minimal b, and so on. Output −1 if there is no solution.

Samples

input	output
0	-1
1	1 2 3

Problem Author: Pierre Fermat
Problem Source: USU Junior Championship March'2005

1350. Canteen

Time limit: 1.0 second
Memory limit: 64 MB

It's dangerous to eat in a canteen — you may be poisoned by a not fresh food. One may fall into a coma because of the canteen's chicken and the other feels OK. And vice versa. The food is cooked from M different food stuffs. There are N different food stuffs in the menu but not all of them are at the distribution. Assume that $K + 1$ students eat the food and we know for each student what products may poison him. The first student eats and he is not poisoned. How the dinner affect on the other students?

Input

The first line contains an integer N ($1 \le N \le 100$). The next N lines contain the food stuffs names — non-empty sequences of Latin letters and digits with length not more than 40 symbols. Then there is a number K ($1 \le K \le 100$) and $K + 1$ blocks describing the menu food stuffs dangerous for the canteen visitors afterwards. The i^{th} block starts with a line with an integer N_i — an amount of dangerous food stuffs and then there are N_i lines with the names of those dangerous stuffs ($0 \le N_i \le N$). The first block describes the food stuffs dangerous for the first student, the next K blocks — for the rest ones. The input ends with the line containing an integer M ($0 \le M \le N$).

Output

K lines — the i^{th} line should contain:

- YES, if the dinner is harmless for the $(i + 1)^{st}$ student,
- NO, if among the food stuffs there is a dangerous one for the $(i + 1)^{st}$ student,
- MAYBE, if there may be different situations under the given conditions.

Sample

input	output
7 Rafinad Kefir Pastila Smetana Chokolade Kljukva Imbir 3 3 Rafinad Kefir Imbir 1 Rafinad 3 Kefir Kljukva Smetana 2 Imbir Smetana 3	YES NO MAYBE

Problem Author: Pavel Egorov
Problem Source: USU Junior Championship March'2005

1351. Good Gnusmas – Dead Gnusmas

Time limit: 1.0 second
Memory limit: 64 MB

A dwarf bibr — a rare animal from the planet Globland. Aborigines of Globland, loathsome gnusmases, like the bibrs' meat. Thet believe that a has eaten the bibr gnusmas becomes a globalizer (i.e. a semi-god, semi-gnusmas that is not afraid of neither acid rains nor radio-active winds of Globland.) Bibrs are very peaceful animals but defending themselves from gnusmases they use all their arsenal of weapons: pistols, submachine guns, projectors and some kinds of the weapons of close fight: fungos, knuckle-dusters and so on.

One nice morning bibr Bob woke up from the howl of anti-gnusmas siren. There were thousands of points on the radar. It meant that loathsome gnusmases scrambled out from their burrows and slowly approached Bob's little house. But Bob was a flash guy and just a week before he had set a new Kakashnikov machine gun on his house's roof. One burst of fire annihilates all the gnusmases that were in the arc of fire and those who survived ran away. He decided to state the value of effectiveness of his bargain and to find out how many gnusmases he would annihilate and how many of them would hide back in their burrows.

Input

The first line contains the machine gun characteristics: D — the range ($1 \le D \le 30000$), X_1, Y_1, X_2, Y_2 ($1 \le |X_1|$, $|Y_1|$, $|X_2|$, $|Y_2| \le 30000$) — coordinates of the right and the left bounds of the arc of fire. The second line contains an integer N ($1 \le N \le 30000$) — an amount of gnusmases that attack Bob's house. The next N lines consist of the gnusmases coordinates X_i, Y_i ($0 \le |X_i|$, $|Y_i| \le 30000$; X_i and Y_i are not equal to zero at the same time). All the numbers are integer. Bob's house coordinates are (0,0). The arc of fire is less than 180 degrees.

Output

N lines with the answers "YES" and "NO". "YES" — if the gnusmas will be annihilated (i.e. he is inside the arc of fire at the distance from Bob's house not greater than the range of the machine gun). "NO" — otherwise.

Sample

input	output
5 1 0 0 1 6 1 1 5 6 3 0 -1 2 0 5 -3 0	YES NO YES NO YES NO

Problem Author: Denis Musin
Problem Source: USU Junior Championship March'2005

1352. Mersenne Primes

Time limit: 1.0 second
Memory limit: 64 MB

Definition. If the number 2^N-1 is prime then it is called a *Mersenne prime number*.

For example, 2^2-1 — the first Mersenne prime, 2^3-1 — the second Mersenne prime, $2^{11213}-1$ — the 23rd, $2^{216091}-1$ — the 31st.

It's a hard problem to find those numbers without a computer. So, Euler in 1772 found the 8th Mersenne prime — $2^{31}-1$ and then for 100 years no Mersenne prime was found! Just in 1876 Lucas showed that $2^{127}-1$ is a prime number. But he didn't find the 9th Mersenne prime, it was the 12th one (the numbers $2^{61}-1$, $2^{89}-1$ and $2^{107}-1$ are prime but it was found out later). A new break-through happened only in 1950's when with the help of the computing machinery Mersenne primes with the powers 521, 607, 1279, 2203 and 2281 were found. All the following Mersenne primes were found with the help of computers. One needn't be a great mathematician to do that. In 1978 and 1979 students Noll and Nickel found the 25th and 26th numbers (21701 and 23209) on the mainframe of their University and they became famous all over the USA. But the modern supercomputers have the limits of their capability. Today the dozens of thousands people all over the world united in one metaproject GIMPS (Great Internet Mersenne Prime Search, www.mersenne.org) look for Mersenne primes. GIMPS found 8 the greatest Mersenne primes. Their powers are 1398269, 2976221, 3021377, 6972593, 13466917, 20996011, 24036583, 25964951. $2^{6972593}-1$ is the 38th Mersenne prime, and for the last 4 numbers one can't tell what are their sequence numbers because not all the lower numbers are checked. Those four numbers are also the greatest known prime numbers.

The latest number $2^{25964951}-1$ was found on February 18, 2005, it contains 7816230 decimal digits. The one who will find a prime number with more than 10 millions digits will get a prize of $100000. You may gain the prize if you join the project.

You are not now to find the 43 th Mersenne prime — the jury won't be able to check your answer. N doesn't exceed 38 in this problem. So, given an integer N you are to find Nth Mersenne prime.

(Information is actual for March, 2005)

Input

The first line contains integer T — an amount of tests. Each of the next T lines contains an integer N.

Output

For each N you should output the power of the Nth by order Mersenne prime.

Sample

input	output
13 18 32 24 21 19 34 27 33 20 30 28 29 22	3217 756839 19937 9689 4253 1257787 44497 859433 4423 132049 86243 110503 9941

Problem Author: Vladimir Yakovlev
Problem Source: USU Junior Championship March'2005

1353. Milliard Vasya's Function

Time limit: 1.0 second
Memory limit: 64 MB

Vasya is the beginning mathematician. He decided to make an important contribution to the science and to become famous all over the world. But how can he do that if the most interesting facts such as Pythagor's theorem are already proved? Correct! He is to think out something his own, original. So he thought out the Theory of Vasya's Functions. Vasya's Functions (VF) are rather simple: the value of the N^{th} VF in the point S is an amount of integers from 1 to N that have the sum of digits S. You seem to be great programmers, so Vasya gave you a task to find the milliard VF value (i.e. the VF with $N = 10^9$) because Vasya himself won't cope with the task. Can you solve the problem?

Input

Integer S ($1 \le S \le 81$).

Output

The milliard VF value in the point S.

Sample

input	output
1	10

Problem Author: Denis Musin
Problem Source: USU Junior Championship March'2005

1354. Palindrome. Again Palindrome

Time limit: 1.0 second
Memory limit: 64 MB

A *word* is the nonempty sequence of symbols $a_1a_2\ldots a_n$. A *palindrome* is the word $a_1a_2\ldots a_n$ that is read from the left to the right and from the right to the left the same way ($a_1a_2\ldots a_n = a_na_{n-1}\ldots a_1$). If $S_1 = a_1a_2\ldots a_n$ and $S_2 = b_1b_2\ldots b_m$, then $S_1S_2 = a_1a_2\ldots a_nb_1b_2\ldots b_m$. The input contains some word S_1. You are to find a nonempty word S_2 of the minimal length that S_1S_2 is a palindrome.

Input

The first input line contains S_1 (it may consist only of the Latin letters). It's guaranteed that the length of S_1 doesn't exceed 10000 symbols.

Output

S_1S_2.

Samples

input	output
`No`	`NoN`
`OnLine`	`OnLineniLnO`
`AbabaAab`	`AbabaAababA`

Problem Author: Denis Nazarov
Problem Source: USU Junior Championship March'2005

1355. Bald Spot Revisited

Time limit: 1.0 second
Memory limit: 64 MB

A student dreamt that he walked along the town where there were lots of pubs. He drank a mug of ale in each pub. All the pubs were numbered with positive integers and one could pass from the pub number n to the pub with a number that divides n. The dream started in the pub number a. The student knew that he needed to get to the pub number b. It's understood that he wanted to drink on the way as much ale as possible. If he couldn't get from the pub number a to the pub number b he woke up immediately in a cold sweat.

Input

The first line contains an integer T — an amount of tests. Then T lines with integers a and b follow ($0 \le T \le 20$; $1 \le a, b \le 10^9$).

Output

For each test in a separate line you are to output the maximal number of mugs that the student could drink on his way.

Sample

input	output
5 30 89 2 16 3 243 1 1 2 2	0 4 5 1 1

Problem Author: Aleksandr Bikbaev
Problem Source: USU Junior Championship March'2005

1356. Something Easier

Time limit: 1.0 second
Memory limit: 64 MB

"How do physicists define prime numbers? Very easily: prime numbers are the number 2 and all the odd numbers greater than 2. They may show that this definition corresponds to the mathematical one: 3 is prime, 5 is prime, 7 is prime… 9? 9 is certainly not prime. Then: 11 is prime, 13 is prime. So 9 is the experiment mistake."

From mathematical analysis course

Once physicist and mathematician argued how many prime numbers one needed for the purpose that their sum was equal to N. One said that it wasn't known and the other that 3 was always enough. The question is how many.

Input

The first line contains T, an amount of tests. Then T lines with integer N follow ($0 \le T \le 20$; $2 \le N \le 10^9$).

Output

For each test in a separate line you should output prime numbers so that their sum equals to N. An amount of such prime numbers is to be minimal possible.

Sample

input	output
7 2 27 85 192 14983 3 7	2 23 2 2 2 83 11 181 14983 3 7

Problem Author: Aleksandr Bikbaev
Problem Source: USU Junior Championship March'2005

1357. Teakettle 1.0 for Dummies

Time limit: 1.0 second
Memory limit: 64 MB

Several students wake up in the morning and want to drink a cup of hot coffee. But they have only one teakettle. In this case, they usually operate according the following instruction:

Students should wake up one after another and act as follows:

- If, when a student wakes up, nobody boils the water, he should fill the teaketle with 200 ml of water and put it on the burner. Water has initial temperature 20 °C.
- If another student boils water already, he should add 200 ml of water in the teakettle. As a result, water temperature equalize according to law $T = (m_1T_1 + m_2T_2) / (m_1 + m_2)$, where m_1, m_2 – mass of water in teakettle and mass of additional portion of water, and T_1, T_2 – their temperatures.
- If another students boil water and the teakettle is full already then other students should wait until the teakettle becomes free. The teakettle capacity is 1 liter (1 kg) of water i.e. 5 students are able to boil water simultaneously.
- As soon as water begins to boil, i.e. the water temperature runs up to 100 °C, the teakettle should be put off the burner and water should be poured into the cups, 200 ml for each student. At the same moment teakettle can be used for boiling next portion of water. Water heats up by $\Delta T = (P\ \Delta t) / (C\ m)$ degrees during Δt seconds, where P – is a power of burner, C = 4190 J/kg·K – specific heat of water, m – water mass.

