Taxes

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

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 $N_2 <= K <= N_3$ 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

```
15
12000000 12
24000000 20
36000000 25
48000000 30
0        35
12000000
12000000
-1
```

Sample Output

```
937233.19
```

Cube in Labyrinth

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

There is a cube on the rectangle X × Y board. The cube side with the side equal to the side of a cell on 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 determine 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.

All the numbers are positive integers; 2<= X,Y <= 10.

Input

The first input line contains two numbers X and Y separated with one or several spaces. Analogously, the second line consists of the numbers A and B, the third line – of the numbers C and D. Then there may be an information about the walls.

After a symbol ‘v’, situated 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 (analogously to the previous paragraph) describing the horizontal walls. 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

```
10 2
1 1
10 1
v
2 1
6 2
h
```

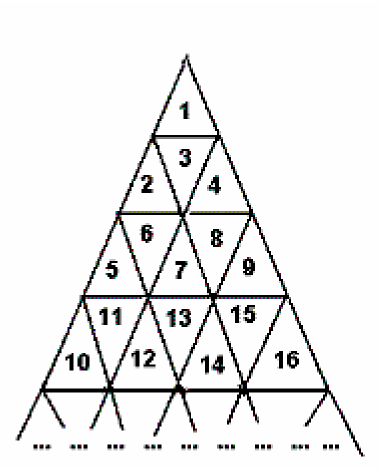
```
4 1
```

Sample Output

```
11
```

Delta-wave

Time Limit: 1.0 second
Memory Limit: 1 000 КБ



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 N and M.

Input

Input contains two integer numbers M and N in the range from 1 to 1,000,000,000 separated with space(s).

Output

Output should contain the length of the shortest route.

Sample Input

```
6 12
```

Sample Output

```
3
```

Minimal coverage

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

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 number M (1 <= M <= 5000). Subsequent lines of input contain pairs of numbers L_i and R_i (abs(L_i), abs(R_i) <= 50,000). Each pair is placed on separate line. Numbers in the pair are separated with space(s). List of pairs is ended with pair of integers "0 0". i <= 100,000

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 "0 0" 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.

Sample Input

```
2 tests

1
-1 0
-5 -3
2 5
0 0

1
-1 0
0 1
0 0
```

Sample Output

```
0

1
0 1
```

Parallelepiped

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

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 <= i <= n <= 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.

The number n is written without a decimal point. All other numbers are written with not more than two digits after a decimal point (if a number is integer a decimal point may be omitted). All the input numbers are integer not greater than 1000.

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

```
1.0 1.0 1.0
1
0.5 0.5 0.5
```

Sample Output

```
0.50
```

Convex hull

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

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) 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 vertexes. Vertexes should be enumerated 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 in the file only once.

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

Sample Input

```
4
3 3
3 1
2 2
4 2
```

Sample Output

```
3 1
4 2
3 3
2 2
```

Sequence Median

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

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 250,000. Each element of the sequence is a positive integer not greater than 2^32 - 1 inclusive.

Output

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

Sample Input

```
4
3
6
4
5
```

Sample Output

```
4.5
```

Archiver

Time Limit: 2.0 second
Memory Limit: 2 000 КБ

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 file 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” (without inverted commas).
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.
Remark. The sample input is just an illustration, it does not satisfy the requirements since it is too short and not a literary text.

Output

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

Sample Input

```
1234567891011121314151617181920212223242526272829303132333435363738394041424344454647484950515253545556575
```

Sample Output

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

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

Deans pyramid

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

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 their lives 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 considered 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

```
2.3660254037 1.3660254038
```

Sample Output

```
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

Dispute

Time Limit: 0.5 second
Memory Limit: 1 000 КБ

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 first line of the input file contains an integer n (0 <= n <= 100,000,000).

Output

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

Sample Input

```
50
```

Sample Output

```
6300
```

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

ACM Diagnostics

Time Limit: 0.5 second
Memory Limit: 1 500 КБ

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 choosing in such way, that the sum of the values of all registers is divided 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, satisfied defined above demands. 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 is really 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 <= L <= 100, 2 <= M <= 50, 1 <= K <= 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

