1000. A+B Problem

Time limit: 1.0 second
Memory limit: 64 MB

Calculate a + b

Input

a and b

Output

a+b

Sample

input	output
1 5	6

Notes

Use + operator

Problem Author: Pavel Atnashev

1001. Reverse Root

Time limit: 2.0 second
Memory limit: 64 MB

The problem is so easy, that the authors were lazy to write a statement for it!

Input

The input stream contains a set of integer numbers A_i ($0 \le A_i \le 10^{18}$). The numbers are separated by any number of spaces and line breaks. A size of the input stream does not exceed 256 KB.

Output

For each number A_i from the last one till the first one you should output its square root. Each square root should be printed in a separate line with at least four digits after decimal point.

Sample

input	output
1427 0 876652098643267843 5276538	2297.0716 936297014.1164 0.0000 37.7757

Problem Author: Prepared by Dmitry Kovalioff

1002. Phone Numbers

Time limit: 2.0 second
Memory limit: 64 MB

In the present world you frequently meet a lot of call numbers and they are going to be longer and longer. You need to remember such a kind of numbers. One method to do it in an easy way is to assign letters to digits as shown in the following picture:

```
1 ij    2 abc   3 def
4 gh    5 kl    6 mn
7 prs   8 tuv   9 wxy
        0 oqz
```

This way every word or a group of words can be assigned a unique number, so you can remember words instead of call numbers. It is evident that it has its own charm if it is possible to find some simple relationship between the word and the person itself. So you can learn that the call number 941837296 of a chess-playing friend of yours can be read as WHITEPAWN, and the call number 2855304 of your favourite teacher is read BULLDOG.

Write a program to find the shortest sequence of words (i.e. one having the smallest possible number of words) which corresponds to a given number and a given list of words. The correspondence is described by the picture above.

Input

Input contains a series of tests. The first line of each test contains the call number, the transcription of which you have to find. The number consists of at most 100 digits. The second line contains the total number of the words in the dictionary (maximum is 50 000). Each of the remaining lines contains one word, which consists of maximally 50 small letters of the English alphabet. The total size of the input doesn't exceed 300 KB. The last line contains call number −1.

Output

Each line of output contains the shortest sequence of words which has been found by your program. The words are separated by single spaces. If there is no solution to the input data, the line contains text “`No solution.`”. If there are more solutions having the minimum number of words, you can choose any single one of them.

Sample

input	output
7325189087 5 it your reality real our 4294967296 5 it your reality real our -1	reality our No solution.

Problem Source: Central European Olympiad in Informatics 1999

1003. Parity

Time limit: 2.0 second
Memory limit: 64 MB

Now and then you play the following game with your friend. Your friend writes down a sequence consisting of zeroes and ones. You choose a continuous subsequence (for example the subsequence from the third to the fifth digit inclusively) and ask him, whether this subsequence contains even or odd number of ones. Your friend answers your question and you can ask him about another subsequence and so on.

Your task is to guess the entire sequence of numbers. You suspect some of your friend's answers may not be correct and you want to convict him of falsehood. Thus you have decided to write a program to help you in this matter. The program will receive a series of your questions together with the answers you have received from your friend. The aim of this program is to find the first answer which is provably wrong, i.e. that there exists a sequence satisfying answers to all the previous questions, but no such sequence satisfies this answer.

Input

Input contains a series of tests. The first line of each test contains one number, which is the length of the sequence of zeroes and ones. This length is less or equal to 10^9. In the second line, there is one non-negative integer which is the number of questions asked and answers to them. The number of questions and answers is less or equal to 5 000. The remaining lines specify questions and answers. Each line contains one question and the answer to this question: two integers (the position of the first and last digit in the chosen subsequence) and one word which is either "`even`" or "`odd`" (the answer, i.e. the parity of the number of ones in the chosen subsequence, where "`even`" means an even number of ones and "`odd`" means an odd number). The input is ended with a line containing −1.

Output

Each line of output containing one integer X. Number X says that there exists a sequence of zeroes and ones satisfying first X parity conditions, but there exists none satisfying $X + 1$ conditions. If there exists a sequence of zeroes and ones satisfying all the given conditions, then number X should be the number of all the questions asked.

Sample

input	output
10 5 1 2 even 3 4 odd 5 6 even 1 6 even 7 10 odd -1	3

Problem Source: Central European Olympiad in Informatics 1999

1004. Sightseeing Trip

Time limit: 0.5 second
Memory limit: 64 MB

There is a travel agency in Adelton town on Zanzibar island. It has decided to offer its clients, besides many other attractions, sightseeing the town. To earn as much as possible from this attraction, the agency has accepted a shrewd decision: it is necessary to find the shortest route which begins and ends at the same place.

Your task is to write a program which finds such a route. In the town there are N crossing points numbered from 1 to N and M two-way roads numbered from 1 to M. Two crossing points can be connected by multiple roads, but no road connects a crossing point with itself. Each sightseeing route is a sequence of road numbers $y_1, \ldots, y_k$, $k \geq 3$. The road y_i ($1 \leq i \leq k-1$) connects crossing points x_i and x_{i+1}, the road y_k connects crossing points x_k and x_1. All the numbers $x_1, \ldots, x_k$ should be different. The length of the sightseeing route is the sum of the lengths of all roads on the sightseeing route, i.e. $L(y_1) + L(y_2) + \ldots + L(y_k)$ where $L(y_i)$ is the length of the road y_i ($1 \leq i \leq k$). Your program has to find such a sightseeing route, the length of which is minimal, or to specify that it is not possible, because there is no sightseeing route in the town.

Input

Input contains T tests ($1 \leq T \leq 5$). The first line of each test contains two integers: the number of crossing points N and the number of roads M ($3 \leq N \leq 100$; $3 \leq M \leq N \cdot (N-1)$). Each of the next M lines describes one road. It contains 3 integers: the number of its first crossing point a, the number of the second one b, and the length of the road l ($1 \leq a, b \leq N$; $a \neq b$; $1 \leq l \leq 300$). Input is ended with a "−1" line.

Output

Each line of output is an answer. It contains either a string "No solution." in case there isn't any sightseeing route, or it contains the numbers of all crossing points on the shortest sightseeing route in the order how to pass them (i.e. the numbers x_1 to x_k from our definition of a sightseeing route), separated by single spaces. If there are multiple sightseeing routes of the minimal length, you can output any one of them.

Sample

input	output
5 7 1 4 1 1 3 300 3 1 10 1 2 16 2 3 100 2 5 15 5 3 20 4 3 1 2 10 1 3 20 1 4 30 -1	1 3 5 2 No solution.

Problem Source: Central European Olympiad in Informatics 1999

1005. Stone Pile

Time limit: 1.0 second
Memory limit: 64 MB

You have a number of stones with known weights w_1, …, w_n. Write a program that will rearrange the stones into two piles such that weight difference between the piles is minimal.

Input

Input contains the number of stones n ($1 \le n \le 20$) and weights of the stones w_1, …, w_n (integers, $1 \le w_i \le 100000$) delimited by white spaces.

Output

Your program should output a number representing the minimal possible weight difference between stone piles.

Sample

input	output
5 5 8 13 27 14	3

Problem Source: USU Championship 1997

1006. Square Frames

Time limit: 2.0 second
Memory limit: 64 MB

The frame consists of the following characters:

Character	ASCII code	Description
┌	218	Left upper corner
┐	191	Right upper corner
└	192	Left bottom corner
┘	217	Right bottom corner
│	179	Vertical (left and right) border line
─	196	Horizontal (top and bottom) border line

N square frames ($1 \le N \le 15$) were sequentially drawn on screen 50 characters wide 20 lines tall. If parts of some frames intersect, only the part of the frame drawn latter remains visible. Each frame lies fully on the screen.

You need to write a program that builds a possible sequence of frames that (if drawn sequentially) would produce the same picture on the screen. Your sequence does not have to be the same with the original sequence used to build the picture on the screen. However, it should not contain more than 2000 frames.

Input

The screen area was originally filled with dots (ASCII 46). Input contains the final picture on the screen after the sequence of frames is drawn.

Output

Your program should write to output the number of frames in the sequence built and the frames coordinates as follows:

K
$X_1\ Y_1\ A_1$
…
$X_k\ Y_k\ A_k$

Here K is the number of frames, X_i and Y_i are coordinates of the upper left frame corner ($0 \le X_i \le 49$, $0 \le Y_i \le 19$) and A_i is the length of the frame side ($2 \le A_i$). All numbers must be delimited with one or more spaces and/or line breaks.

Sample

input	output
(see the figure above)	6 16 11 7 32 14 4 4 8 8 11 6 7 36 11 3 28 8 3

Notes

The input should be read using the cp437 encoding or compatible (e.g. cp866). You can download the sample input here.

Problem Source: USU Championship 1997

1007. Code Words

Time limit: 2.0 second
Memory limit: 64 MB

A transmitter sends over a noisy line some binary code words. The receiver on the other end uses a special technique to recover the original words.

Every word originally consists of symbols 0 and 1. All words have the same length N ($4 \le N \le 1000$). After traveling through the noisy line one (but no more) of the following modifications to a word may occur:

1. Any (but only one) symbol 0 is replaced by 1.
2. Any (but only one) symbol is removed.
3. A symbol (0 or 1) is inserted at any position.

It is known that the original words all have the following property: the sum of positions where symbols 1 are located is a multiple of (N+1) or equal to zero.

Input

Input contains number N followed by received words. The words are delimited with line breaks. There will be no more than 2001 words. There is nothing else in the input data, except maybe for some extra spaces or line breaks.

Output

Your program should print to output the original sequence of words as they were transmitted. The words should be delimited by line breaks.

Sample

input	output
4 0000 011 1011 11011	0000 0110 1001 1111

Problem Source: USU Championship 1997

1008. Image Encoding

Time limit: 2.0 second
Memory limit: 64 MB

There are several ways to encode an image. In this problem we will consider two representations of an image. We assume that the image consists of black and white pixels. There is at least one black pixel and all black pixels are connected with their sides. Coordinates of black pixels are not less than 1 and not greater than 10. An example of such an image is on the figure.

Both representations describe an arrangement of black pixels only.

At the first representation we specify in the first line number of black pixels and coordinates of each black pixel in the following lines. Pixels are listed in order of increasing X. In case of equality of X they are listed in order of increasing Y. Image at the figure is encoded as follows:

6
2 3
2 4
3 3
3 4
4 2
4 3

At the second representation we specify in the first line coordinates of the lowest left black pixel. Each of the following lines contains a description of neighbors for one of the pixels. At first, neighbors of the lowest left pixel are specified, then neighbors of its first neighbor (if it exists) are specified, then neighbors of its second neighbor (if it also exists) follow. When all its neighbors are described the description of the neighbors of its first neighbor follows. The description of the neighbors of its second neighbor follows then and so on.

Each descriptive line contains at most one letter for each neighbor: R for the right, T for the top, L for the left, B for the bottom. If the neighbor was already specified it is not included into the descriptive line and vice-versa. Also, there is only one descriptive line for each pixel. Neighbors are listed counter-clockwise starting with the right. Each descriptive line except the last ends with a comma. The last line ends with a full stop. The image on the figure is encoded as follows:

2 3
RT,
RT,
,
B,
,
.

There are no leading or trailing spaces in any representation. There is exactly one space between X and Y coordinates.

Input

One representation of the image will be given to your program in the input.

Output

Your program has to write other representation of the image to the output.

Samples

input	output
6 2 3 2 4 3 3 3 4 4 2 4 3	2 3 RT, RT, , B, , .
2 3 RT, RT, , B, , .	6 2 3 2 4 3 3 3 4 4 2 4 3

Problem Source: Third Open USTU Collegiate Programming Contest (PhysTech Cup), March 18, 2000

1009. K-based Numbers

Time limit: 0.5 second
Memory limit: 64 MB

Let's consider K-based numbers, containing exactly N digits. We define a number to be valid if its K-based notation doesn't contain two successive zeros. For example:

- 1010230 is a valid 7-digit number;
- 1000198 is not a valid number;
- 0001235 is not a 7-digit number, it is a 4-digit number.

Given two numbers N and K, you are to calculate an amount of valid K based numbers, containing N digits.

You may assume that $2 \le K \le 10$; $N \ge 2$; $N + K \le 18$.

Input

The numbers N and K in decimal notation separated by the line break.

Output

The result in decimal notation.

Sample

input	output
2 10	90

Problem Source: USU Championship 1997

1010. Discrete Function

Time limit: 1.0 second
Memory limit: 64 MB

There is a discrete function. It is specified for integer arguments from 1 to N ($2 \le N \le 100000$). Each value of the function is longint (signed long in C++). You have to find such two points of the function for which all points between them are below than straight line connecting them and inclination of this straight line is the largest.

Input

There is an N in the first line. Than N lines follow with the values of the function for the arguments 1, 2, …, N respectively.

Output

A pair of integers, which are abscissas of the desired points, should be written into one line of output. The first number must be less than the second one. If it is any ambiguity your program should write the pair with the smallest first number.

Sample

input	output
3 2 6 4	1 2

Problem Source: Third Open USTU Collegiate Programming Contest (PhysTech Cup), March 18, 2000

1011. Conductors

Time limit: 2.0 second
Memory limit: 64 MB

Background

Everyone making translations from English to Russian knows an English phrase "Naked conductor runs along the bus". It has two very different meanings.

Problem

Every bus in the Ekaterinburg city has a special man (or woman) called conductor. When you ride the bus, you have to give money to the conductor. We know that there are more than P% conductors and less than Q% conductors of all citizens of Ekaterinburg. Your task is to determine a minimal possible number of Ekaterinburg citizens. By percentage, we know that there are more than P% conductors and less than Q% conductors of all Russian citizens in this city

Input

Two numbers P,Q such that $0.01 \leq P$, $Q \leq 99.99$. Numbers are given with 2 digits precision. These numbers are separated by some spaces or "end of line" symbols.

Output

The minimal number of Ekaterinburg citizens.

Sample

input	output
13 14.1	15

Notes

If there are 15 citizens and 2 conductors among them in Ekaterinburg, then there are 13 1/3 % conductors of all citizens.

Problem Source: USU Championship 1997

1012. K-based Numbers. Version 2

Time limit: 0.5 second
Memory limit: 16 MB

Let's consider K-based numbers, containing exactly N digits. We define a number to be valid if its K-based notation doesn't contain two successive zeros. For example:

- 1010230 is a valid 7-digit number;
- 1000198 is not a valid number;
- 0001235 is not a 7-digit number, it is a 4-digit number.

Given two numbers N and K, you are to calculate an amount of valid K based numbers, containing N digits.

You may assume that $2 \le K \le 10$; $N \ge 2$; $N + K \le 1800$.

Input

The numbers N and K in decimal notation separated by the line break.

Output

The result in decimal notation.

Sample

input	output
2 10	90

1013. K-based Numbers. Version 3

Time limit: 0.5 second
Memory limit: 64 MB

Let's consider K-based numbers, containing exactly N digits. We define a number to be valid if its K-based notation doesn't contain two successive zeros. For example:

- 1010230 is a valid 7-digit number;
- 1000198 is not a valid number;
- 0001235 is not a 7-digit number, it is a 4-digit number.

Given three numbers N, K and M, you are to calculate an amount of valid K based numbers, containing N digits modulo M.

You may assume that $2 \le N, K, M \le 10^{18}$.

Input

The numbers N, K and M in decimal notation separated by the line break.

Output

The result in decimal notation.

Sample

input	output
2 10 100	90

1014. Product of Digits

Time limit: 1.0 second
Memory limit: 64 MB

Your task is to find the minimal positive integer number Q so that the product of digits of Q is exactly equal to N.

Input

The input contains the single integer number N ($0 \le N \le 10^9$).

Output

Your program should print to the output the only number Q. If such a number does not exist print -1.

Sample

input	output
10	25

Problem Source: Ural State University Internal Contest '99 #2

1015. Test the Difference!

Time limit: 2.0 second
Memory limit: 64 MB

There are N ($1 \le N \le 10^5$) dice at the casino's "Royal Flush" storehouse. Some of them are equal, i.e. one can transform one die to another by properly rotating it. Let's say that two dice have the same scheme if it's possible to obtain one of them from another by a series of rotation. In other case (no rotations of the first die lead us to the second die) let's say that dice have different schemes. Your task is to define the dice with the same scheme.

Input

The first line of the input contains the number N. Next N lines contain descriptions of the dice. Each line contains exactly one description of the die. A line describes the die in 6 numbers (separated with spaces): the number of points on the left side of the die, then on the right side, on the top, on the forward side, on the bottom and on the backward side. Certainly, those 6 numbers represent a permutation of integer numbers from 1 to 6 inclusively.

Output

The first line of the output should contain the only number Q of different die's schemes at the storehouse. Next Q lines should contain the numbers of dice with the same scheme. To be more precisely the second line must begin with 1 and after that (separated by spaces) numbers of dice with the same as die #1 scheme must follow. We assume that all dice from the input are numbered from 1 to N. The third line (if not all the dice have the same scheme) must begin with the smallest possible number with the scheme different from the scheme of the die #1. This number (say P) is followed by numbers of dice with the same scheme as the scheme of the die #P. All next lines must be printed in the same manner. Numbers in each line of the output must be sorted in increasing order.

Sample

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

Problem Source: Ural State University Internal Contest '99 #2

1016. Cube on the Walk

Time limit: 2.0 second
Memory limit: 64 MB

A cube placed on some square of a usual chessboard. A cube completely covers one square of the chessboard but not anything more, i.e. size of cube's edge is equal to the size of square's edge. The integer number N ($0 \le N \le 1000$) is written on the each side of the cube. However it does not imply that the same number is written on all sides. On the different sides there are might be different numbers. One can move a cube to the next square by rotating it around the common edge of the cube and the square. During this motion the sum of the numbers on the bottom of the cube is calculated (each number is added as much times as it appeared at the bottom of the cube). Your task is to find the route between two given squares with the minimal sum of numbers on the bottom side. The numbers on the bottom at the beginning and at the end of walk are also counted. The start and the end positions are different.

Input

The only line of the input contains the necessary data set (only spaces used as delimiters). First, the start position is given, and then the end position. Each position is composed from the character (from 'a' to 'h' inclusively, it defines the number of the column on the chessboard) and the digit (from '1' to '8' inclusively, it defines the number of the row). That positions are followed by 6 numbers which are currently written on the near, far, top, right, bottom and left sides of the cube correspondingly.

Output

The only line of the output must contain the minimal sum followed by the optimal route (one of possible routes with minimal sum). The route must be represented by the sequence of cube's positions during the walk. It begins with the start square and ends with the finish square. All square positions on the chessboard should be given in the same format as in input. Use spaces as delimiters.

Sample

input	output
e2 e3 0 8 1 2 1 1	5 e2 d2 d1 e1 e2 e3

Problem Source: Ural State University Internal Contest '99 #2

1017. Staircases

Time limit: 1.0 second
Memory limit: 64 MB

One curious child has a set of N little bricks. From these bricks he builds different staircases. Staircase consists of steps of different sizes in a strictly descending order. It is not allowed for staircase to have steps equal sizes. Every staircase consists of at least two steps and each step contains at least one brick. Picture gives examples of staircase for N=11 and N=5:

Your task is to write a program that reads the number N and writes the only number Q — amount of different staircases that can be built from exactly N bricks.

Input

Number N ($5 \le N \le 500$).

Output

Number Q

Samples

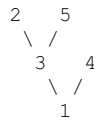
input	output
5	2
212	995645335

Problem Source: Ural State University Internal Contest '99 #2

1018. Binary Apple Tree

Time limit: 1.0 second
Memory limit: 64 MB

Let's imagine how apple tree looks in binary computer world. You're right, it looks just like a binary tree, i.e. any biparous branch splits up to exactly two new branches. We will enumerate by integers the root of binary apple tree, points of branching and the ends of twigs. This way we may distinguish different branches by their ending points. We will assume that root of tree always is numbered by 1 and all numbers used for enumerating are numbered in range from 1 to N, where N is the total number of all enumerated points. For instance in the picture below N is equal to 5. Here is an example of an enumerated tree with four branches:



As you may know it's not convenient to pick an apples from a tree when there are too much of branches. That's why some of them should be removed from a tree. But you are interested in removing branches in the way of minimal loss of apples. So your are given amounts of apples on a branches and amount of branches that should be preserved. Your task is to determine how many apples can remain on a tree after removing of excessive branches.

Input

First line of input contains two numbers: N and Q ($2 \le N \le 100$; $1 \le Q \le N - 1$). N denotes the number of enumerated points in a tree. Q denotes amount of branches that should be preserved. Next $N - 1$ lines contains descriptions of branches. Each description consists of a three integer numbers divided by spaces. The first two of them define branch by it's ending points. The third number defines the number of apples on this branch. You may assume that no branch contains more than 30000 apples.

Output

Output should contain the only number — amount of apples that can be preserved. And don't forget to preserve tree's root ;-)

Sample

input	output
5 2 1 3 1 1 4 10 2 3 20 3 5 20	21

Problem Source: Ural State University Internal Contest '99 #2

1019. Line Painting

Time limit: 2.0 second
Memory limit: 64 MB

The segment of numerical axis from 0 to 10^9 is painted into white color. After that some parts of this segment are painted into black, then some into white again and so on. In total there have been made N re-paintings ($1 \le N \le 5000$). You are to write a program that finds the longest white open interval after this sequence of re-paintings.

Input

The first line of input contains the only number N. Next N lines contain information about re-paintings. Each of these lines has a form:
a_i b_i c_i
where a_i and b_i are integers, c_i is symbol 'b' or 'w', a_i, b_i, c_i are separated by spaces.
This triple of parameters represents repainting of segment from a_i to b_i into color c_i ('w' — white, 'b' — black). You may assume that $0 < a_i < b_i < 10^9$.

Output

Output should contain two numbers x and y ($x < y$) divided by space(s). These numbers should define the longest white open interval. If there are more than one such an interval output should contain the one with the smallest x.

Sample

input	output
4 1 999999997 b 40 300 w 300 634 w 43 47 b	47 634

Problem Source: Ural State University Internal Contest '99 #2

1020. Rope

Time limit: 1.0 second
Memory limit: 64 MB

Plotters have barbarously hammered N nails into an innocent plane shape, so that one can see now only heads. Moreover, pursuing their mean object, they have hammered all the nails into the vertices of a convex polygon. After that they…it is awful… have roped off the nails, so that the shape felt upset (the rope was very thin). They've done it as it is shown in the figure.

Your task is to find out a length of the rope.

Input

There two numbers in the first line of the standard input: N — a number of nails ($1 \le N \le 16$), and a real number R — a radius of heads of nails. All the heads have the same radius. Further there are N lines, each of them contains a pair of real coordinates (separated by a space) of centers of nails. An absolute value of the coordinates doesn't exceed 100. The nails are described either in a clockwise or in a counterclockwise order starting from an arbitrary nail. Heads of different nails don't overlap.

Output

Output a real number with two digits precision (after a decimal point) — a length of the rope.

Sample

input	output
4 1 0.0 0.0 2.0 0.0 2.0 2.0 0.0 2.0	14.28

Problem Author: Alexander Petrov & Nikita Shamgunov
Problem Source: Ural State University Internal Contest October'2000 Junior Session

1021. Sacrament of the Sum

Time limit: 1.0 second
Memory limit: 64 MB

Background

— The Brother of mine, the Head of Monastic Order wants to know tomorrow about the results long-term researches. He wants to see neither more nor less than the Summering Machine! Even moreover, he wants our Machine — only a machine — to demonstrate its comprehension of the Sacrament of the Sum as deeply as it is possible. He wants our Machine to find two numbers that give the sum equal to the Sacred Number 10 000.
— Tsh-sh-sh! This is madness that borders on blasphemy! How can the Machine calculate the Sacred Number? Twenty seven years we work on it, but we've could teach it to tell if the sum of two introduced numbers greater or lower than 10 000. Can an ordinary mortal find two numbers that there sum will be equal to 10 000?
— But we'll have to do it with the help of our Machine, even if it is not capable. Otherwise we'll have… let's say, big problems, if it is possible to call boiling oil like this. However, I have an idea. Do you remember, last week we've entered two numbers -7 and 13 into the Machine, and it answered that their sum is lower than 10 000. I don't know how to check this, but nothing's left for us than to believe to the fruit of our work. Let's enter now a greater number than -7 and start up the Machine again. We'll do like this again and again until we find a number that being added to 13 will give us 10 000. The only thing we are to do is to prepare an ascending list of numbers.
— I don't believe in this… Let's start with the sum that is obviously greater than the Sacred Number and we'll decrease one of the summand. So we have more chances to avoid boilin… big problems.

Haven't come to an agreement, the Brothers went away to their cells. By next day everyone of them has prepared a list of numbers that, to his opinion, could save them… Can both of the lists save them together?

Problem

Your program should decide, if it is possible to choose from two lists of integers such two numbers that their sum would be equal to 10 000.

Input

You are given both of these lists one by one. Format of each of these lists is as follows: in the first line of the list the quantity of numbers N_i of the i-th list is written. Further there is an i-th list of numbers each number in its line (N_i lines).The following conditions are satisfied: $1 \le N_i \le 50\ 000$, each element of the lists lays in the range from -32768 to 32767. The first list is ascending and the second one is descending.

Output

You should write "YES" to the standard output if it is possible to choose from the two lists of integers such two numbers that their sum would be equal to 10 000. Otherwise you should write "NO".

Sample

input	output
4 -175 19 19 10424 3 8951 -424 -788	YES

Problem Author: Leonid Volkov & Alexander Petrov
Problem Source: Ural State University Internal Contest October'2000 Junior Session

1022. Genealogical Tree

Time limit: 1.0 second
Memory limit: 64 MB

Background

The system of Martians' blood relations is confusing enough. Actually, Martians bud when they want and where they want. They gather together in different groups, so that a Martian can have one parent as well as ten. Nobody will be surprised by a hundred of children. Martians have got used to this and their style of life seems to them natural.

And in the Planetary Council the confusing genealogical system leads to some embarrassment. There meet the worthiest of Martians, and therefore in order to offend nobody in all of the discussions it is used first to give the floor to the old Martians, than to the younger ones and only than to the most young childless assessors. However, the maintenance of this order really is not a trivial task. Not always Martian knows all of his parents (and there's nothing to tell about his grandparents!). But if by a mistake first speak a grandson and only than his young appearing great-grandfather, this is a real scandal.

Problem

Your task is to write a program, which would define once and for all, an order that would guarantee that every member of the Council takes the floor earlier than each of his descendants.

Input

The first line of the standard input contains an only number N, $1 \le N \le 100$ — a number of members of the Martian Planetary Council. According to the centuries-old tradition members of the Council are enumerated with integers from 1 up to N. Further, there are exactly N lines, moreover, the i-th line contains a list of i-th member's children. The list of children is a sequence of serial numbers of children in a arbitrary order separated by spaces. The list of children may be empty. The list (even if it is empty) ends with 0.

Output

The standard output should contain in its only line a sequence of speakers' numbers, separated by spaces. If several sequences satisfy the conditions of the problem, you are to write to the standard output any of them. At least one such sequence always exists.

Sample

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

Problem Author: Leonid Volkov
Problem Source: Ural State University Internal Contest October'2000 Junior Session

1023. Buttons

Time limit: 2.0 second
Memory limit: 64 MB

Background

As you surely already know, Yekaterinburg has gotten its right to hold The Summer Olympic Games of the 2032. It is planned that it will be allowed to Russia as a country-organizer to emend a program of the games a bit. So, in order to improve the command result it has been decided to replace the competition in gymnastics by the competition in the new game "Buttons".

The rules of the game are very simple. There's a small heap of K buttons before two players. The players in turns take buttons from the heap, moreover, at a time one can take a number of buttons from 1 up to L. The one who takes the last button is the winner.

The rules of the Olympic Games will be a bit harder then usual. The one, who is to make a first step according to a lot, has an opportunity to fix a number K with the following restriction to it: $3 \le K \le 10^8$ (that is the exact number of buttons that has been prepared for the Olympic tournament). The player who is to make the second step fixes a number L that satisfies the following conditions $2 \le L < K$.

Problem

A very crucial task is given to your team: you are to write a program that should help the second player to make his choice. In other words, given a number K your program is to find a number L that guaranties a victory to the second player with a proper game of both sides.

So, for instance, there are only three buttons in the heap, the choice $L = 2$ provides for the victory of the second player. Really, if the first player takes only one button at his turn, the second one wins, taking the two last buttons. On the contrary, if the first one takes two buttons, the second one wins, taking the last button.

Input

The standard input consists of one line, which contains an only integer number K — a number of buttons in the heap, that has fixed the first player at his turn.

Output

To the standard output you are to write the only number L — the maximal number of buttons that can be taken at a time which provides for the victory of the second player. If there are several those numbers L, you should write the least. If there are no such numbers, you are to write 0 to the standard output.

Sample

input	output
3	2

Problem Author: Leonid Volkov
Problem Source: Ural State University Internal Contest October'2000 Junior Session

1024. Permutations

Time limit: 2.0 second
Memory limit: 64 MB

Background

We remind that the permutation of some final set is a one-to-one mapping of the set onto itself. Less formally, that is a way to reorder elements of the set. For example, one can define a permutation of the set {1,2,3,4,5} as follows:

This record defines a permutation P as follows: P(1) = 4, P(2) = 1, P(3) = 5, etc.

What is the value of the expression P(P(1))? It's clear, that P(P(1)) = P(4) = 2. And P(P(3)) = P(5) = 3. One can easily see that if P(n) is a permutation then P(P(n)) is a permutation as well. In our example (check it by yourself)

It is natural to denote this permutation by $P^2(n) = P(P(n))$. In a general form the definition is as follows: $P(n) = P^1(n)$, $P^k(n) = P(P^{k-1}(n))$.

Among the permutations there is a very important one — that moves nothing:

It is clear that for every k the following relation is satisfied: $(E_N)^k = E_N$. The following less trivial statement is correct (we won't prove it here, you may prove it yourself incidentally):

Let P(n) be some permutation of an N elements set. Then there exists a positive integer k, that $P^k = E_N$.

The least positive integer k such that $P^k = E_N$ is called an order of the permutation P.

Problem

The problem that your program should solve is formulated now in a very simple manner: *"Given a permutation find its order."*

Input

The first line contains the only integer N ($1 \le N \le 1000$), that is a number of elements in the set that is rearranged by this permutation. In the second line there are N integers of the range from 1 up to N, separated by a space, that define a permutation — the numbers P(1), P(2),…, P(N).

Output

You should write the order of the permutation. You may consider that an answer shouldn't exceed 10^9.

Sample

input	output
5 4 1 5 2 3	6

Problem Author: Nikita Shamgunov
Problem Source: Ural State University Internal Contest October'2000 Junior Session

1025. Democracy in Danger

Time limit: 1.0 second
Memory limit: 64 MB

Background

In one of the countries of Caribbean basin all decisions were accepted by the simple majority of votes at the general meeting of citizens (fortunately, there were no lots of them). One of the local parties, aspiring to come to power as lawfully as possible, got its way in putting into effect some reform of the election system. The main argument was that the population of the island recently had increased and it was to longer easy to hold general meetings.

The essence of the reform is as follows. From the moment of its coming into effect all the citizens were divided into K (may be not equal) groups. Votes on every question were to be held then in each group, moreover, the group was said to vote “for” if more than half of the group had voted “for”, otherwise it was said to vote “against”. After the voting in each group a number of group that had voted “for” and “against” was calculated. The answer to the question was positive if the number of groups that had voted “for” was greater than the half of the general number of groups.

At first the inhabitants of the island accepted this system with pleasure. But when the first delights dispersed, some negative properties became obvious. It appeared that supporters of the party, that had introduced this system, could influence upon formation of groups of voters. Due to this they had an opportunity to put into effect some decisions without a majority of voters “for” it.

Let’s consider three groups of voters, containing 5, 5 and 7 persons, respectively. Then it is enough for the party to have only three supporters in each of the first two groups. So it would be able to put into effect a decision with the help of only six votes “for” instead of nine, that would be necessary in the case of general votes.

Problem

You are to write a program, which would determine according to the given partition of the electors the minimal number of supporters of the party, sufficient for putting into effect of any decision, with some distribution of those supporters among the groups.

Input

In the first line an only odd integer K — a quantity of groups — is written ($1 \le K \le 101$). In the second line there are written K odd integers, separated with a space. Those numbers define a number of voters in each group. The population of the island does not exceeds 9999 persons.

Output

You should write a minimal quantity of supporters of the party, that can put into effect any decision.

Sample

input	output
3 5 7 5	6

Problem Author: Leonid Volkov
Problem Source: Ural State University Internal Contest October'2000 Junior Session

1026. Questions and Answers

Time limit: 1.0 second
Memory limit: 64 MB

Background

The database of the Pentagon contains a top-secret information. We don't know what the information is — you know, it's top-secret, — but we know the format of its representation. It is extremely simple. We don't know why, but all the data is coded by integers from 1 up to 5000. The size of the main base (we'll denote it be N) is rather big — it may contain up to 100 000 those numbers. The database is to process quickly every query. The most often query is: "Which element is i-th by its value?"— with i being an integer in a range from 1 to N.

Problem

Your program is to play a role of a controller of the database. In the other words, it should be able to process quickly queries like this.

Input

Input of the problem consists of two parts. At first, a database is written, and then there's a sequence of queries. The format of database is very simple: in the first line there's a number N, in the next N lines there are numbers of the database one in each line in an arbitrary order. A sequence of queries is written simply as well: in the first line of the sequence a number of queries K ($1 \le K \le 100$) is written, and in the next K lines there are queries one in each line. The query "Which element is i-th by its value?" is coded by the number i. A database is separated from a sequence of queries by the string of three symbols "#".

Output

The output should consist of K lines. In each line there should be an answer to the corresponding query. The answer to the query "i" is an element from the database, which is i-th by its value (in the order from the least up to the greatest element).

Sample

input	output
5 7 121 123 7 121 ### 4 3 3 2 5	121 121 7 123

Problem Author: Leonid Volkov
Problem Source: Ural State University Internal Contest October'2000 Junior Session

1027. D++ Again

Time limit: 0.5 second
Memory limit: 64 MB

Background

The language D++, that was perfected by the participants of our March competition, continues improving. Its founders try to make the syntax as clear as it is possible in order to make the programming simpler in the future. Of course, some minimal set of rules is to stay without changes.

Your program is to check the observance of rules, concerning the arrangement of brackets and comments.

Problem

A text of a correct D++ program contains a symbol part, arithmetic expressions and comments. Comments may appear everywhere and may contain any symbols. A comment is always opened by a pair of symbols "(*" and is closed by a pair of symbols "*)". Each comment must be closed. An arithmetic expression in D++ is always opened by "(", is closed by ")" and may contain only symbols "=+-*/0123456789)(" and "end of line" symbols. An arithmetic expression can't start with a pair of symbols "(*". You may run across embedded brackets in an arithmetic expression. In this case these brackets are to be balanced. It means that "((1)))" as well as "(23))((+)" are not correct arithmetic expressions. An arithmetic expression is correct if and only if brackets placed correctly. At last, all the rest of the program text (the result of rejection of all comments and arithmetic expressions from the initial text of the program) may contain every symbol excluding "(" and ")".

We would like to especially notice that the spaces are possible anywhere in a text of a program except when appearing in arithmetic expressions.

Input

Some text is written in the standard input. There are not more than 10000 symbols in the text. There may be Latin letters, digits, brackets, symbols of arithmetic operations, spaces and "end of line" symbols.

Output

Your program should write "YES" to the output if the introduced text is a correct D++ program, and "NO" otherwise.

Sample

input
`Hello, here is a sample D++ program. It contains some arithmetical expressions like (2+2=4), (2+-/*) and ((3+3)*3=20(*this is not true, but you don't have to verify it :-) *)+8) (* the closing bracket in the previous comment is also in order, since this bracket does not belong to any arithmetical expression*)`
output
`YES`

Problem Author: Leonid Volkov, Alexey Lysenko
Problem Source: Ural State University Internal Contest October'2000 Junior Session

1028. Stars

Time limit: 0.25 second
Memory limit: 64 MB

Astronomers often examine star maps where stars are represented by points on a plane and each star has Cartesian coordinates. Let the level of a star be an amount of the stars that are not higher and not to the right of the given star. Astronomers want to know the distribution of the levels of the stars.