You are giver wake up time for each student and you are to simulate this process to find for each student the time when he will be able to drink a cup of coffee. Assume, that students can perform described manipulations in a moment. Power of burner P = 400 Watt.

Input

The first line of input contains single integer *N* – amount of students ($0 \le N \le 500$). Each of the following *N* lines contains a wake up time of corresponding student in format HH:MM:SS. Times are sorted in ascending order. All students wake up at the different moments but during the same day, i.e. all times different and between 00:00:00 and 23:59:59.

Output

For each student you should output the time when he will be able to drink his coffee in the same format HH:MM:SS.

Sample

input	output
4 07:00:00 07:02:00 07:05:30 12:00:00	07:08:23 07:08:23 07:08:23 12:02:48

Problem Author: Vladimir Yakovlev
Problem Source: USU Junior Championship March'2005

1358. Cables

Time limit: 0.25 second
Memory limit: 64 MB

There are N computers in a computer club. It's known which computers are to be connected with a cable in order to make the net work properly. It's left to arrange the computers so that no two cables intersect and distance between every two computer would be greater than one. Regard the computers as points and the cables as line segments. The net is connected, i.e. every two computers are connected with some sequence of cables.

Input

First line contains integer N ($1 \le N \le 1000$). Then N−1 lines follow. In each line there are two integers a_i and b_i, the numbers of computers that are to be connected with a cable ($1 \le a_i, b_i \le N$).

Output

You should output N lines. In the i^{th} line there should be two real numbers — coordinates of the i^{th} computer. The absolute values of the coordinates shouldn't exceed 1000.

Sample

input	output
3 1 2 2 3	0 0 10 0 0 10

Problem Author: Den Raskovalov (text by Aleksandr Bikbaev)
Problem Source: USU Junior Championship March'2005

1359. Construction

Time limit: 1.0 second
Memory limit: 64 MB

A lifting crane broke down. The constructors are to lift down a barrel with and old cement. The barrel is to end up n meters to the left and m meters down from the place where it was. To get a barrel down one can build a ramp of some amount of wooden planks so, that

- both ends of each plank would have integer coordinates;
- one end is always lower than the other one;
- the lowest end of each plank is not to the right from the upper one.

You may assume that barrel doesn't jump during rolling down, i.e. direction of it's velocity changes in a moment and value of velocity keeps unchanged at the turns of a planks. Acceleration of gravity assumed to be equal to 10. You should neglect rotation of the barrel and friction between barrel and ramp. You are to find a minimal time to get a barrel down.

Input

Input contains two integers n and m ($1 \le n, m \le 50$).

Output

Output should contain minimal time in seconds which a barrel would need to roll down on the described construction of wooden planks accurate within 10^{-3}.

Sample

input	output
2 2	0.8614

Problem Author: Den Raskovalov (text by Aleksandr Bikbaev)
Problem Source: USU Junior Championship March'2005

1360. Philosophical Dispute

Time limit: 2.0 second
Memory limit: 64 MB

One day, mathematician and philosopher were engaged in a heated dispute.

Philosopher said:
— Ideal line has only length and no width, therefore, no line can have an area.
Mathematician replied:
— That's as it may be, but still you can fill a square with a line in such a way that there will be no gaps.
And you can't deny that a square has an area, and he grinned.
But Philosopher still wasn't convinced:
— Show me this line, then.
— With pleasure… — responded Mathematician and scribbled some equations on a piece of paper:

— With t increasing, the point (x, y) will move around the square, forming a line.
— So what? — asked Philosopher. How is it going to fill the entire square?
— Indeed, it will, — said Mathematician, — Whichever point inside the square you draw, the line will eventually cross that point.
— No, — replied Philosopher indignantly, — Anyway, I don't believe. When will the line cross this point? — and he put a thick dot inside the square.

Give Philosopher an answer.

Input

The first line of input contains the coordinates (x_0, y_0) of the dot center $(-1 \le x_0, y_0 \le 1)$. The second line contains $\varepsilon \ge 0.0001$ — the radius of the dot (the dot is essentially a small circle).

Output

Any value of t in the segment $[0, 10^{12}]$, which corresponds to the line crossing the dot, or "FAIL", if the line doesn't cross the dot.

Sample

input	output
0.744 0.554 0.01	5.3

Problem Author: Stanislav Vasilyev (idea by Den Raskovalov)
Problem Source: IX Collegiate Students Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1361. Spaceology vs. Chronistics

Time limit: 1.0 second
Memory limit: 64 MB

Every year in the city of Radon-Snark a famous symposium of scientists-spaceologists is held.

Professor A, haunter of the symposium, has decided this time to invite professor B, who does research in the adjacent field of science, applied chronistics.

Unfortunately, professor A has forgotten to meet professor B at the railway station (he was thinking about the exciting future of spaceology and remembered that his friend was coming only when B had already arrived at the city). "There's nothing left to do, I have to go", decided A. He got into his spacemobile and left to the railway station.

Now the world-famous scientist B can't wait even a second (chronistics think that only real action is true, not the some vague argumentation). He has got into his chronomobile immediately, and has left the railway station (A and B have started at the same time). What's left to do is to find out when A would meet B. Note that the spacemobile and chronomobile would not stop before they meet.

Remember that the symposium is just about to start and the strange things that are always associated with the symposium already began to happen. While notable scientists discuss their problems, all the machinery in the city behaves very strange. For example, all the spacemobiles turn to the leftmost road on each junction, and the chronomobiles turn to the rightmost one. Also, leaving the vehicle on the road between two junctions is against the law, so A and B can only meet each other on a junction.

Input

The first line contains one number N ($N \le 100000$) that is the amount of junctions in the city of Radon-Snark. The following N lines describe the junctions. The (i + 1)st line contains a list of junctions that can be reached from the ith junction directly. The roads are listed in order from the leftmost to the rightmost. All roads enter to a junction only from one side, that's why words "leftmost" and "rightmost" have sense. The list is terminated with zero. Each list contains at least one nonzero number. The list can contain a road connecting a junction with itself.

The last input line contains two numbers. The first one specifies the junction where A starts the trip in his spacemobile. The second number is the junction where B starts from.

Output

Output the number of minutes that B will need to meet A. It takes exactly one minute to travel from one junction to another directly reachable junction. Output −1 if they won't meet.

Sample

input	output
7 2 4 0 3 1 0 4 2 0 5 1 6 3 0 6 4 0 2 5 4 0 6 0 1 7	7

Problem Author: Eugeniy Krokhalev (idea by Aleksandr Klepinin)
Problem Source: IX Collegiate Students Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1362. Classmates 2

Time limit: 2.0 second
Memory limit: 64 MB

May be you can remember the "Classmates" problem from the 2004 Urals Programming Contest. The statement of that problem is like follows:

Tanya is a schoolgirl. One day, headmistress asked her to notify the class about next day's lessons being cancelled due to power outage. Tanya successfully carried out that job. She decided to call Lena, then Katya, the Masha. While she was calling Katya, Lena, who did already know the news, called Misha, etc. The whole class knew of the welcome news in almost no time.

The contestants were to determine the minimum time required to pass the news to every student in the class.

Time passed, and Tanya got a summer internship in the "Advertising, Commercials and Media" agency. Like any other firm that offers high salaries for students, ACM has a clearly defined hierarchical structure. The president is the root of the hierarchy. He has some direct subordinates, who themselves can have direct subordinates, etc. One day, Tanya happened to invent the utterly ingenious method for increasing the advertisement efectivenes by 110%. She called her boss immediately, then her friend Lena (who was Tanya's direct subordinate), then Masha and Katya. They, in turn, quickly passed the message to their own colleagues, and so on. So, you are again to determine the minimum time for the information about Tanya's method to spread among the entire ACM agency. There is a peculiar feature of the company's phone network you should take into account. Namely, every employee can only call his/her direct subordinates or immediate boss (this is supposed to prevent girls from chatting over the phone instead of doing their work). Each phone call takes exactly one minute.

Input

The first line of input contains the number N of ACM employees ($N \leq 100000$). Each employee is assigned the unique ID number (these numbers range from 1 to N).

N lines follow, K-th line containing zero-terminated space-delimited list of K-th employee's direct subordinates. The last line contains Tanya's ID number. The hierarchical structure is a tree. I.e., each employee has exactly one direct boss, of course, with exception for the topmost boss.

Output

Output consists of one number — the minimum time, in minutes, that is required to propagate Tanya's idea to all ACM employees.

Sample

input	output
10 2 3 0 4 5 7 0 6 9 0 0 0 8 10 0 0 0 0 0 2	5

Problem Author: Eugeniy Krokhalev
Problem Source: IX Collegiate Students Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1363. Halftones

Time limit: 1.0 second
Memory limit: 64 MB

Ivan Pyatochnikov loves to tinker with electronics. That's why he has a great authority with his friends. Indeed, he can repair whatever piece of hardware you give him. He will deal with any incorrectly configured program. And he will even fix bugs in your code.

Today is Ivan's birthday, and his buddies are preparing a really cool present for him — namely, a genuine shamanic tambourine. Yes, a tambourine. After all, no real geek can live without such an indispensable thing.

Having bought the tambourine and read the manual, Ivan's friends have discovered that a tambourine only becomes effective when there is a picture of owner printed on it, along with some special magical signs. Now the friends are rather depressed, because they only have a dot matrix printer and a file containing the grayscale photo of Ivan. Clearly, any image to be printed on a dot matrix printer shall be monochrome (i.e. shall contain black and white pixels only). Ivan's photo consists of different shades of gray, and friends are afraid that the printout won't resemble the original photo closely enough for the tambourine to work.

You have agreed to help Ivan's buddies prepare the present.

Input

The first line contains two integers N and M ($75 \le N, M \le 300$) — the dimensions of the image. N following lines, each line containing M integers ranging from 0 to 255, define the image itself.

Output

The output shall contain the monochrome image in the form of N lines with M numbers on each line. Each number must be either 0 or 255. For the resulting monochrome image to resemble the original, the following condition must be satisfied for each i and j:

where $A[i, j]$ is the brightness of the original image pixel (i, j) and $B[i, j]$ is the corresponding monochrome image pixel brightness.

Sample

input	output
3 3 255 255 0 0 0 255 254 0 253	255 255 0 0 0 255 255 0 255

Notes

Note that this example is incorrect, because N and M are out of the defined range. Nevertheless, the example shows the correct format of input and output.

Problem Author: Eugeniy Krokhalev (idea by Pavel Atnashev)
Problem Source: IX Collegiate Students Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1364. LaraKiller

Time limit: 1.0 second
Memory limit: 64 MB

A cemetery has a rectangular shape with N rows of graves and M graves in each row. The cemetry is encircled with a high fence. As you know, Lara Croft has penetrated into the cemetry through the sap in the North-Western corner in order to find the hidden treasure. To do that, she has dug a passage under the cemetery according to the following rule:

1. If there was an intact grave straight ahead, then Lara lengthened the passage and ravaged this grave.
2. If there was a cemetry fence or a ravaged grave in the way, then Lara turned 90 degrees right and continued with her questionable affairs.