```
3 10 4
213
```

Sample Output

```
9 6 1
```

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

Bricks

Time Limit: 0.5 second
Memory Limit: 1 000 КБ

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 of the input file contains an integer H, which is the number of brick rows, i.e., the height of the wall (0 <= H <= 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 <= K <= 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 $0 <= L_i, R_i <=$ 10000. The number of all bricks does not exceed 1000000.

Output

You should write a program that outputs “Yes” if all the bricks are stable and “No” otherwise.

Sample Input

```
3
1
10 20
3
1 7
13 17
100 200
2
0 20
60 160
```

Sample Output

```
Yes
```

Problem Author: Idea - Alexey Lakhtin, prepared by Alexey Lakhtin
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

Tray

Time Limit: 0.5 second
Memory Limit: 1 000 КБ

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 1000000. 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 center 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 file.

Sample Input

```
Sample input #1
800 400 200 200 50

Sample input #2
800 400 200 200 51
```

Sample Output

```
Sample output #1
200.0000 200.0000 600.0000 200.0000 400.0000 350.0000

Sample output #2
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

Some words about sport

Time Limit: 0.5 second
Memory Limit: 1 000 КБ

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 <= N <= 100) – the number of pixels on the side of new square monitor. It is followed by N lines, each containing N 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

```
4
1  3  6  10
2  5  9  13
4  8  12 15
7  11 14 16
```

Sample Output

```
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 Aleksandr Klepinin
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

Chase in Subway

Time Limit: 0.5 second
Memory Limit: 1 000 КБ

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 the input file contains the number N of lines in the subway (1 <= N <= 50). Each of the next N lines contains a description of the corresponding line. A description starts with an integer K (2 <= K <= 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 the input file contains surveillance data: an integer M >= 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

```
3
2 61 62
5 75 20 85 50 61
3 10 20 30
3 30 20 85
```

Sample Output

```
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

MDPAR &amp; MIIAR

Time Limit: 1.0 second
Memory Limit: 10 000 КБ

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: "X" denotes a wall, i.e., a cubicle inaccessible both for the speleologist and for water, and a space " " 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 <= W,H <= 1000, 1 <= X <= W, 1 <= Y <= H, D <= 1000.

Output

You should output the line "Can be rescued by himself" (without quotation marks) if the speleologist can reach the surface following the instructions issued by the MIIAR. Otherwise, you should output the line "Rescue operation required".

Sample Input

```
Sample input #1
9 4 8 2 5
##..##..#
##.#.####
##......#
#########

Sample input #2
9 4 8 2 6
##..##..#
##.#.####
##......#
#########
```

Sample Output

```
Sample output #1
Rescue operation required

Sample output #2
Can be rescued by himself
```

Problem Author: Idea - Aleksandr Klepinin, prepared by Aleksandr Klepinin, Ivan Dashkevich
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

Electronic Auction

Time Limit: 0.5 second
Memory Limit: 5 000 КБ

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 (the price is between 0.01 and 10000.00 bibriks and always has exactly 2 digits after the decimal point). 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 file 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.

The number of operations does not exceed 100000. All operations are correct. The last line contains the command ‘QUIT’.

Output

You are to write a program that outputs the profit of the auction in the current month (with exactly 2 digits after the decimal point).

Sample Input

```
BID 0.01
BID 10000
BID 5000
BID 5000
SALE 7000 3
DEL 5000
SALE 3000 3
SALE 0.01 3
QUIT
```

Sample Output

```
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

Hail

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

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 of the input contains two integers: N (3 <= N <= 10), which is the number of polygon vertices, and H (1.00 <= H <= 100.00). 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. 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

```
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
```

Sample Output

```
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

Logarithm

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

Given a set A of N unordered 128-bit numbers you are to compute a value of the function

$$\sum_{k,j=0}^{N-1} \log_{10}(A_k XOR\ A_j),$$

where A_i — the i^{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 <= 5000. In the following N lines there are128-bit numbers Ak 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

```
2
0 0 0 2324
0 2332 0 0
```