For example, look at the map shown on the figure above. Level of the star number 5 is equal to 3 (it's formed by three stars with a numbers 1, 2 and 4). And the levels of the stars numbered by 2 and 4 are 1. At this map there are only one star of the level 0, two stars of the level 1, one star of the level 2, and one star of the level 3.

You are to write a program that will count the amounts of the stars of each level on a given map.

Input

The first line contains an integer N that is a number of stars ($1 \le N \le 15000$). The following N lines contain integers X_i and Y_i that are coordinates of stars ($0 \le X_i, Y_i \le 32000$). There can be only one star at one point of the plane. Stars are listed in ascending order of Y coordinate. Stars with equal Y coordinates are listed in ascending order of X coordinate.

Output

Output N integers, one per line. The first line should contain an amount of stars of the level 0, the second does an amount of stars of the level 1 and so on, the last line contains an amount of stars of the level $N - 1$.

Sample

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

Problem Author: Pavel Zaletsky
Problem Source: Ural Collegiate Programming Contest '99

1029. Ministry

Time limit: 1.0 second
Memory limit: 64 MB

Mr. F. wants to get a document be signed by a minister. A minister signs a document only if it is approved by his ministry. The ministry is an M-floor building with floors numbered from 1 to M. Each floor has N rooms numbered from 1 to N. In each room there is one (and only one) official.

A document is approved by the ministry only if it is signed by at least one official from the M-th floor. An official signs a document only if at least one of the following conditions is satisfied:

a. the official works on the first floor;
b. the document is signed by the official working in the room with the same number but situated one floor below;
c. the document is signed by an official working in a neighbouring room (rooms are neighbouring if they are situated on the same floor and their numbers differ by one).

Each official collects a fee for signing a document. The fee is a positive integer not exceeding 10^9.

You should find the cheapest way to approve the document.

Input

The first line contains integers M and N representing the number of floors in the building and the number of rooms per floor ($1 \le M \le 100$; $1 \le N \le 500$). Each of the next M lines contains N integers that describe fees (the j-th integer at i-th line is the fee required by the official working in the j-th room at the i-th floor).

Output

Output the numbers of rooms in the order they should be visited to approve the document in the cheapest way. If there are more than one way leading to the cheapest cost you may print any of them.

Sample

input	output
3 4 10 10 1 10 2 2 2 10 1 10 10 10	3 3 2 1 1

Notes

You can assume that for each official there always exists a way to get the approval of a document (from the first floor to this official inclusively) paying no more than 10^9.

Problem Author: Evgeny Frolov
Problem Source: Ural Collegiate Programming Contest '99

1030. Titanic

Time limit: 0.5 second
Memory limit: 64 MB

Background

It is a historical fact that during the legendary voyage of “Titanic” the wireless telegraph machine had delivered 6 warnings about the danger of icebergs. Each of the telegraph messages described the point where an iceberg had been noticed. The first five warnings were transferred to the captain of the ship. The sixth one came late at night and a telegraph operator did not notice that the coordinates mentioned were very close to the current ship's position.

Problem

Write a program that will warn the operator about the danger of icebergs!

Input

The input messages are of the following format:

```
Message #<n>.
Received at <HH>:<MM>:<SS>.
Current ship's coordinates are
<X1>^<X2>'<X3>" <NL/SL>
and <Y1>^<Y2>'<Y3>" <EL/WL>.
An iceberg was noticed at
<A1>^<A2>'<A3>" <NL/SL>
and <B1>^<B2>'<B3>" <EL/WL>.
===
```

Here <n> is a positive integer, <HH>:<MM>:<SS> is the time of the message reception, <X1>^<X2>'<X3>" <NL/SL> and <Y1>^<Y2>'<Y3>" <EL/WL> means **“X1 degrees X2 minutes X3 seconds of North (South) latitude and Y1 degrees Y2 minutes Y3 seconds of East (West) longitude.”**

Output

Your program should output a message in the following format:

```
The distance to the iceberg: <s> miles.
```

Where <s> should be the distance between the ship and the iceberg, that is the length of the shortest path on the sphere between the ship and the iceberg. This distance should be printed up to (and correct to) two decimal digits. If this distance is less than (but not equal to!) 100 miles the program should print one more line with the text:

```
DANGER!
```

Sample

input	output
Message #513. Received at 22:30:11. Current ship's coordinates are 41^46'00" NL and 50^14'00" WL. An iceberg was noticed at 41^14'11" NL and 51^09'00" WL. ===	The distance to the iceberg: 52.04 miles. DANGER!

Notes

For simplicity of calculations assume that the Earth is an ideal sphere with the diameter of 6875 miles completely covered with water. Also you can be sure that lines break exactly as it is shown in the input samples. The ranges of the ship and the iceberg coordinates are the same as the usual range for geographical coordinates, i.e. from 0 to 90 degrees inclusively for NL/SL and from 0 to 180 degrees inclusively for EL/WL.

Problem Author: Evgeny Shtykov
Problem Source: Ural Collegiate Programming Contest '99

1031. Railway Tickets

Time limit: 1.0 second
Memory limit: 64 MB

The railway line “Yekaterinburg-Sverdlovsk” with several stations has been built. This railway line can be represented as a line segment, railway stations being points on it. The railway line starts at the station “Yekaterinburg” and finishes at the station “Sverdlovsk”, so stations are numbered starting from “Yekaterinburg” (it has number 1) and “Sverdlovsk” is the last station.

Cost of the ticket between any two stations depends only on a distance between them. The prices for the tickets are specified in the following table.

расстояние X между станциями	price for the ticket
$0 < X \le L_1$	C_1
$L_1 < X \le L_2$	C_2
$L_2 < X \le L_3$	C_3

Direct tickets from one station to another can be booked if and only if the distance between these station does not exceed L_3. So sometimes it is necessary to book several tickets for the parts of the whole way between two required stations.

For example, on the railway line shown at the figure above there are seven stations. The direct ticket from the second station to the sixth one can not be booked. There are several ways to pay for the travel between these stations. One of them is to book two tickets: one ticket at price C_2 to travel between the second and the third stations, and other at price C_3 to travel between the third and the sixth stations. Note, that though the distance between the second and the sixth stations is equal to $2L_2$, the whole travel can not be paid by booking two tickets at price C_2, because each ticket is valid for only one travel and each travel should start and end only at stations.

Your task is to write a program, that will find the minimal cost of the travel between two given stations.

Input

The first line contains integers L_1, L_2, L_3, C_1, C_2 and C_3 ($1 \le L_1 < L_2 < L_3 \le 10^9$; $1 \le C_1 < C_2 < C_3 \le 10^9$). The second line contains an integer N that is the amount of stations ($2 \le N \le 10000$). The third line contains two different integers. They represent serial numbers of stations, the travel between which must be paid. Next $N - 1$ lines contain distances from the first station (“Yekaterinburg”) on the railway line to others. These distances are given as different positive integers and are arranged in the ascending order. The distance from “Yekaterinburg” to “Sverdlovsk” does not exceed 10^9. The distance between any neighboring stations does not exceed L_3. The minimal travel cost between two given stations will not exceed 10^9.

Output

Output the minimal travel cost between two given stations.

Sample

input	output
3 6 8 20 30 40 7 2 6 3 7 8 13 15 23	70

Problem Author: Pavel Zaletsky
Problem Source: Ural Collegiate Programming Contest '99

1032. Find a Multiple

Time limit: 1.0 second
Memory limit: 64 MB

The input contains N positive integers. These integers are not necessarily different (so it may happen that two or more of them will be equal). Your task is to choose a few of given integers ($1 \le few \le N$) so that their sum is a multiple for N, i.e. equals $N \cdot k$ for some integer k.

Input

The first line contains an integer N ($1 \le N \le 10000$). Each of the next N lines contains one integer from the given set. All integers are positive and not greater than 15000.

Output

If the target set of integers can not be found output the single number 0. Otherwise output the number of the chosen integers in the first line followed by the chosen integers themselves (on a separate line each) in arbitrary order.

If there are more than one set of integers with required properties you may output any of them.

Sample

input	output
5 1 2 3 4 1	2 2 3

Problem Author: Dmitry Filimonenkov
Problem Source: Ural Collegiate Programming Contest '99

1033. Labyrinth

Time limit: 1.0 second
Memory limit: 64 MB

Administration of the labyrinth has decided to start a new season with new wallpapers. For this purpose they need a program to calculate the surface area of the walls inside the labyrinth. This job is just for you!

The labyrinth is represented by a matrix $N \times N$. Some matrix cells contain a dot character ('.') that denotes an empty square. Other cells contain a sharp character ('#') that denotes a square filled by monolith block of stone wall. All squares are of the same size 3×3 meters.

The walls are constructed around the labyrinth (except for the upper left and lower right corners, which are used as entrances) and on the cells with a diesis character. No other walls are constructed. There always will be a dot character at the upper left and lower right corner cells of the input matrix.

Your task is to calculate the area of visible part of the walls inside the labyrinth. In other words, the area of the walls' surface visible to a visitor of the labyrinth. Note that there's no holes to look or to move through between any two adjacent blocks of the wall. The blocks are considered to be adjacent if they touch each other in any corner. See picture for an example: visible walls inside the labyrinth are drawn with bold lines. The height of all the walls is 3 meters.

Input

The first line contains an integer N ($3 \le N \le 33$; you see, '3' is a magic digit!). The next N lines contain N characters each. Each line describes one row of the labyrinth matrix. In each line only '.' and '#' characters are used.

Output

Output the exact value of the area of the wallpaper needed.

Sample

input	output
5 ....## ..#.. ..###	198

Problem Author: Vladimir Pinaev
Problem Source: Ural Collegiate Programming Contest '99

1034. Queens in Peaceful Positions

Time limit: 1.0 second
Memory limit: 64 MB

N queens are placed on a chessboard of size $N \times N$. We'll say that these queens are in peaceful position if none of them can attack another. You are to find the total amount of peaceful positions that can be obtained from the given peaceful position by rearranging of **exactly three queens**.

Input

The first line contains an integer N ($4 \le N \le 50$). It is followed by N lines describing positions of queens. Each line contains integers X and Y representing horizontal and vertical coordinates ($1 \le X, Y \le N$).

Output

Output the number of peaceful positions that can be achieved from the initial position by moving of exactly three queens.

Note: the queens are not numbered so if you rearrange them on the chessboard using only squares they already occupied you'll get the same peaceful position, not the new one.

Sample

input	output
4 2 1 1 3 3 4 4 2	0

Problem Author: Dmitry Filimonenkov
Problem Source: Ural Collegiate Programming Contest '99

1035. Cross-stitch

Time limit: 0.5 second
Memory limit: 64 MB

Archaeologists have found a cloth decorated with needlework. This needlework is a cross-stitch made with several threads. The following rules have been observed:

1. The cloth has a grid with square cells.
2. Each stitch covers a diagonal of one cell of the grid. Stitches can lie on both sides of the cloth, but each of them lies only at one side of the cloth (the thread can start, finish and cross the cloth only at the grid vertices).
3. At most one stitch can lie on each diagonal of each cell at each side of the cloth.
4. Each thread makes up several stitches arranged alternately at different sides of the cloth. (It means that two consecutive stitches formed by one thread lay at the different sides of the cloth and are connected in the grid vertex)
5. A needle can go through the cloth only in the vertexes of the grid.

On the figure you can see an example of a pattern made with six stitches. The grid has size 4 × 5. The face of the cloth is drawn on the upper half of the figure. The stitches lying on the face are drawn with solid lines. The rear stitches uncovered with those of the face are drawn with dot-lines. On the lower half of the figure the cloth is oriented as on the upper half. All the rear stitches are drawn with solid lines there. The face stitches, which do not cover rear stitches, are drawn with dot-lines. This cross-stitch cannot be made with less than four threads.

Archaeologists want to know if the pattern was made with the least number of threads. You have to write a program, which will determine the minimal number of threads needed to make the given pattern.

Input

The first line contains integers N and M that are vertical (N) and horizontal (M) sizes of the grid ($1 \le N, M \le 200$). Each of the following $2N$ lines contains M symbols. Each symbol describes one square of the grid. The first N lines correspond to the face of the cloth and the last N lines correspond to the rear of the cloth. The symbols used are ".", "/", "\" and "X" (a dot means an empty square).

For more information see the sample. It corresponds to the cloth drawn at the figure.

Output

Output the minimal number of threads needed to make the described pattern.

Sample

input	output
4 5 ..... .\... ..\.. ....\ .\X..	4

Problem Author: Pavel Zaletsky
Problem Source: Ural Collegiate Programming Contest '99

1036. Lucky Tickets

Time limit: 2.0 second
Memory limit: 64 MB

You are given a number $1 \le N \le 50$. Every ticket has its $2N$-digit number. We call a ticket lucky, if the sum of its first N digits is equal to the sum of its last N digits. You are also given the sum of ALL digits in the number. Your task is to count an amount of lucky numbers, having the specified sum of ALL digits.

Input

Two space-separated numbers: N and S. Here S is the sum of all digits. Assume that $0 \le S \le 1000$.

Output

The amount of lucky tickets.

Sample

input	output
2 2	4

Notes

The tickets are 0101, 0110, 1001, 1010 in the example above

1037. Memory Management

Time limit: 0.4 second
Memory limit: 64 MB

Background

Don't you know that at school pupils' programming contest a new computer language has been developed. We call it D++. Generally speaking it doesn't matter if you know about it or not. But to run programs written in D++ we need a new operating system. It should be rather powerful and complex. It should work fast and have a lot of possibilities. But all this should be done in a future.

And now you are to… No. You should not devise the name for the operating system. You are to write the first module for this new OS. And of course it's the memory management module. Let's discuss how it is expected to work.

Problem

Our operating system is to allocate memory in pieces that we'll call "blocks". The blocks are to be numbered by integers from 1 up to N. When operating system needs more memory it makes a request to the memory management module. To process this request the memory management module should find free memory block with the least number. You may assume that there are enough blocks to process all requests.

Now we should define the meaning of words "free block". At the moment of first request to the memory management module all blocks are considered to be free. Also a block becomes free when there were no requests to it during T minutes.

You may wonder about a notion "request to allocated blocks". What does it mean, "request to allocated block"? The answer is simple: at any time the memory management module may be requested to access a given block. To process this request the memory management module should check if the requested block is really allocated. If it is, the request is considered to be successful and the block remains allocated for T minutes more. Otherwise the request fails.

That's all about the algorithms of the memory management block. You are to implement them for $N = 30\,000$ and $T = 10$ minutes.

Input

Each line of input contains a request for memory block allocation or memory block access. Memory allocation request has a form:

`<Time> +`

where `<Time>` is a nonnegative integer number not greater than 65 000. Time is given in seconds.

Memory block access request has a form:

`<Time> . <BlockNo>`

where `<Time>` meets conditions mentioned above for the memory allocation request and `<BlockNo>` is an integer value in range from 1 to N.

Requests are arranged by their times in a non-decreasing order. There will be no more than 80000 requests.

Output

For each line of input you should print exactly one line with a result of request processing. For memory allocation request you are to write an only integer — a number of allocated block. As it was mentioned above you may assume that every request can be satisfied, there will be no more than N simultaneously allocated blocks. For memory block access request you should print the only character:

- '+' if request is successful (i.e. block is really allocated);
- '-' if request fails (i.e. block with number given is free, so it can't be accessed).

Requests should be processed in that order as they appear in input.

Sample

input	output
1 + 1 + 1 + 2 . 2 2 . 3 3 . 30000 601 . 1 601 . 2 602 . 3 602 + 602 + 1202 . 2	1 2 3 + + - - + - 1 3 -

Problem Author: Alexander Klepinin

Problem Source: Ural State University Internal Contest October'2000 Students Session

1038. Spell Checker

Time limit: 0.5 second
Memory limit: 64 MB

The boss of a firm that you are employed with is dissatisfied with the text processor Word. He wants you to write a better text processor by tomorrow. The interface of the new processor should be clearer, there should be more options, and the resulting text should be more beautiful. You told the boss that this work would take not less than four days. Then your boss asked you to begin with a spell checking program. This program should check capital and small letters. It should detect a mistake in each of the following cases.

1. The first letter in a sentence is small.
2. A capital letter is not the first letter in a word.

A word is a sequence of letters not containing any other symbols or ends of line.

The end of a sentence is defined a full stop, a question-mark or an exclamation mark.

Input

Input contains a text that consists of capital and small letters of the Latin alphabet (A–Z, a–z), digits (0–9), punctuation marks (.,;:-!?) and space characters. The text length is not more than 10000.

Output

Output a number of mistakes in the input text.

Sample

input	output
This sentence iz correkt! -It Has,No mista;.Kes et oll. But there are two BIG mistakes in this one! and here is one more.	3

Problem Author: Alexander Halperin
Problem Source: Ural State University Internal Contest October'2000 Students Session

1039. Anniversary Party

Time limit: 0.5 second
Memory limit: 8 MB

The president of the Ural State University is going to make an 80'th Anniversary party. The university has a hierarchical structure of employees; that is, the supervisor relation forms a tree rooted at the president. In order to make the party fun for all attendees, the president does not want both an employee and his or her immediate supervisor to attend.

The personnel office has ranked each employee with a conviviality rating. Your task is to make up a guest list with the maximal conviviality rating of the guests.

Input

The first line contains an integer N that is a number of employees ($1 \le N \le 6000$). Employees are numbered by integers in a range from 1 to N, Each of the subsequent N lines contains the conviviality rating of the corresponding employee. Conviviality rating is an integer in a range from −128 to 127. After that the supervisor relation tree goes. Each line of the tree specification has the form

```
<L> <K>
```

which means that the K-th employee is an immediate supervisor of L-th employee. Input is ended with the line

```
0 0
```

Output

Output the maximal total conviviality rating of the guests.

Sample

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

Problem Author: Marat Bakirov
Problem Source: Ural State University Internal Contest October'2000 Students Session

1040. Airline Company

Time limit: 0.5 second
Memory limit: 64 MB

An airline company is a sponsor of the 80-th Anniversary celebration at the Ural State University. In return for it the company wants the University to help it. The company serves N airports and carries out flights between some of them. In order to simplify the work the flights are numbered with integers from 1 up to M. If there is a flight between two airports a plane flies in the both directions with the same flight number. There may be only one flight between any two airports. **One can fly between any pair of airports served by company using only its flights.**

The airline company understands that its planes may attract terrorists. In order to create difficulties for terrorists the company wants to number the flights in some special manner. If there are several flights that depart from one airport then the greatest common divisor of their flight numbers should be equal to one. The company turns to you for help (remember, this is a sponsor; you are to work properly).

You should write a program that finds a required numbering or informs that it is impossible to satisfy the requirements. If several numberings are possible, it is sufficient to find any one of them.

Input

The first line contains integers N and M ($2 \le N \le 50$; $1 \le M \le N \cdot (N - 1) / 2$). The next M lines contain information on flights. Each flight is determined by the numbers of the airports that it connects. The numbers of the airports are different integers in range from 1 to N.

Output

Output "YES" in the first line, if it is possible to find a required numbering, and "NO" otherwise. If the answer is "YES", the second line should contain a possible numbering of flights. The numbers are to be ordered as it is done in the input.

Sample

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

Problem Author: Dmitry Filimonenkov
Problem Source: Ural State University Internal Contest October'2000 Students Session

1041. Nikifor

Time limit: 1.0 second
Memory limit: 64 MB

Nikifor has decided to present the dean of the Department of Mathematics and Mechanics with a linearly independent vector system (you know, that we've just celebrated jubilees of the University and of the Department). A store sells M items of N-dimensional vectors. For i-th vector its price c_i is known. Nikifor wants to buy N linearly independent vectors paying for them minimal sum of money.

Write a program that would determine which vectors Nikifor should buy or would inform that it is impossible to satisfy his requirements.

Input

The first line contains integers M and N ($3 \le N \le 50$; $N \le M \le 2000$). The next M lines contain the vectors on sale. All of the coordinates are integers with an absolute value not exceeding 2000. The next M lines contain prices c_i, one number in each line. The prices are positive integers not exceeding 15 000.

Output

In the first line output the minimal amount of money that Nikifor is to pay or the number 0, if Nikifor's requirements cannot be satisfied in this store. If it is possible to make a purchase, then the next N lines should contain the numbers of the vectors that Nikifor should buy. If several sets of such numbers are possible, you should write one of them which is minimal according to the lexicographic order.

Sample

input	output
5 3 1 0 0 0 1 0 0 0 1 0 0 2 0 0 3 10 20 30 10 10	40 1 2 4

Problem Author: Dmitry Filimonenkov
Problem Source: Ural State University Internal Contest October'2000 Students Session

1042. Central Heating

Time limit: 1.0 second
Memory limit: 64 MB

Winter has come, but at the Ural State University heating is not turned on yet. There's one little problem: the University is heated only if all of the valves are opened. There are some technicians at the University. Each of them is responsible for one or more valves. There may be several technicians responsible for the same valve. When a technician gets an instruction to turn on the heating he goes round all of his valves and turns them. It means that if a valve was opened then he closes it, and if it was closed then he opens it. It is well known that every technician earns his money not in vain so **it's impossible to replace any technician by any combination of other technicians**.

Your task is to determine who of the technicians is to get an instruction "to turn on the heating" in order to heat all the Ural State University. Note that there are N technicians and N valves at the University.

Input

The first line contains an integer N ($1 \le N \le 250$). The next N lines contain lists of the valves in charge of each of the technicians. It means that the line number $i + 1$ contains numbers of the valves that the i-th technician is responsible for. Each list of valves is followed by –1.

Output

Output a list of technicians' numbers sorted in ascending order. If several lists are possible, output the shortest one. If it's impossible to turn on the heating at the University, output "No solution".

Sample

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

Problem Author: Evgeny Shtykov
Problem Source: Ural State University Internal Contest October'2000 Students Session

1043. Cover an Arc

Time limit: 1.0 second
Memory limit: 64 MB

A huge dancing-hall was constructed for the Ural State University's 80-th anniversary celebration. The size of the hall is 2000 × 2000 metres! The floor was made of square mirror plates with side equal to one metre. Then the walls were painted with an indelible paint. Unfortunately, in the end the painter flapped the brush and the beautiful mirror floor was stained with the paint. But not everything is lost yet! The stains can be covered with a carpet.

Nobody knows why, but the paint on the floor formed an arc of a circle (a centre of the circle lies inside the hall). The dean of the Department of Mathematics and Mechanics measured the coordinates of the arc's ends and of some other point of the arc (he is sure that this information is quite enough for any student of the Ural State University). The dean wants to cover the arc with a rectangular carpet. The sides of a carpet must go along the sides of the mirror plates (so, the corners of the carpet must have integer coordinates).

You should find the minimal area of such a carpet.

Input

Input consists of six integers. The first two lines contain the coordinates of the arc's ends. The coordinates of an inner point of the arc follow them. Absolute values of all coordinates don't exceed 1000. The points don't belong the same straight line. The arc lies inside the square $[-1000,1000]^2$.

Output

Output the minimal area of the carpet covering this arc.

Sample

input	output
476 612 487 615 478 616	66

Problem Author: Alexander Mironenko
Problem Source: Ural State University Internal Contest October'2000 Students Session

1044. Lucky Tickets. Easy!

Time limit: 2.0 second
Memory limit: 64 MB

The public transport administration of Ekaterinburg is anxious about the fact that passengers don't like to pay for passage doing their best to avoid the fee. All the measures that had been taken (hard currency premiums for all of the chiefs, increase in conductors' salaries, reduction of number of buses) were in vain. An advisor especially invited from the Ural State University says that personally he doesn't buy tickets because he rarely comes across the lucky ones (a ticket is lucky if the sum of its first half of digits is equal to the sum of the second half of digits). So, the way out is found — of course, tickets must be numbered in sequence, but the number of digits on a ticket may be changed. Say, if there were only two digits, there would have been ten lucky tickets (with numbers 00, 11, 22, 33, 44, 55, 66, 77, 88, 99). Maybe under the circumstances the ratio of the lucky tickets to the common ones is greater? And what if we take four digits? A huge work has brought the long-awaited result: in this case there will be 670 lucky tickets. But what to do if there are six or more digits?

So you are to save public transport of our city. Write a program that determines a number of lucky tickets for the given number of digits N.

Input

The only line contains an even integer N that is a number of digits in a ticket ($2 \le N \le 8$).

Output

Output a number of tickets such that the sum of the first half of digits is equal to the sum of the second half of digits.

Samples

input	output
4	670
2	10

Problem Author: Stanislav Vasilyev
Problem Source: Ural State University Internal Contest October'2000 Students Session

1045. Funny Game

Time limit: 1.0 second
Memory limit: 64 MB

There are several airports in one country, and there are flights between some of them. One can fly from any airport to any other, probably with some changes. For any pair of airports there exists only one sequence of flights that connects them.

Two terrorists play a game. They make moves in turn. Each move consists of the following operations. A player mines an airport, chooses a flight and flies away together with his colleague. After the take-off he actuates a radio-controlled fuse. As a result the airport that the terrorists have just left is destroyed, and all the flights to and from this airport are no longer possible. After the aircraft lands the other player makes his move, and so forth. One loses if one cannot make a move.

Given an initial list of flights and the number of an airport where the terrorists are at the start of the game, write a program which would determine who wins if the terrorists play a perfect game (each chooses the best move).

Input

The first line contains two integers: n and k separated with a space. Here n is the number of airports ($n \le 1000$) and k is the number of an airport, which is the starting point of the game ($1 \le k \le n$). The next $n - 1$ lines contain pairs of integers separated with a space. These integers are numbers of airports, connected with a flight (all the flights are symmetric and are mentioned only once). There are at most 20 flights to each airport.

Output

If the player who starts the game wins, the program should write “First player wins flying to airport L” where L is the number of an airport to which the players should fly first. If there are several such airports, the program should find one of them that has the minimal number. Otherwise the program should write “First player loses”.

Sample

input	output
4 3 3 2 3 1 1 4	First player wins flying to airport 2

Problem Author: Dmitry Filimonenkov
Problem Source: Ural State University collegiate programming contest (25.03.2000)

1046. Geometrical Dreams

Time limit: 0.5 second
Memory limit: 64 MB

There is a polygon $A_1A_2\ldots A_N$ (the vertices A_i are numbered in clockwise order). On each side A_iA_{i+1} an isosceles triangle $A_iM_iA_{i+1}$ is built on the outer side of the polygon ($M_iA_i = M_iA_{i+1}$). The angle $A_iM_iA_{i+1}$ is equal to α_i. Here we assume that $A_{N+1} = A_1$.

The set of angles α_i satisfies a condition that the sum of angles in any of its nonempty subsets is not aliquot to 360 degrees.

You are given N, coordinates of vertices M_i and angles α_i (measured in degrees). Write a program, which restores coordinates of the polygon vertices.

Input

The first line contains an integer N ($3 \le N \le 50$). The next N lines contain pairs of real numbers x_i, y_i which are coordinates of points M_i ($-100 \le x_i, y_i \le 100$). And the last N lines of the input consist of degree values of angles α_i. All real numbers in the input contain at most 2 digits after decimal point.

Output

Output N lines with points coordinates, i-th line should contain the coordinates of A_i. Coordinates must be accurate to 2 digits after decimal point. You may assume that solution always exists.

Sample

input	output
3 0 2 3 3 2 0 90 90 90	1 1 1 3 3 1

Problem Author: Dmitry Filimonenkov
Problem Source: Ural State University collegiate programming contest (25.03.2000)

1047. Simple Calculations

Time limit: 1.0 second
Memory limit: 64 MB

There is a sequence of $N + 2$ elements $a_0, a_1, \ldots, a_{N+1}$ ($1 \le N \le 3000$, $-2000 \le a_i \le 2000$). It is known that

$$a_i = (a_{i-1} + a_{i+1})/2 - c_i$$

for each $i = 1, 2, \ldots, N$.

You are given $a_0, a_{N+1}, c_1, \ldots, c_N$. Write a program which calculates a_1.

Input

The first line contains an integer N. The next two lines consist of numbers a_0 and a_{N+1} each having two digits after decimal point, and the next N lines contain numbers c_i (also with two digits after decimal point), one number per line.

Output

Output a_1 in the same format as a_0 and a_{N+1}.

Sample

input	output
1 50.50 25.50 10.15	27.85

Problem Author: Dmitry Filimonenkov
Problem Source: Ural State University collegiate programming contest (25.03.2000)

1048. Superlong Sums

Time limit: 2.0 second
Memory limit: 16 MB

The creators of a new programming language D++ have found out that whatever limit for SuperLongInt type they make, sometimes programmers need to operate even larger numbers. A limit of 1000 digits is so small… You have to find the sum of two numbers with maximal size of 1 000 000 digits.

Input

The first line contains a single integer N that is the length of the given integers ($1 \le N \le 1\ 000\ 000$). It is followed by these integers written in columns. That is, the next N lines contain two digits each, divided by a space. Each of the two given integers is not less than 1, and the length of their sum does not exceed N. The integers may contain leading zeroes.

Output

Output exactly N digits in a single line representing the sum of these two integers.

Sample

input	output
4 0 4 4 2 6 8 3 7	4750

Problem Author: Stanislav Vasilyev and Alexander Klepinin
Problem Source: Ural State University collegiate programming contest (25.03.2000)

1049. Brave Balloonists

Time limit: 2.0 second
Memory limit: 64 MB

Ten mathematicians are flying on a balloon over the Pacific ocean. When they are crossing the equator they decide to celebrate this event and open a bottle of champagne. Unfortunately, the cork makes a hole in the balloon. Hydrogen is leaking out and the balloon is descending now. Soon it will fall into the ocean and all the balloonists will be eaten by hungry sharks.

But not everything is lost yet. One of the balloonists can sacrifice himself jumping out, so that his friends would live a little longer. Only one problem still exists: who is the one to get out. There is a fair way to solve this problem. First, each of them writes an integer a_i not less than 1 and not more than 10000. Then they calculate the magic number N that is the number of positive divisors of the product $a_1*a_2*\ldots*a_{10}$. For example, the number of positive integer divisors of 6 is 4 (they are 1,2,3,6). The hero (a mathematician who will be thrown out) is determined according to the last digit of N. Your task is to find this digit.

Input

Input contains ten integer numbers (each number is in separate line).

Output

Output a single digit from 0 to 9 — the last digit of N.

Sample

input	output
1 2 6 1 3 1 1 1 1 1	9

Problem Author: Stanislav Vasilyev
Problem Source: Ural State University collegiate programming contest (25.03.2000)

1050. Preparing an Article

Time limit: 1.0 second
Memory limit: 64 MB

TeX is the leading typesetting system for mathematics, science, and engineering and has been adopted as standard by the American Mathematical Society. LaTeX was developed later by Leslie Lamport. It is based on TeX and provides a set of higher level commands for production of complex documents. In TeX or LaTeX, any text editor program may be used to enter and modify the input text.

The source text contains the actual text as well as formatting commands beginning with \. Commands are delimited by any non-alphabetic character. One example of beautification by TeX is that it uses `` (two left-single-quotes) and '' (two right-single-quotes) to delimit quotations, rather than the mundane " (one double quote) which is provided by most keyboards. Keyboards typically do not have an oriented double-quote, but they do have a left-single-quote (`) and right-single-quote ('). TeX lets the user type two left-single-quotes (``) to create a left-double-quote and two right-single-quotes ('') to create a right-double-quote.

Now, you have a text only file containing at most 250 lines at most 80 symbols each, as source or input, and you want to use TeX to beautify it. Rather than doing everything by hand, as the first step of automation you want to convert the quotes into the TeX format by using a program. This program will convert the text with double-quotes (") into an identical text except that double quotes have been replaced by the two-character sequences required by TeX for delimiting quotations with oriented double-quotes. The double-quote (") characters should be replaced appropriately by proper double single quotes depending on whether it is an opening or closing quotation mark. Question of nested quotations does not arise. The first " must be replaced by ``, the next by '', the next by ``, the next by '', and so on. An opening double quote must have its closing quote in the same paragraph. If a match is not found in the same paragraph for an opening quote, this quote has to be deleted. Paragraph ends in the source text are marked either by at least one blank line, or a \par command or both. Your program must also be careful about the \" command which is used to produce umlaut or dieresis (\"e leads to ë). These are to be left untouched.

Input

Input will consist of several lines of text containing a number of double quotes ("), as well as some TeX commands. End of input will be marked by an \endinput command.

Output

Output will be an exact replica of the input, except the double quotes are to be modified according to the rules described above.

Sample

input	output
There is no "q in this sentence. \par "Talk child," said the unicorn. She s\"aid, "\thinspace `Enough!', he said." \endinput	There is no q in this sentence. \par ``Talk child,'' said the unicorn.  She s\"aid, ``\thinspace `Enough!', he said.'' \endinput

Notes

Double-quote (") has ASCII code 34,
left-single-quote (`) has ASCII code 96,
right-single-quote (') has ASCII code 39.

Problem Author: Alexander Halperin
Problem Source: Ural State University collegiate programming contest (25.03.2000)

1051. Simple Game on a Grid

Time limit: 1.0 second
Memory limit: 64 MB

There is an infinite grid and an $M \times N$ rectangle of stones on it ($1 \le M, N \le 10000$). The stones are located in the knots of the grid.

A following game for a single player is being played. One stone can jump over another along a vertical or a horizontal line. A stone which had been overjumped is taken away. The purpose of the game is to minimize number of stones on a grid.

Given a pair of numbers M and N separated with one space you are to write a program which should determine a minimal number of the stones left on the grid.

Input

Numbers M and N separated by space.

Output

The minimal number of the stones left on the grid.

Sample

input	output
3 4	2

Problem Author: Stanislav Vasilyev
Problem Source: Ural State University collegiate programming contest (25.03.2000)

1052. Rabbit Hunt

Time limit: 1.0 second
Memory limit: 64 MB

A good hunter kills two rabbits with one shot. Of course, it can be easily done since for any two points we can always draw a line containing the both. But killing three or more rabbits in one shot is much more difficult task. To be the best hunter in the world one should be able to kill the maximal possible number of rabbits. Assume that rabbit is a point on the plane with integer x and y coordinates. Having a set of rabbits you are to find the largest number of rabbits that can be killed with single shot, i.e. maximum number of points lying exactly on the same line. No two rabbits sit at one point.

Input

An input contains an integer N ($3 \le N \le 200$) specifying the number of rabbits. Each of the next N lines in the input contains the x coordinate and the y coordinate (in this order) separated by a space ($-2000 \le x, y \le 2000$).

Output

The output contains the maximal number of rabbits situated in one line.

Sample

input	output
6 7 122 8 139 9 156 10 173 11 190 -100 1	5

Problem Author: Stanislav Vasilyev
Problem Source: Ural State University collegiate programming contest (25.03.2000)

1053. Pinocchio

Time limit: 1.0 second
Memory limit: 64 MB

Mister Geppetto got the commission to make Pinocchio. Client expressed a wish to be unknown and he left material and insisted on finding Pinocchio's nose length as a result of performing the following algorithm:

1. There's a set of N numbered blanks with integer lengths.
2. If the set consists of only one blank, then it's length can be admitted as the length of Pinocchio's nose.
3. Let's choose some 2 blanks
    a. If lengths of the blanks coincide, then one of the blanks is eliminated from the set and algorithm goes back to point 2 to be repeated.
    b. If lengths of the blanks are different, then the piece of the long blank is sawed off and its length must be equal to the length of the other blank. Then the algorithm is repeated from point 2.

Example. There are three blanks in a set with lengths: 2, 3, 4. Then the change of the blank lengths can be shown in the following table. As a result Pinocchio will get the nose with length of 1.

Length of the first blank	Length of the second blank	Length of the third blank	Comments
2	3	4	Initial blank lengths
2	1	4	Sawing off the second blank
2	1	3	Sawing off the third blank
2	1	2	Sawing off the third blank
1	1	2	Sawing off the first blank
-	1	2	The first blank is eliminated
-	1	1	Sawing off the third blank
-	-	1	The second blank is eliminated

Input

The first line contains an integer N ($1 \le N \le 1000$). The other N successive lines contain integers $L_1, L_2, \ldots, L_N$ that are initial blank lengths ($1 \le L_i \le 2^{31} - 1$).