Lara has just found the treasure in the grave (r_T, c_T). Now she is returning home to drink champagne. But Dark Forces have decided to ensnare her. As soon as Lara will reach the grave (r_L, c_L), a new alarm system of "LaraKiller" brand will turn on. It will revive skeletons in some of the ravaged graves. All these skeletons will be revived at the same moment, T seconds later.

Lara realizes that the alarm system turns on. She will try to escape the cemetery (if she will have enough time) or to hide in a ravaged grave. Lara can run along her underground passage not faster than one grave per second in the direction of the exit or backwards. So before these T seconds run out she can hide in any ravaged grave at a distance not more than T from (r_L, c_L).

You are a head programmer of "LaraKiller". You don't want to revive unnecessary skeletons, so you need to find all the ravaged graves where Lara can be located at the moment of skeletons' revival.

Input

The first line contains integers N and M, that are the dimensions of the cemetery ($2 \le N, M \le 100$). The North-Western grave has coordinates (1, 1) and the South-Eastern one has coordinates (N, M). Lara starts to dig her passage from the grave (1, 1) moving to the East, i.e. to the grave (1, 2).

The second line contains integers r_T and c_T that are the treasure grave's coordinates. The third line contains integers r_L and c_L that are Lara's coordinates at the moment the alarm system turns on. $1 \le r_T, r_L \le N$; $1 \le c_T, c_L \le M$. It is guaranteed that the passage from the grave (1, 1) to the grave (r_T, c_T) goes through the grave (r_L, c_L).

The fourth line contains an integer T that is a time of skeletons' revival, in seconds ($0 \le T \le 27000$).

Output

Output coordinates of the graves where Lara can be located T seconds after the alarm system will turn on. The coordinate pair of each grave is to be output in a separate line. The graves should be output in order Lara ravaged them.

Sample

input	output
5 4 4 3 2 2 5	5 2 5 1 4 1 3 1 2 1 2 2 2 3 3 3 4 3

Notes

In the picture you can see the graves where Lara can hide.

Problem Author: Stanislav Vasilyev
Problem Source: IX Collegiate Students Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1365. Testing Calculator

Time limit: 2.0 second
Memory limit: 64 MB

Thursday. April 21. First round of IX Urals Programming Contest Championship has just finished. All teams left the contest area and went to celebrate results of the first round, make sightseeing tour around Ekaterinburg, or just have a little rest after competition.

Jury would like to have a rest too, but it is impossible. It turned out, that one task for the second round is missing. Moreover, the error in contest management system was discovered. No, it is not a critical bug, which distorts results, but a very annoying mistake: one module of the contest management system makes processing of each submission longer by at least 30 seconds.

Some investigations were done. It was found, that nobody knows what this module is intended for. With this module contest management system works very slowly. But without the module it does not work at all.

It seems, that there is no problem. One may take module sources, read them, find specifications for the module, resolve the error, compile the module and put the corrected version into the system. Unfortunately, the module was written 8 years ago when The First Urals Programming Contest was prepared and now sources are not available. They are just lost. Jury feels lost too.

Eureka! We can give out the module to participants of the contest! They are clever enough to run it, discover the logic and write another version of the module, which will run without delay. Moreover, it is an excellent task for the second round!

Input

Jury was able to discover some information about the module. It is known that the module reads input data from the input stream. Each line of the input contains single arithmetic expression, which is evaluated by module. Each expression consists of arithmetic operations of addition (+), multiplication (*), integer division (/) and concatenation.

Arithmetic operations are written in prefix form with unlimited number of operands. This way expression "+3;6;10" has a result of "19". Operation of concatenation does not have any notation and gets executed when no other operation is given. This way expression "3;6;9" is evaluated to "369". It is known also that expressions may contain brackets. For example, expression "(+3;6)(*2;3)" has a result of "96". It may be assumed that logic of expression evaluation does not depend on context. It means, that each subexpression is always evaluated in the same way with no dependency on it's entrance into the whole expression.

Nothing more is known about the module. You see, that there is no information on how arithmetic operations interact with concatenation, brackets and separator character (;). You have to discover it by yourself. To do it, you will be given an existing module. You have to run it, investigate the logic and reproduce it in your solution.

Input data will be located in the input stream. Each line should be interpreted as a separate expression. Expression will consist of the following characters only: "0123456789+*/();".

You may assume, that all intermediate values and result of expression evaluation are in the range from 0 to 10^9. You may also assume that there will be no expressions longer than 50 characters.

Output

Your solution must reproduce the logic of the given module. It should produce output stream with the result of evaluation of expresions from the input. Each line of output must be of the following form: "Expression evaluates to: ", where is the line number and is the result of evaluation of expression in the line of the input stream.

Sample

input	output
(+3;6)(*2;3) (+3;6)(/3;2)	Expression 1 evaluates to: 96 Expression 2 evaluates to: 91

Notes

1. To run the existing module, use http://acm.timus.ru/Supplement/CalcEngine/CalcEngineGUI.html
2. Remember that existing module runs not less than 30 seconds on any set of input data.
3. You may need to update Java Runtime Engine (JRE) used by your browser. To run the module you need JRE of version 1.3 or higher. The latest version of JRE can be downloaded from here: http://java.sun.com

Problem Author: Aleksandr Klepinin
Problem Source: IX Collegiate Students Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1366. Presents

Time limit: 1.0 second
Memory limit: 64 MB

A nice entertainment was invented in the kindergarten. Each child has to bring a present from his home — a big box with something interesting inside. Contents of the boxes shall be kept secret up to the last moment. After that the chidren will exchange their presents.

Children understood that they would have to part with their presents so they stuffed their boxes with useless junk: candy wrappers, husk, broken computer mice and even unnecessary elder brother's fat book with some kind of donkey on the cover.

So, a child didn't mind parting with his box. Moreover, he didn't calm down until he made sure he got rid of his box. Having foisted his box off, no child ever took it back. If some kid didn't get a present in exchange for his own one, he would become very disappointed and his loud cries would attract attention of a nurse who had to take away all those boxes along with the marvellous content!

As a chief information officer of entertainment operations, you are to find out the amount of the present exchange schemes such that each child would be pleased. But there is one hitch… Your hand-book on algorithms was taken by your younger brother to his kindergarten for some purpose. Sixteenth chapter might prove very useful…

Input

One number n ($1 \le n \le 1000$) — the amount of children in the kindergarten.

Output

One number — the amount of exchange schemes. E.g., for three presents A, B and C there are only two exchange schemes:

1. Box A goes to the child B, box B — to the child C and box C — to the child A.
2. Box A goes to the child C, box C — to the child B and box B — to the child A.

Sample

input	output
3	2

Problem Author: Pavel Egorov (idea by Stanislav Vasilyev)
Problem Source: IX Collegiate Students Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1367. Top Secret

Time limit: 2.0 second
Memory limit: 64 MB

In the top secret district of the top secret city, there is a top secret factory which produces top secret Secrets. The top secret spy satellite, equipped with top secret imaging devices, took some top secret photographs. You have to interpret the photograph correctly. This is crucial for success of the top secret mission involving covert manipulation of potential enemy's Secrets.

The photo is repesented by a $W \times H$ matrix of cells, each cell containing one of the symbols:

.	Empty, passable section
-	Horizontal impassable barrier (wall)
|	Vertical impassable barrier (wall)
+	Impassable junction of horizontal and vertical walls
#	The Secret

The Secret is a regular junction (+) with one of the objects of special interest contained inside. This object of interest is accessible from any of the four sides of the junction.

Unfortunately, secret agents are unable to penetrate walls, nor they are allowed to break them. But this is the only limitation — they are secret agents, after all.

We'll give a set of examples to clarify the agents' possibilities.

```
+-+-+ |-+-|
|#|#| |#|#|
+---+ +-+-+
```

In the first example the agent can walk freely from one Secret to another through a hole between the bottom horizontal and middle vertical walls. In the second example the hole is sealed by the junction. Agents can't use holes in the top wall — leaving the photographed area poses a huge risk to the mission. For the sake of simplicity, you can assume that the photographed area is surrounded with the barrier of "+"-es.

```
+-+...+-+ +-+...+-+
|#+---+#| |#+-+-+#|
+---+---+ +---+---+
```

Similarly, in the first example Secrets are connected, and in the second they are not.

```
+-----+ +++++++
+######+ +######+
+++++++ +++++++
```

If you remember that the Secret is a special case of junction, it will become clear why all the secrets in the first example are connected and why every secret in the second example is connected only with its neighbors.

Your task is to construct the secret connectedness graph. That is, the graph which set of vertices is the set of all secrets and the edge (a, b) exists if and only if it is possible for an agent to get from Secret a to Secret b without breaking the walls and leaving the area.

Limitations
The dimensions of the photograph don't exceed 1000 × 1000 cells. It is known that there are no more than 100 Secrets on the photograph.

Input

Input stream format is rather simple: it contains H lines, and each line contains a string W symbols long. These strings of symbols dene the secret photograph of size $W \times H$.

Output

Output the incidency matrix of the secret connectedness graph. Secrets are enumerated from top to bottom, from right to left. The matrix element in the i^{th} row and in the j^{th} column is 1, if i^{th} and j^{th} secrets are connected, and 0 otherwise.

Sample

input	output

+------+ |#+-+--| -++#|..| ##..|#.| -+--+--+	10100 01010 10100 01011 00011

Problem Author: Pavel Egorov
Problem Source: IX Collegiate Students Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1368. Goat in the Garden 3

Time limit: 1.0 second
Memory limit: 64 MB

How do you let your goat walk around your vegetable garden? That becomes a real problem sometimes. You should make sure the goat doesn't die of starvation, and you certainly don't want all your crops to be eaten. The garden is a field consisting of 1 × 1 square elements. You can build a fence in some of the garden's elementary squares. To feed itself, the goat needs to eat all the crops in the area consisting of K ($1 \le K \le 10^6$) squares. The goat initially stands in the point of origin (i.e. in the square with coordinates (0, 0)). You have to put the fence in the minimum number of squares so that the goat will be able to visit exactly K squares. The area is said to be fenced if it is impossible to leave it moving only in horizontal or vertical directions.

Input

The input stream contains one number K — the total area granted to the goat.

Output

The first line shall contain the amount N of boundary (fenced) squares. Each of the following N lines shall contain two numbers — X and Y coordinates of the fenced square, respectively. Squares should be output in the order of traversal.

Sample

input	output
3	7 -1 0 0 1 1 2 2 1 2 0 1 -1 0 -1

Problem Author: Aleksey Lakhtin
Problem Source: IX Collegiate Students Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1369. Cockroach Race

Time limit: 2.0 second
Memory limit: 64 MB

At last, the spring came. Buds swell on the trees, the snow has almost thawn out. More and more often you can hear birds' sonorous twittering from the outside. Less and less students you can see at the USU math-mech department. Even the cockroaches, usual inhabitants of the hostels, show up very rarely.

What's the connection between these phenomena, you may ask.

The answer is the Day of Mathematician and Mechanician celebration, which will begin really soon. At the same time, the traditional cockroach race will take place in the USU. That's what the students are occupied with now—they are training their pets. Everyone wants his pet to become the prize-winner and to receive the proud name of "Magaz".

The race rules are somewhat unusual. Every round, some kind of sweets are placed in N points of the racing area. Together with sweets, M cockroaches are released. N cockroaches that reach these little delights of cockroaches' life, will make it to the next round.