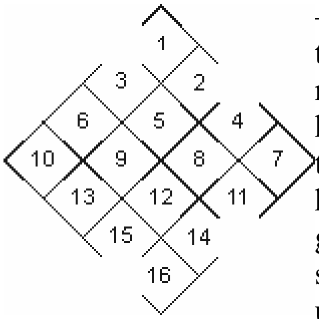
Sample Output

```
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

The Hotel

Time Limit: 1.0 second
Memory Limit: 1 000 КБ



— 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 number N (1 <= N <= 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

```
3
```

Sample Output

```
4 2 1
7 5 3
9 8 6
```

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

Graph decomposition

Time Limit: 0.5 second
Memory Limit: 1 000 КБ

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 sequence of the numbers pairs. Each pair denotes vertices identifiers of one edge. All the identifiers are integers from 1 to 1000. You may assume that the input is correct, i.e 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.

Sample Input

```
Sample input #1
1 2
2 3
3 1
1 10

Sample input #2
1 2
2 3
3 1
4 10
```

Sample Output

```
Sample output #1
1

Sample output #2
0
```

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

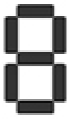
Floor indicator

Time Limit: 0.5 second
Memory Limit: 1 000 КБ

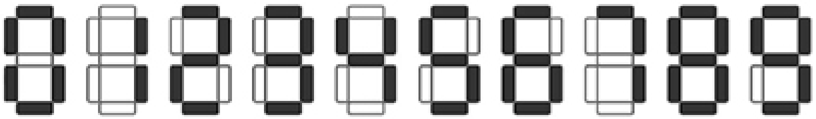
"Let's go!", thought a hotel's manager entering an elevator. He pressed the tenth floor button and meditated. The day was not easy. The manager looked at the floor indicator, saw the number 9, and prepared to get out. But the elevator did not stop. The nine gave place to eight, then to seven. The manager 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 natural numbers starting with 1.

Input

The input 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

```
10
```

Sample Output

```
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

Spy

Time Limit: 0.25 second
Memory Limit: 10 000 КБ

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 S1 = $S_1S_2...S_n$. The machine constructs all cyclic permutations of this line, i.e., S2=$S_2S_3...S_nS_1$, ..., SN=$S_nS_1S_2...S_{n-1}$. Then the set S1, S2,…,SN 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 S1=abracadabra, then the following array is formed:

1. aabracadabr = S11
2. abraabracad = S8
3. abracadabra = S1
4. acadabraabr = S4
5. adabraabrac = S6
6. braabracada = S9
7. bracadabraa = S2
8. cadabraabra = S5
9. dabraabraca = S7
10. raabracadab = S10
11. racadabraab = S3

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 of the input 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 of the output file should contain the initial message.

Sample Input

```
3
rdarcaaaabb
```

Sample Output

```
abracadabra
```

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

Classmates

Time Limit: 4.0 second
Memory Limit: 1 000 КБ

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 <= N <= 10). The second line contains the integer number M (0 <= M <= 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 pupils 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

```
6
7
Tanya Lena
Tanya Katya
Tanya Masha
Lena Natasha
Lena Vitya
Natasha Vitya
Masha Vitya
Tanya
```

Sample Output

```
3
1
Tanya Lena
2
Tanya Masha
Lena Vitya
2
Vitya Natasha
Tanya Katya
```

Problem Author: Idea: Evgeni Kobzev, prepared by Pavel Atnashev, Aleksandr Mironenko
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

Extra spaces

Time Limit: 0.5 second
Memory Limit: 1 000 КБ

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 sequences of up to K spaces (K >= 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 numbers is the length of rows which are replaced by one space during the correspondent round.

Sample Input

```
22
```

Sample Output

```
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

Dirt

Time Limit: 1.0 second
Memory Limit: 5 000 КБ

— 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 in different squares.

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

```
3 7
1 1
3 7
1200121
1212020
1112021
```

Sample Output

```
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

Bottle taps

Time Limit: 3.0 second
Memory Limit: 3 000 КБ

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 <= N <= 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 <= M <= 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

the minimal sum of money that Petrov should spend on obtaining the necessary taps.