Output

Output either Pinocchio nose length, or the word “IMPOSSIBLE” if the nose length is undefined.

Sample

input	output
3 2 3 4	1

Problem Source: Rybinsk State Avia Academy

1054. Tower of Hanoi

Time limit: 1.0 second
Memory limit: 64 MB

Background

“Tower of Hanoi” puzzle is well known. There are 3 pegs with numbers: #1, #2 and #3. N disks of different diameters are set on the first peg in the following order: the lower disk is set, the larger diameter it has. Your aim is to move all disks onto the second peg using the third peg as an auxiliary one. Following the rules within a move it's allowed to replace only one uppermost disk. Besides it's forbidden to put the disk of the bigger diameter onto the disk of the smaller one.

Distribution of disks on the pegs during the game must be assigned as sequence D (D_i is equal to the number of peg where the disk #i is placed on). For instance, sequence D = (3, 3, 1) means that the first and the second disks are set on the third peg and the third disk is on the first peg.

Example. Let's assume that three disks numbered in ascending order of diameters are set on the first peg. Then the movement of the disks can be depicted in the following table:

Step	Peg #1 disks	Peg #2 disks	Peg #3 disks	sequence D
0	1, 2, 3	-	-	1, 1, 1
1	2, 3	1	-	2, 1, 1
2	3	1	2	2, 3, 1
3	3	-	1, 2	3, 3, 1
4	-	3	1, 2	3, 3, 2
5	1	3	2	1, 3, 2
6	1	2, 3	-	1, 2, 2
7	-	1, 2, 3	-	2, 2, 2

Problem

Your aim is to determine (using the given sequence D) the number of moves from the beginning of the game to the given position on condition of performing the optimal algorithm.

Optimal algorithm can be specified with the following recursive procedure.

```
procedure Hanoi(N, From, To_, Temp : integer);
begin
  if N > 0 then
  begin
    Hanoi(N - 1, From, Temp, To_);
    Writeln(N, From, To_);
    Hanoi(N - 1, Temp, To_, From)
  end
end;
```

For example, to move 5 disks it's enough to call `Hanoi(5, 1, 2, 3)`.

Input

The first line contains an integer N that is the number of disks ($1 \le N \le 31$). The other N successive lines contain integers $D_1, \ldots, D_N$ ($1 \le D_i \le 3$).

Output

Output the number of moves from the beginning of the game to the given position. If the given position cannot be achieved performing the optimal algorithm, output -1.

The answer will always be unequivocal, because positions are never repeated during the performing the optimal algorithm .

Samples

input	output
3 3 3 1	3
1 3	-1

Problem Source: Rybinsk State Avia Academy

1055. Combinations

Time limit: 1.0 second
Memory limit: 64 MB

As you have known MMM corporation lab researches the matter of haricot proportions in soup For every day. The ladle is placed down into the soup pan. This ladle holds exactly M haricot seeds of N got into the pan. All the seeds are of different size.

Experimenters calculate the quantity of possible methods to proportion M seeds in the pan with the formula: $C = N! / (M! \cdot (N - M)!)$. The main feature of these experiments is the quantity of different prime divisors of number C.

Lest money would be spent for programmer, MMM corporation board decided to make necessary estimating during the ICPC quarterfinal in Rybinsk. Thus, your aim is to find this quantity.

Input

The only line contains integers N and M that are the number of haricot seeds in the pan and the capacity of the ladle ($1 \le M < N \le 50000$).

Output

Output the quantity of different prime divisors of number C.

Sample

input	output
5 3	2

Notes

In the example $C = 5! / (3! \cdot 2!) = 120 / (6 \cdot 2) = 10 = 2 \cdot 5$.

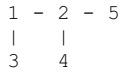
Problem Source: Rybinsk State Avia Academy

1056. Centers of the Net

Time limit: 2.0 second
Memory limit: 64 MB

Background

Computer net is created by consecutive computer plug-up to one that has already been connected to the net. Each new computer gets an ordinal number, but the protocol contains the number of its parent computer in the net. Thus, protocol consists of several numbers; the first of them is always 1, because the second computer can only be connected to the first one, the second number is 1 or 2 and so forth. The total quantity of numbers in the protocol is $N - 1$ (N is a total number of computers). For instance, protocol 1, 1, 2, 2 corresponds to the following net:



The distance between the computers is the quantity of mutual connections (between each other) in chain. Thus, in example mentioned above the distance between computers #4 and #5 is 2, and between #3 and #5 is 3.

Definition. Let the center of the net be the computer which has a minimal distance to the most remote computer. In the shown example computers #1 and #2 are the centers of the net.

Problem

Your task is to find all the centers using the set protocol.

Input

The first line contains an integer N, the quantity of computers ($2 \le N \le 10000$). Successive $N - 1$ lines contain protocol.

Output

Output the ordinal numbers of the determined centers of the net in ascending order.

Sample

input	output
5 1 1 2 2	1 2

Problem Source: Rybinsk State Avia Academy

1057. Amount of Degrees

Time limit: 1.0 second
Memory limit: 64 MB

Create a code to determine the amount of integers, lying in the set [X; Y] and being a sum of exactly K different integer degrees of the integer B.

Example. Let $X = 15$, $Y = 20$, $K = 2$, $B = 2$. By this example three integers are the sum of exactly two integer degrees of number 2:

$$17 = 2^4 + 2^0,$$
$$18 = 2^4 + 2^1,$$
$$20 = 2^4 + 2^2.$$

Input

The first line contains integers X and Y ($1 \le X \le Y \le 2^{31} - 1$). The next two lines contain integers K and B ($1 \le K \le 20$; $2 \le B \le 10$).

Output

Output the amount of integers, lying between X and Y, being a sum of exactly K different integer degrees of B.

Sample

input	output
15 20 2 2	3

Problem Source: Rybinsk State Avia Academy

1058. Chocolate

Time limit: 1.0 second
Memory limit: 64 MB

Fishburg confectionery factory produces **convex** polygon-shaped chocolates. Little Boy and Karlsson bought one and wanted to break it into two fragments. The areas of the fragments must be equal. Create a code to find out the length of breakage-line of a minimal length using the given chocolate form.

Input

The first line contains an integer N that is the number of polygon vertices ($4 \le N \le 50$). The other N lines contain coordinates of the vertices in the counter-clockwise order. Coordinates are real numbers from −100 to 100 and given with at most 3 digits after decimal point.

Output

Output the minimal length of breakage-line with accuracy to 0.0001.

Sample

input	output
4 0 0 4 0 4 3 0 3	3

Problem Source: Rybinsk State Avia Academy

1059. Expression

Time limit: 1.0 second
Memory limit: 64 MB

Petr got an assignment to calculate the value of polynomial P of degree N by its coefficient values and argument x.

Petr has a calculator working in a mode of reverse Polish notation. His calculator is able to add and multiply numbers of any length (capacity). Help Petr to set correct and at the same time the shortest sequence of operands and signs of operations in the order of entering in calculator.

Expression written in a mode of reverse Polish notation consists of operands and signs of operations; the sign of operation is preceded by operands. The brackets in reverse Polish notation are eliminated. The algorithm to compute such an expression is the following:

1. If the expression consists of only one operand, then the value of the expression is the value of this operand.
2. Otherwise:
    - seek for the first operation sign leftmost in the expression
    - process the operation with those two operands that stand to the left from this sign
    - write the result of this operation instead of the sign and the operands
    - repeat the steps 1-2
3. If the rules 1-2 are unacceptable, then the initial expression was incorrect.

For example, the reverse Polish notation “a b c d + * e + /” is the equivalence of ordinary expression “a / (b * (c + d) + e)”.

Input

The only line contains an integer N ($1 \le N \le 1000$).

Output

Output a sequence of minimal length corresponding to polynomial of degree N in reverse Polish notation. The polynomial has a form $a_0x^N+a_1x^{N-1} + \ldots a_N$. This sequence can include only signs of operations “+” and “*”, capital Latin letter “X” and non-negative integers that correspond the coefficients numbers (the integer i denotes the coefficient a_i).

All elements (signs of operations, characters “X” and numbers of coefficients) should be written by one in a line.

Sample

input	output
1	0 X * 1 +

Problem Source: Rybinsk State Avia Academy

1060. Flip Game

Time limit: 2.0 second
Memory limit: 64 MB

Flip game is played on a rectangular 4 × 4 field with two-sided pieces placed on each of its 16 squares. One side of each piece is white and the other one is black and each piece is lying either it's black or white side up. Each round you flip 3 to 5 pieces, thus changing the color of their upper side from black to white and vice versa. The pieces to be flipped are chosen every round according to the following rules:

1. Choose any one of the 16 pieces.
2. Flip the chosen piece and also all adjacent pieces to the left, to the right, to the top, and to the bottom of the chosen piece (if there are any).

Consider the following position as an example:

```
bwbw
wwww
bbwb
bwwb
```

Here “b” denotes pieces lying their black side up and “w” denotes pieces lying their white side up. If we choose to flip the 1st piece from the 3rd row (this choice is shown at the picture), then the field will become:

```
bwbw
bwww
wwwb
wwwb
```

The goal of the game is to flip either all pieces white side up or all pieces black side up. You are to write a program that will search for the minimum number of rounds needed to achieve this goal.

Input

The input consists of 4 lines with 4 characters “w” or “b” each that denote game field position.

Output

Output the minimum number of rounds needed to achieve the goal of the game from the given position. If the goal is initially achieved, then write 0. If it's impossible to achieve the goal, then write the word “Impossible”.

Samples

input	output
bwbw wwww bbwb bwwb	Impossible
bwwb bbwb bwwb bwww	4

Problem Source: 2000-2001 ACM Northeastern European Regional Programming Contest

1061. Buffer Manager

Time limit: 0.5 second
Memory limit: 64 MB

Data blocks being read by DBMS from the hard drive are stored in the main memory in a fixed number of pre-allocated **buffers**. Each buffer can hold one data block. Each buffer can be either **free** (does not contain any useful information) or **occupied** by some data. When DBMS is going to read data block from the hard drive it has to decide which buffer to use for data storing. If there are any free buffers, then one of them is used for that purpose. If there are no free buffers, then one of the occupied buffers has to be flushed to become free, unless it was **locked** by some part of DBMS.

The choice of the buffer to flush is critical to DBMS performance. A lot of different choice algorithms were developed. Your DBMS is going to implement the Advanced Buffer Management algorithm which reads a number of consecutive data blocks from the hard drive into consecutive memory buffers.

Buffers are numbered from 1 to N, where N is a total number of buffers. Each buffer can be in any one of the following states: free, occupied or locked. Each occupied buffer is assigned an integer from 1 to 9 – the **worthiness** of the currently stored information in that buffer. The worthiness of free buffers is considered to be zero. Locked buffers cannot be neither used nor flushed and their worthiness is undefined.

Having received the request to read K data blocks from the hard drive, Buffer Manager has to choose K consecutive (i.e. numbered from L to $L + K - 1$) non-locked buffers with minimal possible sum of their worthiness, or to report that it is impossible. The latter can happen if some buffers are locked or total number of buffers is less than K.

Your task is to write a program that models the processing of one request to Buffer Manager using the above algorithm.

Input

The first line contains integers N and K ($1 \le N \le 100000$; $1 \le K \le 10000$).

Starting from the second line there is a description of a buffers' state. The state of each buffer is represented by a single character:

- `0` – when the corresponding buffer is free.
- `1` – when the corresponding buffer is occupied and has worthiness of 1.
- `2` – when the corresponding buffer is occupied and has worthiness of 2.
- ...
- `9` – when the corresponding buffer is occupied and has worthiness of 9.
- `*` – when the corresponding buffer is locked.

Those characters are situated on the consecutive lines grouped by 80 characters per line without any spaces. Thus, each line starting from the second one contains exactly 80 characters with a possible exception for the last line.

Output

Output the integer L that is the buffer number where first of the K blocks from the hard drive shall be read to ensure the minimal possible total worthiness of the blocks that have to be flushed. If there are more than one such value for L, then output the smallest one.

Output a single number 0 if it's impossible to find K consecutive non-locked buffers.

Samples

input	output
100 53 21657452160918534777558003938597858072075231699543 41**7363*9*94664808*4777717089 09825185827659480548	0
100 10 21657452160918534777558003938597858072075231699543 41**7363*9*94664808*4777717089 09825185827659480548	36

Problem Source: 2000-2001 ACM Northeastern European Regional Programming Contest

1062. Triathlon

Time limit: 2.0 second
Memory limit: 64 MB

Triathlon is an athletic contest consisting of three consecutive sections that should be completed as fast as possible as a whole. The first section is swimming, the second section is riding bicycle and the third one is running.

The speed of each contestant in all three sections is known. The judge can choose the length of each section arbitrarily provided that no section has zero length. As a result sometimes she could choose their lengths in such a way that some particular contestant would win the competition.

Input

The first line contains an integer N, denoting the number of contestants ($1 \le N \le 100$). Then N lines follow, each line contains integers V_i, U_i and W_i, denoting the speed of i^{th} contestant in each section ($1 \le V_i, U_i, W_i \le 10000$)

Output

For every contestant output one line, that contains word “Yes” if the judge could choose the lengths of the sections in such a way that this particular contestant would win (i.e. she is the only one who would come first), or word “No” if this is impossible.

Sample

input	output
9 10 2 6 10 7 3 5 6 7 3 2 7 6 2 6 3 5 7 8 4 6 10 4 2 1 8 7	Yes Yes Yes No No No Yes No Yes

Problem Source: 2000-2001 ACM Northeastern European Regional Programming Contest

1063. Domino Puzzle

Time limit: 2.0 second
Memory limit: 64 MB

Background

***Dominoes**, game played with small, rectangular blocks of wood or other material, each identified by a number of dots, or pips, on its face. The blocks usually are called bones, dominoes, or pieces and sometimes men, stones, or even cards. The face of each piece is divided, by a line or ridge, into two squares, each of which is marked as would be a pair of dice... The principle in nearly all modern dominoes games is to match one end of a piece to another that is identically or reciprocally numbered.*

ENCYCLOPÆDIA BRITANNICA

Problem

Consider an arbitrary set of domino pieces where each piece is marked with two digits from 1 to 6. Some sets can be completely laid out in a row matching one end of a piece to another that is identically numbered, while others cannot. For example, the set consisting of 5 pieces: (1, 5), (1, 6), (5, 5) and (2, 4) twice, cannot be laid out in a row. However, if we add (2, 5) piece to the above set we could lay out the resulting set in the following row:

However, we are interested in a row having the smallest sum of digits on its pieces. In our example, instead of the piece (2, 5) with a sum of 7, we could add two pieces (1, 2) with a total sum of 6 to lay out the following row:

Your task is to write a program that for a given domino set will find an additional (possibly empty) set with the smallest possible sum of digits, so that a row could be laid out with both sets combined.

Input

The first line contains a single integer N representing the total number of pieces in the domino set ($2 \le N \le 100$). The following N lines describe pieces. Each piece is represented on a separate line in a form of two digits from 1 to 6 separated by a space. The digits of a piece can be written in any order.

Output

On the first line write the smallest sum of digits of the additional set or 0 if that set is empty. On the second line write the total number of pieces in the additional set or 0 if that set is empty. Then write the pieces of the additional set in the same format as in input.

If there are a number of additional sets with the same smallest sum of digits exist then write any one of them.

Samples

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

Problem Source: 2000-2001 ACM Northeastern European Regional Programming Contest

1064. Binary Search

Time limit: 1.0 second
Memory limit: 64 MB

The program fragment below performs binary search of an integer number in an array that is sorted in a nondescending order:

Pascal

```
procedure BinarySearch(x: integer; N: integer; A: array of integer);
var
    p, q, i, L: integer;
begin
    p := 0;     { Left border of the search  }
    q := N - 1; { Right border of the search }
    L := 0;     { Comparison counter         }
    while p <= q do begin
        i := (p + q) div 2;
        inc(L);
        if A[i] = x then begin
            writeln('Found item i = ', i, ' in L = ', L, ' comparisons');
            exit
        end;
        if x < A[i] then
            q := i - 1
        else
            p := i + 1
    end
end;
```

C++

```
void BinarySearch(int x, int N, int* A)
{
    int p = 0;     // Left border of the search
    int q = N - 1; // Right border of the search
    int L = 0;     // Comparison counter
    while (p <= q) {
        int i = (p + q) / 2;
        ++L;
        if (A[i] == x) {
            printf("Found item i = %d in L = %d comparisons\n", i, L);
            return;
        }
        if (x < A[i])
            q = i - 1;
        else
            p = i + 1;
    }
}
```

Python

```
def BinarySearch(x: int, N: int, A: list):
    p = 0      # Left border of the search
    q = N - 1  # Right border of the search
    L = 0      # Comparison counter
    while p <= q:
        i = (p + q) // 2
        L += 1
        if A[i] == x:
```

```
        print('Found item i =', i, 'in L =', L, 'comparisons')
        return

    if x < A[i]:
        q = i - 1
    else:
        p = i + 1
```

Before `BinarySearch` was called, N was set to some integer number from 1 to 10000 inclusive and array A was filled with a nondescending integer sequence of length N.

It is known that the procedure has terminated with the message "`Found item i = XXX in L = XXX comparisons`" with some known values of i and L.

Your task is to write a program that finds all possible values of N that could lead to such message. However, the number of possible values of N can be quite big. Thus, you are asked to group all consecutive values of N into intervals and write down only first and last value in each interval.

Input

A single line contains integers i and L ($0 \le i \le 9999$; $1 \le L \le 14$).

Output

On the first line of the output write the single integer number K representing the total number of intervals for possible values of N. Then K lines shall follow listing those intervals in an ascending order.

Each line shall contain two integers A_i and B_i ($A_i \le B_i$) separated by a space, representing first and last value of the interval.

If there are no possible values of N exist, then the output shall contain the single 0.

Samples

input	output
9000 2	0
10 3	4 12 12 17 18 29 30 87 94

Problem Source: 2000-2001 ACM Northeastern European Regional Programming Contest

1065. Frontier

Time limit: 2.0 second
Memory limit: 64 MB

Lilliputian frontier is a convex polygon with non-zero area. The vertices of this polygon are guard towers, which are connected by straight lines. This frontier is too long and expensive for Lilliputia to maintain; therefore the Lilliputian government has decided to revise it to make it shorter. However, they don't want to build new guard towers, but to use existing ones as a part of a new frontier.

Each day frontier guards inspect the frontier. They go from one guard tower to the next one, traversing the frontier clockwise. Guard towers are numbered from 1 to N according to this inspection order. Frontier revision should not change this way of inspection and the area of Lilliputia shall remain non-zero.

For example, the frontier that is shown on the picture (axes are in kilometer scale) is traversed by 1 - 2 - 3 - 4 - 5 - 1 route, which is 57.89 kilometers long. To make the frontier as short as possible Lilliputia should revise it so that the frontier is traversed by 2 - 3 - 4 - 2 route, thus reducing its length to 27.31 kilometers.

However, Lilliputia has a number of historical monuments which are its major pride. The historical monuments shall be kept inside Lilliputia at any cost and they should not end up on the frontier. So, the task is to design the shortest frontier that will preserve all historical monuments inside Lilliputia.

On the sample picture two historical monuments marked “A” and “B” are shown. The desire to keep them inside Lilliputia will lead to the shortest frontier with a traverse path 1 - 2 - 3 - 4 - 1 having 51.78 kilometers in length.

Input

The first line contains integers N and M that are a total number of guard towers on the Lilliputian frontier and a total number of historical monuments that are situated inside Lilliputia ($3 \le N \le 50$; $0 \le M \le 1000$).

Next N lines contain guard towers' coordinates in a clockwise order followed by M lines that contain historical monuments' coordinates. All coordinates are represented as two integers (for X and Y correspondingly) separated by a space. Coordinates are given in a kilometer scale and each coordinate does not exceed 10000 by an absolute value. All guard towers are located at distinct points.

Output

Output the minimal possible length of the Lilliputian frontier (in kilometers) accurate to two digits to the right of the decimal point.

Samples

input	output
5 0 8 9 0 -7 -8 -7 -8 1 -8 9	27.31
5 2 8 9 0 -7 -8 -7 -8 1 -8 9 -4 -3 -1 -5	51.78

Problem Source: 2000-2001 ACM Northeastern European Regional Programming Contest

1066. Garland

Time limit: 2.0 second
Memory limit: 64 MB

The New Year garland consists of N lamps attached to a common wire that hangs down on the ends to which outermost lamps are affixed. The wire sags under the weight of lamp in a particular way: each lamp is hanging at the height that is 1 millimeter lower than the average height of the two adjacent lamps.

The leftmost lamp is hanging at the height of A millimeters above the ground. You have to determine the lowest height B of the rightmost lamp so that no lamp in the garland lies on the ground though some of them may touch the ground.

You shall neglect the lamp's size in this problem. By numbering the lamps with integers from 1 to N and denoting the i^{th} lamp height in millimeters as H_i we derive the following equations:

- $H_1 = A$
- $H_i = (H_{i-1} + H_{i+1})/2 - 1$, for all $1 < i < N$
- $H_N = B$
- $H_i \geq 0$, for all $1 \leq i \leq N$

The sample garland with 8 lamps that is shown on the picture has $A = 15$ and $B = 9.75$.

Input

The input consists of a single line with two numbers N and A separated by a space. N ($3 \leq N \leq 1000$) is an integer representing the number of lamps in the garland, A is a real number representing the height of the leftmost lamp above the ground in millimeters ($10 \leq A \leq 1000$).

Output

Output the single real number B accurate to two digits to the right of the decimal point representing the lowest possible height of the rightmost lamp.

Samples

input	output
8 15	9.75
692 532.81	446113.34

Problem Source: 2000-2001 ACM Northeastern European Regional Programming Contest

1067. Disk Tree

Time limit: 2.0 second
Memory limit: 64 MB

Hacker Bill has accidentally lost all the information from his workstation's hard drive and he has no backup copies of its contents. He does not regret for the loss of the files themselves, but for the very nice and convenient directory structure that he had created and cherished during years of work.

Fortunately, Bill has several copies of directory listings from his hard drive. Using those listings he was able to recover full paths (like "WINNT\SYSTEM32\CERTSRV\CERTCO~1\X86") for some directories. He put all of them in a file by writing each path he has found on a separate line.

Your task is to write a program that will help Bill to restore his state of the art directory structure by providing nicely formatted directory tree.

Input

The first line contains single integer N that denotes a total number of distinct directory paths ($1 \le N \le 500$). Then N lines with directory paths follow. Each directory path occupies a single line and does not contain any spaces, including leading or trailing ones. No path exceeds 80 characters. Each path is listed once and consists of a number of directory names separated by a back slash ("\").

Each directory name consists of 1 to 8 uppercase letters, numbers, or the special characters from the following list: exclamation mark, number sign, dollar sign, percent sign, ampersand, apostrophe, opening and closing parenthesis, hyphen sign, commercial at, circumflex accent, underscore, grave accent, opening and closing curly bracket, and tilde ("`!#$%&'()-@^_`{}~`").

Output

Output the formatted directory tree. Each directory name shall be listed on its own line preceded by a number of spaces that indicate its depth in the directory hierarchy. The subdirectories shall be listed in lexicographic order immediately after their parent directories preceded by one more space than their parent directory. Top level directories shall have no spaces printed before their names and shall be listed in lexicographic order. See sample below for clarification of the output format.

Sample

input	output

7
WINNT\SYSTEM32\CONFIG
GAMES
WINNT\DRIVERS
HOME
WIN\SOFT
GAMES\DRIVERS
WINNT\SYSTEM32\CERTSRV\CERTCO~1\X86
	GAMES
	 DRIVERS
	HOME
	WIN
	 SOFT
	WINNT
	 DRIVERS
	 SYSTEM32
	  CERTSRV
	  CERTCO~1
	  X86
	  CONFIG

Problem Source: 2000-2001 ACM Northeastern European Regional Programming Contest

1068. Sum

Time limit: 2.0 second
Memory limit: 64 MB

Your task is to find the sum of all integer numbers lying between 1 and *N* inclusive.

Input

The input consists of a single integer *N* that is not greater than 10000 by it's absolute value.

Output

Write a single integer number that is the sum of all integer numbers lying between 1 and *N* inclusive.

Sample

input	output
-3	-5

Problem Source: 2000-2001 ACM Northeastern European Regional Programming Contest (test tour)

1069. Prufer Code

Time limit: 0.25 second
Memory limit: 8 MB

A tree (i.e. a connected graph without cycles) with vertices is given ($N \geq 2$). Vertices of the tree are numbered by the integers 1,…, N. A Prufer code for the tree is built as follows: a leaf (a vertex that is incident to the only edge) with a minimal number is taken. Then this vertex and the incident edge are removed from the graph, and the number of the vertex that was adjacent to the leaf is written down. In the obtained graph once again a leaf with a minimal number is taken, removed and this procedure is repeated until the only vertex is left. It is clear that the only vertex left is the vertex with the number N. The written down set of integers ($N-1$ numbers, each in a range from 1 to N) is called *a Prufer code* of the graph.

Your task is, given a Prufer code, to reconstruct a tree, i.e. to find out the adjacency lists for every vertex in the graph.

You may assume that $2 \leq N \leq 7500$

Input

A set of numbers corresponding to a Prufer code of some tree. The numbers are separated with a spaces and/or line breaks.

Output

Adjacency lists for each vertex. Format: a vertex number, colon, numbers of adjacent vertices separated with a space. The vertices inside lists and lists itself should be sorted by vertex number in an ascending order (look at sample output).

Sample

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

Problem Author: Magaz Asanov
Problem Source: Ural State Univerisity Personal Contest Online February'2001 Students Session

1070. Local Time

Time limit: 1.0 second
Memory limit: 64 MB

Soon the USU team will go to Vancouver to participate in the final of the ACM International Collegiate Programming Contest. They will be to take four different planes (three changes on the way)!

By the way, our team plans to return from Vancouver, so the two-way tickets are bought. The departure time (local time of the airport of departure) and the time of the arrival (local time of the destination airport) are printed on the tickets.

For example, the departure at 15.42 and the arrival at 16.23, and a return flight departs at 08.10 and arrives at 17.51.

Your task is to help to our team to find out how much does the time of the first airport differs from the one of the second. It is known that time in different airports differs by an integer amount of hours. The time of flights there and back may differ from each other not more than by 10 minutes.

The duration of a flight doesn't exceed 6 hours. The difference between airport local times is not greater than 5 hours.

Input

There are two lines, each of them contains two numbers. The first line consists of the departure time and the arrival time of the flight there, the second one — the departure and the arrival times of the back flight. Numbers in the lines are separated with a space, an amount of minutes is separated from an amount of hours with a point.

Output

Your program should write a non-negative integer (without extra zeroes) that corresponds to the difference in time between the two airports.

Sample

input	output
23.42 00.39 08.10 17.11	4

Problem Author: Magaz Asanov & Stanislav Vasilyev
Problem Source: Ural State Univerisity Personal Contest Online February'2001 Students Session

1071. Nikifor 2

Time limit: 1.0 second
Memory limit: 64 MB

Nikifor has a number x. He doesn't need it. He needs a number y. Nikifor tries to obtain the required number by erasing some digits from x. But he is not lucky in the meanwhile. May be he is to choose an appropriate number system?

Write a program that reads numbers x and y, and determines a minimal radix of a number system that it is possible to obtain in it the number y from x by erasing some digits. If it is impossible, your program should write to an output a message "No solution".

Input

The only line contains integers x and y ($1 \le y < x \le 1\ 000\ 000$), separated with a space.

Output

Output either the message "No solution", if there is no appropriate number system, or an integer, not less than 2, that is an answer in the problem.

Sample

input	output
127 16	3

Problem Author: Dmitry Filimonenkov
Problem Source: Ural State Univerisity Personal Contest Online February'2001 Students Session

1072. Routing

Time limit: 1.0 second
Memory limit: 64 MB

There is a TCP/IP net of several computers. It means that:

1. Each computer has one or more net interfaces.
2. Each interface is identified by its IP-address and a subnet mask — these are two four-byte numbers with a point after each byte. A subnet mask has a binary representation as follows: there are k 1-bits, then — m 0-bits, k+m=8*4=32 (e.g., 212.220.35.77 — is an IP-address and 255.255.255.128 — is a subnet mask).
3. Two computers belong to the same subnet, if and only if (IP_1 AND $NetMask_1$) = (IP_2 AND $NetMask_2$), where IP_i and $NetMask_i$ — are an IP-address and subnet mask of i-th computer, AND — is bitwise.
4. A packet is transmitted between two computers of one subnet directly.
5. If two computers belong to different subnets, a packet is to be transmitted via some other computers. The packet can pass from one subnet to another only on computer that has both subnets interfaces.

Your task is to find the shortest way of a packet between two given computers.

Input

The first line contains a number N — an amount of computers in the net, then go N sections, describing interfaces of each computer. There is a number K in the first line of a section — that is an amount of interfaces of the computer, then go K lines — descriptions of the interfaces, i.e. its IP-address and a subnet mask. The last line of an input contains two integers — the numbers of the computers that you are to find a way between them.

You may assume that $2 \le N \le 90$ and $K \le 5$.

Output

The word “Yes” if the route exists, then in the next line the computer numbers passed by the packet, separated with a space. The word “No” otherwise.

Sample

input	output
6 2 10.0.0.1 255.0.0.0 192.168.0.1 255.255.255.0 1 10.0.0.2 255.0.0.0 3 192.168.0.2 255.255.255.0 212.220.31.1 255.255.255.0 212.220.35.1 255.255.255.0 1 212.220.31.2 255.255.255.0 2 212.220.35.2 255.255.255.0 195.38.54.65 255.255.255.224 1 195.38.54.94 255.255.255.224 1 6	Yes 1 3 5 6

Problem Author: Evgeny Kobzev
Problem Source: Ural State Univerisity Personal Contest Online February'2001 Students Session

1073. Square Country

Time limit: 1.0 second
Memory limit: 64 MB

There live square people in a square country. Everything in this country is square also. Thus, the Square Parliament has passed a law about a land. According to the law each citizen of the country has a right to buy land. A land is sold in squares, surely. Moreover, a length of a square side must be a positive integer amount of meters. Buying a square of land with a side a one pays a^2 quadrics (a local currency) and gets a square certificate of a landowner.

One citizen of the country has decided to invest all of his N quadrics into the land. He can, surely, do it, buying square pieces 1×1 meters. At the same time the citizen has requested to minimize an amount of pieces he buys: "It will be easier for me to pay taxes," — he has said. He has bought the land successfully.

Your task is to find out a number of certificates he has gotten.

Input

The only line contains a positive integer $N \leq 60\ 000$, that is a number of quadrics that the citizen has invested.

Output

The only line contains a number of certificates that he has gotten.

Sample

input	output
344	3

Problem Author: Stanislav Vasilyev
Problem Source: Ural State Univerisity Personal Contest Online February'2001 Students Session

1074. Very Short Problem

Time limit: 1.0 second
Memory limit: 64 MB

There are no long stories here, no long introductions and no complicated settings. All you need is to read from an input a real number in a format that is specified below, and to write it to an output in another format that is described below as well.

Here a syntactical description of the used notions follows.

<digit> ::= 0|1|2|3|4|5|6|7|8|9
<unsigned integer number> ::= <digit>|<digit><unsigned integer number>
<sign> ::= +|-
<integer number> ::= <unsigned integer number>|<sign><unsigned integer number>
<exponent symbol> ::= e|E
<exponent> ::= <exponent symbol><integer number>
<simple unsigned real number> ::= <unsigned integer number>|.<unsigned integer number>|<unsigned integer number>.<unsigned integer number>
<simple real number> ::= <simple unsigned real number>|<sign><simple unsigned real number>
<real number> ::= <simple real number>|<simple real number><exponent>

Input

consists of one or several pairs of lines. The first line of a pair contains an arbitrary set of symbols S. A length of the string S doesn't exceed 100 symbols. The second line of a pair contains a integer number N ($0 \le N \le 100$). An input is ended with the pair of lines, the first one of them contains the only symbol "#".

Output

For each pair of lines your program should write to an output a message: "Not a floating point number", if the string S is not a correct <real number> according to the given above specification. If S represents a correct <real number>, your program should write to an output this number formatted as a <simple real number> with N symbols after a decimal point. While forming the result you ought to note:

1. An integral part of a number should be non-empty.
2. There shouldn't be leading zeroes in a non-zero integral part of a number. A zero integral part of a number should consist of an only zero.
3. A fractional part of a number should contain exactly N symbols.
4. There shouldn't be a sign "+" before a positive number.
5. You shouldn't round off a number.

It is guaranteed that a length of a result will not exceed 200 symbols.

Sample

input	output
10.23 0 .04 1 -0.051e0 1 1.1e30 10 -1.1E-30 1 2468097632.1358642324268913e-2 20 e23 3 1 e3 1 #	10 0.0 0.0 110000000000000000000000000000.0000000000 0.0 24680976.32135864232426891300 Not a floating point number Not a floating point number

Problem Author: Alexander Klepinin
Problem Source: Ural State Univerisity Personal Contest Online February'2001 Students Session

1075. Thread in a Space

Time limit: 1.0 second
Memory limit: 64 MB

There are three points in a 3-dimensional space: A, B and C. All the coordinates of the points are integer numbers with absolute values not greater than 1000. A solid ball with a center in the point C is firmly fixed. A radius of the ball is R, a positive integer number. Distances from the point C to the points A and B are strictly greater than R.

It is necessary to stretch a thread of minimal length between points A and B. Surely, the thread should be outside of the ball.

You are to find out a length of the thread.

Input

The first three lines contain coordinates of the points A, B and C respectively. The fourth one contains a radius R of the ball.

Output

should contain a minimal length of the thread to within 2 symbols after a decimal point. You should output answer with two or more digits.

Sample

input	output
0 0 12 12 0 0 10 0 10 10	19.71

Problem Author: Alexander Mironenko
Problem Source: Ural State Univerisity Personal Contest Online February'2001 Students Session

1076. Trash

Time limit: 1.0 second
Memory limit: 64 MB

You were just hired as CEO of the local junkyard.One of your jobs is dealing with the incoming trash and sorting it for recycling.The trash comes every day in N containers and each of these containers contains certain amount of each of the N types of trash. Given the amount of trash in the containers find the optimal way to sort the trash. Sorting the trash means putting every type of trash in separate container. Each of the given containers has infinite capacity. The effort for moving one unit of trash from container i to j is 1 if $i \neq j$ otherwise it is 0.You are to minimize the total effort.

Input

The first line contains the number N ($1 \leq N \leq 150$), the rest of the input contains the descriptions of the containers.The $(1 + i)$-th line contains the description of the i-th container the j-th amount ($0 \leq$ amount ≤ 100) on this line denotes the amount of the j-th type of trash in the i-th container.

Output

You should write the minimal effort that is required for sorting the trash.

Sample

input	output
4 62 41 86 94 73 58 11 12 69 93 89 88 81 40 69 13	650

Problem Author: Jivko Ganev

1077. Travelling Tours

Time limit: 1.0 second
Memory limit: 64 MB

There are N cities numbered from 1 to N ($1 \le N \le 200$) and M two-way roads connect them. There are at most one road between two cities. In summer holiday, members of DSAP Group want to make some traveling tours. Each tour is a route passes K different cities ($K > 2$) T_1, T_2, …, T_K and return to T_1. Your task is to help them make T tours such that:

1. Each of these T tours has at least a road that does not belong to ($T-1$) other tours.
2. T is maximum.

Input

The first line of input contains N and M separated with white spaces. Then follow by M lines, each has two number H and T which means there is a road connect city H and city T.

Output

You must output an integer number T — the maximum number of tours. If $T > 0$, then T lines followed, each describe a tour. The first number of each line is K — the amount of different cities in the tour, then K numbers which represent K cities in the tour.

If there are more than one solution, you can output any of them.