During the race all spectators have an unique opportunity to place bets and to win a lot of money. But the totalizator organizers are puzzled, they cannot understand how to calculate the probabilities of cockroaches' victories quickly and without mistakes. This is absolutely required to make the maximum profit out of their enterprise. Math-mech is rather big department and everyone here wants to participate.

You are to determine, for each of N pieces of sweet, which of the cockroaches is closest to that piece. This will help to determine the race leaders.

Input

The first line contains an integer M ($1 \le M \le 100000$). M lines follow, containing coordinates of the cockroaches at the present moment. ($M + 2$)nd line contains an integer N ($0 \le N \le 10000$). N following lines contain coordinates of sweet pieces. All coordinates are floating point numbers ($-10000.0 \le x, y \le 10000.0$). The distance between any two cockroaches is not less than 10^{-3}. Also the distance between any two sweets is not less than 10^{-3}.

Output

For each piece of "Cockroach Sweets" you should output all cockroaches closest to that piece in ascending order of their numbers separated by spaces.

Sample

input	output
4 0 0 1 0 0 1 1 2 2 0 0 0 2	1 3 4

Problem Author: Den Raskovalov
Problem Source: IX Collegiate Students Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1370. Magician

Time limit: 1.0 second
Memory limit: 64 MB

Denis Mednoepole is a famous magician. For example, he can read spectators' minds and tell the random number that a spectator has got by means of Denis's magic machine. Any spectator can check this: he or she just rotates the wheel of the machine and looks at the 10-digit number that has appeared in the window of the machine. Within two seconds Denis will read the spectator's mind and tell the number.

You must write a program that will tell this number even faster — within one second. You need to know that the magic machine has a wheel with N digits written along the edge of the wheel without spaces between them. All digits are of the same width. The window is 10 digits wide, so it shows 10 successive digits. Each time the wheel turns "one digit" an audible click is produced. So you know how many clicks Denis Mednoepole heard while the spectator was rotating the wheel.

Input

The first line contains two integers separated with a space, N and M. Here N ($11 \le N \le 1000$) is the number of the digits written on the wheel, and M ($1 \le M \le 32767$) is the number of clicks produced by the wheel. The next N lines contain all the digits written on the wheel, one digit per line. The digits are given in the clockwise order. The first ten digits correspond to the number visible in the window before the rotation. The wheel can be rotated only counter-clockwise, thus after one click the first digit will go out of site and the 11-th digit will become visible.

Output

Write the number that will appear in the window of the magic machine after the rotation. The number must have exactly 10 digits and may contain leading zeroes.

Sample

input	output
12 7 1 2 3 4 5 6 7 8 9 0 7 7	8907712345

Problem Author: Stanislav Vasilyev
Problem Source: IX Collegiate Students Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1371. Cargo Agency

Time limit: 1.0 second
Memory limit: 64 MB

Advanced Carriage Messaging company does business in cargo delivery, so it has a complicated network of branch offices. Business is successfull, and it was decided to establish new offices to extend further the delivery network. But first of all company management department wants to analyze the efficiency of delivery between offices. You were asked to do this analysis, because of your renowned experience and knowledge.

Delivery works in the following way. There exists exactly one route of cargo delivery from any office to another office (possibly via intermediate ones). The times $t[i, j]$ of cargo delivery between two offices (with numbers i and j) have been measured. These times are available only for offices which have direct communication. Direct cargo delivery for other offices is impossible. You are asked to calculate the average delivery time between offices, i.e. the following value: sum ($t[i, j]$) / (N*(N − 1)), where the sum is taken for $1 \le i, j \le N$ and $i \ne j$.

Input

The first line of input contains one integer N ($2 \le N \le 50000$) — the number of branch offices of ACM company. Each of next N−1 lines contains three numbers a_i, b_i, c_i. Numbers a_i, b_i ($1 \le a_i, b_i \le N$) are numbers of offices which have a direct communication between them. Integer number c_i ($0 \le c_i \le 1000$) is a cargo delivery time between these offices.

Output

Output must contain a single number: average cargo delivery time between branch offices of ACM company with a precision of 4 decimal digits.

Sample

input	output
4 1 2 1 2 3 1 2 4 1	1.5

Problem Author: Aleksandr Klepinin (idea by Den Raskovalov)
Problem Source: IX Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1372. Death Star

Time limit: 1.0 second
Memory limit: 64 MB

A long time ago in a galaxy far, far away…

Well, let's skip these well known entry words of the famous movie. No doubt, anyone knows them.

Anyway, a long time ago in a galaxy far, far away there was a space station named "Death star". For those of you who do not remember it, we would remind that it mostly looked like a big metal ball with lots of turrets on its surface. And somewere between turrets there were towers with maneuring engines on a top of each tower; that allowed the station to turn in space. All the engines were the same: standard krypton engines with autonomous power supply. And all the towers were the same. Each of the engines could be in two distinct states: switched off and switched on to a maximum power. But it was not a problem since each engine could be rotated individually relative to the station.

Unfortunately, the engine control cable was damaged after one of the space battles: it was melted by the direct hit of a rebel laser weapon. So all the manoeuvre drives became uncontrollable. All of them which were at maximal power just kept working. Switched off engines remained in that state. Death Star won the battle, but stuck manoeuvre drives made a strange effect: instead of smooth and majestic movement through space, station makes unpredictable rotations which are unsuitable for the Imperial fleet flagship.

It makes Emperor annoyed. No, not just annoyed but furious, because it seems to him that the station rotates faster and faster, and he already feels nauseous.

You have to fix the situation as soon as possible, otherwise you are not guaranteed to stay alive for long.

Engine control cable can be fixed in a week. You see, it is too long. So you have to establish another engine tower with a maintenance engine to stop the rotation. And you have to do this really quickly.

You may have thought that you would need to solve a lot of complex differential equations and to write engine controlling program. No, not at all. As a station engineer, you must know that jet thrust of a maintenance engine does not change over time. It depends on jet nozzle size only. So you only have to find the suitable place for the engine tower, calculate jet nozzle size, set up thrust vector and switch the maintenance engine on. No matter how you do it. Just make it in such a way that the working krypton engines won't further speed up the rotation.

Input

In the first line of input, some technical data on the Death Star station are given. You know, these are top secret data, but without them you will definitely fail your mission.

The first number gives a radius of station in meters (it ranges from 100 to 2000), then total station mass is given (in kilograms, not greater than 40000000, you may assume that the mass is uniformly distributed throughout the station), followed by the height of a standard engine tower (in meters above the station surface; maintenance engine will be placed on the very same tower, and you may place this maintenance engine tower anywhere on the station surface).

Next line of input contains krypton engine description. It starts with a number F — standard jet thrust of krypton engine in newtons (ranges from 1 to 100000). Then an integer number N ($1 \le N \le 9$) follows. It is a number of krypton engines currently working at maximum power.

Each of the following N lines contains description of working krypton engine placements. Each line contains 4 numbers. These are: longitude (from 0 to 360 degrees) and latitude (from −90 to 90 degrees) of engine placement relative to the station. Other two numbers define the jet thrust vector of the engine: if this vector is applied to the center of the station, then it would intersect station surface at the point with longitude and latitude given by these two numbers.

Output

Print 5 numbers with a precision of eight decimal digits: two numbers (longitude and latitude) must define place to establish the maintenance engine tower, then two numbers must define jet thrust vector (in notation described above) of the maintenance engine, and the last number must define jet thrust in newtons. You should choose the placement with the least jet thrust among different possible maintenance engine placements.

Sample

input	output
100.3 1000000.45 15.5 156.5 1 0.0 0.0 90.0 0.0	0.00 0.00 270.00 0.00 156.50

Problem Author: Pavel Egorov (idea by Aleksandr Mironenko)
Problem Source: IX Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1373. Pictura ex Machina

Time limit: 1.0 second
Memory limit: 64 MB

Don't you want to be a millionair? It is so simple! You only need to automate some prevalent handwork, and then you save enough money to buy so long wanted beautiful red Ferrari.

For example, why not to automate a process of dime novels? Or why not to automate a process of writing songs for pop-stars? No! It is too simple for real programmer, it is below you! So in this task you will need to automate something spiritual: the process of paintings creation! Just think about it: home computer will be able to create beautiful landscapes, artful still lifes and even delicate self-portraits!

Frankly speaking, painting algorithm is already developed and implemented by your colleague, great programmer Mr.Petrov. His algorithm makes a description of brushstrokes, so it is enough to load this description into plotter, and in a couple of hours the picture will be ready.

Meanwhile a frame for the picture may be prepared. But this is where Mr. Petrov fails: he can not calculate size of the picture given the brushstrokes description produced by his algorithm…

It could be just trivial if each brushstroke was a line segment. But the major innovation of Petrov's algorithm is that his rushstrokes has a form of an angle. To make this kind of brushstroke one needs to draw (by a brush, or a plotter) a line segment of X millimeters length in one direction, then turn 90 degrees clockwise and draw another line segment of X millimeters length. This is what innovative Petrov's brushstroke looks like.

Input

Input contains N ($0 \le N \le 100000$) lines of numbers. Each line describes single brushstroke. Description consists of four integers: coordinates of the start and coordinates of the end points of the brushstroke. All integers do not exceed 10000 in absolute value.

Output

Output should contain 2 numbers: the width and the height of the smallest frame for the picture, produced by brushstrokes description given in input. Numbers must be separated with single space. And they should have a precision of 4 decimal digits.

Notice, that picture must not be rotated, even if it allows to fit the picture into the frame of smaller size. Your task is not to save materials, from which the frame is made. You just have to prepare suitable frame for the picture. Also make sure that all parts of each brushstroke are inside the frame.

Sample

input	output
0 0 1 1 1 1 2 0 2 0 0 0 2 0 3 0	3.0000 2.0000

Problem Author: Pavel Egorov
Problem Source: IX Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1374. Misere

Time limit: 1.0 second
Memory limit: 16 MB

Naked man walks along the streed. Policeman stops him ans asks:
"Are you alright? Why naked?"
Man answers with a question:
"Listen, does a misere ever gets spoiled by 7, 9 and J?"
Policeman: "Never!"
Man: "And I thought so…"

It is six o'clock. Morning. Or night. Kitchen. Four students play a game of preference. Or do not play. Indeed, they do not play. But why? What happened? What are they busy with? Oh… They dispute about something. Oh, got it!

One of them bid a misere contract. And believes that he will not get any trick. But other players are trying to convince him of being wrong. Their voices are already hoarse. They even opened all the cards! Made discarded cards known! But nothing helps…

Then one of them tells:
— Aren't we students of Mathematical and mechanical department? Let's write a program, which will look over all possible plays and justify us.

Nobody objected. But because of they were bad students or maybe due to sleepless night the program was never written… Will you help them?

Oh, you need to know some basic facts about preference. Precisely, you only need to know rules of playing a misere.

Preference is played with 32 cards: the Ace (A), King (K), Queen (Q), Jack (J), 10, 9, 8 and 7s (in the order of decreasing rank) from a standard 4-suit 52-card deck. Suits are usual: Spades, Clubs, Diamonds and Hearts. Preference is played by three players sitting at the round table (in fact, the shape of the table does not matter; the only important thing is that each player has right and left neighbour and that players are numbered clockwise).

After cards are dealt, bidding takes place. We omit bidding rules here, they are not important for this problem. One of the possible result of bidding is a misere contract. It means, that player will try to win no tricks with his cards.

After the bidding each player has 10 cards. So 10 tricks are played. Each trick is played in the following way. The player, who leads the first trick, puts any of his cards on the table face up. Then his left neighbour puts his card. Then remaining player puts his card.