Sample Input

```
4
10
11
12
13
3
17 2 1 3
```

```
25 3 2 3 4
15 2 3 4
3 1 3 4
```

Sample Output

```
25
```

Problem Author: Idea: Evgeni Kobzev, prepared by Pavel Atnashev, Aleksandr Mironenko
Problem Source: VIII Collegiate Students Urals Programming Contest. Yekaterinburg, March 11-16, 2004

Fuses

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

"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 of the input file contains one number - A. The second line contains B. A and B are both integers; 1 <= A <= B <= 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.

Sample Input

```
Sample input #1
```

```
1
5

Sample input #2
100
200
```

Sample Output

```
Sample output #1
3

Sample output #2
50
```

Problem Author: Den Raskovalov
Problem Source: The 10th Collegiate Programming Contest of the High School Pupils of the Sverdlovsk Region (October 16, 2004)

Fireball

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

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.

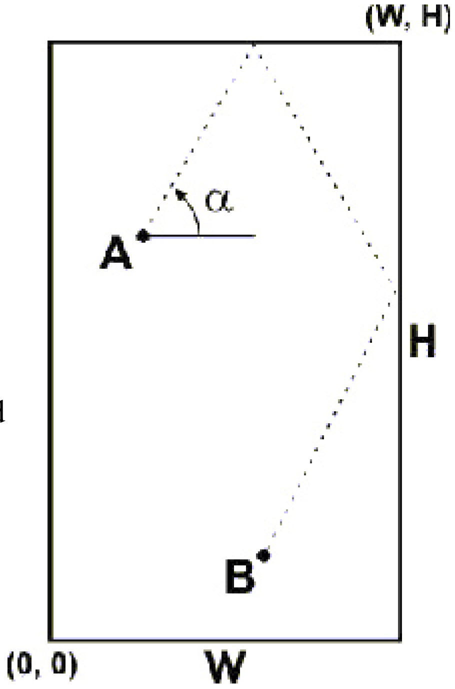
However this fact had not made any harm to its ability of research work. For example right now 3rd-level wizard Vitka Korneev tests a new battle-spell of fireball in his lab in SCITS. Oh… fireball is such a ball of fire that is used for… m-m-m… for neutralization of undesirable consequences.

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 100 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

```
1000 10
3
101 5 128 8
```

Sample Output

```
45.00
```

Hint

Characters and background are taken from books "Monday Begins on Saturday" (Arkady and Boris Strugatsky) and trilogy "Night Watch", "Day Watch" and "Twilight 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)

The galactic story

Time Limit: 1.0 second
Memory Limit: 3 000 КБ

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

The input file contains full history tree description in the following format. In the first line there is a number N (N <= 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 <= 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

```
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
```

Sample Output

```
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)

Intervals

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

— 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 <= N <= 10000). The reading of ith hyperfield intensity sensor is integer k_i (-10000 <= k_i <= 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 <= j)?: The number of queries Q is expected to be rather large (0 <= Q <= 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

```
5
1
2
3
-1
4
3
1 5
4 4
1 4
```

Sample Output

```
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)

Vladislava

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

— 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 <= N, M <= 5000) — amounts of artifacts and landing grounds respectively. Then there is positive real R <= 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, $|W_i| <= 90$, $|L_i| <= 180$. The next N lines contain artifacts coordinates — latitude w_i and longitude l_i — real numbers, $|w_i| <= 90$, $|l_i|$ <= 180.

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.

Sample Input

```
Sample input #1
```

```
1 2
1
0 0
1 1
0 0

Sample input #2
2 1
1
0 0
0 90
0 45
```

Sample Output

```
Sample output #1
0.00

Sample output #2
1.57
0.79
```

Problem Author: Aleksandr Bikbaev
Problem Source: The 10th Collegiate Programming Contest of the High School Pupils of the Sverdlovsk Region (October 16, 2004)

Genie bomber

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

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 <= N <= 100). The following N lines contain the coordinates x_i, y_i of city centers (x_i, y_i are integers, $|x_i|$, $|y_i|$ <= 10000). City centers don't coincide with each other.