Sample

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

Problem Author: Nguyen Xuan My (Converted by Dinh Quang Hiep and Tran Nam Trung)
Problem Source: From the third contest at Department of Mathematics and Informatics - Natural Sciences College - National University of HaNoi.

1078. Segments

Time limit: 1.0 second
Memory limit: 64 MB

A number of segments are lying on a line. Every segment is given with the coordinates of its endpoints. Segments are numbered from 1 to N ($0 < N < 500$). We assume, that one segment is inside another, if the two segments are different, the first one is fully contained in the second one, and their endpoints do not coincide. Write a program, which finds the numbers of the segments in the longest sequence of segments which are contained in. In the sequence, every segment except the last is inside the next segment in the sequence.

Input

The first line contains one integer N. Next, there are N lines, with two integers on every line, which are the coordinates of the left and the right endpoints of the corresponding segment. These coordinates are integers in the interval [–10000, 10000]. We assume that, the given segments are numbered according to their place in the input.

Output

The first line must contain one integer, equal to the number of segments in the found sequence. The following line must contain the numbers of the segments in this sequence. These numbers must be outputted, in the order in which the segments' lengths increase, starting from the smallest. If there are more than one output sequences, write any of them.

Sample

input	output
4 -2 2 -1 1 -3 3 4 5	3 2 1 3

Problem Author: Emil Kelevedzhiev
Problem Source: Winter Mathematical Festival Varna '2001 Informatics Tournament

1079. Maximum

Time limit: 2.0 second
Memory limit: 64 MB

Consider the sequence of numbers a_i, $i = 0, 1, 2, \ldots$, which satisfies the following requirements:

- $a_0 = 0$
- $a_1 = 1$
- $a_{2i} = a_i$
- $a_{2i+1} = a_i + a_{i+1}$

for every $i = 1, 2, 3, \ldots$.

Write a program which for a given value of n finds the largest number among the numbers $a_0, a_1, \ldots, a_n$.

Input

You are given several test cases (not more than 10). Each test case is a line containing an integer n ($1 \le n \le 99\,999$). The last line of input contains 0.

Output

For every n in the input write the corresponding maximum value found.

Sample

input	output
5 10 0	3 4

Problem Author: Emil Kelevedzhiev
Problem Source: Winter Mathematical Festival Varna '2001 Informatics Tournament

1080. Map Coloring

Time limit: 1.0 second
Memory limit: 64 MB

We consider a geographical map with N countries numbered from 1 to N ($0 < N < 99$). For every country we know the numbers of other countries which are connected with its border. From every country we can reach to any other one, eventually crossing some borders. Write a program which determines whether it is possible to color the map only in two colors — red and blue in such a way that if two countries are connected their colors are different. The color of the first country is red. Your program must output one possible coloring for the other countries, or show, that such coloring is impossible.

Input

On the first line is written the number N. On the following N lines, the i-th line contains the countries to which the i-th country is connected. Every integer on this line is bigger than i, except the last one which is 0 and marks that no more countries are listed for country i. If a line contains 0, that means that the i-th country is not connected to any other country, which number is larger than i.

Output

The output contains exactly one line. If the coloring is possible, this line must contain a list of zeros and ones, without any separators between them. The i-th digit in this sequence is the color of the i-th country. 0 corresponds to red color, and one — to blue color. If a coloring is not possible, output the integer −1.

Sample

input	output
3 2 0 3 0 0	010

Problem Author: Emil Kelevedzhiev
Problem Source: Winter Mathematical Festival Varna '2001 Informatics Tournament

1081. Binary Lexicographic Sequence

Time limit: 0.5 second
Memory limit: 64 MB

Consider all the sequences with length ($0 < N < 44$), containing only the elements 0 and 1, and no two ones are adjacent (110 is not a valid sequence of length 3, 0101 is a valid sequence of length 4). Write a program which finds the sequence, which is on K-th place ($0 < K < 10^9$) in the lexicographically sorted in ascending order collection of the described sequences.

Input

The first line of input contains two positive integers N and K.

Output

Write the found sequence or −1 if the number K is larger then the number of valid sequences.

Sample

input	output
3 1	000

Problem Author: Emil Kelevedzhiev
Problem Source: Winter Mathematical Festival Varna '2001 Informatics Tournament

1082. Gaby Ivanushka

Time limit: 1.0 second
Memory limit: 64 MB

Once upon a time there lived a tsar that has a daughter — Vasilisa the Beautiful. There were many of the young men that wanted to marry her but she repelled all suitors. The tsar was tired of her jigs, he got angry and issued an order: "The first who solves my puzzle, will marry Vasilisa!" Ivanushka decided to try his fortune. He came to the tsar and the tsar told him: "This is a program for you. Input *N* numbers and the program will tell you who you should marry. I give you a day to think." Ivanuska looked at the program and got upset: there were unknown letters, omnifarious symbols. The time passed. Ivanushka has thought out nothing.

The program was as follows.

C++

```
#include <cstdio>

const int N = ...;

int A[N];

int Q(int l, int r)
{
    if (l >= r)
        return 0;

    int m;
    int c = 0;
    int x = A[l];
    int i = l - 1;
    int j = r + 1;
    while (true)
    {
        do
        {
            --j;
            ++c;
        }
        while (A[j] > x);

        do
        {
            ++i;
            ++c;
        }
        while (A[i] < x);

        if (i < j)
        {
            int t = A[i];
            A[i] = A[j];
            A[j] = t;
        }
        else
        {
            m = j;
            break;
        }
    }

    return c + Q(l, m) + Q(m + 1, r);
```

```
}

int main()
{
    for (int i = 0; i < N; ++i)
        scanf("%d", &A[i]);

    if (Q(0, N - 1) == (N * N + 3 * N - 4) / 2)
        printf("Vasilisa the Beautiful\n");
    else
        printf("Koschei the Immortal\n");
    return 0;
}
```

Pascal

```
const
    N = ...;

var
    A: array [1..N] of integer;

function Q(l, r: integer): integer;
var
    m, c: integer;
    i, j, t, x: integer;
begin
    if l >= r then
        exit;

    c := 0;
    x := A[l];
    i := l - 1;
    j := r + 1;
    while true do
    begin
        repeat
            dec(j);
            inc(c)
        until A[j] <= x;

        repeat
            inc(i);
            inc(c)
        until A[i] >= x;

        if i < j then
        begin
            t := A[i];
            A[i] := A[j];
            A[j] := t
        end
        else
        begin
            m := j;
            break
        end
    end;

    Q := c + Q(l, m) + Q(m + 1, r)
```

```
end;

var
    i: integer;

begin
    for i := 1 to N do
        read(A[i]);

    if Q(1, N) = (N * N + 3 * N - 4) div 2 then
        writeln('Vasilisa the Beautiful')
    else
        writeln('Koschei the Immortal')
end.
```

Python

```
def Q(l: int, r: int) -> int:
    if l >= r:
        return 0

    c = 0
    x = A[l]
    i = l - 1
    j = r + 1
    while True:
        while True:
            j -= 1
            c += 1
            if A[j] <= x:
                break

        while True:
            i += 1
            c += 1
            if A[i] >= x:
                break

        if i < j:
            A[i], A[j] = A[j], A[i]
        else:
            m = j
            break

    return c + Q(l, m) + Q(m + 1, r)


N = ...
A = [int(x) for x in input().split()][0:N]

if Q(0, N - 1) == (N * N + 3 * N - 4) / 2:
    print('Vasilisa the Beautiful')
else:
    print('Koschei the Immortal')
```

Now you know this program. You may try to help Ivanushka.

Input

The only line contains an integer N that is a value of constant from the tsar's program ($1 \le N \le 1000$).

Output

Output N integers in range from -10^9 to 10^9. The tsar's program given those integers should output a message "Vasilisa the Beautiful". The integers should be separated with a space. If several variants are possible choose any you like.

Sample

input	output
3	3 7 19

Problem Author: Nikita Shamgunov
Problem Source: The 3rd high school children programming contest, USU, Yekaterinburg, Russia, March 4, 2001

1083. Factorials!!!

Time limit: 1.0 second
Memory limit: 64 MB

Definition 1. $n!!\ldots!$ = $n(n-k)(n-2k)\ldots(n \bmod k)$, if k doesn't divide n; $n!!\ldots!$ = $n(n-k)(n-2k)\ldots k$, if k divides n (There are k marks ! in the both cases).

Definition 2. X mod Y — a remainder after division of X by Y.

For example, 10 mod 3 = 1; 3! = 3·2·1; 10!!! = 10·7·4·1.

Given numbers n and k we have calculated a value of the expression in the first definition. Can you do it as well?

Input

contains the only line: one integer n, $1 \le n \le 10$, then exactly one space, then k exclamation marks, $1 \le k \le 20$.

Output

contains one number — $n!!\ldots!$ (there are k marks ! here).

Sample

input	output
9 !!	945

Problem Author: Oleg Katz
Problem Source: The 3rd high school children programming contest, USU, Yekaterinburg, Russia, March 4, 2001

1084. Goat in the Garden

Time limit: 1.0 second
Memory limit: 64 MB

Someone has let a goat in a square kitchen-garden and had bound it to a stake. The stake is driven into the ground in the very midst of the square. The goat is hungry as a hunter and very voracious, and eats everything that can be reached without leaving the square and tearing off the rope. What area of the kitchen-garden will be ate round?

Input

contains lengths of the garden sides and a cord length in meters (positive integers not exceeding 100, located in one line and separated with a space).

Output

should contain an area of the kitchen-garden (in square meters to within 3 symbols after a decimal point), ate round by the goat.

Sample

input	output
10 6	95.091

Problem Author: Irina Danilina
Problem Source: The 3rd high school children programming contest, USU, Yekaterinburg, Russia, March 4, 2001

1085. Meeting

Time limit: 2.0 second
Memory limit: 64 MB

K friends has decided to meet in order to celebrate their victory at the programming contest. Unfortunately, because of the tickets rise in price there is a problem: all of them live in different parts of the city, and they are to choose a place of meeting so that they wouldn't pay too much for the tickets. You are to help them make the best choice.

All stops are enumerated with integers 1, …, *N* inclusive. There are *M* tram routes in the city (the friends take only trams and do not go on foot from stop to stop). For each route numbers of its stops are known. For each friend we know an amount of money he has and whether he has a month tram ticket. A ticket price equals 4 rubles.

You are to find out a stop number, such that all of the friends might come there and the sum of money they spend for their tramps would be minimal. Naturally, they may change routes (it means that each one may make changes on his way to the required stop). Note, that changing the route one has to pay for a new ticket: the friends are honest people — they do always pay for tickets. Everyone pays for a ticket from his own money. No one is to leave money for the return tickets.

Input

The first line contains two integers *N* and *M*; $1 \le N, M \le 100$ (*N* is a number of stops, *M* is a number of routes). The next *M* lines define the routes in the following sort: there is an integer *L* in the beginning of a line — that is an amount of stops of the corresponding route ($2 \le L \le 100$). Then *L* integers defining stops numbers of the route follow. The numbers are separated with a space. A route is defined by its stops along the line in one direction. The next line contains an integer *K* ($1 \le K \le 100$), that is an amount of friends. The next *K* lines contain information about each of them (one line for one person): there is a positive integer in the beginning of a line that is an amount of money (in rubles) the person has, then a number of a stop that he goes there from his home on foot, then 0 (if this person has no month ticket) or 1 (if he has). The numbers in a line are separated with a space. No one of the friends has more than 1000 rubles.

Output

Output a number of a stop that is a meeting point (if there are several numbers choose the minimal one) and a total sum of money (in rubles) that the friends has paid for their trips to the appointed place. The numbers should be separated with a space. If the friends won't be able to meet at one stop, output the only number 0.

Sample

input	output
4 3 2 1 2 2 2 3 2 3 4 3 27 1 0 15 4 0 45 4 0	4 12

Problem Author: Alexander Somov
Problem Source: The 3rd high school children programming contest, USU, Yekaterinburg, Russia, March 4, 2001

1086. Cryptography

Time limit: 2.0 second
Memory limit: 64 MB

While preparing this problem set the jury has run into the following problem: it was necessary to send by e-mail the texts of the problems. As it is well known, e-mail is not reliable, messages are sent not enciphered, there is a danger that someone can intercept them. The members of the program committee wanted no participant know the texts of the problems before the start of the contest. That's why they resorted to cryptography methods in order to save the texts of the problems from an unsanctioned reading. The jury gas worked up a new way of enciphering of a text. It is not patented yet, so it's kept secret. However, we'll reveal you one secret: the new algorithm is based on the work with prime numbers. In particular, in uses a calculation of n-th by order prime number.

Several members of the program committee independently have worked up programs that make such calculations, but these programs produce different answers. Each one of the programmers is sure that his program works correctly. That's why the jury has reached the deadlock and can't continue working. The contest is about not to take place.

You are to help to the jury and to save the contest. We want you to write a program that calculates the n-th by order prime number. The main thing is that your program should work correctly.

Input

First line contains a positive integer k. Then k positive integers follow (one in each line). The numbers don't exceed 15000.

Output

For each number n you should output the n-th by order prime number. Each number should be in its line.

Sample

input	output
4 3 2 5 7	5 3 11 17

Notes

The *prime number* is a positive integer that has exactly two different positive divisors, i.e. 1 is not a prime number.

Problem Author: folklore
Problem Source: The 3rd high school children programming contest, USU, Yekaterinburg, Russia, March 4, 2001

1087. The Time to Take Stones

Time limit: 1.0 second
Memory limit: 64 MB

You probably know the game where two players in turns take 1 to 3 stones from a pile. Looses the one who takes the last stone. We'll generalize this well known game. Assume that both of the players can take not 1, 2 or 3 stones, but k_1, k_2, …, k_m ones. Again we'll be interested in one question: who wins in the perfect game. It is guaranteed that it is possible to make next move irrespective to already made moves.

Input

The first line contains two integers: n and m ($1 \le n \le 10000$; $1 \le m \le 50$) — they are an initial amount of stones in the pile and an amount of numbers k_1, …, k_m. The second line consists of the numbers k_1, …, k_m, separated with a space ($1 \le k_i \le n$).

Output

Output 1, if the first player (the first to take stones) wins in a perfect game. Otherwise, output 2.

Sample

input	output
17 3 1 3 4	2

Problem Author: Anton Botov
Problem Source: The 3rd high school children programming contest, USU, Yekaterinburg, Russia, March 4, 2001

1088. Ilya Murometz

Time limit: 1.0 second
Memory limit: 64 MB

Misfortune has come to the Russian land. Foul Idol has dragged away good fellow Alyosha Popovich and has chained him with iron chains to the magical stone, so that Alyosha canTt even step. It is necessary to rescue Alyosha from the captivity but nobody is able to: the Tsar is on the war with his detachment. Ilya Murometz has thought a bit and says: "I'll go to rescue my friend Alyosha from Foul Idol alone, and if he is to die there, so am I". Ilya has said these words, and has leaped on to his good horse and has gone seeking for Alyosha.

Ilya rides his horse and comes to a stone, but not the magic one that he has been looking for, but to the road stone. As usual, it is written there: "If you go left — you'll get `time limit', if you go right — you'll get `wrong answer'". Ilya Murometz sits down and falls apart. Suddenly a little bird sits near and asks: "Why are you upset, good fellow?" Ilya Murometz tells her about his problem and the bird tells him: "It's always here like this: if you go left — in an hour you'll see such road stone; and so until you come to the sea. Foul Idol has made this. He wants you to get lost here. Go home, Ilya Murometz, don't try your doom!" And the bird disappears, as there hasn't been the one. Ilya Murometz thinks a lot, but decides not to betray his friendship. He leaps on to his good horse and continues his way. And he rides, and he rides, and he rides.

Ilya Murometz dismounts near a road stone that is like the first one and falls apart more than the first time. Suddenly he sees the magic stone — it is close to him, but Ilya Murometz can't leave the road — he'll disappear in the marsh. He sees the sea very close to him — he can even count: there's D hours of the way from the magic stone to the sea, and from himself he'll have to ride E hours. But from his father he new that from the first stone to the sea (no matter where he turns) there's F hours way.

He looks closely and sees that the road leads to the moorings, and the moorings are enumerated with integers. Ilya looks very-very closely and he finds out that there is a rule of the enumeration: if one rides from the first stone always to the right — he comes to the mooring #1, if at the last moment one turns left — he comes to the mooring #2, and if one always turns right, one before the last time he turns left and then turns right — he comes to the mooring #3! Ilya Murometz sees, that the nearest mooring to his stone is #E_p, and the nearest one to the magic stone is #D_p. But the roads branch, and there are other moorings.

The good horse turns his head and tells to Ilya Murometz: "Ilyusha! I won't gallop more than H hours." Help to Ilya Murometz to find out, wether he saves Alyosha Popovich, or he stays himself in the foreign land.

Input

The first line contains 6 integers: D, E, F, D_p, E_p, H. $0 \le D, E, F, H \le 30$; $1 \le E_p, D_p \le 1\ 073\ 741\ 824$. All numbers are separated with a space.

Output

should contain the word “YES”, if Ilya Murometz will reach Alyosha Popovich on his horse, and otherwise — the word “NO”.

Sample

input	output
1 2 3 2 6 4	YES

Problem Author: Oleg Katz
Problem Source: The 3rd high school children programming contest, USU, Yekaterinburg, Russia, March 4, 2001

1089. Verification with the Dictionary

Time limit: 0.5 second
Memory limit: 64 MB

Your English teacher recently told you that she dreams about an automated spelling correction system to correct and count the mistakes that her students make. The 8th of March is soon and you decided that if you write such a program and present it to your teacher, then she would be so happy that she might get more favorable to you at the exams. The task is to replace the words that differ from the ones in the dictionary not more than in one letter and to count the amount of fixed mistakes. Your teacher must have forgotten that her students might also skip or add letters. Excellent, you don't have to worry about these cases!

Input

In the first and the next lines up until the separator line containing the only character `#' there are dictionary words, each word in its own line. In the lines following the separator line there is a text that you are to correct. There are no more than 100 words in the dictionary. The length of a word in the dictionary doesn't exceed 8 characters. The text to correct contains no more than 1000 words of the length up to 16 characters. The text consists of one or more lines of length up to 80 characters. The cumulative length of all lines doesn't exceed 10000 characters. All words are written in lower case. Words might only contain Latin letters from `a' to `z' (e.g. `one-seventh' consists of two words: `one' and `seventh'). The dictionary is compiled so that for each word no more than one variant of correction is possible. There's exactly one end of line character at the end of the text.

Output

Output the corrected text keeping the initial formatting: punctuation marks, line breaks and words that are absent in the dictionary. In the next line output the number of fixed mistakes.

Sample

input
country occupies surface covers russia largest europe part about world # the rushia is the larjest cauntry in the vorld. it ockupies abaut one-seventh of the earth's surfase. it kovers the eastern park of yurope and the northern park of asia.
output
the russia is the largest country in the world. it occupies about one-seventh of the earth's surface. it covers the eastern part of europe and the northern part of asia. 11

Problem Author: Anton Botov
Problem Source: The 3rd high school children programming contest, USU, Yekaterinburg, Russia, March 4, 2001

1090. In the Army Now

Time limit: 1.0 second
Memory limit: 64 MB

The sergeant ordered that all the recruits stand in rows. The recruits have formed K rows with N people in each, but failed to stand according to their height. The right way to stand in a row is as following: the first soldier must be the highest, the second must be the second highest and so on; the last soldier in a row must be the shortest. In order to teach the young people how to form rows, the sergeant ordered that each of the recruits jump as many times as there are recruits before him in his row who are shorter than he. Note that there are no two recruits of the same height.

The sergeant wants to find which of the rows will jump the greatest total number of times in order to send this row to work in the kitchen. Help the sergeant to find this row.

Input

The first line contains integers N and K ($2 \le N \le 10000$; $1 \le K \le 20$). Each of the following K lines contains N different integers from 1 to N. The recruits in each row are numbered according to their height (1 — the highest, N — the shortest). Each line shows the order in which the recruits stand in the corresponding row. The first integer in a line is the number of the first recruit in a row and so on. Therefore a recruit jumps as many times as there are numbers which are greater than his number in the line before this number.

Output

You should output the number of the row in which the total amount of jumps is the greatest. If there are several rows with the maximal total amount of jumps you should output the minimal of their numbers.

Sample

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

Problem Author: Nikita Shamgunov
Problem Source: USU Open Collegiate Programming Contest March'2001 Senior Session

1091. Tmutarakan Exams

Time limit: 1.0 second
Memory limit: 64 MB

University of New Tmutarakan trains the first-class specialists in mental arithmetic. To enter the University you should master arithmetic perfectly. One of the entrance exams at the Divisibility Department is the following. Examinees are asked to find K different positive integers that have a common divisor greater than one. All integers in this set should not exceed S. The numbers K and S are announced at the beginning of the exam. To exclude copying (the Department is the most prestigious in the town!) each set of numbers is credited only once (to the person who submitted it first).

Last year these numbers were $K = 25$ and $S = 49$ and, unfortunately, nobody passed the exam. Moreover, it was proved later by the best minds of the Department that there do not exist sets of numbers with the required properties. To avoid embarrassment this year, the dean asked for your help. You should find the number of sets of K different positive integers, each of the integers not exceeding S, which have a common divisor greater than one. Of course, the number of such sets equals the maximal possible number of new students of the Department.

Input

The only line contains integers K and S ($2 \le K \le S \le 50$).

Output

Output the maximal possible number of the Department's new students if this number does not exceed 10000 which is the maximal capacity of the Department, otherwise you should output 10000.

Sample

input	output
3 10	11

Notes

In the example the following sets satisfy the conditions:

1. (2, 4, 6);
2. (2, 4, 8);
3. (2, 4, 10);
4. (2, 6, 8);
5. (2, 6, 10);
6. (2, 8, 10);
7. (3, 6, 9);
8. (4, 6, 8);
9. (4, 6, 10);
10. (4, 8, 10);
11. (6, 8, 10).

Problem Author: Stanislav Vasilyev
Problem Source: USU Open Collegiate Programming Contest March'2001 Senior Session

1092. Transversal

Time limit: 1.0 second
Memory limit: 64 MB

Consider a square table of size $(2N + 1) \times (2N + 1)$ with a cells each containing the sign “+” or the sign “-”. We call an arbitrary set of $2N + 1$ cells a transversal if each line and each column of the table contain exactly one cell belonging to the set.

By one operation you are allowed to change signs to opposite in all cells of one transversal. You are asked to determine if it is possible to obtain a table containing not more than $2N$ cells with the sign “+” by a sequence of such operations.

Input

The first line contains a positive integer N not exceeding 20. The next $2N + 1$ lines contain the table. They consist of the symbols “+” and “-” without spaces between them.

Output

Output “No solution” if a necessary sequence of operations does not exist. Otherwise output in the first line “There is solution:” and in the next lines a sequence of operations that leads to the required result. Each of these lines should describe one transversal and should contain the integers from 1 to $2N + 1$. Number K at position S means that the transversal includes the cell at the intersection of the line number S with the column number K.

If there exist many sequences of operations you may output any one.

Sample

input	output
1 +++ ++- +-+	There is solution: 1 2 3 2 3 1 1 3 2 3 1 2

Problem Author: Dmitry Filimonenkov
Problem Source: USU Open Collegiate Programming Contest March'2001 Senior Session

1093. Darts

Time limit: 1.0 second
Memory limit: 64 MB

Maybe you know about the game called Darts. The goal of the game is to hit a circular target, which is called a dartboard. To do so you should take a dart and throw it to the dartboard. Of course, it takes time to learn how to hit the dartboard. And your task is to speed the learning by writing a program that determines if a dart will hit or miss the dartboard.

Let's make some assumptions about the objects under consideration.

1. A dartboard is a circular part of a plane. It is defined by coordinates of its center (C_x, C_y and C_z), radius R, and coordinates of a vector that is orthogonal to the plane of the dartboard (N_x, N_y and N_z).
2. A dart is a point that moves according the following equations:
   $M_x = S_x + V_x \cdot t$
   $M_y = S_y + V_y \cdot t$
   $M_z = S_z + V_z \cdot t - (g/2) \cdot t^2$
   Here S_x, S_y, and S_z are coordinates of initial position of the dart, V_x, V_y, and V_z are components of initial velocity of the dart, M_x, M_y, and M_z are coordinates of the point where the dart will be at time t, g is the acceleration of gravity and is assumed to be equal to 10.
3. You may assume that the dart is thrown with a nonzero horizontal component (i.e. V_x or V_y is non-zero).
4. You may assume no friction.
5. To hit the dartboard the dart should hit it strictly inside boundary from any side.
6. Initially the dart is not inside the dartboard's boundary.

Input

Input contains 13 real numbers. They are (according to notation given before):

C_x C_y C_z N_x N_y N_z R
S_x S_y S_z V_x V_y V_z

All numbers are not greater than 500 in their absolute values and given with at most 4 digits after decimal point. $R > 0$.

Output

Output should contain the only word “HIT” if dart hits the dartboard, and word “MISSED” otherwise.

Sample

input	output
47 0 -72 1 0 1 4.25 0 0 0 10 0 10	HIT

Problem Author: Alexander Klepinin
Problem Source: USU Open Collegiate Programming Contest March'2001 Senior Session

1094. E-screen

Time limit: 0.25 second
Memory limit: 64 MB

A new one-line electronic screen (e-screen) especially designed for quick input and change of information was installed at a supermarket. All information is entered by an operator. Every time the operator presses a button corresponding to a symbol this symbol is shown on the e-screen at the position where the cursor is located at that moment (therefore the symbol that was shown at that position earlier is erased) and then the cursor moves one position to the right.

The keyboard contains letters a-z, A-Z, digits 0-9, punctuation signs (:;-!?.,), and the space button. There are also two keys that move the cursor one position to the right and to the left without erasing anything. The width of the screen is 80 symbols. When the cursor reaches left or right edge of the screen it is automatically placed at the first position to the left.

The new e-screen had worked perfectly when it was run by its seller, but when the seller had gone it was found that nobody could operate the e-screen properly. Besides, the e-screen was installed in such a place that the operator could not see it. Your task is to make a program emulation of the e-screen so that the operator could see the results of his or her actions.

Input

The **single line** contains a sequence of the buttons pressed by the operator (the length of this sequence is from 0 to 10000). The symbol '>' stands for the move of the cursor one position to the right and the symbol '<' stands for the move of the cursor one position to the left.

Output

Output the line that would be shown on the e-screen after pressing the given sequence of the buttons. Assume that at the beginning the e-screen contains 80 spaces and the cursor is placed at the first position to the left.

Sample

input
>><<<Look for clothes at the <<<<<<<<<<<<<<<second flo or. <<<<<<<Fresh pizza and <<<<<<<<<<<<<<<<hamburger a t a shop right to <<<<<<<<<<<<<the entrance. Call <<<< <<<<<< 123<-456<-8790 <<<<<<<<<<<<<<<<to order <<<<<<< <<<<<<<<<<computers< and office<<<<<<< chairs.
output
Look for second hamburger at computer and chairs.790

Notes

In the sample the single input line is splitted to several lines, but in all tests there is a single line.

Problem Author: Stanislav Vasilyev
Problem Source: USU Open Collegiate Programming Contest March'2001 Senior Session

1095. Nikifor 3

Time limit: 1.0 second
Memory limit: 64 MB

Nikifor has a certain positive integer containing each of the digits 1, 2, 3, 4 in its decimal form. He asks you to rearrange the digits of this integer in such a way that the new integer divides by 7.

Input

The first line contains an integer N that is a number of integers to be checked ($1 \le N \le 10000$). The next N lines contain these integers. Each integer is positive and has no more than 20 digits.

Output

For each of the N integers output an integer divisible by 7 that can be obtained from the corresponding integer from the input data by a rearrangement of the digits. If such rearrangement does not exist you should output 0 in the corresponding line. In the case of several valid rearrangements you may output any one of them.

Sample

input	output
2 1234 531234	4123 354123

Problem Author: Dmitry Filimonenkov
Problem Source: USU Open Collegiate Programming Contest March'2001 Senior Session

1096. Get the Right Route Plate!

Time limit: 1.0 second
Memory limit: 64 MB

Everybody who had ridden a Ekaterinburg bus could notice that on the other side of the plate with the number of the route there was a number of another route.

One day the driver of a new bus came to the storehouse and found that there was no plate with the number of the route he had been assigned to ride. The storekeeper simply gave him a random plate and advised to change it for a plate from another bus. Any driver will agree to change his plate for another if this plate has the number of his route. Help the new driver to find a shortest sequence of changes that will enable him to get a plate with the number of his route.

Input

The first line contains an integer K that is a number of the acting buses excluding the new bus ($1 \le K \le 1000$). The next K lines contain the number of the route of the corresponding bus and the number on the other side of its plate. Numbers of routes are integers from 1 to 2000.

The last line of the input contains integers T, S_1 and S_2 that are the number of the route of the new bus and the numbers on the plate given by the storekeeper ($1 \le T, S_1, S_2 \le 2000$; $T \ne S_1$; $T \ne S_2$).

Output

If it is impossible to get the needed number by a sequence of changes, output “IMPOSSIBLE”. Otherwise output the least necessary number of changes $M > 0$ in the first line and then M lines with sequentially numbers of buses (not routes!) with drivers of which the plates must be changed. The buses are numbered from 1 to K as they are described in the input. If there are several optimal solutions, you can output any one.

Sample

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

Problem Author: Stanislav Vasilyev
Problem Source: USU Open Collegiate Programming Contest March'2001 Senior Session

1097. Square Country 2

Time limit: 1.0 second
Memory limit: 64 MB

The Square Parliament of the Square country has decreed that the National Square Park be created. Of course, the Park should occupy a large square. Unfortunately, at the moment a lot of square citizens have invested (with the help of last championship's participants) their quadrics into land so that a part of the country is already occupied. Maybe, it is now impossible to find a land for the Park without affecting interests of the private owners. In this case some of the pieces of land must be expropriated.

To avoid social unrest the Parliament has to locate the Park so that the interests of as less important as possible citizens were affected. It is better to expropriate land from a thousand of simple citizens than from one member of the Parliament or from one bank-owner.

All pieces of land that are occupied by square citizens are marked with integers from 2 to 100 according to importance of the owner: the property of the Square President is marked with 100, the property of great businessmen are marked with 99, the property of members of the Parliament is marked with 98, and so on.

Besides, some pieces of land belong to the members of (not square) Jury which created this problem. This land is marked with number 255 and cannot be expropriated at all.

Input

The first line contains integers L and A, which are the length of a side of the Square country, and the length of a side of the Park ($1 \le A \le L \le 10000$). The next line contains an integer M that is a number of occupied pieces of land ($1 \le M \le 100$). According to the Square Rules a piece of land is a square with integer coordinates of corners and its sides are parallel to the axes. The coordinates of the lower left corner of the Square country itself are (1, 1).

The next M lines contain information about occupied pieces of land: importance of the owner, length of the square's side and the coordinates of the lower left corner. The importance is an integer from 2 to 100 or integer 255. The side and the coordinates are integers from 1 to L. Each piece of land is contained in the country and may intersect another piece of land only along its boundary. All land which is not contained in the occupied pieces is free.

Output

If it is possible to create the Park on free land, output an integer 1. Otherwise, output the least possible importance of owners whose land must be expropriated (an integer from 2 to 100). The number and area of expropriated pieces of land are not important. You should only take into account importance of the most important of the affected land-owners.

If it is impossible to create the Park not involving land of the Jury, output “IMPOSSIBLE”.

Samples

input	output
5 3 6 94 2 4 1 3 1 1 1 2 1 1 2 2 2 2 1 100 1 2 4 255 1 5 5	3
5 3 1 255 1 3 3	IMPOSSIBLE

Notes

The figure illustrates the first example. The Park can be created on the land marked with grey color. To do this the property of land-owners with importance 2 and 3 must be expropriated.

Problem Author: Stanislav Vasilyev
Problem Source: USU Open Collegiate Programming Contest March'2001 Senior Session

1098. Questions

Time limit: 1.0 second
Memory limit: 64 MB

Holding a collegiate programming contest is a very exhausting work. There is a well-known proverb that one fool can ask so many questions that a hundred clever men will not answer. And during a collegiate programming contest questions are asked by one hundred clever people.

The jury of the Third Urals Collegiate Programming Contest being clever enough has found a simple way to make its work easier. We have invented a simple algorithm that will help us answer ALL your numerous questions! Moreover, this algorithm guarantees that the same questions will have the same answers (this would be hardly possible if we undertake such a task ourselves). According to this algorithm a member of the jury starts to delete characters of the question in the following order:

1. Starting from the first character he or she counts out $N-1$ characters (spaces, punctuation marks etc. are considered to be characters too) and deletes the Nth character.
2. If a string ends the count continues from the beginning of the string.
3. After deleting a character the count restarts from the character that would be the ($N+1$)-st in the previous count.
4. If the last remaining character is a question-mark ("?") then the answer to the question is "Yes". If it is a space then the answer is "No". Any other character will lead to "No comments" answer.

You should help the jury and write a program that will do a hard work of answering your questions tomorrow. The number N is secret and will not be announced even after the end of the contest. Your program should use $N = 1999$.

For example, taking a string "Is it a good question?" (its length is 22) the characters will be counted in the following way: "Is it a good question?Is it … quest" and "i" will be deleted. Then the count restarts from "on?Is it…" etc., until "s" will be left (thus the answer is "No comments", as usual).

Input

The input is a question, that is any text containing at least one character (end of line is not a character). Each character of the input (excepting the ends of lines) is a part of the question. You should read the question from the input.

The size of the input is not more than 30000.

Output

The answer.

Samples

input	output
Does the jury of this programming contest use the algorithm described in this problem to answer my questions?	Yes
At least, will anybody READ my question?	No
This is UNFAIR!	No comments

Notes

There are no spaces in the sample inputs except for those between words in one line. Thus the first question contains 108 characters, the second contains 40 and the third contains 14.

Problem Author: Stanislav Vasilyev
Problem Source: This problem was offered to participants of the Third Urals Collegaite Programming Contest at the trial tour, the day before main contest

1099. Work Scheduling

Time limit: 0.5 second
Memory limit: 64 MB

There is a certain amount of night guards that are available to protect the local junkyard from possible junk robberies. These guards need to be scheduled in pairs so that each pair guards in a different night. The junkyard CEO ordered you to write a program which given the guards characteristics determines the maximum amount of scheduled guards (the rest will be fired). Please note that each guard can be scheduled with only one of his colleagues and no guard can work alone.

Input

The first line of the input contains one number $N \leq 222$ which is a number of night guards. Unlimited number of lines consisting of unordered pairs (i, j) follow, each such pair means that guard #i and guard #j can work together, because it is possible to find uniforms that suit both of them (The junkyard uses different parts of uniforms for different guards i.e. helmets, pants, jackets. It is impossible to put small helmet on a guard with a big head or big shoes on guard with small feet). The input ends with Eof.

Output

You should output one possible optimal assignment. On the first line of the output write the even number C, the amount of scheduled guards. Then output C/2 lines, each containing 2 integers (i, j) that denote that i and j will work together.

Sample

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

Problem Author: Jivko Ganev

1100. Final Standings

Time limit: 1.0 second
Memory limit: 16 MB

Old contest software uses bubble sort for generating final standings. But now, there are too many teams and that software works too slow. You are asked to write a program, which generates exactly the same final standings as old software, but fast.

Input

The first line of input contains only integer $1 < N \le 150000$ — number of teams. Each of the next N lines contains two integers $1 \le ID \le 10^7$ and $0 \le M \le 100$. ID — unique number of team, M — number of solved problems.

Output

Output should contain N lines with two integers ID and M on each. Lines should be sorted by M in descending order as produced by bubble sort (see below).