Each player must follow the suit of the first card in a trick, if possible. Otherwise, he may discard any card from his hand. The player, who put the card of the highest rank with a suit of the first card in a trick, wins that trick. The player who wins a trick leads the next trick.

Successfull playing of the misere contract gives a big profit to the player. But each won trick gives a great loss. Now you know the actual reason for the dispute!

Your task is to justify the students. Given cards of each player and a number of the player, who leads the first trick, you are to determine, whether the first player, who declared the misere contract, is able to play the declared contract regardless of the other players' strategy.

Input

Each of the first three lines of the input contain cards of the corresponding player. Each card is described with two characters: a suit (one of the S, C, D or H) and a rank (one of the 7, 8, 9, 10, J, Q, K or A). Card descriptions are separated with space(s). The fourth line contains the number of the player, who leads the first trick.

Output

Output should contain exactly one line. In case when the misere contract can be played successfully regardless of the strategy of other players, this line should contain ";)". Otherwise it should contain ";(".

Sample

input	output
7S 8S 9S 10S JS QS KS AS 7C 8C 9C 10C JC QC KC AC 7D 8D 9D 10D JD QD KD AD 7H 8H 9H 10H JH QH 1	;)

Problem Author: Den Raskovalov
Problem Source: IX Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1375. Bill Clevers

Time limit: 1.0 second
Memory limit: 64 MB

System administrator with a well sounding name of Bill Clevers pays great attention to operating system security vulnerabilities. He truly believes that a system protection must be rather original to be reliable. So he decided to design a new public key cryptography system. No sooner said than done. One sleepless night, a little use of his favourite shamanic tambourine (did you hear about this important tool of system administrator?), a couple of broken computers, and a brand new cryptographic system is designed!

In this crypto-system, public key is a pair of numbers (k, p) where p must be a prime number and $0 \le k \le p-1$. Private key is a pair of numbers (x, y), which satisfies the following relation:

$$x^2 + y^2 = k \pmod{p}$$

Of course, Bill does not think that his cryptographic system is better than RSA, but it is definitely much less known to potential hackers.

Everything could be just fine, if Bill had not (unfairly!) restricted access to some Internet sites, which were used by the remarkable programmer Bob Buggins to create his software masterpieces. Evidently, this unfair act of Bill must be circumvented somehow. To do it, Bob just needs to recover Bill's own private key from the corresponding public key. But Bill's cryptographic system appears strong enough! That long-awaited private key still can't be found by Bob…

Input

Input contains two integers: k and p. Number p is a prime number, $2 \le p \le 10^6$; $0 \le k \le p-1$.

Output

Output any pair of integers x and y ($0 \le x \le p-1$, $0 \le y \le p-1$) such that $x^2 + y^2 = k \pmod{p}$. If there is no such a pair, output "NO SOLUTION".

Sample

input	output
1 3	0 2

Problem Author: Den Raskovalov
Problem Source: IX Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1376. Dean's Pyramid 2

Time limit: 1.0 second
Memory limit: 64 MB

At Ural Championship 2004 there was a task "Dean's pyramid". Briefly it looks like the following:

> There is an egiptian like glass pyramid on the table of Ural State University mathematics and mechanics faculty dean. The faculty legend says that before weak student gets sent down, dean gives the student the last chance. He puts unsigned list of dismissed students at one side of the table, and puts the pyramid at another side. If the student is able to move the pyramid to the locus of deans signature on the dismissing listplace, by rolling the pyramid over the edges for no more than 70 times, the student stays at the faculty (for the last time).
>
> The task was to determine, how close the pyramid can be moved to the target point under the following conditions. The pyramids base is square and its lateral faces are regular triangles. You can roll the pyramid by turning it from one face to the adjacent one over some edge. During this turning the edge should not slide on surface of the table. If after the turn over some edge the pyramid stands on its base, the next turn can be performed either over the same edge or over the opposite edge of the pyramids base only. There are no restrictions on the rolling from the triangle faces.

After Ural Championship 2004 weak students felt secured: now they know the secrets of dean's pyramid and the simplest algorithm to find the points on the table where the pyramid could be. To learn this algorithm is much, much simpler than to learn theorems. And you never get sent down! The only doubt disturbed them: what if authors of the task garbled something...

And they were right: looking at the hidden camera picture it was found that side faces of the pyramid were not regular triangles, but only isosceles one! And strange rule of rolling the pyramid could be just a fiction of task authors... Weak students feel distress again...

You see, this could be a legend. And could be a truth. The pyramid is so real, and it looks so beautiful at the dean's table... So help the students for the last time. Try to determine the best way of rolling the pyramid to put it as close as possible to the point of dean's signature.

You are given a number N — the maximum number of rolls allowed by the dean. You are also given an exact size of the pyramid: the length of an edge of square face A_1 and the length of a side edge A_2. Both lengths are given in centimetres. It was estimated by eye that a value of A_2/A_1 is in a range from 0.9 to 1.9. It is known that the dean is severe enough, so he puts dismissing list far enough from the pyramid. You may assume that the pyramid is originally placed not closer than 20.05 * A_1 to the point of the dean's signature.

Pyramid originally stands on its base with a center of the base placed at the point (0,0) and with edges of the base parallel to the axes of coordinate system. You are to find the sequence of rolls by which the pyramid gets to the point (X,Y) as close as possible. After the last roll the pyramid again must stand on its base, though edges of the base could be arbitrary oriented. But the distance between the center of the pyramid base and the point (X,Y) must be as small as possible. In the case of tie, you also have to minimize the number of rolls. You are allowed to make less than N rolls. The pyramid can be rolled from one face to another only via their common edge. And the edge should not slide on surface of the table during the roll.

Input

Input contains the numbers A_1 and A_2 (in centimetres, both in range from 0.1 to 10). Two numbers X and Y follows. They are the coordinates of sacramental point of the dean's signature. The number N ($2 \le N \le 32$) ends the input.

Output

You have to conform the following format. First of all pring the distance between the center of the pyramid base and the signature point in the pyramid's final placement. This distance must be printed correct to three places of decimals. Then print the number M ($0 \le M \le N$) of rolls required to get the printed distance.

Then print M+1 numbers — the numbers of the faces on which the pyramid stands after each roll. Separate these numbers with spaces. And do not forget to start this list with the number of the pyramid base on which it stands initially.

Faces of the pyramid are numbered according to the following scheme. The pyramid's base has number 0. The side faces are numbered clockwise by 1, 2, 3 and 4 being looked from the top of the pyramid standing on it's base. The face of the pyramid which looks to positive direction of 0Y axis has a number 1.

Sample

input	output
10.0 10.0 1000.0 0.0 5	972.679 5 0 2 3 4 1 0

Problem Author: Aleksandr Mironenko
Problem Source: IX Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1377. Lara Croft

Time limit: 1.0 second
Memory limit: 64 MB

A cemetery has a form of rectangle. There are N rows of graves, M ones in each row. The cemetery is enclosed with a high fence.

Lara Croft has penetrated into the cemetery through the sap at the Northwestern corner. It takes one night for Lara to dig a subway under one of the graves. If there is an intact grave straight ahead then Lara will lengthen the passage during the next night and will ravage the grave. If there is a cemetery fence or a ravaged grave on the way, then Lara will turn 90 degrees clockwise and will continue with her questionable affairs.

Treasures are located in two graves only. And we exactly know in which ones. But Lara doesn't. Lara has bought a package of champagne today. It means, that today she has found one of those graves. We wonder how long will it take her to find the other one?

Input

The first line contains integers N and M that are the sizes of the cemetery ($2 \le N, M \le 100$). The North-Western grave has coordinates (1, 1) and the South-Eastern one — (N, M). Lara starts with the grave (1, 1) moving to the East, i.e. towards the grave (1, 2).

The second and the third lines contain integers (r_1, c_1) and (r_2, c_2) that are the treasure graves coordinates ($1 \le r_i \le N$; $1 \le c_i \le M$). The order of graves is arbitrary, i.e. Lara may reach the first or the second grave earlier.

Output

Output an amount of days that Lara will spend reaching for another grave with treasures.

Sample

input	output
5 4 2 2 5 3	6

Notes

In the example Lara will find the treasures on the 9'th and the 15'th days.

Problem Author: Stanislav Vasilyev
Problem Source: IX Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1378. Artificial Intelligence

Time limit: 1.0 second
Memory limit: 64 MB

In his novel "2001: A Space Odissey", Arthur C. Clarke described a situation when a spaceship computer strengthened with an artificial intelligence decided to get rid of humans being aboard. To achieve its goal, the computer invented some tricks. In particular it was able to make a man leave the spaceship without a spacesuit...

In fact, many people just like science fiction writers thought that in the year 2000 the artificial intelligence would be definitely invented. And computers would communicate with humans as equals. Now it is evident that even the problem of human speech recognition (we do not speak about recognizing the meaning of what's said!) is hard enough.

Your task is to make a big step forward and to write a program to recognize photos. There is a photo of a man. You are to process the photo and recognize the nationality of the man. During this competition we will give you photos of Flatland citizens only. So there are only three nationalities: Circle, Square and Triangle.

Input

The first line of the input contains two numbers W and H — the width and the height of the photo ($32 \le W, H \le 1000$). Then H lines of W numbers of 0 or 1 follow. It is a black-white photo of a black figure (circle, square or triangle) drawn on a white background in a standard photo editor. There is only one figure at the photo. The figure does not touch the borders of the photo. It does not exceed the bounds of the photo either. Size of the biggest side or the radius of the figure is not less than 30 pixels. The least angle of the triangle is not less than 10 degrees.

Output

Just print one of the following words: **circle**, **triangle** or **square**, depending on what figure is drawn at the picture.

Sample

input	output
32 32 00000000000000000000000000000000 01111111111111111111111111111110 00000000000000000000000000000000	square

Problem Author: Stanislav Vasilyev (idea by Aleksandr Klepinin)
Problem Source: IX Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1379. Cups Transportation

Time limit: 1.0 second
Memory limit: 64 MB

It was decided to prepare specially designed cups with a competition logotype of Ural Championship 2005 for each participant and for any observer, who wish to have such a cup.

Having a habit to do such important things at the last moment, designer finished his job with a cup design two days before the Championship. So there is a shortage of time. One day is needed to manufacture the cups and to put Championship logotype onto them. So there are 24 hours left to deliver the cups from an enterprise to contest area.

Of course, it is impossible to deliver ten million cups (this is the amount of cups ordered by organizing committee) in a one trip. But it is desirable to deliver as many of cups as possible in a one trip.

Organizing committee ordered one big delivery truck. One would try to load the truck by the cups for it's full load. But some roads have weight limit: if the weight of the truck exceeds the road weight limit, it can not pass the road. So it is possible, that fully loaded truck can not go along the shortest route of delivery, it has to use longer route with acceptable weight limit. Moreover, it is possible, that the fully loaded truck will not deliver the cups in time. And it is definitely unacceptable.

So, how many cups can be loaded into the truck, so that this precious load could be delivered in time without violation of road weight limits?

Input

The first line contains two numbers: the total number of road map nodes N ($1 \le N \le 500$) and the total number of roads M. Next M lines contain information about the roads. Each road is described on a separate line in the following way. Two numbers of road map nodes connected by the road go first. They are followed by the time required to pass the road. Finally the weight limit for the road goes. It is known that each road connects two different road map nodes. And for any two road map nodes there is at most one road connecting them. All numbers are separated with a one or more space(s).