The last line of the input file contains two numbers – radius R of the genie's area of destruction and the city radius r (R, r <= 10000).

Output

Output should only contain one number – the maximum number of cities that can be destroyed by one bottled genie.

Sample Input

```
Sample input #1
3
0 0
0 4
4 0
3 1
```

```
Sample input #2
5
0 0
0 1
0 2
0 3
0 4
1 1
```

Sample Output

```
Sample output #1
2

Sample output #2
1
```

Problem Author: Aleksandr Bikbaev
Problem Source: The 10th Collegiate Programming Contest of the High School Pupils of the Sverdlovsk Region (October 16, 2004)

Genie bomber 2

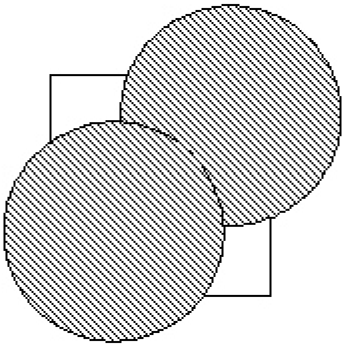
Time Limit: 1.0 second
Memory Limit: 1 000 КБ

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 <= x <= 1, 0 <= y <= 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 <= 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 <= x_i, y_i, r_i <= 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 <= x <= 1, 0 <= y <= 1). Precision of the result is required to be 1% or better.

Sample Input

```
Sample input #1
1
0.5 0.5 0.5

Sample input #2
2
0.2 0.2 0.5
0.8 0.8 0.5
```

Sample Output

```
Sample output #1
78.539816
```

```
Sample output #2
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)

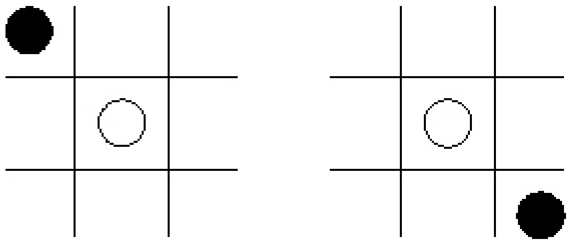
Checkers

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

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

In the output file there must be the only number – the number of move on which a set was lost, or the word "Draw" (of course, without quotation marks), if a set was not lost by either demon.

Sample Input

```
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
```

Sample Output

```
5
```

Problem Author: Den Raskovalov
Problem Source: The 10th Collegiate Programming Contest of the High School Pupils of the Sverdlovsk Region (October 16, 2004)

White Thesis

Time Limit: 0.5 second
Memory Limit: 1 000 КБ

"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 natural number triplet (A, B, C) satisfying the following property: $A^2 + B^2$ is divisile by C. The hypothesis also states that all three White Thesis components lie between the squares of two consecutive natural numbers N and N+1.

Input

contains one integer N (2 <= N <= 30000).

Output

The output file should contain three different integers A, B and C, which satisfy these two conditions:
$(A^2 + B^2)$ is a multiple of C;
N^2 <= A, B, C <= $(N+1)^2$
If two or more such triplets exist, your program should output any one of them. If there are no such triplets, then the program should output "No solution" (without quotes).

Sample Input

```
Sample input #1
2
Sample input #2
1000
```

Sample Output

```
Sample output #1
8 6 4

Sample output #2
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)

Problem of Ben Betsalel

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

"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 natural number in the form of ratio of square and cube of some natural numbers. But why does this problem always have a solution?... Ok, you will see :)

Input

In the input stream there is only one natural number n (1 <= n <= 10^9).

Output

In the first line of the output stream there must be number m. In the second – number k. m^2 should be divisible by k^3, and m^2/k^3=n, 1 <= m, k <= 10^{100}.