Sample

input	output
8 1 2 16 3 11 2 20 3 3 5 26 4 7 1 22 4	3 5 26 4 22 4 16 3 20 3 1 2 11 2 7 1

Notes

Bubble sort works following way:

```
while (exists A[i] and A[i+1] such as A[i] < A[i+1]) do
   Swap(A[i], A[i+1]);
```

Problem Author: Pavel Atnashev
Problem Source: Tetrahedron Team Contest May 2001

1101. Robot in the Field

Time limit: 1.0 second
Memory limit: 64 MB

There is a field [−*N*..*N*]×[−*N*..*N*]. At initial moment, robot stands at point (0, 0). It starts moving in (1, 0) direction. Robot moves according to a program. Program is a correct boolean expression. It contains operators NOT, AND, OR (NOT has highest priority, OR - lowest), brackets '(', ')', constants 'TRUE' and 'FALSE', and registers 'A', ..., 'Z'. Initially, all robot's registers are FALSE. Robot moves forward until it reaches a fork. Then, robot evaluate the expression and turns right if it is TRUE and turns left if it is FALSE. Besides, there are some points in the field, standing on which makes one of robot's registers to invert. You are asked to print robot's route until it falls out of the field.

Input

First line contains boolean expression. The length of expression ≤ 250. Second line contains three integers $1 \le N \le 100$, $0 \le M \le 100$, $0 \le K \le 100$. M — number of forks, K — number of register inverting points. Then follows M lines, each of them contains two integers X, Y — coordinates of forks. Then follows K lines, each of them contains two integers X, Y and character C — coordinates of register inverting point and name of register, which inverts. You may assume, that there is no fork at point (0, 0). You may assume, that no two objects (forks or register inverting points) coincide. You may assume, that after some moves robot falls out of the field.

Output

You should print robot's route to output, every pair of coordinates in separate line.

Sample

input	output
NOT((A OR NOT B) AND (A OR B)) OR NOT (A AND NOT B OR TRUE) 1 5 2 1 0 1 1 1 -1 -1 -1 -1 1 0 1 A -1 0 D	0 0 1 0 1 -1 0 -1 -1 -1 -1 0 -1 1 0 1 1 1

Problem Author: Pavel Atnashev
Problem Source: Tetrahedron Team Contest May 2001

1102. Strange Dialog

Time limit: 1.0 second
Memory limit: 16 MB

One entity named "one" tells with his friend "puton" and their conversation is interesting. "One" can say words "out" and "output", besides he calls his friend by name. "Puton" can say words "in", "input" and "one". They understand each other perfect and even write dialogue in strings without spaces.

You have N strings. Find which of them are dialogues.

Input

In the first line of input there is one non-negative integer $N \leq 1000$. Next N lines contain non-empty strings. Each string consists of small Latin letters. Total length of all strings is no more than 10^7 characters.

Output

Output consists of N lines. Line contains word "YES", if string is some dialogue of "one" and "puton", otherwise "NO".

Sample

input	output
6 puton inonputin oneputonininputoutoutput oneininputwooutoutput outpu utput	YES NO YES NO NO NO

Problem Author: Katya Ovechkina
Problem Source: Tetrahedron Team Contest May 2001

1103. Pencils and Circles

Time limit: 1.0 second
Memory limit: 64 MB

One old man and one old woman sit at the table and write letters. On the table in front of them at the coordinates (x_i, y_i) stay N pencils. Diameter of pencils is 0 and all of them stay vertically. Man and woman decide to divide pencils into two equal groups. In order to do this old man give thread and put it on the table so that thread lay in a circle. Circle must cross over three pencils. He wants one group to get into the circle and other group not to get into the circle.

Input

At the first line of the input is one odd integer $3 \le N \le 5000$. At the next N lines are coordinates of pencils. There are no three pencils staying in one line and there are no four pencils staying on one circle. All coordinates are integers that are not greater than 10^8 by absolute value.

Output

You must output 6 integers — coordinates of three pencils over which cross thread. Inside of circle must be $(N-3)/2$ pencils. On the circle must be these three pencils. And outside of the circle must be others $(N-3)/2$ pencils. If there is no solution you must output "No solution". If there are more than one solution output any of them.

Sample

input	output
7 0 0 1 0 2 -1 2 1 1 1 0 2 -3 -1	0 0 1 0 2 1

Problem Author: Katya Ovechkina
Problem Source: Tetrahedron Team Contest May 2001

1104. Don’t Ask Woman about Her Age

Time limit: 1.0 second
Memory limit: 64 MB

Mrs Little likes digits most of all. Every year she tries to make the best number of the year. She tries to become more and more intelligent and every year studies a new digit. And the number she makes is written in numeric system which base equals to her age. To make her life more beautiful she writes only numbers that are divisible by her age minus one. Mrs Little wants to hold her age in secret.

You are given a number consisting of digits 0, …, 9 and Latin letters A, …, Z, where A equals 10, B equals 11 etc. Your task is to find the minimal number k satisfying the following condition: the given number, written in k-based system is divisible by $k-1$.

Input

Input consists of one string containing no more than 10^6 digits or uppercase Latin letters.

Output

Output the only number k, or "No solution." if for all $2 \le k \le 36$ condition written above can't be satisfied. By the way, you should write your answer in decimal system.

Sample

input	output
A1A	22

Problem Author: Igor Goldberg
Problem Source: Tetrahedron Team Contest May 2001

1105. Observers Coloring

Time limit: 0.5 second
Memory limit: 64 MB

Nikifor told us that once he solved problem at mathematical tournament of S.Petersburg's secondary school N239 in 1994. Nikifor said that he solved a problem from the moment T_0 to the moment T_1. He remembers that N observers appeared in the room. The i-th observer entered the room at the moment $t_{0,i}$ and went out at the moment $t_{1,i}$. At every moment there was at least one observer in the room.

When the tournament was finished, Nikifor claimed that it is possible to color some observers, and the summary time when there was only one colored observer in the room is not less than 2/3 of the time when Nikifor solved problem.

You are to answer whether Nikifor right or not.

Input

The first line of input contains real numbers T_0 and T_1 ($T_0 < T_1$). The second line contains number N — number of observers ($N < 10000$). Next N lines contain real numbers $t_{0,i}$ and $t_{1,i}$ ($T_0 \le t_{0,i} < t_{1,i} \le T_1$).

Output

If Nikifor is not right output should contain the only number 0. If Nikifor is right you should write to the first line the quantity of colored observers, and next lines should contain their numbers. Do not write more than one number in a line. You may write these numbers in any order. If there are more than one solution exist you may find any of them.

Sample

input	output
0.0 20.0 7 1.0 1.5 0.0 10.0 9.0 10.0 18.0 20.0 9.0 18.0 2.72 3.14 19.0 20.0	3 2 5 7

Problem Author: Dmitry Filimonenkov feat Igor Goldberg
Problem Source: Tetrahedron Team Contest May 2001

1106. Two Teams

Time limit: 1.0 second
Memory limit: 64 MB

The group of people consists of N members. Every member has one or more friends in the group. You are to write program that divides this group into two teams. Every member of each team must have friends in another team.

Input

The first line of input contains the only number N ($N \leq 100$). Members are numbered from 1 to N. The second, the third,…and the (N+1)th line contain list of friends of the first, the second, …and the Nth member respectively. This list is finished by zero. Remember that friendship is always mutual in this group.

Output

The first line of output should contain the number of people in the first team or zero if it is impossible to divide people into two teams. If the solution exists you should write the list of the first group into the second line of output. Numbers should be divided by single space. If there are more than one solution you may find any of them.

Sample

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

Problem Author: Dmitry Filimonenkov
Problem Source: Tetrahedron Team Contest May 2001

1107. Warehouse Problem

Time limit: 1.0 second
Memory limit: 64 MB

The are N different types of goods at the warehouse. Types are numbered by numbers $1 \ldots N$. Employees of this warehouse made K different sets of these goods. We'll say that two sets are “similar” if one of them is obtained by deleting one good from the second set or by replacing one good to another.

E.g. Set “1 2 3 4” is similar to sets “3 2 1”, “1 2 5 3 4”, “1 2 3 4 2” and “1 5 4 3” and is not similar to “1 2”, “1 1 2 2 3 4” and “4 5 3 6”.

This warehouse serves M shops ($0 < N < M < 101$), sending them sets of goods. Every two sets sent to the shop should not be similar. It is possible not to send any set to one or more shops.

You are to write program that determines how to distribute all K sets to these M shops.

Input

The first line contains numbers N, K, M. Then K lines describing every set of goods follow, $K \leq 50000$. Each of these lines is started with the number of goods in the set, then numbers of goods are written. Number of goods in any set is more than 0 and less than 101. All numbers in these lines are separated by exactly one space.

Output

The first line of the output should contain word YES if the solution exists or NO contrary. If the answer is YES write the numbers of the shops where sets should be sent to. In the second line you have to write number of the shop where the first set should be sent to, the third — for the second set, etc. If there are more than one solution exist you may find any of them.

Sample

input	output
8 20 12 5 1 3 5 6 4 5 1 3 5 6 3 4 5 6 3 3 4 5 6 3 4 4 4 6 5 8 4 7 7 7 7 3 7 7 7 2 2 2 3 2 2 7 3 1 2 3 3 1 2 4 10 1 2 3 4 5 6 7 8 7 6 10 8 7 6 5 4 3 2 1 2 1 20 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 1 3 5 7 5 4 6 4 6 4 5 6 4 6 4 6 6 6 6 6 6 6 6 3 6 6 6 1 1 1 2	YES 2 1 9 1 6 2 4 5 3 7 8 5 4 8 7 9 1 1 2 3

Problem Author: Dmitry Filimonenkov
Problem Source: Tetrahedron Team Contest May 2001

1108. Heritage

Time limit: 2.0 second
Memory limit: 64 MB

Your rich uncle died recently, and the heritage needs to be divided among your relatives and the church (your uncle insisted in his will that the church must get something). There are N relatives ($N \leq 18$) that were mentioned in the will. They are sorted in descending order according to their importance (the first one is the most important). Since you are the computer scientist in the family, your relatives asked you to help them. They need help, because there are some blanks in the will left to be filled. Here is how the will looks:

```
Relative #1 will get 1/... of the whole heritage,
Relative #2 will get 1/... of the whole heritage,
...
Relative #N will get 1/... of the whole heritage.
```

The logical desire of the relatives is to fill the blanks in such way that the uncle's will is preserved (i.e the fractions are non-ascending and the church gets something) and the amount of heritage left for the church is minimized.

Input

The only line of input contains the single integer N ($1 \leq N \leq 18$).

Output

Output the numbers that the blanks need to be filled (on separate lines), so that the heritage left for the church is minimized.

Sample

input	output
2	2 3

Problem Author: Pavlin Peev

1109. Conference

Time limit: 0.5 second
Memory limit: 64 MB

On the upcoming conference were sent M representatives of country A and N representatives of country B (M and $N \leq 1000$). The representatives were identified with 1, 2, …, M for country A and 1, 2, …, N for country B. Before the conference K pairs of representatives were chosen. Every such pair consists of one member of delegation A and one of delegation B. If there exists a pair in which both member #i of A and member #j of B are included then #i and #j can negotiate. Everyone attending the conference was included in at least one pair. The CEO of the congress center wants to build direct telephone connections between the rooms of the delegates, so that everyone is connected with at least one representative of the other side, and every connection is made between people that can negotiate. The CEO also wants to minimize the amount of telephone connections. Write a program which given M, N, K and K pairs of representatives, finds the minimum number of needed connections.

Input

The first line of the input contains M, N and K. The following K lines contain the choosen pairs in the form of two integers p_1 and p_2, p_1 is member of A and p_2 is member of B.

Output

The output should contain the minimum number of needed telephone connections.

Sample

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

Problem Source: Bulgarian National Olympiad Day #1

1110. Power

Time limit: 0.5 second
Memory limit: 64 MB

You are given the whole numbers N, M and Y. Write a program that will find all whole numbers X in the interval $[0, M-1]$ such that $X^N \bmod M = Y$.

Input

The input contains a single line with N, M and Y ($0 < N < 999$, $1 < M < 999$, $0 < Y < 999$) separated with one space.

Output

Output all numbers X separated with space on one line. The numbers must be written in ascending order. If no such numbers exist then output -1.

Sample

input	output
2 6 4	2 4

Problem Source: Bulgarian National Olympiad Day #1

1111. Squares

Time limit: 0.5 second
Memory limit: 64 MB

You are given n ($1 \le n \le 50$) squares and point P. The distance between P and square is the shortest line segment that connects P with the contour or the internal area of the square. If P is inside the square then the distance is zero. It is possible some squares to be points i.e. to have vertices that coincide. Write a program that will sort the squares in ascending order according the distance from P.

Input

The first line contains the integer n. The following n lines contain four integers in the range $(-9999, 9999)$. The first two numbers define the x and y coordinates of one of the vertices of the square, the next two numbers define the opposite vertex. The last line contains the x and y coordinates of P.

Output

The output should be a line containing the ids of the squares sorted according to the distance from P. The ids are defined according to the order in which the squares are given in the input. Use ids to break ties i.e. if two squares are the same distance from P then write the square with the lowest id first. Using 10^{-14} precision when comparing the distances is accurate enough.

Sample

input	output
2 0 0 1 1 0 3 1 4 0 0	1 2

1112. Cover

Time limit: 0.5 second
Memory limit: 64 MB

You are given N line segments on a line. Every segment is defined with its endpoints A_i and B_i ($A_i < B_i$, $1 \le i \le N$). Some of the segments probably intersect. Write a program, which removes minimum number of the given segments, so that none of the left segments have common interior point.

Input

The first line of input contains the integer N ($1 \le N \le 99$). Each of the following N lines, contains integers A_i and B_i ($-999 \le A_i < B_i \le 999$).

Output

On the first line write the integer P, equal to the number of segments, which are left after your program removes the excess segments. The following P lines should contain the coordinates of the left and the right endpoints of the segments which are left. These coordinates must be separated with one space. Coordinates of the left endpoints must be written in their ascending order. If the problem has more the one solution, write only one of them no matter which.

Sample

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

Problem Source: Bulgarian National Olympiad Day #2

1113. Jeep

Time limit: 0.5 second
Memory limit: 64 MB

Our jeep is in the desert and must reach a point which is N kilometres away from it. The terrain is tough, the car is old, and it seems like the fuel flows out, and because of this on every kilometre travelled, one litre of fuel is spent. But the jeep has fuel-cans and fuel tank with total capacity of $M < N$ litres. On other hand, at the beginning of the road there is unlimited amount of fuel, and everywhere in the desert there are empty cisterns, in which the jeep, passing near them, can leave unlimited amount of fuel.

Write a program which calculates the minimum amount of fuel in litres which is needed to reach the destination point.

Input

Two integers N and M are written on the only line of input. It is known that $5M \geq N > 0$. ($N < 32000$).

Output

The result (the minimum liters of fuel, eventually rounded up), must be written on the only line of output.

Sample

input	output
1000 500	3837

Problem Source: Bulgarian National Olympiad Day #2

1114. Boxes

Time limit: 0.6 second
Memory limit: 64 MB

N boxes are lined up in a sequence ($1 \le N \le 20$). You have A red balls and B blue balls ($0 \le A \le 15$, $0 \le B \le 15$). The red balls (and the blue ones) are exactly the same. You can place the balls in the boxes. It is allowed to put in a box, balls of the two kinds, or only from one kind. You can also leave some of the boxes empty. It's not necessary to place all the balls in the boxes. Write a program, which finds the number of different ways to place the balls in the boxes in the described way.

Input

Input contains one line with three integers N, A and B separated by space.

Output

The result of your program must be an integer written on the only line of output.

Sample

input	output
2 1 1	9

Problem Source: First competition for selecting the Bulgarian IOI team.

1115. Ships

Time limit: 1.0 second
Memory limit: 64 MB

The military intelligence of one country found out that N ($N < 100$) battle ships of neighboring enemy country are situated in M rows ($1 < M < 10$). The intelligence knows the lengths $l_1, l_2, \ldots, l_N$ of the battle ships which are whole numbers in the interval [1, 100], and wants to know in which rows the ships are situated. The only thing that is known about the M rows are their lengths — $L_1, L_2, \ldots, L_M$. Assume that the ships touch their neighbours in the rows and that every row contains at least one ship. Write program that will find one possible ordering of the ships in rows.

Input

The first line of the input contains N and M. The next N lines contain the lengths of the ships. The next M lines contain the lengths of the rows.

Output

The output should contain M pairs of lines. The first line of each pair should contain the amount of the ships in the current row, the following line should contain the lengths of the ships from the current row. The order of the M row descriptions should be the same as the order in which the rows are given in the input.

Sample

input	output
5 2 4 10 2 5 3 11 13	3 5 4 2 2 10 3

Problem Source: First competition for selecting the Bulgarian IOI team.

1116. Piecewise Constant Function

Time limit: 0.5 second
Memory limit: 64 MB

SKB Kontur has been developing a new *SKB Kontur Framework* for the last three months. One of the latest wrinkles is that data will be presented with the help of piecewise constant functions. Your team is to implement an operation of "subsettraction" of the functions. A function is called piecewise constant if its domain can be divided into intervals and the function is constant on each interval. We also assume that the function value at the left-end point of each interval of constancy is equal to its value on the interval. In fig.1 there is a piecewise constant function with three intervals of constancy. Note that the function value at the point B is U and at the points C, E and on the interval (C,D) - the function value is not defined.

A result of the subsettraction operation of two piecewise constant functions F_1 @ F_2 is a piecewise constant function F defined as follows:

- $F(x) = F_1(x)$ if F_1 is defined and F_2 is not defined;
- $F(x)$ is not defined if F_1 and F_2 are defined both;
- $F(x)$ is not defined if F_1 is not defined.

In Fig.3 there is the result of the operation of the subsettraction of the two functions from Fig.1 and Fig.2.

Input

contains two lines of the same format. Each line characterizes one piecewise constant function. A line begins with an integer N ($1 \le N \le 15000$). Then characterizations of constancy intervals follow in the ascending order with respect to their left ends. Each interval is given by three integer numbers A, B, Y ($|A| < 32000$, $|B| < 32000$, $|Y| \le 100$, $A < B$), where A is the left end of an interval, B is the right end of an interval and Y is the function value on the interval. It's known that that no two intervals from one line intersect. If two intervals are adjacent, the function values on the intervals are different.

Output

contains one line of the same format (see the input specification). This line should describe a result of the operation of subsettraction of the two input piecewise constant functions.

Sample

input	output
3 -1 1 2 1 3 4 4 6 3 2 -2 2 1 5 7 5	2 2 3 4 4 5 3

Problem Author: Oleg Kaz
Problem Source: USU Open Collegiate Programming Contest October'2001 Junior Session

1117. Hierarchy

Time limit: 1.0 second
Memory limit: 64 MB

During long years of work in SKB Kontur a certain hierarchy of employees has been developed. Each person except ordinary employees has exactly two direct subordinates and not more than one direct superior. There is no subordinate of ordinary employees (see figure).

Each employee has his own number. Of course, different employees have different numbers. It's known as well that either an employee has the maximal number or there is another employee whose number is greater by one. Similarly, either an employee has a number "1" or there is another employee whose number is less by one. The number of intermediate levels of employees between an arbitrary employee who has subordinates (an ordinary employee) and the employee who has no superiors (the main superior) is the same for all ordinary employees.

The things have come round so that each employee who has subordinates has got a number greater than the number of one of his subordinates and less than the number of the other. Moreover, if his number is greater than his superior's then the numbers of his subordinates are also greater than the number of his superior. And conversely, if his number is less, then his subordinates' numbers are less too.

A special system of intracorporate message exchange has been worked out. A message from an employee with the number i can be addressed directly only to the employees $i-1$ and $i+1$. Moreover, this is done the same day (it takes 0 days to do that) if the employees are direct superior and subordinate. Otherwise, the message delivery takes an amount of days that is equal to the number of intermediate employees between the sender and recipient. For example, a message from the employee 2 to the employee 4 is being delivered as follows. The employee 2 sends the message to the employee 3, and the employee 3 addresses it to the employee 4. This process takes one day because the first step (2->3) takes 0 days, and the second one (3->4) takes 1 day.

Input

The only line contains two positive integers: the number of an employee who sends a message and the number of the recipient. Each of the numbers doesn't exceed $2^{31}-1$.

Output

You should output the only number — the number of days necessary to deliver the message.

Sample

input	output
1 5	2

Problem Author: Alexander Somov
Problem Source: USU Open Collegiate Programming Contest October'2001 Junior Session

1118. Nontrivial Numbers

Time limit: 2.0 second
Memory limit: 64 MB

Specialists of SKB Kontur have developed a unique cryptographic algorithm for needs of information protection while transmitting data over the Internet. The main advantage of the algorithm is that you needn't use big numbers as keys; you may easily do with natural numbers not exceeding a million. However, in order to strengthen endurance of the cryptographic system it is recommended to use special numbers - those that psychologically seem least "natural". We introduce a notion of *triviality* in order to define and emphasize those numbers.

Triviality of a natural number N is the ratio of the sum of all its proper divisors to the number itself. Thus, for example, triviality of the natural number 10 is equal to 0.8 = (1 + 2 + 5) / 10 and triviality of the number 20 is equal to 1.1 = (1 + 2 + 4 + 5 + 10) / 20. Recall that *a proper divisor* of a natural number is the divisor that is strictly less than the number.

Thus, it is recommended to use as nontrivial numbers as possible in the cryptographic protection system of SKB Kontur. You are to write a program that will find the less trivial number in a given range.

Input

The only line contains two integers I and J, $1 \le I \le J \le 10^6$, separated with a space.

Output

Output the only integer N satisfying the following conditions:

1. $I \le N \le J$;
2. N is the least trivial number among the ones that obey the first condition.

Sample

input	output
24 28	25

Problem Author: Leonid Volkov
Problem Source: USU Open Collegiate Programming Contest October'2001 Junior Session

1119. Metro

Time limit: 0.5 second
Memory limit: 64 MB

Many of SKB Kontur programmers like to get to work by Metro because the main office is situated quite close the station Uralmash. So, since a sedentary life requires active exercises off-duty, many of the staff — Nikifor among them — walk from their homes to Metro stations on foot.

Nikifor lives in a part of our city where streets form a grid of residential quarters. All the quarters are squares with side 100 meters. A Metro entrance is situated at one of the crossroads. Nikifor starts his way from another crossroad which is south and west of the Metro entrance. Naturally, Nikifor, starting from his home, walks along the streets leading either to the north or to the east. On his way he may cross some quarters diagonally from their south-western corners to the north-eastern ones. Thus, some of the routes are shorter than others. Nikifor wonders, how long is the shortest route.

You are to write a program that will calculate the length of the shortest route from the south-western corner of the grid to the north-eastern one.

Input

There are two integers in the first line: N and M ($0 < N,M \le 1000$) — west-east and south-north sizes of the grid. Nikifor starts his way from a crossroad which is situated south-west of the quarter with coordinates (1, 1). A Metro station is situated north-east of the quarter with coordinates (N, M). The second input line contains a number K ($0 \le K \le 100$) which is a number of quarters that can be crossed diagonally. Then K lines with pairs of numbers separated with a space follow — these are the coordinates of those quarters.

Output

Your program is to output a length of the shortest route from Nikifor's home to the Metro station in meters, rounded to the integer amount of meters.

Sample

input	output
3 2 3 1 1 3 2 1 2	383

Problem Author: Leonid Volkov
Problem Source: USU Open Collegiate Programming Contest October'2001 Junior Session

1120. Sum of Sequential Numbers

Time limit: 0.5 second
Memory limit: 64 MB

There is no involute formulation concerning factitiously activity of SKB Kontur in this problem. Moreover, there is no formulation at all.

Input

There is the only integer S, $1 \le S \le 10^9$.

Output

Your program is to output two positive integers A and N separated with a space such that:

1. $S = A + (A + 1) + \ldots + (A + N - 1)$.
2. You are to choose a pair with the maximal possible value of N.

Sample

input	output
14	2 4

Notes

14 = 2 + 3 + 4 + 5

Problem Author: Leonid Volkov
Problem Source: USU Open Collegiate Programming Contest October'2001 Junior Session

1121. Branches

Time limit: 1.0 second
Memory limit: 64 MB

SKB Kontur has a lot of branches scattered all over the city. The company management decided to create a guide that will help clients to choose which of the branches they need. You are asked for help in this work.

The city is represented in the form of a grid of blocks. Each block is a square whose sides are streets and whose corners are cross-roads. We suppose that all branches are located exactly at cross-roads. The branches of SKB Kontur are of different types: service centers, warehouses, shops, training centers and so on.

Let's mark service centers with number 1, warehouses with number 2, shops with number 4, training centers with number 8 and so on. There are not more than 11 types of branches, and two branches of the same type cannot be located at the same cross-road. Each cross-road is assigned a number equal to the sum of numbers with which the branches located at this cross-road are marked. Crossroads at which there are no branches of SKB Kontur are assigned 0.

Let the distance between two cross-roads be equal to the number of street segments which one has to go from the first cross-road to the second (see picture). For example, the distance from a corner of a block to the opposite corner of this block is 2. For each cross-road at which there are no branches of SKB Kontur you have to find the sum of the numbers corresponding to the types of the branches nearest to this cross-road. For example, suppose that there are no branches at a given cross-road and at distance 1 from it, there is a branch of type 16 at distance 2, there are also two branches of type 8 and one of type 4 at distance 2 in other directions and there are no more branches at distance 2 from this cross-road. Then we should output number 28=16+8+4 for this cross-road. We do not take into consideration branches that are at distances greater than 5 from a given cross-road. Thus, if a cross-road does not have branches of SKB Kontur that are located at distances less than 6 from it then we should output 0 for this cross-road.

Input

The first line contains positive integers H and W not exceeding 150. They are numbers of "vertical" and "horizontal" streets, correspondingly. The next H lines contain W numbers each, the i-th number in the j-th line describing types of the branches located at the cross-road of the i-th "vertical" and the j-th "horizontal" street.

Output

You should output H lines containing W numbers each, the i-th number in the j-th line being equal to the sum of the numbers corresponding to the types of the branches nearest to the corresponding cross-road if there are no branches at this cross-road and -1 otherwise.

Sample

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

Problem Author: Leonid Volkov, Alexander Somov
Problem Source: USU Open Collegiate Programming Contest October'2001 Junior Session

1122. Game

Time limit: 1.0 second
Memory limit: 64 MB

At SKB Kontur we have to work much. So there is no sin in taking a rest and playing from time to time. Consider for example the following famous one-player game.

We have a 4×4 field. There are chips with one side painted white and another side painted black on the field. Some of the chips are with their white side up and the others are with their white side down at the moment. Each move consists in turning over a chip together with all the chips that are adjacent to it vertically and horizontally (i.e. 5 chips altogether). The aim is to come to the position in which all the chips are with the same side up.

Naturally, one is easily bored with this game because interesting and unexpected positions become fewer as time goes on. That is why a modified version of the game is now more popular at SKB Kontur. In this version a move consists in turning over a fixed combination of chips within a 3×3 square. For example, a move may consist in turning over all the diagonal neighbors of a chosen chip.

The combination of chips is chosen arbitrarily; it may be assigned in the form of a 3×3 field in which the central cell corresponds to the cell at which a move as made. For example, in picture at the left the upper combination corresponds to a standard game and the lower combination is for the game described in the previous paragraph. Note that a combination can be asymmetrical. Each move is made at one of the cells of the playing field (i.e. the central cell of the 3×3 move-defining square is selected among the field's cells). Prescriptions to turn over chips at cells which are outside the 4×4 field are ignored.

In this game it would be nice to know if it is possible to reach a position in which all the chips are with the same side up and if it's possible to do this then in how many moves. You are to write a program which answers these questions.

Input

The first four lines describe the initial arrangement of chips. A symbol "W" stands for a chip which lies with its white side up and a symbol "B" stands for a chip which lies with its black side up. The next three lines describe a move: the chips that are to be turned over are shown by "1" and others are shown by "0".

Output

If it is impossible to reach the aim of the game you should output the word "Impossible", otherwise you should output the minimal number of moves necessary to come to the final position.

Sample

input	output
WWWW WBBW WBWW WWWW 101 010 101	Impossible

Problem Author: Leonid Volkov, Oleg Kats, Alexander Somov
Problem Source: USU Open Collegiate Programming Contest October'2001 Junior Session

1123. Salary

Time limit: 1.0 second
Memory limit: 64 MB

All employees of SKB Kontur like to get their salaries. Often and in large quantities. But the company management is of a bit different opinion and pays out strictly once a month. After some consultations the employees decided that if one of the parameters (frequency of payment) was fixed it was possible to change the second parameter (amount of the money paid out). They contrived the following scheme. A group of employees who proudly call themselves mathematics and mechanics faculty graduates visits the management and using their mathematical authority claims that the computers in the company's accounts department will work more efficiently if salaries of all the employees take the form of palindromes. As you know, a numerical palindrome is a number that does not change when you read it from right to left. For example, 12344544321 is a palindrome and 12345543210 is not. Of course, the management had to agree with this proposal, but upon one condition: each employee had to re-count his or her salary so that the salary took the form of the least possible palindrome that is greater than or equal to the original salary. You are asked to help the employees of SKB Kontur.

Input

consists of one string containing the original salary of an employee. The string is not longer than 2001 symbols.

Output

should consist of one string containing the new salary calculated according to the above rules.

Sample

input	output
12341321	12344321

Problem Author: Leonid Volkov, Oleg Kats
Problem Source: USU Open Collegiate Programming Contest October'2001 Junior Session

1124. Mosaic

Time limit: 0.25 second
Memory limit: 64 MB

There's no doubt that one of the most important and crucial things to do in this world is to bring up children. May be, if you study properly and reach good results at the competition you'll get a position of nanny in a kindergarten. But you are to get ready for it! Let's consider some problems that a nanny has to solve in a kindergarten.

Everyone knows the game "Mosaic". Playing the game, one is to lay out pictures of different colored pieces. Let there be M different boxes and N mosaic pieces of each of the M colors. After playing the game children rarely put the pieces back to their boxes correctly so that the color of the box and the colors of its pirces would be the same. A nanny has to do that.

Children have already put the mosaic pieces to the boxes but possibly not correctly. There are N pieces in each box. Some pieces (possibly all of them) are located in wrong boxes (i.e. boxes with pieces of a different color). Moving a hand once one can take a piece from one box to another or simply move the hand to another box. You may start from any box you like. The movement towards the first box is not taken into account. Find out the minimal number of movements one needs to put all the mosaic pieces to their boxes.

Input

The first line contains integers $2 \le M \le 500$ (the number of colors) and $2 \le N \le 50$ (the number of pieces of each color). Each of the next M lines contains N integers in the range from 1 to M (the i+1-st line contains colors of pieces located in the i-th box).

Output

the minimal possible number of hand movements that one has to make in order to take all the pieces to their boxes.

Sample

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

Problem Author: Stanislav Vasilyev
Problem Source: VI Ural State University Collegiate Programming Contest (21.10.2001)

1125. Hopscotch

Time limit: 0.25 second
Memory limit: 64 MB

Nikifor likes to play hopscotch in the kindergarten. The playing field is a rectangle $M \times N$ partitioned into cells 1×1 meter. Nikifor hops from one cell to another possibly not adjacent cell. Each cell is colored black or white. Each time Nikifor hops into a cell, all cells whose centers are at an integer amount of meters away from Nikifor's cell center reverse their colors. You are given the final colors of the playing field cells. You also know the number of times Nikifor has been at each cell. Your task is to restore the initial colors of the cells.

Input

The first line contains two nonnegative integers M and N that do not exceed 50. The next M lines contain a character table $M \times N$. This table describes the final coloring of the field. Character 'W' denotes the white color of the cell, and 'B' denotes the black color. There are no other characters in the table. The next M lines contain a matrix with nonnegative integer elements. Each element shows how many times Nikifor has been at the corresponding cell. Numbers in the lines are separated with a space and do not exceed $2 \cdot 10^9$.

Output

should consist of M lines. The lines should contain a character table that shows the initial coloring of the playing field.

Sample

input	output
6 6 BWBBWW BWBBWB BBWWBW BBBBBW BBWWWW BBWBBW 2 0 12 46 2 0 3 0 0 0 0 200 4 2 1 1 4 2 4 2 1 1 4 4 0 0 0 0 0 0 2 56 24 4 2 2	WWBBWW WBWWBW WBBBBW WBWWBW WBWWBW WBWWBW

Problem Author: Dmitry Filimonenkov
Problem Source: VI Ural State University Collegiate Programming Contest (21.10.2001)

1126. Magnetic Storms

Time limit: 0.5 second
Memory limit: 64 MB

The directory of our kindergarten decided to be attentive to the children's health and not to let them walk outdoors during magnetic storms. Special devices that measure and display magnetic intensity were ordered. If the readout exceeded some certain level the children were told to go indoors. They disliked it because they couldn't play their games up to the end. The nannies hated it because they had to dress and undress children many times.

After a while it became clear that one could try to forecast magnetic intensity because long periods of quietude alternated with short periods of plenty of sharp peaks (this is called a magnetic storm). Hence a new modification of the devices was ordered.

The new devices were to remember the situation within several last hours and to display the maximal intensity during the period. If the intensity was low within the last 6 hours the magnetic field was regarded to be quiet; the children were let outdoors and played all the prescript time. Otherwise new peaks were probable and the children spent their time indoors.

Your task is to write a program for a new version of the device. As a matter of fact you are to solve just the main problem of modification. All the rest is already done.

You are given a number M which is length of a period (in seconds) within which peaks are to be stored and displayed. A sequence of measured magnetic intensity values is given to you as well. Each measurement is a number within the range from 0 to 100000.

Your are to output a sequence of values displayed by the device. The first number of the sequence is the maximal element of the first M input numbers, the second number is the maximal element of the 2nd, …, M+1-st input numbers and so on.

We hope that the new devices with your program won't go back on nannies and children will not walk during magnetic storms.

Input

The first line contains a number M, $2 \le M \le 14000$. Then values (N integers) measured by the device follow each one in its line. There is a number -1 in the end. $M \le N \le 25000$.

Output

a sequence of readouts, each one in its line.

Sample

input	output
3 10 11 10 0 0 0 1 2 3 2 -1	11 11 10 0 1 2 3 3

Problem Author: Alexander Mironenko
Problem Source: VI Ural State University Collegiate Programming Contest (21.10.2001)

1127. Colored Bricks

Time limit: 0.4 second
Memory limit: 64 MB

There are lots of cubic bricks in the kindergarten. Children like to build toy brick towers and then to drop them. It is clear that the higher tower has been built the more interesting it is to drop it. The tower is built by placing bricks one onto another and aligning their sides. The tower is based on one brick. Thus the height of a tower is the number of the bricks it is built of. Each side of a brick is painted in one color. So the kids build colored towers. In order to train the children's sense of beauty nannies teach them to build the towers in such a way that each side of the tower would be one-color. Thus the kids would like to build a tower with one-color sides as high as possible.

Every nanny can easily solve this problem. Try your best to do it as well.

Input

The first line contains a number N ($1 < N \le 10^3$) — the number of bricks. The next N lines contain descriptions of bricks. Each brick is described with a string of 6 capital latin letters denoting the color of the corresponding side (A — Azure, B — Blue, C — Cyan, G — Green, O — Orange, R — Red, S — Scarlet, V — Violet, W — White, Y — Yellow). The colors of the sides are given in the following order: *front, right, left, rear, top, bottom.* A brick never has two sides of the same color.

Output

Output the only number — the maximal height of a toy tower that can be built of the given brick set.

Sample

input	output
4 GYVABW AOCGYV CABVGO OVYWGA	3

Problem Author: Ekaterina Vasilyeva
Problem Source: VI Ural State University Collegiate Programming Contest (21.10.2001)

1128. Partition into Groups

Time limit: 0.5 second
Memory limit: 64 MB

There are N children in the kindergarten. Unfortunately, the children quarrel though not often. Each child has not more than three adversaries. Is it possible to partition the children into two groups (possibly not equal), so that each child would have not more than one adversary in his or her group?

Input

The first line contains an integer N, $0 < N \leq 7163$. The next N lines contain lists of adversaries of each child. A line starts with the amount of the corresponding child's adversaries, then the numbers of the adversaries follow. The numbers in each line are separated with a space.

Output

The first line contains the number of children in the smaller group. The next line contains the list of children in the group. The numbers in the second line are separated with a space. If the groups are of the same size then you are to describe the group that contains the child number one. Note that the output may contain the only number 0. If there are several possible partitions it's sufficient to output an arbitrary one. If there's no possible partition you are to output the only string “NO SOLUTION”.