Road map nodes are numbered with an integers in a range from 1 to N. Enterprise, where the cups are manufactured, is always placed in a node 1. And Urals Championship area always in at node N. Road passage time is given in minutes, and it does not exceed 1440 (24 hours). Weight limit is given in grammes, and it does not exceed 10^9. One cup has a weight of 100 grammes, and the empty truck has a weight of 3 tonnes (1 tonne = 1000 kilogrammes; 1 kilogramme = 1000 grammes).

Output

Print the only number: the total amount of the cups (it must be as much as possible), which can be delivered in the first trip of the truck in a time of 24 hours.

Sample

input	output
3 3 1 2 10 3000220 2 3 20 3000201 1 3 1 3000099	2

Problem Author: Pavel Egorov
Problem Source: IX Urals Programming Contest. Yekaterinburg, April 19-24, 2005

1380. Ostap's Chess

Time limit: 1.0 second
Memory limit: 64 MB

The well-known grand-master Bender arrives at the city of Petrozavodsk. Among others, there were Ha, Ru and Po to come to the simultaneous chess-game. Ru had lost the game N ($1 \le N \le 10^9$) times, Po had lost the game M ($1 \le M \le 10^{1000}$) times, and Ha hadn't counted his losses as all. The friends had noticed that Bender played in his own style: the chess-pieces occasionally disappeared from the board, or new pieces occasionally appeared. Bender has no shame, and he can do the cheats with any chess-piece. Then Ha got tired of these games, and he decided to start writing down the moves, having written down the current position first. You have to tell from Ha's notes whether Ostap cheated during play, or not.

The rules of chess, as known by Ostap, are much like the common ones, but Ostap doesn't know such concepts as *en passant*, *castling* and *promotion*. Otherwise, the pieces move just like the usual chess-pieces do. A player's move is considered legal if the only piece moved is this player's piece, and his king isn't in check at the end of the move. A king is considered in check when its location is under attack of a hostile piece, and checkmated when, in addition to this, there is no legal moves. Stalemate (a draw) is a position when neither king is in check, and there is no legal moves. The initial position may be senseless; it needn't be verified before the first move.

Input

The current position is given as 64 numbers, delimited with spaces or new-lines. The number 0 corresponds to an empty square, 1 — to pawn, 2 — to knight, 3 — to bishop, 4 — to rock, 5 — to queen, 6 — to king. The positive numbers correspond to white pieces, and their negative counterparts — to black ones. A number $1 \le P \le 60$ follows, which gives the number of moves written down by Ha. Following is the color of player to move, either 'White' or 'Black'. At last, the P moves are given, formatted as square-square, where a square is a letter a-h (a corresponding to the leftmost file, h to the rightmost) followed by a number 1-8 (1 corresponding to the bottommost rank, 8 to the topmost).

Output

For each move, your program must output either of the following messages: "Incorrect" for illegal move, "Mate" for checkmate, "Check" for check, "Draw" for stalemate, "Correct" for neither of the above. After an illegal move, checkmate or stalemate, the program must output an empty line followed by the current position, formatted in the same way as the input, without performing the illegal move. The moves following the illegal move, checkmate or stalemate must be ignored.

Samples

input	output
-4 -2 -3 -5 -6 -3 -2 -4 -1 -1 -1 -1 -1 -1 -1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 4 2 3 5 6 3 2 4 2 White e2-e4 e7-e4	Correct Incorrect -4 -2 -3 -5 -6 -3 -2 -4 -1 -1 -1 -1 -1 -1 -1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 4 2 3 5 6 3 2 4
-4 -2 -3 -5 -6 -3 -2 -4 -1 -1 -1 -1 -1 -1 -1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 4 2 3 5 6 3 2 4 8 White e2-e4 e7-e5 d1-f3 h7-h6 f1-c4 g7-g6 f3-f7 e8-f7	Correct Correct Correct Correct Correct Correct Mate -4 -2 -3 -5 -6 -3 -2 -4 -1 -1 -1 -1 0 5 0 0 0 0 0 0 0 0 -1 -1 0 0 0 0 -1 0 0 0 0 0 3 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 4 2 3 0 6 0 2 4

0 0 0 0 0 6 0 -6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 White h6-a6	Correct 0 0 0 0 0 6 0 -6 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Problem Author: Ivan Popelyshev
Problem Source: Petrozavodsk summer training camp, August 2005.

1381. Cockroaches in the Building

Time limit: 1.5 second
Memory limit: 64 MB

Petya lives in a many-storied building where each flat occupies an entire story. Petya has studied local flora and fauna for many years. Having performed montly metering, he notices the following peculiarities:

- The number of cockroaches in each of the flats is an integer ranged from 0 to *N*, where *N* is a non-negative number not exceeding 30.
- The number of cockroaches in a flat during a month is a function of three arguments: the number of cockroaches in this flat and the flats above and below this one during the previous month.

Petya becomes interested in inducing the ways the insects' population size changes under these laws. To avoid the special cases of ground floor and top floor, where the laws might be more complicated, Petya decided to solve this problem for an infinite-storied skyscraper. Let's consider that the number of cockroaches can decrease if they can be distributed among the flats (having no more than *N* in any) in such a way that in the next month the total number of cockroaches in the building will decrease. Only a finite number of cockroaches can exist in the building; thus, all the flats, except a finite number of them, are free from these insects. Likewise, the number of cockroaches can increase if they can be distributed among the flats in such a way that in the next month the total number of cockroaches in the building will increase (and may possibly become infinite).

Input

The first line of the input contains the single number *N* — the maximum number of cockroaches in a flat. Following are samples of the function that defines the changes in the number of cockroaches in a flat. The samples are organized in a table having *N*+1 block, where each block has *N*+1 lines, and each line has *N*+1 numbers ranged from 0 to *N*. *k*'th number of *i*'th line of *j*'th block contains the number of cockroaches in the flat for the current month if in the previous month there were *j* cockroaches in it, *i* cockroaches in the flat below, and *k* cockroaches in the flat above. The indices of blocks, lines, and numbers are zero-based. Petya has studied that if there were no cockroaches in some flat and in the flats below and above it in the previous month then there are no cockroaches in it in the current month.

Output

The output must contain either:

- =, if the number of cockroaches can neither increase or decrease, or
- <, if the number of cockroaches can decrease, but not increase, or
- >, if the number of cockroaches can increase, but not decrease, or
- <>, if the number of cockroaches can both increase and decrease.

Samples

input	output
1 0 1 0 1 0 0 1 1	=
2 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 0	<

Problem Author: Andrew Rumyantsev
Problem Source: Petrozavodsk summer training camp, August 2005.

1382. Game with Cards

Time limit: 0.5 second
Memory limit: 64 MB

Tyomitch plays the following game with N of his friends. Tyomitch leaves the room. His friends write numbers from 1 to N on cards, and each of the friends takes a card in a way that Tyomitch doesn't know which card each one has. Let's number the friends from 1 to N. After Tyomitch comes back to the room, each of his friends makes 2 statements of the following form (examples given for i'th friend):

1. I have the card number a_i.
2. b_i'th friend has the card number c_i ($b_i \neq i$).

Exactly one of these statements is true, and the other one is false. It's known that no two friends said that friend b has card c, and nobody said that friend b has card c if b admitted that he has this very card. The task for Tyomitch is to determine for each of his friends which of his statements is true.

Input

The first line of the input contains the number N ($2 \leq N \leq 1000$). Each of the following N lines contains a triple a_i, b_i, c_i — the statements of Tyomitch's friends.

Output

The only line of output must contain N numbers separated with spaces, being the numbers of the true statement (either 1 or 2) for each of the friends. It is known that a solution exists.

Sample

input	output
5 3 4 3 1 3 2 3 2 5 2 5 4 3 4 1	1 2 2 2 2

Problem Author: Alexander Ipatov, special thanks to Dmitry Ivankov
Problem Source: Petrozavodsk summer training camp, August 2005.

1383. Flower-garden Designs

Time limit: 2.0 second
Memory limit: 64 MB

There were N people to take part in the competition for a flower-garden design. Each of them had proposed his design, the finite sequence of points in plane which are the suggested locations for flowers. To save the main jury from needless labor of considering identical designs, the pre-jury wants to find the designs which only differ in rearrangement of points and their affine transformation that doesn't change the orientation (that is, the radius-vector of each point is multiplied by a matrix with positive determinant and translated by a fixed vector).

Input

The first line of input contains the single number N ($N \le 10000$). The N designs follow. Each design is represented as the length of the sequence M, followed by coordinates of points (M pairs of integers whose absolute value doesn't exceed 1000, each pair on a line by itself). The sum of all sequences' lengths doesn't exceed 200000.

Output

The first line of output must contain the number of different design classes. The following lines must list the classes as one-based indices of designs, terminated with zero.

Sample

input	output
7 5 1 2 0 0 6 0 0 4 2 7 5 1 2 3 9 0 1 2 3 9 2 5 -43 -37 -73 -47 -3 3 -23 -7 -3 63 3 0 0 1 0 0 1 3 0 0 1 0 3 0 3 10 3 3 7 5 2 3 6 1 6 5 6 7	4 4 6 0 5 7 0 1 3 0 2 0

Problem Author: Andrew Rumyantsev
Problem Source: Petrozavodsk summer training camp, August 2005.

1384. Goat in the Garden 4

Time limit: 1.0 second
Memory limit: 64 MB

Tyomitch takes his favorite goat for a stroll through his neighbor's cabbage garden. Tyomitch had to go away for a while, and he decided to drive a stake into the ground and to bound the goat to the stake to prevent it from eating up all the neighbor's cabbage. To save his pet from starving to death, Tyomitch wishes to select such a spot for the stake, and such a length for the rope, that the goat could deal with as large area of the garden as possible. However, there's a little problem: the goat, when left alone with the garden, tries to pierce the garden's fence with its horns. The goat will succeed if it can reach the fence with its horns and the rope has even a little slack at this moment. Tyomitch wants to avoid the traces of the uninvited guests getting noticed by the neighbors, so he tries to bound the goat in such a way that the fence would remain safe and sound. Your task is to help him. You only have to find the length for the rope, and Tyomitch will locate the stake on his own.

Input

The neighbor's garden is polygonal with N vertices, and it may be non-convex. The first line of the input contains the number N ($3 \le N \le 25$). The next N lines give the coordinates of the vertices, listed counter-clockwise. (i+1)'th line will give the coordinates x_i and y_i, being integers between 0 and 1000 inclusive. The garden is so large that you can consider the goat a point.

Output

Your output must be the single number R — the length for the rope, rounded to 2 digits after the decimal point.

Sample

input	output
3 0 0 200 0 0 200	58.58

Problem Author: Alexander Ipatov
Problem Source: Petrozavodsk summer training camp, August 2005.

1385. Interesting Number

Time limit: 1.0 second
Memory limit: 64 MB

Tyomitch calls the number with $2N$ digits (without leading zeroes) "interesting", if it's divisible by both the number formed from its first N digits and the number formed from its last N digits. For example, 1020 is "interesting" (divisible by 10 and 20) and 2005 is not. Tyomitch wants to know how many "interesting" $2N$-digit numbers exist. You are to help him.

Input

Input contains an integer N ($1 \le N \le 10000$).

Output

Output the number of "interesting" $2N$-digit numbers.

Sample

input	output
1	14

Notes

11, 12, 15, 22, 24, 33, 36, 44, 48, 55, 66, 77, 88, 99.

Problem Author: Alexander Ipatov
Problem Source: Petrozavodsk summer training camp, August 2005.