Sample Input

```
Sample input #1
18

Sample input #2
1
```

Sample Output

```
Sample output #1
12
2

Sample output #2
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)

Bureaucracy

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

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 <= N <= 100) and amount of days in a week according to calendar of Fools Country L (1 <= L <= 100). The next line contains the numbers of bureaucrat's visiting days A_i (1 <= A_i <= 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 ith 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 <= K <= 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 during finite time period then your program should output a single line "No Solution" (without quotation marks, of course). 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

```
2 7
1 2
0
1 0
1
1 0
2 0
```

Sample Output

```
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)

Automobiles

Time Limit: 3.0 second
Memory Limit: 16 000 КБ

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

```
10 5
..####....
..o..o....
..####.#o.
......##..
.o#####...
1
4
```

Sample Output

```
Experiment #1: North: 0, South: 1, East: 0, West: 0
```

Problem Author: Aleksandr Klepinin
Problem Source: USU Championship 2004

Babies

Time Limit: 1.0 second
Memory Limit: 16 000 КБ

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 ($1 \le k \le 1\ 000\ 000$). The second line consists of the numbers of girls that are liked by boys: if the ith boy likes some girls, her number is at the ith position; if the ith boy likes nobody, there is 0 at the ith 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 ith element of the sequence is the number of a girl that is a pair-mate of the ith boy. The numbers are separated with a space.

Sample Input

```
3
3 0 0
0 2 0
```

Sample Output

```
3 2 1
```

Problem Author: Magaz Asanov
Problem Source: USU Championship 2004

Cucaracha

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

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

```
1 2 90 10
21 2
```

Sample Output

```
31.4159
```

Problem Author: Pavel Egorov
Problem Source: USU Championship 2004

Device

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

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 ≤ 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

```
56.846841
53.36673
1124.427
```

Sample Output

```
56.847
60.631
```

Problem Author: Stanislav Vasilyev
Problem Source: USU Championship 2004

Enterprise

Time Limit: 5.0 second
Memory Limit: 1 000 КБ

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 minimal work load of the workshops.

Input

The first line contains two integers N and M ($1 \leq N, M \leq 1000$) – 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 ($1 \leq K_i \leq 100$, $0 \leq P_i, Q_i \leq 1000$) — 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 i-th broom's production is the linear with respect to i 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.

Sample Input

```
Sample input #1
2 10
6 20 15
100 100 100

Sample input #2
```

```
2 10
5 30 14
1 20 20
```

Sample Output

```
Sample output #1
Minimum possible cost: 505.00

Sample output #2
Maximum possible amount: 6
Minimum possible cost: 130.00
```

Problem Author: Magaz Asanov and Pavel Egorov
Problem Source: USU Championship 2004

Fairy tale

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

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

```
5
64631
```

Sample Output

```
646310554187
```

Problem Author: Pavel Atnashev
Problem Source: USU Championship 2004

Gigantic turnip

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

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

Input. The first line contains integer N(1 ≤ N ≤ 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

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`”.

Sample Input

```
Sample input #1
2
1 -1 2 0
2 1 0 -2

Sample input #2
4
1 3 2 0
2 1 0 4
1 -1 4 1
2 3 2 –2

Sample input #3
4
1 2 2 0
2 1 1 -2
1 4 4 0
2 3 3 -1
```

Sample Output

```
Sample output #1
2.0000

Sample output #2
No solution

Sample output #3
Any
```

Problem Author: Pavel Egorov
Problem Source: USU Championship 2004

HTML

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

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 "<span class=comment>" and "</span>"
2. All the key words must be enclosed in the pair of tags "<span class=keyword>" and "</span>"
3. All the strings must be enclosed in the pair of tags "<span class=string>" and "</span>"
4. All the numbers must be enclosed in the pair of tags "<span class=number>" and "</span>"
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, strings or numbers come upon the strings then they are assumed as a part of the string and not as key words, strings 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

```
Begin
  writeln('Hello world!');
end.
```

Sample Output

```
<span class=keyword>Begin</span>
  writeln(<span class=string>'Hello world!'</span>);
<span class=keyword>end</span>.
```

Hint

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

Intervals of monotonicity

Time Limit: 1.0 second
Memory Limit: 1 000 КБ

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 values 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

```
1 10
1 2 3 4 2 1 -1 3 6 7
```

Sample Output

```
3
```

Problem Author: Aleksandr Klepinin
Problem Source: USU Championship 2004