Sample

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

Problem Author: Dmitry Filimonenkov
Problem Source: VI Ural State University Collegiate Programming Contest (21.10.2001)

1129. Door Painting

Time limit: 0.25 second
Memory limit: 64 MB

There are many rooms, corridors and doors between them in the kindergarten. Some repairs are planned to be made soon. The doors are agreed to be painted in bright cheerful colors: green and yellow. The matron of the kindergarten wants the doors to satisfy the following condition: the sides of an arbitrary door must have the different colors. The number of green doors in each of the lodgings must differ from the number of yellow doors not more than by one. Given the plan of the kindergarten suggest your scheme of door painting.

Input

The first line contains the number of lodgings $N \le 100$ in the kindergarten. The next N lines contain description of the door configuration (k+1-st line contains a description of the k-th lodging). Each of the N lines starts with the number of doors that connect this lodging with adjacent ones. Then there are numbers of adjacent lodgings separated with a space (these numbers follow in ascending order).

Output

should contain a required painting scheme or the word “Impossible” if it is impossible to satisfy the requirements. The colors of the K-th room doors should be put in the K-th line in the same order as they were in the input data. The green color is denoted by G, yellow — by Y.

Sample

input	output
5 3 2 3 4 3 1 3 5 4 1 2 4 5 3 1 3 5 3 2 3 4	G Y G Y G Y G Y Y G Y G G G Y Y

Problem Author: Magaz Asanov
Problem Source: VI Ural State University Collegiate Programming Contest (21.10.2001)

1130. Nikifor's Walk

Time limit: 0.25 second
Memory limit: 64 MB

Little Nikifor wouldn't stay long without movement. It's boring to run in one direction for a long time, as well. A wise nanny knows that when Nikifor goes playing outdoors he moves along vectors a_1, a_2, …, a_n; each time his displacement is either equal to the next in turn vector or to the vector opposite to it. A pedagogical influence of the nanny with Nikifor is rather strong, so each time she can point out which one of the two possible directions he should choose.

The nanny knows that length of each of the vectors a_1, a_2, …, a_n doesn't exceed L. Nikifor starts his walk from the nanny and she wants him to move off her not farther than by the square root of 2 multiplied by L (sqrt(2)L) in the end of his walk. What directions should she point out in order not to let the child move too far off her pedagogical influence in the end of the walk?

Input

The first line contains an integer n, $0 < n \le 10000$. The second line contains a non-negative integer L, $L < 100$. The next n lines contain coordinates of the vectors. The coordinates are integer.

Output

The first line is to contain the word "YES" if the nanny can cope with her task, and "WRONG ANSWER" otherwise. If the answer is "YES" then the next line should consist of n symbols "+" or "-". There is the symbol "+" at the i-th position if Nikifor runs along the vector a_i, and there's a symbol "-" if Nikifor runs along the vector $-a_i$. If there are several solutions it's enough to output an arbitrary one.

Sample

input	output
4 5 5 0 0 5 0 0 -3 4	YES +-++

Problem Author: Dmitry Filimonenkov
Problem Source: VI Ural State University Collegiate Programming Contest (21.10.2001)

1131. Copying

Time limit: 0.25 second
Memory limit: 64 MB

A new educating program was received by the kindergarten. Of course, children have discovered it immediately and want to play with it as soon as possible. In order to let them do it the program has to be copied to all the N computers that the kindergarten had bought just before the default of 1998. At the moment the program is installed only on one computer. Other computers do not have floppy drives and are not connected with a local network. The only way to transfer information from one computer to another is to copy it using a null-modem cable (a cable that connects two computers directly). So, if the program is installed on a computer, it can be copied to some other (but only one!) computer within an hour. There are only K null-modem cables in the kindergarten. Your task is to estimate the minimal time necessary for copying the program to all the computers in the kindergarten.

Input

The only input line contains two integers separated with a space: N and K ($1 \le N \le 10^9$; $1 \le K \le 10^9$).

Output

You are to output the minimal time (in hours) necessary for copying of the program to all the computers.

Sample

input	output
8 3	4

Problem Author: Stanislav Vasilyev, Alexander Mironenko
Problem Source: VI Ural State University Collegiate Programming Contest (21.10.2001)

1132. Square Root

Time limit: 1.0 second
Memory limit: 64 MB

The number x is called a square root of a modulo n (root(a,n)) if $x*x = a \pmod{n}$. Write the program to find the square root of number a by given modulo n.

Input

One number K in the first line is an amount of tests ($K \le 100000$). Each next line represents separate test, which contains integers a and n ($1 \le a, n \le 32767$, n is prime, a and n are relatively prime).

Output

For each input test the program must evaluate all possible values root(a,n) in the range from 1 to $n - 1$ and output them in increasing order in one separate line using spaces. If there is no square root for current test, the program must print in separate line: 'No root'.

Sample

input	output
5 4 17 3 7 2 7 14 31 10007 20011	2 15 No root 3 4 13 18 5382 14629

Problem Author: Michael Medvedev

1133. Fibonacci Sequence

Time limit: 1.0 second
Memory limit: 64 MB

is an infinite sequence of integers that satisfies to Fibonacci condition $F_{i+2} = F_{i+1} + F_i$ for any integer i. Write a program, which calculates the value of F_n for the given values of F_i and F_j.

Input

The input contains five integers in the following order: i, F_i, j, F_j, n.
$-1000 \le i, j, n \le 1000$, $i \ne j$,
$-2 \cdot 10^9 \le F_k \le 2 \cdot 10^9$ ($k = \min(i, j, n), \ldots, \max(i, j, n)$).

Output

The output consists of a single integer, which is the value of F_n.

Sample

input	output
3 5 -1 4 5	12

Notes

In the example you are given: $F_3 = 5$, $F_{-1} = 4$; you asked to find the value of F_5. The following Fibonacci sequence can be reconstructed using known values:

$$\ldots, F_{-1} = 4, F_0 = -1, F_1 = 3, F_2 = 2, F_3 = 5, F_4 = 7, F_5 = 12, \ldots$$

Thus, the answer is: $F_5 = 12$.

Problem Source: Quarterfinal, Central region of Russia, Rybinsk, October 17-18 2001

1134. Cards

Time limit: 1.0 second
Memory limit: 64 MB

Each of the n cards has numbers written on the both sides of it. The first card has 0 and 1 on it, the second has 1 and 2, …, the n-th has $(n - 1)$ and n. First-grade pupil Nick takes cards one by one in random order and reads the number on one of the sides. Nick is not very good with numbers, so it is possible that he makes a mistake. Your task is to find out if he was mistaken, i.e. if the given sequence of numbers is possible for some order of taking cards.

Input

The first line contains integers n, the total number of cards, and m, the number of the cards that were taken ($2 \le n \le 1000$; $1 \le m \le n$). In the second line m integers are listed (the sequence read by Nick). All these integers are from 0 to n.

Output

Write “YES” if the given sequence of numbers is possible for some order of taking cards, “NO” otherwise.

Sample

input	output
5 4 2 0 1 2	NO

Problem Source: Quarterfinal, Central region of Russia, Rybinsk, October 17-18 2001

1135. Recruits

Time limit: 1.0 second
Memory limit: 64 MB

N recruits are standing in front of a sergeant who orders to turn left. Some of the soldiers turn left, while the others turn right. In a second each recruit seeing the face of another recruit understands that a mistake was made and turns around. This happens at the same time to each pair of soldiers facing each other. The process continues until the formation becomes stable. Write a program, which finds out the number of times when a pair of soldiers turned around. If the process is infinite then the program should write the word “NO”.

Example:
Legend:
‘<’: a recruit facing left;
‘>’: a recruit facing right.

Formation	Comments	Number of turns
> > < < > <	Initial formation	2
> < > < < >	One second has passed	2
< > < > < >	Two seconds have passed	2
< < > < > >	Three seconds have passed	1
< < < > > >	Final formation	Total: 7

Input

The first line contains the number of recruits (*N*). The rest of the input contains only ‘<’, ‘>’ and line break characters. There is exactly *N* ‘<’ and ‘>’ characters in the input. Each line of the input may have up to 255 characters.
$1 \le N \le 30000$.

Output

Write the number of turns.

Sample

input	output
6 >><<><	7

Problem Source: Quarterfinal, Central region of Russia, Rybinsk, October 17-18 2001

1136. Parliament

Time limit: 1.0 second
Memory limit: 64 MB

A new parliament is elected in the state of MMMM. Each member of the parliament gets his unique positive integer identification number during the parliament registration. The numbers were given in a random order; gaps in the sequence of numbers were also possible. The chairs in the parliament were arranged resembling a tree-like structure. When members of the parliament entered the auditorium they took seats in the following order. The first of them took the chairman's seat. Each of the following delegates headed left if his number was less than the chairman's, or right, otherwise. After that he took the empty seat and declared himself as a wing chairman. If the seat of the wing chairman has been already taken then the seating algorithm continued in the same way: the delegate headed left or right depending on the wing chairman's identification number.

The figure below demonstrates an example of the seating of the members of parliament if they entered the auditorium in the following order: 10, 5, 1, 7, 20, 25, 22, 21, 27.

During its first session the parliament decided not to change the seats in the future. The speech order was also adopted. If the number of the session was odd then the members of parliament spoke in the following order: the left wing, the right wing and the chairman. If a wing had more than one parliamentarian then their speech order was the same: the left wing, the right wing, and the wing chairman. If the number of the session was even, the speech order was different: the right wing, the left wing, and the chairman. For a given example the speech order for odd sessions will be 1, 7, 5, 21, 22, 27, 25, 20, 10; while for even sessions — 27, 21, 22, 25, 20, 7, 1, 5, 10.

Determine the speech order for an even session if the speech order for an odd session is given.

Input

The first line of the input contains N, the total number of parliamentarians. The following lines contain N integer numbers, the identification numbers of the members of parliament according to the speech order for an odd session.

The total number of the members of parliament does not exceed 3000. Identification numbers do not exceed 65535.

Output

The output should contain the identification numbers of the members of parliament in accordance with the speech order for an even session.

Sample

input	output
9 1 7 5 21 22 27 25 20 10	27 21 22 25 20 7 1 5 10

Problem Source: Quarterfinal, Central region of Russia, Rybinsk, October 17-18 2001

1137. Bus Routes

Time limit: 1.0 second
Memory limit: 64 MB

Several circular bus routes were in the city of Fishburg. None of the routes shared the same section of road, though common stops and intersections were possible. Fishburg old residents stated that it was possible to move from any stop to any other stop (probably making several transfers). The new mayor of the city decided to reform the city transportation system. He offered that there would be only one circular route going through all the sections where buses moved in the past. The direction of movement along the sections must be the same and no additional sections should be used.

Write a program, which creates one of the possible new routes or finds out that it is impossible.

Input

The first line contains the number of old routes n ($1 \le n \le 100$). Each of the following n lines contains the description of one route: the number of stops m ($2 \le m \le 200$) and the list of that stops. Bus stops are identified by integers from 1 to 1000. A route is represented as a sequence of $m + 1$ bus stop identifiers: $l_1, l_2, \ldots, l_m, l_{m+1} = l_1$ that are sequentially visited by a bus moving along this route. A route may be self-intersected. A route always ends at the same stop where it starts.

Output

The output contains the number of stops in the new route k and the new route itself in the same format as in the input. The last (k+1)-th stop must be the same as the first. If it is impossible to make a new route according to the problem statement then write 0 (zero) to the output.

Sample

input	output
3 6 1 2 5 7 5 2 1 4 1 4 7 4 1 5 2 3 6 5 4 2	15 2 5 4 2 3 6 5 7 4 1 2 1 4 7 5 2

Notes

Here is a picture for the example:

Problem Source: Quarterfinal, Central region of Russia, Rybinsk, October 17-18 2001

1138. Integer Percentage

Time limit: 1.0 second
Memory limit: 64 MB

Applying for a new job, programmer N. Smart required that his new salary (in rubles, positive integer) would be greater than his previous salary by integer percentage. What could be the highest possible number of previous jobs for mister Smart, if his latest salary did not exceed n rubles and his first salary was exactly s rubles?

Example. Let $n = 10, s = 2$, then $m = 5$. The sequence 2, 4 (+100%), 5 (+25%), 8 (+60%), 10 (+25%) is the longest (although not unique) sequence that satisfies to the problem statement. Salary increase percentage is written inside the brackets.

Input

Two integers n and s separated by one or more spaces. $1 \le n, s \le 10000$.

Output

A single integer m — the maximum number of N. Smart's previous jobs.

Sample

input	output
10 2	5

Notes

if $n = s$, the answer is 1.

Problem Source: Quarterfinal, Central region of Russia, Rybinsk, October 17-18 2001

1139. City Blocks

Time limit: 1.0 second
Memory limit: 64 MB

The blocks in the city of Fishburg are of square form. N avenues running south to north and M streets running east to west bound them. A helicopter took off in the most southwestern crossroads and flew along the straight line to the most northeastern crossroads. How many blocks did it fly above?

Note. A block is a square of minimum area (without its borders).

Input

The input contains N and M separated by one or more spaces. $1 < N, M < 32000$.

Output

The number of blocks the helicopter flew above.

Samples

input	output
4 3	4
3 3	2

Notes

The figures for samples:

Problem Source: Quarterfinal, Central region of Russia, Rybinsk, October 17-18 2001

1140. Swamp Incident

Time limit: 1.0 second
Memory limit: 64 MB

An attempt was made to count all the cranberries in the swamps located in the surroundings of Rybinsk. It appeared convenient to split the surface of the swamp into regular hexagonal cells. One of these cells was considered as the central one, where the helicopter hovered. Three directions were marked (see picture).

After that a hired student landed in the central cell and wandered around for a certain time. He counted the berries and recorded his movements as he walked. Movements were recorded as a sequence of transitions from one cell to another through their common side along one of the marked (or reverse) directions. The route consisted of linear sections determined by directions (*X*, *Y*, or *Z*) and lengths (signed nonzero integers). A movement in the marked direction is represented with positive numbers, in the reverse direction — with negative numbers.

Your task is to write a program, which determines a route from the last cell visited by the student back to the central cell, having the least possible number of cells in it.

Input

The first line of the input contains n — the length of the route ($n > 0$). Each of the following n lines contains a letter denoting a direction (*X*, *Y*, or *Z*) and a signed integer l ($l \neq 0$) denoting the length of the section (in cells). The letter and the number are separated by one space.

While wandering, the student moved away from the central cell for no more than 100 cells in each of marked and reverse directions. The total length of the route does not exceed 32000 linear sections.

Output

The output must contain the description of a route from the last cell visited by the student back to the central cell, having the least possible number of cells in it.

The first line of the output must contain m — the length of the route (number of sections in the back route, $m \geq 0$). The following m lines of the output must contain the description of the sections of the back route in the same format as in the input.

Sample

input	output
4 Z -2 Y 3 Z 3 X -1	2 Y -2 Z -2

Problem Source: Quarterfinal, Central region of Russia, Rybinsk, October 17-18 2001

1141. RSA Attack

Time limit: 1.0 second
Memory limit: 64 MB

The RSA problem is the following: given a positive integer n that is a product of two distinct odd primes p and q, a positive integer e such that gcd(e, (p-1)*(q-1)) = 1, and an integer c, find an integer m such that $m^e = c \pmod n$.

Input

One number K (K <= 2000) in the first line is an amount of tests. Each next line represents separate test, which contains three positive integer numbers – e, n and c (e, n, c <= 32000, n = p*q, p, q – distinct odd primes, gcd(e, (p-1)*(q-1)) = 1, e < (p-1)*(q-1)).

Output

For each input test the program must find the encrypted integer m.

Sample

input	output
3 9 187 129 11 221 56 7 391 204	7 23 17

Problem Author: Michael Medvedev

1142. Relations

Time limit: 1.0 second
Memory limit: 64 MB

Background

Consider a specific set of comparable objects. Between two objects a and b, there exits one of the following three classified relations:

$$a = b$$
$$a < b$$
$$b < a$$

Because relation '=' is symmetric, it is not repeated above.

So, with 3 objects (a, b, c), there can exist 13 classified relations:

$$a = b = c \quad a = b < c \quad c < a = b \quad a < b = c$$
$$b = c < a \quad a = c < b \quad b < a = c \quad a < b < c$$
$$a < c < b \quad b < a < c \quad b < c < a \quad c < a < b$$
$$c < b < a$$

Problem

Given N, determine the number of different classified relations between N objects.

Input

Includes many integers N (in the range from 2 to 10), each number on one line. Ends with −1.

Output

For each N of input, print the number of classified relations found, each number on one line.

Sample

input	output
2 3 -1	3 13

1143. Electric Path

Time limit: 1.0 second
Memory limit: 64 MB

Background

At the team competition of the 10th national student informatics Olympic, which is organized at Hanoi National University, there are N teams participating. Each team is assigned to work in a camp. On the map, it can be seen that the camps are positioned on the vertices of a convex polygon with N vertices: $P_1, P_2, \ldots, P_N$ (the vertices are enumerated around the polygon in counter-clockwise order.) In order to achieve absolute safety providing electricity to the camps, besides an electric supplying system, the host organization set up a path from a reserved electricity generator (which is placed in one of the camps) to every camp once, and *the path's total length is minimum*.

Problem

Given the coordinates of the polygons' vertices (the camps' positions), determine the length of the electric path corresponding to the host organization's arrangement.

Input

The first line contains the integer N ($1 \le N \le 200$). The i'th line of the next N lines contains two real numbers x_i, y_i, separated by a space, with no more than 3 digits after the decimal points, are vertex P_i's coordinates on the plane (with $i = 1, 2, \ldots, N$). The length of the path connecting two vertex (x_i, y_i) and (x_j, y_j) is computed with the formula: sqrt($(x_i - x_j)^2 + (y_i - y_j)^2$).

Output

The only line should contain real number L (written in real number format, with 3 digits after the decimal point), which is the total length of the electric path.

Sample

input	output
4 50.0 1.0 5.0 1.0 0.0 0.0 45.0 0.0	50.211

Problem Source: The competition for selecting the Vietnam IOI team

1144. The Emperor's Riddle

Time limit: 1.0 second
Memory limit: 4 MB

Background

In the olden times there was a young emperor who was the bravest, wisest, richest, most handsome in the whole world. He had proven himself in endless of battles, quests, and victories but his court was not happy because he had not appointed a queen yet. However, choosing a queen was not easy because of his high status and standard, the emperor wanted a girl not only beautiful but smart and kind as well. Lightning Knight - that was the young Emperor's name - sent his most trusted knights out to seek for a girl like that; and after a long time searching, the men brought back two of the most beautiful and intelligent girls in all the lands. They were two princess sisters from a faraway land. The older sister - Van Trinh - was mysterious and beautiful like the moon, while Thuy Linh - the younger one - was bright and lovely as the sun. They were both famous for being kind, gentle, and intelligent to their people, and as many girls before them, they both fell truly, madly, deeply in love with the handsome emperor at first sight.

Now, the Emperor had to face the hardest test of all: to pick just one in these two sisters to become his rightful and beloved queen and lay the world under her feet. After countless sleepless nights, the Emperor sought out a just solution. He thought of a riddle and announced to the two princesses and the court that he would marry the first one who bring the right answer to his desk.

Problem

At the same time with the above event, the Emperor had just won the most important battle to unite all the lands in the world. That was two good news in such a short time. Being the rich and generous emperor he was, the Emperor wanted to reward to all the brave and loyal generals with boxes of gold. The distribution was not easy and that's why he chose it as the riddle for Van Trinh and Thuy Linh. Centuries has passed since then, the Emperor and queen might have died and their romance might have been forgotten from our world, but the riddle still remains as one of the hardest tasks in the ancient books.

The Emperor wants to reward N boxes of gold to M generals. The i-th box has the value of A_i. Now the Emperor wants to give N boxes to M generals so that the difference of gold between the general who receives the most gold and the general who receives the least gold is as small as possible. Note: a general can receive more than one box, and he must receive the whole box (i.e.: not half or 1/3 of box).

Input

The 1st line contains three positive integers N, M and K ($N \leq 10000$, $M \leq 1000$ and $N \geq M$). K is the maximum result that the emperor accepts. The 2nd line contains N positive integers $0 < A_1, A_2, \ldots, A_N \leq 1000$.

Output

The 1st line contains one integer which is the minimum difference your program can find. In the next M lines, the i-th line contains the index of boxes rewarded to the i-th general.

Sample

input	output
10 3 4 12 95 16 37 59 50 47 3 41 95	4 6 7 9 1 8 10 4 3 5 2

Problem Author: HNT

1145. Rope in the Labyrinth

Time limit: 0.5 second
Memory limit: 64 MB

A labyrinth with rectangular form and size $m \times n$ is divided into square cells with sides' length 1 by lines that are parallel with the labyrinth's sides. Each cell of the grid is either occupied or free. It is possible to move from one free cell to another free cells that share a common side with the cell. One cannot move beyond the labyrinth's borders. The labyrinth is designed pretty specially: for any two cells there is only one way to move from one cell to the other. There is a hook at each cell's center. In the labyrinth there are two special free cells, such that if you can connect the hooks of those two cells with a rope, the labyrinth's secret door will be automatically opened. The problem is to prepare a shortest rope that can guarantee, you always can connect the hooks of those two cells with the prepared rope regardless their position in the labyrinth.

Input

The first line contains integers n and m ($3 \le n, m \le 820$). The next lines describe the labyrinth. Each of the next m lines contains n characters. Each character is either "#" or ".", with "#" indicating an occupied cell, and "." indicating a free cell.

Output

Print out in the single line the length (measured in the number of cells) of the required rope.

Sample

input	output
7 6 ####### #.#.### #.#.### #.#.#.# #.....# #######	8

1146. Maximum Sum

Time limit: 0.5 second
Memory limit: 64 MB

Given a 2-dimensional array of positive and negative integers, find the sub-rectangle with the largest sum. The sum of a rectangle is the sum of all the elements in that rectangle. In this problem the sub-rectangle with the largest sum is referred to as the *maximal sub-rectangle*. A sub-rectangle is any contiguous sub-array of size 1 × 1 or greater located within the whole array.

As an example, the maximal sub-rectangle of the array:

$$\begin{array}{rrrr} 0 & -2 & -7 & 0 \\ 9 & 2 & -6 & 2 \\ -4 & 1 & -4 & 1 \\ -1 & 8 & 0 & -2 \end{array}$$

is in the lower-left-hand corner and has the sum of 15.

Input

The input consists of an $N \times N$ array of integers. The input begins with a single positive integer N on a line by itself indicating the size of the square two dimensional array. This is followed by N^2 integers separated by white-space (newlines and spaces). These N^2 integers make up the array in row-major order (i.e., all numbers on the first row, left-to-right, then all numbers on the second row, left-to-right, etc.). N may be as large as 100. The numbers in the array will be in the range [−127, 127].

Output

The output is the sum of the maximal sub-rectangle.

Sample

input	output
4 0 -2 -7 0 9 2 -6 2 -4 1 -4 1 -1 8 0 -2	15

1147. Shaping Regions

Time limit: 0.5 second
Memory limit: 64 MB

N opaque rectangles ($1 \le N \le 1000$) of various colors are placed on a white sheet of paper whose size is A wide by B long. The rectangles are put with their sides parallel to the sheet's borders. All rectangles fall within the borders of the sheet so that different figures of different colors will be seen.

The coordinate system has its origin (0, 0) at the sheet's lower left corner with axes parallel to the sheet's borders.

Input

The order of the input lines dictates the order of laying down the rectangles. The first input line is a rectangle “on the bottom”. First line contains A, B and N, space separated ($1 \le A, B \le 10000$). Lines 2, …, $N + 1$ contain five integers each: *llx*, *lly*, *urx*, *ury*, color: the lower left coordinates and upper right coordinates of the rectangle whose color is *color* ($1 \le color \le 2500$) to be placed on the white sheet. The color 1 is the same color of white as the sheet upon which the rectangles are placed.

Output

The output should contain a list of all the colors that can be seen along with the total area of each color that can be seen (even if the regions of color are disjoint), ordered by increasing color. Do not display colors with no area.

Sample

input	output
20 20 3 2 2 18 18 2 0 8 19 19 3 8 0 10 19 4	1 91 2 84 3 187 4 38

1148. Building Towers

Time limit: 1.0 second
Memory limit: 4 MB

There are N bricks in a toy box which have 1-unit height and 2-unit width. The teacher organizes a tower-building game. The tower is built of the bricks. The tower consists of H levels. The bottom level contains M bricks, every next level must contain exactly one brick less or greater than the level just below it.

Here is an example of a tower with H=6, M=2, N=13.

The tower with H levels can be represented by the array of H integers, which are the numbers of bricks in each level from the bottom to the top. Consider all different towers with exactly H levels and exactly M bricks in the bottom level that can be built using not more than N bricks. We can number these towers in such way that corresponding arrays will be ordered lexicographically.

Your task is to find a tower with specific number in the sense of described order.

Input

The first line of the input contains positive numbers N, H and M ($N \le 32767$, $H \le 60$, $M \le 10$). Each of the following lines contains integer K, which is the interested tower number. The last line contains number -1. The towers are numbered starting from 1.

Output

The first line of the output should contain the total number of different towers that can be built. Each following line should contain an array describing the tower with number K given in respective line of input. The numbers in the arrays should be separated by at least one space.

Sample

input	output
22 5 4 1 10 -1	10 4 3 2 1 2 4 5 4 5 4

1149. Sinus Dances

Time limit: 1.0 second
Memory limit: 64 MB

Let A_n = sin(1–sin(2+sin(3–sin(4+…sin(n))…)
Let S_n = (…(A_1+n)A_2+n–1)A_3+…+2)A_n+1
For given N print S_N

Input

One integer N. $1 \le N \le 200$

Output

Line containing S_N

Sample

input	output
3	((sin(1)+3)sin(1-sin(2))+2)sin(1-sin(2+sin(3)))+1

Problem Author: Vladimir Gladkov
Problem Source: Ural Collegiate Programming Contest, April 2001, Perm, Test Round

1150. Page Numbers

Time limit: 1.0 second
Memory limit: 64 MB

John Smith has decided to number the pages in his notebook from 1 to *N*. Please, figure out the number of zeros, ones, twos, …, nines he might need.

Input

One number N ($1 \le N < 10^9$).

Output

Output 10 lines. The first line should contain the number of zeros needed, the second line should contain the number of ones needed, …, the tenth line should contain the number of nines needed.

Sample

input	output
12	1 5 2 1 1 1 1 1 1 1

Problem Author: Eugene Bryzgalov
Problem Source: Ural Collegiate Programming Contest, April 2001, Perm, English Round

1151. Radiobeacons

Time limit: 0.5 second
Memory limit: 64 MB

N radiobeacons are located on the plane. Their exact positions are unknown but we know that $N \le 10$ and that their coordinates are integers from 1 to 200. Each beacon produces unique signal, that distinguishes it from the other beacons.

In some different places, we should call them *checkpoints*, coordinates of which are well known, there were conducted measurements. As a result of these measurements distances from checkpoints to some of beacons became known. Here we should note, that the distance between points A and B equals $\max(|A_x - B_x|, |A_y - B_y|)$.

You need to get positions of all beacons basing on coordinates of the checkpoints and results of measurements, if that is possible.

Input

First line contains an integer M, the number of checkpoints. $1 \le M \le 20$. Then M lines follow, each of them contains an information received from one of the checkpoints, formatted as follows:

<X_i>,<Y_i>:<ID_1>-<R_1>[,<ID_2>-<R_2>][,…]

where X_i, Y_i are coordinates of checkpoints, ID_k is ID of beacon k, R_k is a distance from checkpoint i to beacon k. Coordinates of checkpoints are integers from 1 to 200. Each checkpoint measures at least one signal. IDs of beacons are integers from 1 to 30000.

Output

Output should consist of N lines describing positions of the beacons. If there is exactly one position (x_k, y_k) of the k'th beacon satisfying the results of measurements and such that $1 \le x_k, y_k \le 200$, you should output <ID_k>:<x_k>,<y_k> in the k'th line (ID_k is ID of the k'th beacon). In the other case you should output <ID_k>:UNKNOWN in the k'th line. All lines should be ordered by ID_k in an ascending order.

Sample

input	output
2 15,15:16-7,5-3 10,10:5-2,16-2	5:12,12 16:UNKNOWN

Notes

It is guaranteed that measurements are consistent, i.e. the solution always exists, but may be ambiguous.

Problem Source: Ural Collegiate Programming Contest, April 2001, Perm, English Round

1152. False Mirrors

Time limit: 2.0 second
Memory limit: 64 MB

Background

We wandered in the labyrinth for twenty minutes before finally entering the large hall. The walls were covered by mirrors here as well. Under the ceiling hung small balconies where monsters stood. I had never seen this kind before. They had big bulging eyes, long hands firmly holding riffles and scaly, human-like bodies. The guards fired at me from the balconies, I shot back using my BFG-9000. The shot shattered three mirrors filling the room with silvery smoke. Bullets drummed against my body-armor knocking me down to the floor. Falling down I let go a shot, and got up as fast as I fell down by rotating on my back, like I did in my youth while break dancing, all this while shooting three more times. Three mirrors, three mirrors, three mirrors…

Sergey Lukjanenko, "The Labyrinth of Reflections"

Problem

BFG-9000 destroys three adjacent balconies per one shoot. (*N*-th balcony is adjacent to the first one). After the shoot the survival monsters inflict damage to Leonid (main hero of the novel) — one unit per monster. Further follows new shoot and so on until all monsters will perish. It is required to define the minimum amount of damage, which can take Leonid.

Input

The first line contains integer *N*, amount of balconies, on which monsters have taken a circular defense. $3 \le N \le 20$. The second line contains *N* integers, amount of monsters on each balcony (not less than 1 and no more than 100 on each).

Output

Output minimum amount of damage.

Sample

input	output
7 3 4 2 2 1 4 1	9

Problem Author: Eugene Bryzgalov
Problem Source: Ural Collegiate Programming Contest, April 2001, Perm, English Round

1153. Supercomputer

Time limit: 2.0 second
Memory limit: 64 MB

To check the speed of JCN Corporation new supercomputer it was decided to figure out the sum of first N ($N < 10^{600}$) positive integers. Unfortunately, by the time the calculation was finished the Chief Programmer forgot the value of N he entered. Your task is to write the program (for personal computer), which would determine the value of N by the result calculated on supercomputer.

Note: JCN Corporation manufactures only reliable computers, and its programmers write only correctly working programs.

Input

One line containing the result of calculations on the supercomputer.

Output

Выведите N, the number entered by Chief Programmer.

Sample

input	output
28	7

Problem Author: Eugene Bryzgalov
Problem Source: Ural Collegiate Programming Contest, April 2001, Perm, English Round

1154. Mages Contest

Time limit: 1.0 second
Memory limit: 64 MB

The Powers of Light and the Powers of Darkness had gathered the best elemental mages of the Middle-earth: the Lords of Fire, Earth, Air and Water.

Sometime during 24 hours there exists the Moment of Power for each one of the elements, when the mastery of corresponding mage is in its maximum point. In contrary there exists the Moment of Weakness, when the mastery of the mage is minimum. In between these moments the mage's mastery changes linearly.

There can be several mages fighting on each side. The mages cumulative mastery is defined as sum of their individual masteries. The win is given to the side, which mages' cumulative mastery is the largest. The larger the advantage of one side over another, the easier its win is, and the smaller casualties are.

The Supreme Master, who is to declare the time of the Contest secretly wishes the Powers of Light to win, and tries to make this win as easy as possible. Assume that the Contest is held momentary, and the mastery of the mages doesn't change during it. You are to help the Supreme Master in selecting the time of the Contest.

Input

The first four lines contain information about the Moments of Weakness and the Moments of Power for mages of each of the elements. Each line contains five parameters separated by spaces: element code, the time of the Moment of Power, the mastery in the Moment of Power, the time of the Moment of Weakness, the mastery in the Moment of Weakness.

Element code is one of four capital letters: “A” for Air, “E” for Earth, “F” for Fire and “W” for Water. The time is HH:MM:SS formatted and lies between 00:00:00 and 23:59:59. The Moment of Power is not equal to the Moment of Weakness. The mastery in these moments is a positive integer less than or equal to 10000.

Then two more lines follow, which determine respectively the cast of the Powers of Light and the cast of the Powers of Darkness. Each line consists of symbols “A”, “E”, “F”, “W” representing one mage of the corresponding element. The number of the mages from each side is not less than 1 and is not greater than 1000.

Output

The first line should contain HH:MM:SS formatted time of the Contest. In the second line there should be one number with two decimal points, which represents the advantage of the Powers of Light over the Powers of Darkness. The time of the Contest should lie between 00:00:00 and 23:59:59. If there exist several moments can be chosen as the time of the Contest, then choose the earliest one.

If there is no way the Powers of Light can win, then write “We can't win!”

Samples

input	output
A 10:00:00 130 18:00:00 40 E 14:00:00 150 21:30:00 25 F 06:00:00 105 18:00:00 70 W 23:00:00 140 02:00:00 20 A WWW	02:00:00 25.00
A 10:00:00 130 18:00:00 40 E 14:00:00 150 21:30:00 25 F 06:00:00 105 18:00:00 70 W 23:00:00 140 02:00:00 20 A WWWF	We can't win!

Problem Author: Eugene Bryzgalov
Problem Source: Ural Collegiate Programming Contest, April 2001, Perm, English Round

1155. Troubleduons

Time limit: 0.5 second
Memory limit: 64 MB

Archangel of the Science is reporting:
"O, Lord! Those physicists on the Earth have discovered a new elementary particle!"
"No problem, we'll add another parameter to the General Equation of the Universe."

As physics develops and moves on, scientists find more and more strange elementary particles, whose properties are more than unknown. You may have heard about muons, gluons and other strange particles. Recently scientists have found new elementary particles called troubleduons. These particles are called this way because scientists can create or annihilate them only in couples. Besides, troubleduons cause trouble to scientists, and that's why the latter want to get rid of them. You should help scientists get rid of troubleduons.

Experimental set consists of eight cameras, situated in the vertices of a cube. Cameras are named as A, B, C, …, H. It is possible to generate or annihilate two troubleduons in neighbouring cameras. You should automate the process of removing troubleduons.

Input

The only line contain eight integers ranging from 0 to 100, representing number of troubleduons in each camera of experimental set.

Output

Output sequence of actions leading to annihilating all troubleduons or "IMPOSSIBLE", if you cannot do it. Actions should be described one after another, each in a separate line, in the following way: name of the first camera, name of the second camera (it should be a neighborough to the first one), "+" if you create troubleduons, "-" if you destroy them. Number of actions in the sequence should not exceed 1000.

Samples

input	output
1 0 1 0 3 1 0 0	EF- EA- AD+ AE- DC-
0 1 0 1 2 3 2 2	IMPOSSIBLE

Problem Source: Ural Collegiate Programming Contest, April 2001, Perm, English Round

1156. Two Rounds

Time limit: 2.0 second
Memory limit: 64 MB

There are two rounds in the Urals Championship. The competitors have to solve N problems on each round. The jury had been working hard and finally managed to prepare $2N$ problems for the championship. But it appeared that among those problems there were some, which have the analogous solutions. One shouldn't assign such a problems for the same round. Please, help the jury form sets of tasks for each of the rounds.

Input

First line contains two numbers: N, the number of tasks for a round, and M, the number of pairs of tasks which should not be assigned for one round ($1 \le N \le 50$; $0 \le M \le 100$). Then M lines follow, each of them contains two numbers of analogous tasks.

Output

Output two lines, containing numbers of tasks assigned for each round. If there is no solution, output the only word "IMPOSSIBLE". If there are more than one solution you may assume anyone of them.

Sample

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

Problem Author: Eugene Bryzgalov
Problem Source: Ural Collegiate Programming Contest, April 2001, Perm, English Round

1157. Young Tiler

Time limit: 1.0 second
Memory limit: 64 MB

One young boy had many-many identical square tiles. He loved putting all his tiles to form a rectangle more, than anything in the world — he has learned the number of all rectangles he can form using his tiles. On his birthday he was presented a number of new tiles. Naturally, he started forming rectangles from these tiles — the thing he loved most of all! Soon he has learned all rectangles he could form with a new number of tiles.