1386. Maze

Time limit: 0.5 second
Memory limit: 64 MB

One of the most popular games of all times is the "Maze". The game is played on a $N \times M$ table. The player can make the instructions: 'left', 'right', 'up', 'down'. For each cell of the table and each instruction the game-master has defined the destination cell that the player moves to; that is, the player is given the map of the maze. Once a game was interrupted, and the master has forgotten which cell the player was in. Fortunately, a full record of the gameplay has remained, which is the sequence of the instructions made by the player. You are to write a program determining the cells that the player can be currently in.

Input

The first line of the input contains two numbers N and M ($1 \le N,M \le 100$). Four blocks of N lines each follow. Each line contains M pairs, being the new coordinates of the player after making k'th instruction standing in the cell (j, i), where i is the number of pair in the line, j is the number of line in the block, and k is the number of block. Following is the number S ($1 \le S \le 4000$) of the instructions made by the player. The last line contains the S numbers of the instructions made.

Output

The first line of the output must contain the number L of the cells that player can be in after making the given sequence of instructions. Each of the next L lines must contain the coordinates of these cells, ordered first by the first coordinate, and then by the second.

Sample

input	output
2 3 1 2 1 3 1 3 2 2 2 3 2 3 2 1 2 2 2 3 2 1 2 2 2 3 1 1 1 1 1 2 2 1 2 1 2 2 1 1 1 2 1 3 1 1 1 2 1 3 4 1 2 3 4	2 1 1 1 2

Problem Author: Andrew Khalyavin
Problem Source: Petrozavodsk summer training camp, August 2005.

1387. Vasya's Dad

Time limit: 1.0 second
Memory limit: 64 MB

Vasya’s dad is good in maths. Lately his favorite objects have been "beautiful" directed graphs. Dad calls a graph "beautiful" if all the following conditions are true:

1. The graph contains exactly N vertices and $N-1$ edges.
2. Exactly one vertex has no entering edges.
3. The graph contains no directed cycles.

Dad calls two "beautiful" graphs isomorphic, if the vertices of the first graph can be renumbered in such way that it turns into the second one.

Dad picks an integer N, stocks up blank paper, and draws a "beautiful" graph on each sheet. He verifies that no two drawn graphs are isomorphic.

Given the number N, you are to find the number of sheets that Vasya's dad has to stock up.

Input

Input contains the single integer N ($1 \le N \le 50$).

Output

Output the number of "beautiful" graphs with N vertices.

Samples

input	output
3	2
5	9

Problem Author: Alexander Ipatov
Problem Source: Petrozavodsk summer training camp, August 2005.

1388. Photo

Time limit: 1.0 second
Memory limit: 64 MB

A river flows down the plain along the line $y = 0$. There are trees growing in the plain to the both sides of the river, but not on the river itself. A dam is positioned on the river in the point (0,0). It is necessary to make two matching photos of the two river-sides from the dam. The photos are considered matching if the arrangement of the trees on them match (only the horizontal arrangement is considered, not the distance to the trees).

If a tree is obscured with another one, the tree isn't present in the photo. No two trees occupy one point. Sizes of the photo-camera and the trees are negligible.

The photographing occurs in the following way: first a camera is used whose spanning angle is arbitrary close to 180 degrees (the photo-film is a line, and trees in front of it are centrally projected onto the film, center is point (0,0)); then a segment with trees from one river-side only is cut out of the film, and scaled arbitrarily.

Input

Each line of the input, except the last one, contains the coordinates of the trees (two integers from −20000 to 20000). The number of the trees doesn't exceed 10^5.

The last line contains two zeros.

Output

The output must contain a single number, being the maximum number of trees on two matching photos of the two river-sides.

Samples

input	output
-1 1 0 1 2 1 3 1 7 2 3 -2 1 -1 0 -1 0 -5 -4 -2 -3 -1 0 0	4
0 1 1 1 2 1 -3 -1 -2 -2 -1 -3 0 0	3

Problem Author: Andrew Rumyantsev
Problem Source: Petrozavodsk summer training camp, August 2005.

1389. Roadworks

Time limit: 1.0 second
Memory limit: 64 MB

Once upon a time there was a king. One day the king counted up the collected taxes and decided to spend the money for the road maintenance. There were N cities in that kingdom and M two-way roads connected them in such way that one could travel from a city to others using these roads. The road network was catastrophic without repairing, so the king made up his mind to repair as many roads as possible during the summer, before the money depreciated. The inhabitants of the kingdom were shocked to know that all the ways they used to go would be blocked for summer. So the king promised that at most one road from a city would be blocked. Help the king to fulfil his plan without displeasing the citizens.

Input

The first line of input contains two natural numbers N and M ($2 \le N \le 10^5$, $M = N - 1$), separated with a space. Each of the next M lines describes a road in the form (a_i, b_i), where a_i and b_i are numbers of the cities connected with i'th road ($1 \le a_i, b_i \le N$).

Output

The first line of output should contain the only integer K being the maximum number of roads that the king can close for maintenance without raising disorders in his kingdom. The next K lines should describe these roads in the same form as they were given in the input.

Sample

input	output
4 3 1 2 2 3 3 4	2 1 2 3 4

Problem Author: Dmitry Ivankov & Alexander Ipatov
Problem Source: Petrozavodsk summer training camp, August 2005.

1390. Shots at Walls

Time limit: 3.0 second
Memory limit: 64 MB

A new pistol is being tested. The pistol can fire shots with variant bullet speeds. In some points of time shots are fired from the point of origin with certain horizontal speeds, and in some other points of time walls are built on a horizontal platform. The walls are non-singular segments lying on lines that do not go through the point of origin. The walls may intersect. For processing of the test results, you are to determine the time that each shot bullet had been flying for. You can assume that the speed of the bullet after shot is constant.

Input

Each line of the input begins with either "shot", "wall", or "end" (without quotes). The number of lines doesn't exceed 50000. After "shot", the two coordinates of speed of the bullet are listed; the speed cannot be zero. After "wall", the four numbers follow, being the coordinates of wall's beginning and end. "end" denotes the end of the input. All the coordinates are integers whose absolute values doesn't exceed 10000. All the events are listed in chronological order, and time intervals between the events exceed the time needed to build a wall, or the time needed for bullet to reach the next wall or end of the proving ground.

Output

For each of the shots, you must output the single number, on a line by itself: the time that the bullet had been flying for, with precision of 10^{-6}. If the bullet doesn't hit any wall, you must output "Infinite" instead of a number.

Sample

input	output
shot 1 0 wall 1 0 0 1 shot 1 1 shot -1 3 wall 1 0 -1 2 shot -1 3 wall 1 1 -1 1 shot -1 3 wall 2 3 2 -3 wall 3 -2 -3 -2 shot 1 -1 shot 40 -39 shot 9999 -10000 shot -1 -1 shot -3000 -2000 shot -3001 -2000 shot -3000 -2001 shot 1 0 shot 1 1 wall -1 2 10 -10 shot -1 1 shot 0 1 shot 1 1 shot 1 0 shot 1 -1 wall 0 -10000 -10000 0 shot -2 -1 end	Infinite 0.50000000000000000000 Infinite 0.50000000000000000000 0.33333333333333333333 2.00000000000000000000 0.05000000000000000000 0.00020000000000000000 2.00000000000000000000 0.00100000000000000000 Infinite 0.00099950024987506247 1.00000000000000000000 0.50000000000000000000 1.00000000000000000000 0.90909090909090909091 0.43478260869565217391 0.83333333333333333333 2.00000000000000000000 3333.33333333333333333

Problem Author: Andrew Rumyantsev
Problem Source: Petrozavodsk summer training camp, August 2005.

1391. Snake

Time limit: 3.0 second
Memory limit: 64 MB

Everyone knows that snakes have hard time living in mazes. Even if a snake lives alone, it can perish by running into a wall or its own tail. A certain participant of snakes competition called Vasya decided to teach his snake to get out from distant areas of the maze. Such sub-mazes are dangerous because the snake has little chance to get out from them alive, and of course the longer the snake the less chance it has. Vasya trains his snake in the following way: when it's young and its length is 2, he lets it into a practice dangerous maze. The snake's goal is to get out from the maze as soon as possible. If the snake survives, then the training will be repeated as soon as the snake reaches the length of 3. The training goes so on until the snake either perishes or matures at the length of 18.

The maze is a rectangle with width W and length H, each cell of which is either obstructed 'X' or free '.'. The maze is surrounded with impassable stones '*' with the exception of the only entrance '#' located in the border of the maze. Here's a simple 4-by-3 maze for your reference:

```
***#**
*.X.X*
*.X..*
*....*
******
```

The snake of length L is a sequence of L cells. Any two consecutive cells have a common side. All the cells in the sequence are different. The snake can creep in 3 ways relative to its current direction: forward, to the left or to the right. All the cells of snake's body move at once, each moving into the place of preceding one, except for the head cell. Here are the examples of snake's movement:

- `321. -> .321`
- `321 -> .32`
  `...    ..1`
- `12 -> 23`
  `.3    1.`
- `12 -> 23`
  `43    14`

The snake creeps through exactly one cell per unit of time, or perishes if it has nowhere to creep into.

Input

The first line of the input contains H and W specifying the size of the maze, where $1 \le H \le 300$ and $1 \le W \le 30$. The second line contains h_0 and w_0 being the coordinates of the entrance cell; h_0 equals either 1 or H, or w_0 equals either 1 or W. Following are H lines of W characters each, specifying the maze outline ('X' for obstruction and '.' for free cell). Time is counted starting from 0; initially the snake has its head at (h_0, w_0) and all other body cells outside the maze. Time is counted until snake's head is again at (h_0, w_0). Even though the maze is a practice one, no snake of length 18 can get out from it alive.

Output

The output must contain 16 lines, where i'th line is either the best time needed for a snake of length i+1 to get out from the maze, or −1 if it can't get out alive.

Sample

input	output
9 9 1 5 XXXX.XXXX XXX...XXX XX..X..XX ....XX..X X.X.X.X.X ..XX..... X...XXX.. XXXXX.... X.....XXX	10 10 10 22 22 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

Problem Author: Dmitry Ivankov
Problem Source: Petrozavodsk summer training camp, August 2005.

1392. Dreaming of Stars

Time limit: 0.5 second
Memory limit: 64 MB

For the whole last week Ilya Petrov spent his time sitting at the computer and playing his favourite game "Perimeter". But today he suddenly realized that he does not want to look at the monitor. "Hm… It is so strange… What will I do now?" thought Ilya. "Oh! Astronomy! — most wonderful of all sciences. This is what I need!" — he exclaimed. So he did not lose time, he came to a store and bought a big black telescope. This night he spent admiring the beauty of distant galaxies, unreachable (so far) for a human…

The magic of the moment was broken suddenly as Ilya noticed two planets, which have stuck each other. "What is it?" — exclaimed Ilya. He was really amazed. But he thought a little, took a paper and wrote down the following.

"Two planets have stuck each other if they have at least two common points. In this case we will say that each of the planets is reachable from another one. Then an archipelago is a group of planets for which the following conditions are true:

- Each of planets of archipelago is reachable from any other planet of the archipelago (possibly in a several steps).
- There are no planets of the archipelago reachable from any planet outside of archipelago.

Clearly, any set of planets can be uniquely split into archipelagos. You have to find the number of archipelagos for a given set of planets and from which planets each archipelago consists of. The solution of this task will make the humans more powerful and flawless race. So this task is to be solved immediately."