Here we should notice that boy can easily count the number of rectangles, but he has difficulty counting the number of tiles — there are too much of them for such a young boy. But it will not be difficult for you to determine how many tiles he has now, knowing how many rectangles he could form before, how many rectangles he can form now, and how many tiles he got as a birthday present.

You are given numbers M, N and K. You should find the smallest number L, such as you can form exactly N different rectangles using all L tiles, and form exactly M different rectangles using $L - K$ tiles.

Input

One line containing three integers: M, N, K ($1 \le M, N \le 50$; $1 \le K \le 9999$).

Output

If L is less than or equal to 10000, then print that number (if there is a number of such L, you should print the smallest one). If there is no solution or smallest L is greater than 10000, print 0.

Sample

input	output
2 3 1	16

Problem Source: Ural Collegiate Programming Contest, April 2001, Perm, English Round

1158. Censored!

Time limit: 2.0 second
Memory limit: 64 MB

The alphabet of Freeland consists of exactly N letters. Each sentence of Freeland language (also known as Freish) consists of exactly M letters without word breaks. So, there exist exactly N^M different Freish sentences.

But after recent election of Mr. Grass Jr. as Freeland president some words offending him were declared unprintable and all sentences containing at least one of them were forbidden. The sentence S contains a word W if W is a substring of S i.e. exists such k >= 1 that S[k] = W[1], S[k+1] = W[2], ... , S[k+len(W)-1] = W[len(W)], where k+len(W)-1 <= M and len(W) denotes length of W. Everyone who uses a forbidden sentence is to be put to jail for 10 years.

Find out how many different sentences can be used now by freelanders without risk to be put to jail for using it.

Input

The first line contains three integer numbers: N - the number of letters in Freish alphabet, M - the length of all Freish sentences and P - the number of forbidden words ($1 \le N \le 50$, $1 \le M \le 50$, $0 \le P \le 10$).

The second line contains exactly N different characters - the letters of the Freish alphabet (all with ASCII code greater than 32).

The following P lines contain forbidden words, each not longer than min(M, 10) characters, all containing only letters of Freish alphabet.

Output

Output the only integer number - the number of different sentences freelanders can safely use.

Sample

input	output
3 3 3 QWE QQ WEE Q	7

Problem Author: Nick Durov
Problem Source: ACM ICPC 2001. Northeastern European Region, Northern Subregion

1159. Fence

Time limit: 1.0 second
Memory limit: 64 MB

Workers are going to enclose a new working region with a fence. For their convenience the enclosed area has to be as large as possible. They have N rectangular blocks to build the fence. The length of the i-th block is L_i meters. All blocks have the same height of 1 meter. The workers are not allowed to break blocks into parts. All blocks must be used to build the fence.

Input

The first line contains one integer N ($3 \le N \le 100$). The following N lines describe fence blocks. Each block is represented by its length in meters (integer number, $1 \le L_i \le 100$).

Output

Write one non-negative number S - maximal possible area of the working region (in square meters). S must be written with two digits after the decimal point. If it is not possible to construct the fence from the specified blocks, write 0.00.

Sample

input	output
4 10 5 5 4	28.00

Problem Author: Nick Durov
Problem Source: ACM ICPC 2001. Northeastern European Region, Northern Subregion

1160. Network

Time limit: 1.0 second
Memory limit: 64 MB

Andrew is working as system administrator and is planning to establish a new network in his company. There will be N hubs in the company, they can be connected to each other using cables. Since each worker of the company must have access to the whole network, each hub must be accessible by cables from any other hub (with possibly some intermediate hubs).

Since cables of different types are available and shorter ones are cheaper, it is necessary to make such a plan of hub connection, that the maximum length of a single cable is minimal. There is another problem - not each hub can be connected to any other one because of compatibility problems and building geometry limitations. Of course, Andrew will provide you all necessary information about possible hub connections.

You are to help Andrew to find the way to connect hubs so that all above conditions are satisfied.

Input

The first line contains two integer: N - the number of hubs in the network ($2 \le N \le 1000$) and M — the number of possible hub connections ($1 \le M \le 15000$). All hubs are numbered from 1 to N. The following M lines contain information about possible connections - the numbers of two hubs, which can be connected and the cable length required to connect them. Length is a positive integer number that does not exceed 10^6. There will be no more than one way to connect two hubs. A hub cannot be connected to itself. There will always be at least one way to connect all hubs.

Output

Output first the maximum length of a single cable in your hub connection plan (the value you should minimize). Then output your plan: first output P - the number of cables used, then output P pairs of integer numbers - numbers of hubs connected by the corresponding cable. Separate numbers by spaces and/or line breaks.

Sample

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

Problem Author: Andrew Stankevich
Problem Source: ACM ICPC 2001. Northeastern European Region, Northern Subregion

1161. Stripies

Time limit: 1.0 second
Memory limit: 64 MB

Our chemical biologists have invented a new very useful form of life called *stripies* (in fact, they were first called in Russian - *polosatiki*, but the scientists had to invent an English name to apply for an international patent). The stripies are transparent amorphous amebiform creatures that live in flat colonies in a jelly-like nutrient medium. Most of the time the stripies are moving. When two of them collide a new stripie appears instead of them. Long observations made by our scientists enabled them to establish that the weight of the new stripie isn't equal to the sum of weights of two disappeared stripies that collided; nevertheless, they soon learned that when two stripies of weights m_1 and m_2 collide the weight of resulting stripie equals to $2 \cdot \mathrm{sqrt}(m_1 m_2)$. Our chemical biologists are very anxious to know to what limits can decrease the total weight of a given colony of stripies.

You are to write a program that will help them to answer this question. You may assume that 3 or more stipies never collide together.

Input

The first line contains one integer N ($1 \le N \le 100$) - the number of stripies in a colony. Each of next N lines contains one integer ranging from 1 to 10000 - the weight of the corresponding stripie.

Output

The output must contain one line with the minimal possible total weight of colony with the accuracy of two decimal digits after the point.

Samples

input	output
2 72 50	120.00
3 72 30 50	120.00

Problem Author: Nick Durov
Problem Source: ACM ICPC 2001. Northeastern European Region, Northern Subregion

1162. Currency Exchange

Time limit: 1.0 second
Memory limit: 64 MB

Several currency exchange points are working in our city. Let us suppose that each point specializes in two particular currencies and performs exchange operations only with these currencies. There can be several points specializing in the same pair of currencies. Each point has its own exchange rates, exchange rate of A to B is the quantity of B you get for 1A. Also each exchange point has some commission, the sum you have to pay for your exchange operation. Commission is always collected in source currency.

For example, if you want to exchange 100 US Dollars into Russian Rubles at the exchange point, where the exchange rate is 29.75, and the commission is 0.39 you will get (100 - 0.39) * 29.75 = 2963.3975RUR.

You surely know that there are N different currencies you can deal with in our city. Let us assign unique integer number from 1 to N to each currency. Then each exchange point can be described with 6 numbers: integer A and B - numbers of currencies it exchanges, and real RAB, CAB, RBA and CBA - exchange rates and commissions when exchanging A to B and B to A respectively.

Nick has some money in currency S and wonders if he can somehow, after some exchange operations, increase his capital. Of course, he wants to have his money in currency S in the end. Help him to answer this difficult question. Nick must always have non-negative sum of money while making his operations.

Input

The first line contains four numbers: N - the number of currencies, M - the number of exchange points, S - the number of currency Nick has and V - the quantity of currency units he has. The following M lines contain 6 numbers each - the description of the corresponding exchange point - in specified above order. Numbers are separated by one or more spaces. $1 \le S \le N \le 100$, $1 \le M \le 100$, V is real number, $0 \le V \le 10^3$.

For each point exchange rates and commissions are real, given with at most two digits after the decimal point, $10^{-2} \le$ rate $\le 10^2$, $0 \le$ commission $\le 10^2$.

Let us call some sequence of the exchange operations simple if no exchange point is used more than once in this sequence. You may assume that ratio of the numeric values of the sums at the end and at the beginning of any simple sequence of the exchange operations will be less than 10^4.

Output

If Nick can increase his wealth, output YES, in other case output NO.

Samples

input	output
3 2 1 10.0 1 2 1.0 1.0 1.0 1.0 2 3 1.1 1.0 1.1 1.0	NO
3 2 1 20.0 1 2 1.0 1.0 1.0 1.0 2 3 1.1 1.0 1.1 1.0	YES

Problem Author: Nick Durov
Problem Source: ACM ICPC 2001. Northeastern European Region, Northern Subregion

1163. Chapaev

Time limit: 2.0 second
Memory limit: 64 MB

Legendary divisional commander Vasiliy I. Chapaev was fond of playing this beautiful game with his aide-de-camp Petka during their (scanty) spare time. The game is played as follows. There are eight white and eight red draughts on the board at the beginning of the game. The red side starts the game by making the first turn. The turn consists of selecting an arbitrary draught of player's colour and pushing it with a flick into a certain direction. Then this draught begins to move in this direction until it completely falls off the board. If it hits or even just touches another draught of any colour during the movement, the latter is immediately removed from the board being considered killed. In any case the moving draught continues its movement without changing its speed or direction. After the moving draught has completed its movement across the board (has fallen off) the other player has to make his turn. If there are no draughts of player's colour left he is considered to lose the game.

The historians have a record of an initial position in one of such games. Unfortunately, the result of this game is unknown. Your task is to establish the truth taking for granted that both Chapaev and Petka always used the optimal strategy.

Input

Each of two lines contains eight pairs of numbers - the coordinates of centres of red and white draughts respectively. The draughts are considered to be cylinders of radius 0.4 and height 0.15. The coordinates are calculated so that the board is a square 8x8 with vertices (0, 0), (0, 8), (8, 0) and (8, 8). There will be no draught that overlaps or touches another one. Also each piece in the initial position is completely contained within the limits of the board.

Output

Output RED or WHITE corresponding to the winning side.

Sample

input	output
0.5 7.5 1.5 7.5 2.5 7.5 3.5 7.5 4.5 7.5 5.5 7.5 6.5 7.5 7.5 7.5 0.5 0.5 1.5 0.5 2.5 0.5 3.5 0.5 4.5 0.5 5.5 0.5 6.5 0.5 7.5 0.5	RED

Problem Author: Nick Durov
Problem Source: ACM ICPC 2001. Northeastern European Region, Northern Subregion

1164. Fillword

Time limit: 1.0 second
Memory limit: 64 MB

Alex likes solving fillwords. Fillword is a word game with very simple rules. The author of the fillword takes rectangular grid (M cells width, N cells height) and P words. Then he writes letters in the cells of the grid (one letter in one cell) so that each word can be found on the grid and the following conditions are met:

- no cell belongs to more than one word
- no cell belongs to any word more than once

Some word W (let us consider its length being k) is found on the grid if you can find such sequence of cells (x_1, y_1), (x_2, y_2), …, (x_k, y_k) that:

- (x_i, y_i) and (x_{i+1}, y_{i+1}) are neighbors ($|x_i-x_{i+1}| + |y_i-y_{i+1}| = 1$) for each i = 1, 2, …, k-1
- W[i] is written in the cell with coordinates (x_i, y_i).

The task is to find all the words on the grid. After they are found, you see that the letters in some cells are not used (they do not belong to any found word). You make up a secret word using these letters and win a big prize.

Your task is to help Alex to solve fillwords. You should find out which letters will be left after he finds all the words on the grid. The most difficult task - to make up a secret word out of them - we still reserve to Alex.

Input

The first line contains three integers - N, M ($2 \le M, N \le 10$) and P ($P \le 100$). Next N lines contain M characters each, and represent the grid. The following P lines contain words that are to be found on the fillword grid.

Fillword will always have at least one solution. All characters occurring in fillword will be capital English letters.

Output

Output letters from, which a secret word should be made up. Letters should be output in lexicographical order.

Sample

input	output
3 3 2 EBG GEE EGE BEG GEE	EEG

Problem Author: Alex Selivanov
Problem Source: ACM ICPC 2001. Northeastern European Region, Northern Subregion

1165. Subnumber

Time limit: 1.0 second
Memory limit: 64 MB

George likes arithmetics very much. Especially he likes the integers series. His most favourite thing is the infinite sequence of digits, which results as the concatenation of all positive integers in ascending order. The beginning of this sequence is 1234567891011121314… Let us call this sequence S. Then S[1] = 1, S[2] = 2, …, S[10] = 1, S[11] = 0, …, and so on.

George takes a sequence of digits A and wants to know when it first appears in S. Help him to solve this difficult problem.

Input

The first line contains A - the given sequence of digits. The number of digits in A does not exceed 200.

Output

Output the only number - the least k such that A[1] = S[k], A[2] = S[k+1], ... A[len(A)] = S[k + len(A) – 1], where len(A) denotes the length of A (i.e. the number of digits in it).

Sample

input	output
101	10

Problem Author: Nikita Shamgunov
Problem Source: ACM ICPC 2001. Northeastern European Region, Northern Subregion

1166. Funny Card Game

Time limit: 1.0 second
Memory limit: 64 MB

Of course all of you want to know how to become ACM world champions. There is no exact answer to this question but it is well known that the champions of the last two ACM World Finals enjoyed playing the following funny card game. Two or more players can play this game simultaneously. It is played using a standard 54-card deck. At the beginning the players sit forming a circle. One of the players shuffles the deck and then he deals the cards in clockwise order starting from the neighbour on his left. He deals the top card of the deck to the current player each time. He does it until each player gets five cards. Then he takes the top card of the deck and lays it onto the table face up and he also lays the remainder of the deck nearby (these cards are laid face down preserving their original order). The card laid by the shuffler is considered as the first turn of the game (as if it was made by the shuffler to the player on his left).

The normal game flow as following: the player should cover the last laid card with the card of the same suit or value. If he has none, he takes one card from the top of the deck and again checks this condition. If still there are no matching cards, the move will go to the next player (his left neighbour). But for some cards special rules are applied:

1. If the laid card is 6, the player takes one card from the top of the deck and skips his turn
2. If the laid card is 7, the player takes two cards from the top of the deck (if there is only one card in the deck, he takes just it) and skips his turn
3. If the laid card is Ace the player skips his turn without taking any cards
4. If the player lays Queen, he himself announces the suit of the card it should be covered with
5. Eight is one of the most powerful weapons in this game. When it is laid, the next player has to cover it in any case. If he cannot cover it with his cards he has to take cards from the deck, until he is able to cover it.
6. And the most important card in the game is the King of Spades. If it's laid, the next player takes 4 cards from the top of the deck (if there is not enough cards in the deck, he takes all of them) and skips his turn.

You may assume that the deck is looped and the laid cards are immediately moving to the bottom of the deck. So it can happen that the player has to cover the card by itself.

We should say some words about Jokers. Jokers can have any card value by the wish of the player who has it. If the player lays the joker, he assigns a definite card value and suit for it, so this Joker has this assigned value until another player takes it from the deck (if it ever happens). The player is free to use or not to use the Joker whenever he wants (if it is his turn to play, of course).

If the player is left without any cards in his hand, he is considered a winner and the game continues without him (his left neighbour becomes the left neighbour of his right neighbour).

If there is only one player left, he is the looser, so he is called a Japanese Fool (it is a Russian name of this game).

We are interested in the following situation. Consider the moment when only two players are left in the game. If one of them has a special combination of cards, it may happen that he can lay out all his cards in a some sequence so that the other player won't get a move (he'll just have to take cards from the deck and skip turns) and will be the loser — provided the first one finds the winning sequence.

You will be given the position of the game in which only two players are left. Your task will be to determine whether such a winning sequence for the first player exists or not.

We will consider that the first player have already taken all cards from the deck that he had to (if any), so he cannot take any cards from the deck. We will also consider that if the last laid card is a skip-turn card, it was the second player who skipped the turn.

Input

The first line contains cards of the first player separated by spaces. The second line contains the last laid face up card.

The card description consists of two characters. The first of them corresponds to the card value (2-9 for digits, T for 10, J for Jack, Q for Queen, K for King and A for Ace). The next describes the suit of the card and may be one of the following: S for Spades, C for Clubs, D for Diamonds or H for Hearts. Joker is represented by a character '*'. If the last laid card is Queen, it is followed by a suit letter. If the last laid card is a joker, then the '*' is followed by an actual card description (the card specified by the player who laid the Joker).

Output

The first line should contain a single word YES or NO signalling whether the winning sequence exists. If the answer is positive the second line must contain the winning sequence of cards separated by spaces. As in the input, the Joker is to be followed by a card specification and the Queen should follow a suit letter. If there is more than one solution you may output an arbitrary one.

Sample

input	output
6C QD 6S KS 7S * *QHS	YES 7S KS 6S 6C *6D QDS

Notes

Note that the Queen could be covered with other Queen only if the latter has the announced suit.

Problem Author: Andrew Lopatine
Problem Source: ACM ICPC 2001. Northeastern European Region, Northern Subregion

1167. Bicolored Horses

Time limit: 1.0 second
Memory limit: 64 MB

Every day, farmer Ion (this is a Romanian name) takes out all his horses, so they may run and play. When they are done, farmer Ion has to take all the horses back to the stables. In order to do this, he places them in a straight line and they follow him to the stables. Because they are very tired, farmer Ion decides that he doesn't want to make the horses move more than they should. So he develops this algorithm: he places the 1st P_1 horses in the first stable, the next P_2 in the 2nd stable and so on. Moreover, he doesn't want any of the **K** stables he owns to be empty, and no horse must be left outside. Now you should know that farmer Ion only has black or white horses, which don't really get along too well. If there are **i** black horses and **j** white horses in one stable, then the coefficient of unhappiness of that stable is **i*j**. The total coefficient of unhappiness is the sum of the coefficients of unhappiness of every of the K stables.

Determine a way to place the **N** horses into the **K** stables, so that the total coefficient of unhappiness is minimized.

Input

On the 1st line there are 2 numbers: **N** ($1 \le N \le 500$) and **K** ($1 \le K \le N$). On the next N lines there are N numbers. The i-th of these lines contains the color of the i-th horse in the sequence: 1 means that the horse is black, 0 means that the horse is white.

Output

You should only output a single number, which is the minimum possible value for the total coefficient of unhappiness.

Sample

input	output
6 3 1 1 0 1 0 1	2

Notes

Place the first 2 horses in the first stable, the next 3 horses in the 2nd stable and the last horse in the 3rd stable.

Problem Author: Mugurel Ionut Andreica
Problem Source: Romanian Open Contest, December 2001

1168. Radio Stations

Time limit: 1.0 second
Memory limit: 64 MB

Along the surface of Romania, there are **K** radio stations positioned at different points and altitudes. Each of these radio stations has a given broadcast radius, that is, the maximum distance it can send its signal to. The government wants to place a radio receiver somewhere on the map, so that it will receive the signals from all the K radio stations: this means that the distance to every radio station should be less or equal to the broadcast radius of that radio station.

The map of Romania is given as a **M*N** matrix, where the value in row i and column j represents the altitude of the corresponding zone. The side of a square in this matrix is 1. All the K radio stations are placed at distinct coordinates on the map and at the same height as the corresponding zone (plus, they are placed exactly in the center of their square). The radio receiver can be placed in the center of any square not occupied by a radio station, at the same altitude of the square or it can be placed higher with an integer number of meters. The radio receiver cannot be placed in a square occupied by a radio station.

Your task is to decide how many possibilities to place the radio receiver the government has. Note that if the radio receiver may be placed in row **i** and column **j** at altitudes **h1** and **h2** ($h1 \neq h2$), this counts as 2 different possibilities.

Input

The first line of input contains 3 integers: **M**, **N** ($1 \leq M, N \leq 50$) and **K** ($1 \leq K \leq \min(M*N-1, 1000)$), representing the dimensions of the map and the number of radio stations. Next there are M lines each containing N integers, which are the altitudes of the zones on the map (no altitude will be higher than 32000 or lower than 0). After that, there will be K lines, each containing 3 numbers: **i, j** and **R**. i and j will be the location of the radio station on the map and R will be its broadcast radius (R is a real number, not larger than 100000 and not less than 0).

No two radio stations will be placed on the same square.

Output

You should output one integer number, which is the total number of valid possibilities to place the radio receiver on the map.

Sample

input	output
5 5 3 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10 5 4 3 2 1 1 1 4.3 5 5 4.3 5 1 4.3	4

Notes

The radio receiver can be placed at position (3, 2), with extra height 0 and extra height 1, and at position (3, 3), with extra height 0 and extra height 1. So, there are 4 possible ways to place the receiver.

When you compute distances, be aware that they are distances in a 3D space.

Problem Author: Mugurel Ionut Andreica
Problem Source: Romanian Open Contest, December 2001

1169. Pairs

Time limit: 1.0 second
Memory limit: 64 MB

A Romanian software company has bought **N** computers, which are going to be connected so that they may form a network. A connection can be made between any 2 distinct computers and is bidirectional (if the 2 computers are labeled i and j, then data can be sent both from i to j and from j to i). Your job is to determine a way to connect all the N computers, in such a way that every 2 computers will be able to send data between them (directly or using other computers as intermediate devices).

There is only one extra requirement: the network must contain exactly **K** critical pairs. A pair (i, j) is critical if there exists a connection which, if removed, data communication between i and j will become impossible.

Input

The input consists of 2 integer numbers: N ($1 \leq N \leq 100$), the number of computers the network will contain and K ($0 \leq K \leq N*(N-1)/2$), the number of critical pairs the network will contain.

Output

You should output the connections which form the network, one connection per line. A connection is described by a pair (i, j), which means that i and j are directly connected. The 2 numbers of the pair should be separated by a blank. If you cannot build a network which contains exactly K critical pairs, then you should output -1.

Sample

input	output
7 12	1 2 1 3 2 3 3 4 4 5 4 6 4 7 5 6 5 7 6 7

Problem Author: Mugurel Ionut Andreica
Problem Source: Romanian Open Contest, December 2001

1170. Desert

Time limit: 1.0 second
Memory limit: 64 MB

Sergeant Gica (this is a Romanian name) has just been announced that he is going to be sent out in the desert for training. He will have to walk **L** meters away from the army's headquarters, on a straight line. Since he is a bit lazy and because he has been allowed to pick up his own route, sergeant Gica takes a look at the map of the region. On the map, the army's headquarters are at the coordinates (0, 0). There are also **N** rectangular regions which have strictly positive coordinates marked on the map. Each rectangle has a given "delay coefficient". This means that the time needed to cross a line segment which is contained inside the rectangle, is equal to the product between the length of the segment and the delay coefficient of the rectangle. The total time needed to travel all the L meters is equal to the sum of the times needed to travel every segment of the straight line which lies inside the zones marked on the map (including the desert).

Note that the desert (the space between the N rectangular zones) has its own delay coefficient.

Input

The 1st line contains **N** ($1 \leq N \leq 500$), the number of rectangular zones marked on the map. The next N lines contain 5 numbers each: **x1 y1 x2 y2 c**. (x1, y1) are the coordinates of the lower left corner of the rectangle, (x2, y2) are the coordinates of the upper right corner of the rectangle and c is the delay coefficient of the rectangle. Then there is one more line which contains 2 numbers: **c0** (the delay coefficient of the desert) and **L** (the number of meters sergeant Gica has to travel).

All numbers are positive integers not greater than 32000. L will be larger than the distance from (0, 0) (the headquarters) to any point located on the edges of the rectangles (this means that the sergeant's "trip" will always end up in the desert and not inside any of the rectangular zones marked on the map).

Output

On the 1st line, you should output the minimum time needed to travel L meters on a straight line, starting from (0, 0) and ending at a point with positive coordinates. On the 2nd line you should output the coordinates of the point where sergeant Gica should go to (this point must be L meters away from (0, 0) and must have positive coordinates). If there are more solutions with the same minimum time, you may output any one of them.

You should output all 3 numbers with 6 decimal digits.

Sample

input	output
1 1 1 2 2 1 2 3	4.585786 2.121320 2.121320

Notes

The sergeant walks a total of 1.414.. meters inside the rectangular zone and (3-1.414..) meters outside the zone (through the desert). Thus, the total time is: 1.414.. * 1 + (3 - 1.414..) * 2.

Problem Author: Mugurel Ionut Andreica
Problem Source: Romanian Open Contest, December 2001

1171. Lost in Space

Time limit: 1.0 second
Memory limit: 4 MB

An astronaut of the RSA (Romanian Space Agency) was "forgotten" on the **N-th** level of a Romanian Space Station. He wants to go down to the 1st level, where the communication devices are located, so he can call a space ship to take him home. Unfortunately, the astronaut doesn't know how long the ship will take until it gets to the space station. That's why he wants to gather as much space food as possible before he sends out the message to the ship.

Every level of the space station contains **16** rooms, arranged in 4 rows and 4 columns (numbered accordingly). Every room contains an amount of space food (a number between 1 and 255).

The astronaut may move freely inside the space station, but he is not allowed to visit the same room twice. Moreover, if he went to a lower level, he is not allowed to move back to a higher level. From every room, he could move either north, south, east or west (on the same level) or down (on the same row and column, but at the level below), if there is a door to the level below in that room. For each level, a map is given which tells which rooms have doors to the level below them.

Whenever he enters a room, the astronaut gets the amount of space food found in that room. The food ratio of the astronaut is defined as the total quantity of food gathered during his trip inside the space station divided by the number of days he spends inside the space station before he sends out the signal to the space ship. It is considered that the astronaut spends the 1st day in the room he begins his trip, on the N-th level, and that he gets the amount of space food found in this room during this day. He is only allowed to move once per day.

You have to find a path from the N-th level to the 1st level, which has the maximum food ratio. Note that the astronaut does not have to call the ship as soon as he gets to a room on the 1st level. He may move around the level first, gather the necessary food and only then call the ship.

Input

The first line of input contains a single number: N ($1 \leq N \leq 16$), the number of levels of the space station. For each level, there will be 8 lines of input containing its description. The first four lines will contain four integers, representing the amount of space food found in the corresponding room on that level (the number found on the j-th position on the i-th line represents the amount of food in the room found on row i and column j on the level). The next four lines will contain four integers, in the range 0..1. 1 means that there is a door from that room to the level below, 0 means that there isn't one. Level 1 will have only 0s on these four lines (there is no level below level 1).

The order of the levels in the input will be from top to bottom (from level N to level 1). The last line of input will contain 2 numbers **r** and **c**, representing the row and column of the room the astronaut is initially in, on level N.

Output

On the 1st line, you should output the maximum possible food ratio for the astronaut, with 4 decimal digits. On the 2nd line you should output the length of his path (print 0 if the astronaut never gets out of the room he is initially found in). If the length of his path is **L**, L > 0, then on the 3rd line of output you should output L characters: 'N', 'E', 'S', 'W' or 'D', each character corresponding to one direction of movement (north, east, south, west and down). If there are more solutions with the same maximum food ratio, then you may output any of them.

Note that, if L is the length of the astronaut's path, then he spends L+1 days before he calls the space ship.

Every test case is guaranteed to have at least 1 solution.

Sample

input	output
2 1 20 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 20 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1	8.6000 4 EDSW

Problem Author: Mugurel Ionut Andreica
Problem Source: Romanian Open Contest, December 2001

1172. Ship Routes

Time limit: 1.0 second
Memory limit: 16 MB

A Romanian tourist went on a trip to the Mediterranean Sea. He arrived to one of the cities of the 3 islands he is going to visit. Every island has exactly **N** cities and they are all ports. The tourist plans to begin his journey from the city he is in, visit all the other **3*N-1** cities exactly once and then return to the starting city so he may go back home after that.

Unfortunately, there are cannibals along the roads on all the 3 islands. That's why travelling on the road between 2 cities on the same island is very dangerous and, consequently, prohibited. Hopefully, there are always ship routes. Every pair of cities which are not on the same island is connected by such a ship route. There are no routes between cities which are on the same island.

The tourist wants to know in how many ways he can plan his journey through the 3 islands.

Input

The input contains a single number: N ($1 \leq N \leq 30$), the number of cities on each of the 3 islands.

Output

You should output a single number: the number of ways the tourist can plan his trip. Note that 2 trips are identical if the successions of the 3*N cities are identical or if the succession of the 3*N cities of the first trip is the same as the succession of the 3*N cities of the 2nd trip, read backwards (for instance, if every island had 1 city, numbered according to the island's number, the trips 1-2-3-1 and 1-3-2-1 would be identical).

Sample

input	output
2	16

Problem Author: Mugurel Ionut Andreica
Problem Source: Romanian Open Contest, December 2001

1173. Lazy Snail

Time limit: 1.0 second
Memory limit: 64 MB

Here, in Romania, all snails are lazy. Take Wally the Snail, for example. He has to visit **N** friends which are located at distinct coordinates in the plane. But since he is so lazy, he doesn't want to leave his house. He said that he will go visit his friends if someone can show him the right path to follow.

He wants to leave his house, visit all of his friends exactly once and then return to his house. Between 2 friends' houses or between his house and a friend's house, he walks on the straight line which connects them. 'Is that all ?', someone asked. Wally realized that this would be too easy, so he added that, during his trip, no two line segments along which he travels should cross (except for every 2 consecutive segments, which cross at one end). You must find a path for Wally, so he can go visit all of his friends (although he doesn't want to).

Input

On the 1st line of input, there will be 2 real numbers: X and Y, separated by a blank, representing the coordinates of Wally's house. On the 2nd line, there will be an integer number: N ($2 \leq N \leq 1000$), the number of friends Wally has to visit. On the next N lines, there will be 3 numbers, separated by blanks: X , Y and ID. ID will be an integer number, representing the ID of one of Wally's friends. X and Y will be 2 real numbers, representing the coordinates of Wally's friend's house (they will be given with at most 3 decimal digits and will be in the range -100000 .. 100000).

All IDs are unique, between 1 and N. No 3 friends (including Wally) have their houses on the same straight line.

Output

You should output N+2 lines: the IDs of the friends whose houses Wally is about to visit, in the order he visits them. Start with Wally's ID, continue with the ID of the friend he visits first and so on. Finish with Wally's ID. Wally has ID 0.

If there is no solution, then print a single line, containing the number -1.

Sample

input	output
0 0 3 3 3 1 6 0 2 6 2 3	0 1 3 2 0

Problem Author: Mugurel Ionut Andreica
Problem Source: Romanian Open Contest, December 2001

1174. Weird Permutations

Time limit: 0.5 second
Memory limit: 64 MB

Three Romanian programmers developed this new algorithm which generates all the N! permutations with N elements in a specific order, they called the **transposition order**. The algorithm starts with the permutation 1 2 3 .. N. Then it chooses a pair of two adjacent elements (that is, two elements which are located on consecutive positions) and switches them. This way, they get a new permutation. They do the same for this new permutation and they obtain a new one and so on, until all the N! permutations are generated. You realize that the algorithm must be pretty smart in order to generate all the N! permutations exactly once (without repetitions).

Hopefully, your task will not be to write such an algorithm. In fact, you are given the files perm.pas and perm.cpp, which are two implementations of this algorithm (in Pascal and C++). They read the integer **N** ($1 \le N \le 100$) from the keyboard and print to the file **perm.txt** all the N! permutations, one per line, in the order in which the algorithm generates them.

What you have to do is, given a permutation, to find out its index in the list of permutations generated by the algorithm.

Perm.pas

```
const
  fileout = 'perm.txt';
  MAXN = 100;

var
  fout :text;
  n, i :integer;
  permut :array [1..MAXN] of integer;
  position :array [1..MAXN] of integer;
  dir :array [1..MAXN] of integer;

procedure PrintPermutation;
begin
  for i := 1 to n do
    write(fout, ' ', permut[i]);
  writeln(fout);
end;

procedure Switch(p1, p2 :integer);
var
  xch :integer;
begin
  xch := permut[p1];
  permut[p1] := permut[p2];
  permut[p2] := xch;
  position[permut[p1]] := p1;
  position[permut[p2]] := p2;
end;

procedure GeneratePermutation(nn :integer);
var ii :integer;
begin
  if (nn = n + 1) then
    PrintPermutation
  else
  begin
    GeneratePermutation(nn + 1);
    for ii := 1 to nn - 1 do
    begin
      Switch(position[nn],
             position[nn] + dir[nn]);
      GeneratePermutation(nn + 1);
    end;
    dir[nn] := -dir[nn];
  end;
end;

begin
  readln(n);
  for i := 1 to n do
  begin
    permut[i] := i;
    position[i] := i;
    dir[i] := -1;
  end;

  assign(fout, fileout);
  rewrite(fout);

  GeneratePermutation(1);

  close(fout);
end.
```

Perm.cpp

```
#include <stdio.h>

const char *fileout = "perm.txt";
const int MAXN = 100;

FILE *fout;
int n, i;
int permut[MAXN + 1];
int position[MAXN + 1];
int dir[MAXN + 1];

void PrintPermutation()
{
  for (i = 1; i <= n; i++)
    fprintf(fout, " %d", permut[i]);
  fprintf(fout, "\n");
}

void Switch(int p1, int p2)
{
  int xch = permut[p1];
  permut[p1] = permut[p2];
  permut[p2] = xch;
  position[permut[p1]] = p1;
  position[permut[p2]] = p2;
}

void GeneratePermutation(int nn)
{
  int ii;

  if (nn == n + 1)
    PrintPermutation();
  else
  {
    GeneratePermutation(nn + 1);
    for (ii = 1; ii <= nn - 1; ii++)
    {
      Switch(position[nn],
             position[nn] + dir[nn]);
      GeneratePermutation(nn + 1);
    }
    dir[nn] = -dir[nn];
  }
}

int main()
{
  scanf("%d", &n);
  for (i = 1; i <= n; i++)
  {
    permut[i] = i;
    position[i] = i;
    dir[i] = -1;
  }

  fout = fopen(fileout, "wt");

  GeneratePermutation(1);

  fclose(fout);
  return 0;
}
```

Input

The first line contains a single integer: N ($1 \leq N \leq 100$). The 2nd line contains N integers separated by blanks. They are the given permutation with N elements.

Output

Print one single integer, which will be the index of the permutation in the list of N! permutations generated by the algorithm described above.

Sample

input	output
4 2 3 1 4	17

Notes

Run the 2 given programs for N=4 and you will notice that the permutation 2 3 1 4 will be on the 17th line of the file perm.txt.

Problem Author: Mugurel Ionut Andreica. The Pascal and C++ programs of the three programmers are improved (and, obviously, modified) versions of two programs written by Frank Ruskey (I found them on the Web). So, thanks Frank!
Problem Source: Romanian Open Contest, December 2001

1175. Strange Sequence

Time limit: 1.0 second
Memory limit: 2 MB

You have been asked to discover some important properties of one strange sequences set. Each sequence of the parameterized set is given by a recurrent formula:

$$X_{n+1} = F(X_{n-1}, X_n),$$

where n > 1, and the value of F(X,Y) is evaluated by the following algorithm:

1. find $H = (A_1*X*Y + A_2*X + A_3*Y + A_4)$;
2. if $H > B_1$ then H is decreased by C until $H \le B_2$;
3. the resulting value of H is the value of function F.

The sequence is completely defined by nonnegative constants A_1, A_2, A_3, A_4, B_1, B_2 and C.

One may easily verify that such sequence possess a property that $X_{p+n} = X_{p+q+n}$ for appropriate large enough positive integers p and q and for all $n \ge 0$. You task is to find the minimal p and q for the property above to hold. Pay attention that numbers p and q are well defined and do not depend on way minimization is done.

Input

The first line contains seven integers: A_1, A_2, A_3, A_4, B_1, B_2 and C. The first two members of sequence (X_1 and X_2) are placed at the second line. You may assume that all intermediate values of H and all values of F fit in range [0..100000].

Output

An output should consist of two integers (p and q) separated by a space.

Sample

input	output
0 0 2 3 20 5 7 0 1	2 3

Problem Author: Alexander Klepinin
Problem Source: Third USU personal programming contest, Ekaterinburg, Russia, February 16, 2002

1176. Hyperchannels

Time limit: 1.0 second
Memory limit: 64 MB

The Galaxy Empire consists of N planets. Hyperchannels exist between most of the planets. New Emperor urged to extend hyperchannels network in such a way, that he can move from any planet to any other using no more than one channel. One can pass through the channel only in one direction.