Input

The first line of input contains a number K ($0 < K \le 1000$) — the number of the planets in a set. Then K lines with planet center coordinates X, Y, Z and planet radius R follow. All numbers are integer and they do not exceed 1000 by absolute value. Absence of coinciding planets is guaranteed. All the planets are balls.

Output

Output must contain exactly P lines. Here P is the number of archipelagos in a given set of planets. The i-th line of output should contain the list of planets constituting the i-th archipelago. Each planet in a list is present by its number (planets are numbered by integers starting from 0). Planet numbers in a list must be printed in increasing order with a comma and a space as a separator. Do not print a comma after last number in a list.

Archipelagos must be printed in the following order. If the least number of the planet of one archipelago is less than the least number of the planet of another, then the first archipelago must precede the second in output.

Samples

input	output
2 1 1 1 1 1 3 1 1	0 1
3 1 1 1 1 1 3 1 1 1 4 1 1	0 1, 2
3 1000 1000 1000 1 999 1000 1000 1 1 1 1 1	0, 1 2

Problem Author: Fedor Fominykh
Problem Source: The 1st Collegiate Programming Contest of the High School Pupils of the Novouralsk Town (April, 2005)

1393. Average Common Prefix

Time limit: 1.5 second
Memory limit: 16 MB

Let **T** denote some string of length **n** consisting of capital Latin letters. Let **Shift(T, k)** denote the left cyclic shift of **T** by **k-1** positions. The permutation array for **T** is an array **P[1..n]** such that **Shift(T, P[1])**, **Shift(T, P[2])**, ..., **Shift(T, P[n])** is a list of cyclic shifts of **T** sorted in lexicographical order.

For given two strings **v** and **w** we define **LCP(v, w)** as the length of their longest common prefix. The **Average LCP** of the string **T** is the average length of longest common prefix between two consecutive shifts:

Example. **T** = 'MISSISSIPPI', **n** = 11:

i	P[i]	Shift(T, P[i])	LCP
1	11	'IMISSISSIPP'	1
2	8	'IPPIMISSISS'	1
3	5	'ISSIPPIMISS'	4
4	2	'ISSISSIPPIM'	0
5	1	'MISSISSIPPI'	0
6	10	'PIMISSISSIP'	1
7	9	'PPIMISSISSI'	0
8	7	'SIPPIMISSIS'	2
9	4	'SISSIPPIMIS'	1
10	6	'SSIPPIMISSI'	3
11	3	'SSISSIPPIMI'	

Average LCP of 'MISSISSIPPI' is 1.3

Input

The first line of the input contains integer **n** (1 < **n** < 250001). The second line contains string **T**.

Output

The only line of the output should contain the **Average LCP** of **T** with at least 3 digits after the decimal point.

Sample

input	output
11 MISSISSIPPI	1.300

Problem Author: Ilya Grebnov

1394. Ships. Version 2

Time limit: 1.0 second
Memory limit: 64 MB

The military intelligence of one country found out that N ($N < 100$) battle ships of neighboring enemy country are situated in M rows ($1 < M < 10$). The intelligence knows the lengths $l_1, l_2, \ldots, l_N$ of the battle ships which are whole numbers in the interval [1, 100], and wants to know in which rows the ships are situated. The only thing that is known about the M rows are their lengths — $L_1, L_2, \ldots, L_M$. Assume that the ships touch their neighbours in the rows and that every row contains at least one ship. Write program that will find one possible ordering of the ships in rows.

Input

The first line of the input contains N and M. The next N lines contain the lengths of the ships. The next M lines contain the lengths of the rows.

Output

The output should contain M pairs of lines. The first line of each pair should contain the amount of the ships in the current row, the following line should contain the lengths of the ships from the current row. The order of the M row descriptions should be the same as the order in which the rows are given in the input.

Sample

input	output
5 2 4 10 2 5 3 11 13	3 5 4 2 2 10 3

Notes

This problem is the same as 1115 "Ships" but with harder tests.

Problem Author: Prepared by Vladimir Yakovlev

1395. Pascal vs. C++. Version 2

Time limit: 1.0 second
Memory limit: 4 MB
Language limit: C, C++, Pascal

Background

This problem is a hardcore version of the problem "Pascal vs. C++". We, Timus Top Coders, dedicate it to those, who still believe in the power of human intellect. In the fact, that there is no limit to perfection. In the fact, that all the languages are equal. In the freedom of choice. In the unlimited programming! We are happy we could create this problem. And we hope you will be proud after you solve it. Enjoy!

Problem

A sequence S consists of N elements S[i] indexed from 1 to N. You should take a maximal number of different elements from this sequence which are successive terms of some increasing arithmetical progression. The order of these elements in the sequence S does not matter. And may Pascal, C++ and Java be with you ;)

Input

The first line contains the integer number N ($2 \le N \le 10000$). The second line contains N integers S[i] ($1 \le S[i] \le 10^9$).

Output

The first line should contain the maximal number of taken elements. The second line should contain the indexes of these elements in the sequence S. The indexes may be listed in any order and should be separated by single spaces. If the problem has several solutions, you should output any of them.

Sample

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

Notes

We recommend you to solve this problem on C++, because in this specific case local C++ compiler produces more effective binaries than Pascal and Java compilers.

Problem Author: Ilya Grebnov, Dmitry Kovalioff, Nikita Rybak
Problem Source: Timus Top Coders: Second Challenge

1396. Maximum. Version 2

Time limit: 1.0 second
Memory limit: 64 MB

Consider the sequence of numbers a_i, $i = 0, 1, 2, \ldots$, which satisfies the following requirements:

- $a_0 = 0$
- $a_1 = 1$
- $a_{2i} = a_i$
- $a_{2i+1} = a_i + a_{i+1}$

for every $i = 1, 2, 3, \ldots$.

Write a program which for a given value of n finds the largest number among the numbers $a_0, a_1, \ldots, a_n$.

Input

You are given several test cases (not more than 10 000). Each test case is a line containing an integer n ($1 \le n < 10^{18}$). The last line of input contains 0.

Output

For every n in the input write the corresponding maximum value found.

Sample

input	output
5 10 0	3 4

Notes

This problem is the same as 1079 "Maximum" but with bigger limitations.

Problem Author: Prepared by Vladimir Yakovlev

1397. Points Game

Time limit: 1.0 second
Memory limit: 64 MB

Two students are playing the following game. There are $2 \cdot n$ points on the plane, given with their coordinates (x_i, y_i). Each move player paints the point with his own color (first with white, second with black). The first student makes odd moves, second student makes even moves. When all points are painted (each student made n moves), the game finishes. Each student gets amount of points (real number) that equals to the sum of all distances among pairs of points, colored with his color. Student who get more points becomes a winner. The students play optimally. Find and print the difference between points amount of winner and looser.

Input

Contains multiple test cases. The first line of each case contains positive integer number n ($n \le 500$). Next $2 \cdot n$ lines contain points' coordinates (x_1, y_1), (x_2, y_2), …, (x_{2n}, y_{2n}).

Output

For each test case output the difference between the points of winner and looser. Output the difference with three digits after decimal point.

Sample

input	output
2 0 0 0 1 1 0 1 1 2 0 0 1 0 0 3 1 5	0.000 1.937

Problem Author: Michael Medvedev
Problem Source: Open All-Ukrainian Collegiate Programming Contest 2006

1398. Bishop and Pawn

Time limit: 1.0 second
Memory limit: 64 MB

There are a white bishop and black pawn on a chessboard. Moves are made in accordance with the usual chess rules. White moves first. Black wins if he can promote his pawn to a queen and the white bishop cannot capture the queen by the subsequent move. The game ends in a draw if it's Black's turn to move but the pawn cannot move forward. In other cases, White wins. It is required to tell the result of the game if both sides play optimally.

Input

The first and second lines show the positions of the white bishop and black pawn, respectively, by means of the standard chess notation. The rank in which the pawn is initially positioned may have the number from 2 to 7, and the bishop is initially positioned at any square different from the pawn's square.

Output

Output WHITE if White wins, DRAW in the case of a draw, and BLACK if Black wins.

Sample

input	output
a1 c2	WHITE

Problem Author: Magaz Asanov
Problem Source: Practice tour of Urals Championship 2005

1399. Economical Director

Time limit: 2.0 second
Memory limit: 64 MB

The director of the household appliance chain “Rodonit” was calculating the month’s losses. The time was over when buyers were taking away the purchased goods from a store immediately after buying them. Now they demanded a free delivery and wanted to be telephoned in advance in order to agree upon the time of delivery.

There are ten “Rodonit” stores in the city. The director understands that it is not always rational to deliver items directly from the stores where they were bought. On the outskirts of the city, there is a warehouse of household appliances, and next to the warehouse there is a garage. Every night employees compile a list of goods that have been bought during the day, with information about the buyers: names, addresses, and telephone numbers. These goods are delivered during the next day. The company has only one lorry for delivering goods. The lorry has a limited carrying capacity, therefore several trips are sometimes necessary to deliver the goods. The employees arrange the delivery schedule: which items should the lorry driver take from the warehouse for each trip and the order of visits to the buyers.

In order to save the fuel, the total length of the trips must be minimal. The employees use a city map to measure the distances between objects (which are the warehouse and the houses of the buyers from the list) and start the complicated optimization process. The following assumptions are made:

1. The distance between the warehouse and the garage is 0.
2. Let $D(i, j)$ be the distance between objects i and j. Then for any objects i, j, k
    a. $D(i, i) = 0$.
    b. $D(i, j) = D(j, i)$.
    c. $D(i, k) \le D(i, j) + D(j, k)$.
3. In the end of each trip the lorry must return to the garage.
4. The sum of the masses of the goods carried by the lorry at one time must not exceed its carrying capacity.

The director must pay extra wages to the employees for their night work. In order to economize, he decided to employ a programmer who would write a program producing the delivery schedule.

Input

In the first line there are the number of buyers $M \le 20$, the number of items $N \le 50$, and the carrying capacity of the lorry $L_{max} \le 3000$. In the following lines there is a matrix D of size $(M+1)\times(M+1)$ containing distances between the objects (the warehouse is assigned the number 0 and the buyers are assigned numbers from 1 to M). Each of the following N lines describes the corresponding item: its mass and the number of the buyer (an integer from 1 to M). All the masses and distances are integers in the range from 1 to 100.

Output

In the first line output the number of trips that should be made. Then describe each trip giving the following information. The first line must contain the numbers of the items delivered during this trip, given in an arbitrary order and separated with a space (these numbers are in the range from 1 to N). The second line must contain the maximal load of the lorry during this trip (which is the total mass of the items being delivered). In the third line, output the number of objects in the order of traveling separated with a space. In the fourth line, output the total length of the trip. In the last line output the total length of all the trips. The blocks describing the trips must be separated from each other and from the first and last lines by an empty line.

Sample

input	output

7 10 5 0 2 3 4 5 6 5 4 2 0 4 5 6 7 6 5 3 4 0 3 4 5 4 1 4 5 3 0 3 4 1 2 5 6 4 3 0 1 2 3 6 7 5 4 1 0 3 4 5 6 4 1 2 3 0 3 4 5 1 2 3 4 3 0 3 1 5 2 1 3 1 4 2 5 1 6 2 7 1 5 2 2 1 1	4 1 10 4 0 1 0 4 4 5 6 8 5 0 4 5 6 0 14 2 5 0 2 0 6 3 7 9 5 0 3 7 2 0 10 34

Notes

Your program will pass a test if it produces an answer that is no worse than the answer produced by the jury's program for the same data.

Problem Author: Alexander Ipatov