The last channel-establishing ship is located on the base near planet A. This ship can't pass through the existing channel, it always establishes a new one. But presence of two channels connecting the same planets in one direction makes navigation too difficult, almost impossible. The problem is to find a route for this ship to establish all necessary channels with no excessive ones. In the end of this route ship should return to the base.

Input

First line contains integer $N \leq 1000$ and number of the planet A ($A \leq N$) where the base is situated. Each of the following N lines contain N numbers, the j-th number of the i-th line equals to 1 if there exists channel from planet i to planet j, and equals to 0 otherwise. It is known, that Empire can fulfill its need of hyperchannels by establishing no more than 32000 new ones.

Output

Output should contain the sequence in which channels should be established. Each line should contain two integers — numbers of source and destination planet of channel. You may assume, that solution always exists.

Sample

input	output
4 2 0 0 1 0 0 0 1 0 1 1 0 1 0 0 1 0	2 4 4 1 1 2 2 1 1 4 4 2

Problem Author: Pavel Atnashev
Problem Source: Third USU personal programming contest, Ekaterinburg, Russia, February 16, 2002

1177. Like Comparisons

Time limit: 1.0 second
Memory limit: 64 MB

Development team of new DBMS asks you to write subroutine for the ‘like’ operator.
‘Like’ operator works as following. It returns true if text string matches specified template. Template is a text string containing any symbols or following special sequences:

%	matches any number of any characters
_	matches any single character
[c1-c2]	matches any single character in the range c1-c2
[c1c2c3…cN]	matches any single character of the set {c1,c2,c3,…,cN}
[^c1-c2]	matches any single character not in the range c1-c2
[^c1c2c3…cN]	matches any single character not in the set {c1,c2,c3,…,cN}

Input

First line contains number of tests $N \leq 1000$. Next N lines contain comparisons in the following format:

```
'string' like 'template'
```

String or template may contain any symbols with ASCII codes 32-255. Inner entrance of apostrophe symbol (ASCII 39) into string or template is encoded by double apostrophe symbol. Maximal length of string or template is 100 symbols.

Output

For each of N comparisons output single 'YES' or 'NO' at a line.

Sample

input	output
15 'abcde' like 'a' 'abcde' like 'a%' 'abcde' like '%a' 'abcde' like 'b' 'abcde' like 'b%' 'abcde' like '%b' '25%' like '_5[%]' '_52' like '[_]5%' 'ab' like 'a[a-cdf]' 'ad' like 'a[a-cdf]' 'ab' like 'a[-acdf]' 'a-' like 'a[-acdf]' '[]' like '[[]]' '''''' like '_''' 'U' like '[^a-zA-Z0-9]'	NO YES NO NO NO NO YES YES YES YES NO YES YES YES NO

Problem Author: Pavel Atnashev
Problem Source: Third USU personal programming contest, Ekaterinburg, Russia, February 16, 2002

1178. Akbardin's Roads

Time limit: 1.0 second
Memory limit: 64 MB

Great Akbardin decided to build new roads in his caliphate. He wants to build minimal number of roads so that one can travel from any town to any other using only these roads. But this problem is too difficult for him and his mathematicians. So, at first, they decided to build straight roads between towns in such a way, that every town becomes connected with only one other. Because crossroads make movement dangerous, no two roads should intersect.

You task is to make plan of the roads being given coordinates of towns.

Input

First line contains an even integer N ($N \le 10000$) — the number of towns. Each of the next N lines contains pair of integers — coordinates of i-th town x_i, y_i ($-10^9 < x_i, y_i < 10^9$). No three towns lay on one line.

Output

Output $N/2$ lines with description of one road on each. Road is identified by pair of towns it connects. If there are several correct answers, output any of them.

Sample

input	output
4 0 2 1 1 3 4 4 4	1 3 2 4

Problem Author: Pavel Atnashev
Problem Source: Third USU personal programming contest, Ekaterinburg, Russia, February 16, 2002

1179. Numbers in Text

Time limit: 1.0 second
Memory limit: 64 MB

During building of roads, Akbardin read many statistical reports. Each report contained a lot of numbers. But different reports contained numbers in different numeric systems. And Akbardin asks his mathematicians a question – in what numeric system text contains maximal amount of numbers. Number is a sequence of digits, with non-digits to the left and right. Capital Latin letters are used in k-based system with k > 10 ('A' = 10, 'B' = 11, …, 'Z' = 35).

You task is to help mathematicians to solve this problem and save their heads.

Input

Text consists of digits, capital Latin letters, spaces and line breaks. Size of input doesn't exceed 1 Mb.

Output

Output should contain two integers: base of numeric system K ($2 \le K \le 36$) and amount of numbers. If more than one answer is possible, output the one with a less K.

Sample

input	output
01234B56789 AZA	11 4

Problem Author: Pavel Atnashev
Problem Source: Third USU personal programming contest, Ekaterinburg, Russia, February 16, 2002

1180. Stone Game

Time limit: 1.0 second
Memory limit: 64 MB

Two Nikifors play a funny game. There is a heap of N stones in front of them. Both Nikifors in turns take some stones from the heap. One may take any number of stones with the only condition that this number is a nonnegative integer power of 2 (e.g. 1, 2, 4, 8 etc.). Nikifor who takes the last stone wins. You are to write a program that determines winner assuming each Nikifor does its best.

Input

An input contains the only positive integer number N (condition $N \leq 10^{250}$ holds).

Output

The first line should contain 1 in the case the first Nikifor wins and 2 in case the second one does. If the first Nikifor wins the second line should contain the minimal number of stones he should take at the first move in order to guarantee his victory.

Sample

input	output
8	1 2

Problem Author: Dmitry Filimonenkov
Problem Source: Third USU personal programming contest, Ekaterinburg, Russia, February 16, 2002

1181. Cutting a Painted Polygon

Time limit: 1.0 second
Memory limit: 64 MB

There is a convex polygon with vertices painted in three colors: Red (R), Green (G) and Blue (B). It is known that all the colors are present and any two neighbor vertices have different colors. You are to find out whether it is possible to cut this polygon with noncrossing diagonals so that each of the obtained triangles would have all vertices of different colors: one red, one green and one blue vertex. Point out a possible way of the cutting if the cutting is possible.

Input

The first line contains a number N of the polygon vertices ($4 \le N \le 1000$). There are N symbols of the set {'R', 'G', 'B'} in the second line that specify a color for the correspondent vertex.

Output

The first line should contain either a number of drawn diagonals in case the required cutting is possible or the number 0 otherwise (cutting is impossible). In the first case the following lines should contain a description of the drawn diagonals. The description of a diagonal takes one line and consists of diagonal vertices numbers. The numbers are separated with a space. If there are several possible cuttings that satisfy the requirements you may output any of them.

Sample

input	output
7 RBGBRGB	4 1 3 3 7 5 7 5 3

Problem Author: Dmitry Filimonenkov
Problem Source: Third USU personal programming contest, Ekaterinburg, Russia, February 16, 2002

1182. Team Them Up!

Time limit: 1.0 second
Memory limit: 64 MB

Your task is to divide a number of persons into two teams, in such a way, that:

- everyone belongs to one of the teams;
- every team has at least one member;
- every person in the team knows every other person in his team;
- teams are as close in their sizes as possible.

This task may have many solutions. You are to find and output any solution, or to report that the solution does not exist.

Input

For simplicity, all persons are assigned a unique integer identifier from 1 to N.

The first line contains a single integer number N ($2 \le N \le 100$) - the total number of persons to divide into teams, followed by N lines - one line per person in ascending order of their identifiers. Each line contains the list of distinct numbers A_{ij} ($1 \le A_{ij} \le N$, $A_{ij} \ne i$) separated by spaces. The list represents identifiers of persons that i^{th} person knows. The list is terminated by 0.

Output

If the solution to the problem does not exist, then write a single message “No solution” (without quotes). Otherwise write a solution on two lines. On the first line write the number of persons in the first team, followed by the identifiers of persons in the first team, placing one space before each identifier. On the second line describe the second team in the same way. You may write teams and identifiers of persons in a team in any order.

Samples

input	output
5 3 4 5 0 1 3 5 0 2 1 4 5 0 2 3 5 0 1 2 3 4 0	No solution
5 2 3 5 0 1 4 5 3 0 1 2 5 0 1 2 3 0 4 3 2 1 0	3 1 3 5 2 2 4

Problem Author: Vladimir Kotov, Roman Elizarov
Problem Source: 2001-2002 ACM Northeastern European Regional Programming Contest

1183. Brackets Sequence

Time limit: 1.0 second
Memory limit: 64 MB

Let us define a regular brackets sequence in the following way:

1. Empty sequence is a regular sequence.
2. If S is a regular sequence, then (S) and [S] are both regular sequences.
3. If A and B are regular sequences, then AB is a regular sequence.

For example, all of the following sequences of characters are regular brackets sequences:

```
(), [], (()), ([]), ()[], ()[()]
```

And all of the following character sequences are not:

```
(, [, ), )(, ([)], ([(]
```

Some sequence of characters '(', ')', '[', and ']' is given. You are to find the shortest possible regular brackets sequence, that contains the given character sequence as a subsequence. Here, a string $a_1a_2...a_n$ is called a subsequence of the string $b_1b_2...b_m$, if there exist such indices $1 \le i_1 < i_2 < ... < i_n \le m$, that $a_j=b_{i_j}$ for all $1 \le j \le n$.

Input

The input contains at most 100 brackets (characters '(', ')', '[' and ']') that are situated on a single line without any other characters among them.

Output

Write a single line that contains some regular brackets sequence that has the minimal possible length and contains the given sequence as a subsequence.

Sample

input	output
`([(]`	`()[()]`

Problem Author: Andrew Stankevich
Problem Source: 2001-2002 ACM Northeastern European Regional Programming Contest

1184. Cable Master

Time limit: 1.0 second
Memory limit: 64 MB

Inhabitants of the Wonderland have decided to hold a regional programming contest. The Judging Committee has volunteered and has promised to organize the most honest contest ever. It was decided to connect computers for the contestants using a “star” topology - i.e. connect them all to a single central hub. To organize a truly honest contest, the Head of the Judging Committee has decreed to place all contestants evenly around the hub on an equal distance from it.

To buy network cables, the Judging Committee has contacted a local network solutions provider with a request to sell for them a specified number of cables with equal lengths. The Judging Committee wants the cables to be as long as possible to sit contestants as far from each other as possible.

The Cable Master of the company was assigned to the task. He knows the length of each cable in the stock up to a centimeter, and he can cut them with a centimeter precision being told the length of the pieces he must cut. However, this time, the length is not known and the Cable Master is completely puzzled.

You are to help the Cable Master, by writing a program that will determine the maximal possible length of a cable piece that can be cut from the cables in the stock, to get the specified number of pieces.

Input

The first line contains two integers N and K, separated by a space. N ($1 \le N \le 10000$) is the number of cables in the stock, and K ($1 \le K \le 10000$) is the number of requested pieces. The first line is followed by N lines with one number per line, that specify the length of each cable in the stock in meters. All cables are at least 1 meter and at most 100 kilometers in length. All lengths are written with a centimeter precision, with exactly two digits after a decimal point.

Output

Write the maximal length (in meters) of the pieces that Cable Master may cut from the cables in the stock to get the requested number of pieces. The number must be written with a centimeter precision, with exactly two digits after a decimal point.

If it is not possible to cut the requested number of pieces each one being at least one centimeter long, then the output must contain the single number “0.00” (without quotes).

Sample

input	output
4 11 8.02 7.43 4.57 5.39	2.00

Problem Author: Vladimir Pinaev, Roman Elizarov
Problem Source: 2001-2002 ACM Northeastern European Regional Programming Contest

1185. Wall

Time limit: 1.0 second
Memory limit: 64 MB

Once upon a time there was a greedy King who ordered his chief Architect to build a wall around the King's castle. The King was so greedy, that he would not listen to his Architect's proposals to build a beautiful brick wall with a perfect shape and nice tall towers. Instead, he ordered to build the wall around the whole castle using the least amount of stone and labor, but demanded that the wall should not come closer to the castle than a certain distance. If the King finds that the Architect has used more resources to build the wall than it was absolutely necessary to satisfy those requirements, then the Architect will loose his head. Moreover, he demanded Architect to introduce at once a plan of the wall listing the exact amount of resources that are needed to build the wall.

Your task is to help poor Architect to save his head, by writing a program that will find the minimum possible length of the wall that he could build around the castle to satisfy King's requirements.

The task is somewhat simplified by the fact, that the King's castle has a polygonal shape and is situated on a flat ground. The Architect has already established a Cartesian coordinate system and has precisely measured the coordinates of all castle's vertices in feet.

Input

The first line contains two integers N and L separated by a space. N ($3 \le N \le 1000$) is the number of vertices in the King's castle, and L ($1 \le L \le 1000$) is the minimal number of feet that King allows for the wall to come close to the castle.

Next N lines describe coordinates of castle's vertices in a clockwise order. Each line contains two integers X_i and Y_i separated by a space ($-10000 \le X_i, Y_i \le 10000$) that represent the coordinates of i^{th} vertex. All vertices are different and the sides of the castle do not intersect anywhere except for vertices.

Output

Write the single number that represents the minimal possible length of the wall in feet that could be built around the castle to satisfy King's requirements. You must present the integer number of feet to the King, because the floating numbers are not invented yet. However, you must round the result in such a way, that it is accurate to 8 inches (1 foot is equal to 12 inches), since the King will not tolerate larger error in the estimates.

Sample

input	output
9 100 200 400 300 400 300 300 400 300 400 400 500 400 500 200 350 200 200 200	1628

Problem Author: Sergey Volchenkov, Roman Elizarov
Problem Source: 2001-2002 ACM Northeastern European Regional Programming Contest

1186. Chemical Reactions

Time limit: 1.0 second
Memory limit: 64 MB

Bill teaches chemistry in the school and has prepared a number of tests for his students. Each test has a chemical formula and a number of possible reaction outcomes that his students are to choose one correct from. However, Bill wants to make sure that he has not made any typos while entering his tests into a computer and that his students won't easily throw away wrong answers simply by counting a number of chemical elements on the left and on the right side of the equation, which should be always equal in a valid reaction.

You are to write a program that will help Bill. The program shall read the description of the test for the students that consists of the given left side of the equation and a number of possible right sides, and determines if the number of chemical elements on each right side of the equation is equal to the number of chemical elements on the given left side of the equation.

To help you, poor computer folks, that are unaware of the complex world of chemistry, Bill has formalized your task. Each side of the equation is represented by a string of characters without spaces, and consists of one or more chemical sequences separated by a '+' (plus) characters. Each sequence has an optional preceding integer multiplier that applies to the whole sequence and a number of elements. Each element is optionally followed by an integer multiplier that applies to it. An element in this equation can be either distinct chemical element or a whole sequence that is placed in round parenthesis. Every distinct chemical element is represented by either one capital letter or a capital letter that is followed by a small letter.

Even more formally, using notation that is similar to BNF, we can write:

- <formula> ::= [<number>] <sequence> { '+' [<number>] <sequence> }
- <sequence> ::= <element> [<number>] { <element> [<number>] }
- <element> ::= <chem> | '(' <sequence> ')'
- <chem> ::= <uppercase_letter> [<lowercase_letter>]
- <uppercase_letter> ::= 'A'..'Z'
- <lowercase_letter> ::= 'a'..'z'
- <number> ::= '1'..'9' { '0'..'9' }

Every distinct chemical element is said to occur in the given formula for some total number X, if X is the sum of all separate occurrences of this chemical element multiplied to all numbers that apply to it. For example, in the following chemical formula:

```
C2H5OH+3O2+3(SiO2)
```

- `C` occurs for a total of 2 times.
- `H` occurs for a total of 6 times (5 + 1).
- `O` occurs for a total of 13 times (1 + 3*2 + 3*2).
- `Si` occurs for a total of 3 times.

All multipliers in the formula are integer numbers that are at least 2 if explicitly specified and are 1 by default. Each chemical formula is at most 100 characters long, and every distinct chemical element is guaranteed to occur for a total of no more than 10000 times in each formula.

Input

The first line represents a chemical formula that is to be tested as the left side of the equation. The second line contains a single integer number N ($1 \le N \le 10$), which is the number of right sides of the equation that are to be tested. Each one of the following N lines represents one such formula.

Output

You are to write N lines — one line per each possible answer of the chemical test for Bill's students that is given in the input. For each right-hand side formula that is encountered in the input, you should output:

```
<left_formula>==<right_formula>
```

if the total number of occurrences of each distinct chemical element on the left-hand side equals to the total number of occurrences of this chemical element on the right-hand side. Otherwise write:

```
<left_formula>!=<right_formula>
```

Here `<left_formula>` must be replaced exactly (character by character) with the original left-hand side formula as it is given in the first line of the input, and `<right_formula>` must be replaced exactly with each right-hand side formula as they are given in the input. Do not place any spaces in the lines you write.

Sample

input	output

C2H5OH+3O2+3(SiO2) 7 2CO2+3H2O+3SiO2 2C+6H+13O+3Si 99C2H5OH+3SiO2 3SiO4+C2H5OH C2H5OH+3O2+3(SiO2)+Ge 3(Si(O)2)+2CO+3H2O+O2 2CO+3H2O+3O2+3Si	C2H5OH+3O2+3(SiO2)==2CO2+3H2O+3SiO2 C2H5OH+3O2+3(SiO2)==2C+6H+13O+3Si C2H5OH+3O2+3(SiO2)!=99C2H5OH+3SiO2 C2H5OH+3O2+3(SiO2)==3SiO4+C2H5OH C2H5OH+3O2+3(SiO2)!=C2H5OH+3O2+3(SiO2)+Ge C2H5OH+3O2+3(SiO2)==3(Si(O)2)+2CO+3H2O+O2 C2H5OH+3O2+3(SiO2)!=2CO+3H2O+3O2+3Si

Problem Author: Joseph Romanosky, Roman Elizarov
Problem Source: 2001-2002 ACM Northeastern European Regional Programming Contest

1187. Statistical Trouble

Time limit: 1.0 second
Memory limit: 64 MB

Your team was hired by the international corporation ACM (Analytical Calculation Maxims). Every year ACM creates and conducts various surveys. Surveys themselves are simple forms with a list of questions and a list of possible answers for every question. Surveys are distributed around the globe, where field agents question the target group of people. All the answers are gathered in the ICPC (International Computation and Processing Center), where teams of well-paid analysts mine raw data in search for relevant correlations. The raw data for each individual survey consists of lots of lines of answers. Each line corresponds to every questioned person and for every question lists answers that the person has made on that particular survey.

The first step of analysis that your team was hired to automate is to create cross tables that correlate answers on interesting pairs of questions. In its most simple way, given a pair of questions, cross table has a row for every possible answer on the first question, and has a column for every possible answer on the second question. Each cell of the cross table contains a number of lines in the raw data that has both answers for the corresponding questions at the same time.

However, your task is complicated by the fact that you are to compute and output not only simple cross table values, but also total values for every row and column in the cross table (that is the sum of values in the corresponding row and column) that are placed in an additional last column and last row, as well as a percentage distributions for every row and column. Percentage distribution for a row is an additional number in every cell in that row that shows percent ratio of the value in that cell to the total value for that row, unless the total value is zero (in that case percentage distribution for this row is not defined). The same applies to the percentage distributions of columns. Thus, the cross table in your output will have at most three values in every cell (the value itself, row-wise percent, and column-wise percent). Please note, that percentage distributions also apply to totals. For example, in the total column for every row the row-wise percent will be always 100%, unless the total value for the row is zero (in that case row-wise percents are not defined), and column-wise percent shows percents ratio of the total value for this row to the total number of lines in the raw data (which is the value that can be found in the last column of the last row).

Percents are rounded to integers on output. Percent that has a non-zero fractional part is rounded to either the smallest integer number greater than the resulting percent, or the largest integer number smaller than the resulting percent, in such a way, that the sums of all corresponding row-wise percents by row (without row totals) or column-wise percents by column (without column totals) are equal to 100% unless they are undefined. There are various rounding algorithms that produce results satisfying the above constraints. You are free to use any rounding algorithm as long as the above constraints are satisfied.

Input

The input consists of 3 sections: survey description, survey results, and cross table descriptions.

The first line of the input contains the name of the survey, which is at most 100 characters long. Subsequent lines describe all the questions in the survey. On the first line of every question there is a 3-character question code (capital letters and digits only) followed by a space, and followed by the question name, which is at most 80 characters long. Each subsequent line for a question describes one possible answer on the question and starts with a space, followed by a single-character code for the answer (capital letter, digit, or character '.', '*', or '@'), followed by a space and followed by an answer description, which is at most 40 characters long. The list of questions is terminated by the line with a single character '#'. All answer codes are unique within the question, and all question codes are unique within the input. There are at least 2 and at most 10 possible answers per question and at least 2 and at most 100 questions.

Next lines in the input describe survey results. Every line contains a character per question (in the order they appear in the input) that gives the answer code for the corresponding question. The characters follow one another without any delimiters. This section is terminated by the line with a single character '#'. There is at least one line with answers in the section and at most 10000 answers in total (the number of lines times the number of questions).

Next lines in the input describe cross tables that are to be created. Each cross table description occupies one line. That line contains the code for the first question, followed by a space, followed by the different code for the second question, followed by a space, and followed by the cross table name, which is at most 100 characters long. This section is terminated by the line with a single character '#'. There are at most 100 cross table descriptions in the input.

The input has no trailing spaces on any line. All names do not start or end with a space, but may contain spaces.

Output

Write to the output a cross table for every cross table description in the input in the order they appear in the input. On the first line of the cross table write the survey name, followed by a space, followed by a '-' (dash) character, followed by a space, followed by the cross table name. Then write the description of the first question, and the description of the second question exactly as they appear in the input and in the same format. Then write an empty line, followed by the table itself. The table contains exactly `1+3*(N1+1)` lines and exactly `6*(N2+2)` characters on every line, where `N1` is the number of possible answers for the first question, and `N2` is the number of possible answers for the second question.

The table has one line for column headings, and `N1+1` rows (3 lines per row). The first `N1` of these rows correspond to the answers on the first question in the order they appear in the input, and the last row is for column totals. The table also has `N2+2` columns, where each column is 6 characters wide. The first column is for row headings; the subsequent `N2` columns correspond to the answers on the second question in the order they appear in the input, and the last column is for row totals. All information in the cells (including headings) is aligned to the right and is padded on the left with spaces to become 6 characters wide.

The heading for the first column is empty. The headings for the subsequent `N2` columns are composed from the second question code, followed by a ':' (colon) character, and followed by the corresponding answer code. The heading for the last column is the string "TOTAL" (without quotes). The headings for the first `N1` 3-line rows of the cross table are composed from the first question code, followed by a ':' (colon) character, and followed by the corresponding answer code. The heading for the last row is the string "TOTAL" (without quotes). Row headings are situated on the first line of the corresponding row. The subsequent 2 lines in the heading column of every row must be blank.

All non-heading cells in the table contain computed values and percents. On the first line of every cell the corresponding cross table integer value is situated. The second line contains properly rounded to integers row-wise percent, with a mandatory trailing '%' (percent) character, or a single '-' (dash) character if the corresponding row-wise percent is not defined. The third line contains column-wise percent in the same format. All cross tables in the output must be separated by a single empty line.

Sample

input	output

```
New Year Phone Survey for ACM ICPC
Q01 Hello!
 H Hello!
 Y Yes!
 * Uhm...
 . (silence)
 @ (other)
Q02 How are you?
 H Hello!
 Y Yes!
 F Fine!
 Q Who are you?
 @ (other)
BYE Happy New Year!
 Y You too.
 * (censored)
 @ (other)
 . (hang up)
#
.@.
HH@
.@.
YFY
HQ*
H@.
YYY
.H@
HFY
HH@
#
Q01 Q02 Health vs greeting style
Q02 BYE Politeness matrix
#
```

```
New Year Phone Survey for ACM ICPC - Health vs greeting style
Q01 Hello!
 H Hello!
 Y Yes!
 * Uhm...
 . (silence)
 @ (other)
Q02 How are you?
 H Hello!
 Y Yes!
 F Fine!
 Q Who are you?
 @ (other)

       Q02:H Q02:Y Q02:F Q02:Q Q02:@ TOTAL
 Q01:H     2     0     1     1     1     5
          40%    0%   20%   20%   20%  100%
          66%    0%   50%  100%   33%   50%
 Q01:Y     0     1     1     0     0     2
           0%   50%   50%    0%    0%  100%
           0%  100%   50%    0%    0%   20%
 Q01:*     0     0     0     0     0     0
           -     -     -     -     -     -
           0%    0%    0%    0%    0%    0%
 Q01:.     1     0     0     0     2     3
          33%    0%    0%    0%   67%  100%
          34%    0%    0%    0%   67%   30%
 Q01:@     0     0     0     0     0     0
           -     -     -     -     -     -
           0%    0%    0%    0%    0%    0%
 TOTAL     3     1     2     1     3    10
          30%   10%   20%   10%   30%  100%
         100%  100%  100%  100%  100%  100%

New Year Phone Survey for ACM ICPC - Politeness matrix
Q02 How are you?
 H Hello!
 Y Yes!
 F Fine!
 Q Who are you?
 @ (other)
BYE Happy New Year!
 Y You too.
 * (censored)
 @ (other)
 . (hang up)

       BYE:Y BYE:* BYE:@ BYE:. TOTAL
 Q02:H     0     0     3     0     3
           0%    0%  100%    0%  100%
           0%    0%  100%    0%   30%
 Q02:Y     1     0     0     0     1
         100%    0%    0%    0%  100%
          33%    0%    0%    0%   10%
 Q02:F     2     0     0     0     2
         100%    0%    0%    0%  100%
          67%    0%    0%    0%   20%
 Q02:Q     0     1     0     0     1
           0%  100%    0%    0%  100%
           0%  100%    0%    0%   10%
 Q02:@     0     0     0     3     3
           0%    0%    0%  100%  100%
           0%    0%    0%  100%   30%
 TOTAL     3     1     3     3    10
          30%   10%   30%   30%  100%
         100%  100%  100%  100%  100%
```

Problem Author: Roman Elizarov

Problem Source: 2001-2002 ACM Northeastern European Regional Programming Contest

1188. Library

Time limit: 1.0 second
Memory limit: 64 MB

Castaway Robinson Crusoe is living alone on a remote island. One day a ship carrying a royal library has wrecked nearby. Usually Robinson brings any useful stuff from the shipwreck to his island, and this time he has brought a big chest with books.

Robinson has decided to build a bookcase for these books to create his own library. He cut a rectangular niche in the rock for that purpose, hammered in wooden pegs, and placed wooden planks on every pair of pegs that have the same height, so that all planks are situated horizontally and suit to act as shelves.

Unfortunately, Robinson has discovered that one especially old and big tome does not fit in his bookcase. He measured the height and width of this tome and has decided to redesign his bookcase in such a way, as to completely fit the tome on one of the shelves, taking into account locations of other shelves and the dimensions of the niche. With each shelf in the bookcase, one of the following operations should be made:

1. Leave the shelf on its original place.
2. Move the shelf to the left or to the right.
3. Shorten the shelf by cutting off a part of the plank and optionally move it to the left or to the right.
4. Move one of the pegs to a different place at the same height and move the shelf to the left or to the right.
5. Shorten the shelf by cutting off a part of the plank, move one of the pegs to a different place at the same height, and optionally move the shortened shelf to the left or to the right.
6. Remove the shelf from the bookcase along with both supporting pegs.

We say that the shelf is properly supported by its pegs, if exactly two distinct pegs support the shelf and the center of the shelf is between its pegs or coincides with one of the pegs. The original design of Robinson's library has all the shelves properly supported by their pegs and lengths of all shelves are integer number of inches. The Robinson may only cut an integer number of inches from the planks, because he has no tools for more precise measurements. All remaining shelves after the redesign must be properly supported by their pegs.

You are to find the way to redesign Robinson's library to fit the special old tome without changing original design too much. You have to minimize the number of pegs that are to be removed from their original places during the redesign (operations 4 and 5 remove one peg, and operation 6 removes two pegs). If there are different ways to solve the problem, then you are to find the one that minimizes the total length of planks that are to be cut off (operations 3 and 5 involve cutting something from the planks, and operation 6 counts as if cutting off the whole plank). Width of planks and diameter of pegs shall be considered zero.

The tome may not be rotated. The tome should completely (to all its width) stand on one of the shelves and may only touch other shelves, their pegs or niche's edge.

Input

The first line contains four integers X_N, Y_N, X_T, and Y_T, separated by spaces. They are, correspondingly, width and height of the niche, and width and height of the old tome in inches ($1 \le X_N, Y_N, X_T, Y_T \le 1000$).

The second line contains a single integer N ($1 \le N \le 100$) that represents the number of the shelves. Then N lines follow. Each line represents a single shelf along with its two supporting pegs, and contains five integer numbers y_i, x_i, l_i, x_{1_i}, x_{2_i}, separated by spaces, where:

- y_i ($0 < y_i < Y_N$) — the height of the i^{th} shelf above the bottom of the niche in inches.
- x_i ($0 \le x_i < X_N$) — the distance between the left end of the i^{th} shelf and the left edge of the niche in inches.
- l_i ($0 < l_i \le X_N - x_i$) — the length of the i^{th} shelf in inches.
- x_{1_i} ($0 \le x_{1_i} \le l_i/2$) — the distance between the left end of the i^{th} shelf and its leftmost supporting peg in inches.
- x_{2_i} ($l_i/2 \le x_{2_i} \le l_i$; $x_{1_i} < x_{2_i}$) — the distance between the left end of the i^{th} shelf and its rightmost supporting peg in inches.

All shelves are situated on different heights and are properly supported by their pegs. The problem is guaranteed to have a solution for the input data.

Output

Output two integers separated by a space. The first one is the minimal number of pegs that are to be removed by Robinson from their original locations to place the tome. The second one is the minimal total length of planks in inches that are to be cut off during the redesign that removes the least number of pegs.

Samples

input	output

11 8 3 4 4 1 1 7 1 4 4 3 7 1 6 7 2 6 3 4 2 0 3 0 3	0 0
11 8 4 6 4 1 1 7 1 4 4 3 7 1 6 7 2 6 3 4 2 0 3 0 3	1 3

Problem Author: Elena Kryuchkova, Roman Elizarov
Problem Source: 2001-2002 ACM Northeastern European Regional Programming Contest

1189. Pairs of Integers

Time limit: 1.0 second
Memory limit: 64 MB

You are to find all pairs of integers such that their sum is equal to the given integer number N and the second number results from the first one by striking out one of its digits. The first integer always has at least two digits and starts with a non-zero digit. The second integer always has one digit less than the first integer and may start with a zero digit.

Input

The input consists of a single integer N ($10 \le N \le 10^9$).

Output

Write the total number of different pairs of integers that satisfy the problem statement. Then write all those pairs. Write one pair on a line in ascending order of the first integer in the pair. Each pair must be written in the following format:

$X + Y = N$

Here X, Y, and N must be replaced with the corresponding integer numbers. There should be exactly one space on both sides of '+' and '=' characters.

Sample

input	output
302	5 251 + 51 = 302 275 + 27 = 302 276 + 26 = 302 281 + 21 = 302 301 + 01 = 302

Problem Author: Vladimir Lelyukh, Roman Elizarov
Problem Source: 2001-2002 ACM Northeastern European Regional Programming Contest

1190. Bar of Chocolate

Time limit: 1.0 second
Memory limit: 64 MB

On the wrapper of a bar of chocolate, the producer must specify all the components of the product. Strictly speaking, producers should also specify the weight fractions of all the components, but most of them do not do so because the consumers won't buy a chocolate if they read, for example, "Water 80%". However, producers prefer to specify the weight fractions of some components. For example, the inscription "Whole nuts 90%" is likely to attract buyers. But the buyer must remember that the components are always written in non-increasing order of weight fractions. Accordingly, the beautiful inscription "Cocoa-butter, Water, Cocoa-powder 40%", evidently, is a lie.

Write a program that determines whether an inscription on a bar of chocolate is a lie.

Input

The first line contains the number of components N ($1 \le N \le 5000$). Each of the following N lines contains a description of one component. Each description starts with the name of the component, which is no longer than 16 symbols and may contain only upper- or lowercase English letter, hyphens, and underline characters. All the components have different names. Then there is a space followed by the number 0 or 1; 0 means that the producer didn't specify the weight fraction of the component. If the number is 1, it is followed by a space and the weight fraction given in hundredths of percent. All specified weight fractions are given in non-increasing order. Weight fractions, both real and specified on the wrapper, are integers from 1 to 10000 (in hundredths of percent).

Output

Output NO if the inscription is certainly a lie; output YES if the inscription is consistent and can be true.

Sample

input	output
4 Water 0 Cocoa-butter 0 Cocoa-powder 1 4000 Lecithin 0	NO

Problem Author: Leonid Volkov
Problem Source: Fifth High School Children Programming Contest, Ekaterinburg, March 02, 2002

1191. Catch the thief!

Time limit: 1.0 second
Memory limit: 64 MB

A thief is fleeing a place of crime. A cop follows him, with a time lag of L minutes. They run equally fast, thus the lag between them remains constant. Finally, feeling tired, the thief reaches a tram stop. He wants to take a tram of a specific route; trams follow each other with an interval of exactly K_1 minutes on this route at any time of day and night. When the tram comes, the thief boards it. The policeman comes to the same tram stop. If the thief is still there waiting for the tram to arrive, the policeman arrests him. If the thief is gone, the cop himself waits for a tram of the same route. The thief leaves his tram at some stop and starts waiting for a tram of another route (trams of that route keep an interval of exactly K_2 minutes). When a tram arrives, the thief gets on it and continues his way. Of course, the cop leaves his tram also at this stop and, if the thief is still there, arrests him. If the thief managed to leave, the policeman waits for a tram of the same route that the thief used and follows the thief…

This process continues until either the policeman arrests the thief or the thief, having used N trams, reaches his secret cover place where he is safe.

Although the speeds of the cop and the thief remain equal all the time and the speeds of their trams are equal, it may happen that the policeman is lucky to overtake the thief. For instance, if $L < K_1$, then it may happen that the policeman reaches the first stop when the thief is still waiting for a tram there. Other situations are also possible. Write a program that determines whether the cop can catch the thief.

Input

The input consists of two lines: in the first line there are the numbers L ($0 < L < 100$) and N ($0 < N < 100$) delimited with a space. In the second line there are time intervals K_1, K_2, …, K_N ($0 < K_i < 100$) between trams of the corresponding routes. These numbers are also separated with spaces. All numbers in the input data are integers.

Output

Output NO if the cop has no chance to overtake the thief before he reaches his cover place; output YES if he still has such a chance.

Samples

input	output
8 3 6 4 3	NO
15 4 7 3 13 6	YES

Problem Author: Leonid Volkov
Problem Source: Fifth High School Children Programming Contest, Ekaterinburg, March 02, 2002

1192. Ball in a Dream

Time limit: 1.0 second
Memory limit: 64 MB

A little boy likes throwing balls in his dreams. He stands on the endless horizontal plane and throws a ball at an angle of a degrees to the plane. The starting speed of the ball is V m/s. The ball flies some distance, falls down, then jumps off, flies again, falls again, and so on.

As far as everything may happen in a dream, the laws of the ball's motion differ from the usual laws of physics:

- the ball moves in the gravity field with acceleration of gravity equal to 10 m/s^2;
- the rebound angle equals the angle of fall;
- after every fall, the kinetic energy of the ball decreases by a factor of K;
- there is no air in the dream;
- "Pi" equals to 3.1415926535.

Your task is to determine the maximal distance from the point of throwing that the ball can fly.

Input

The input contains three numbers: $0 \le V \le 500000$, $0 \le a \le 90$, and $K > 1$ separated by spaces. The numbers V and a are integers; the number K is real.

Output

The output should contain the required distance in meters rounded to two fractional digits.

Sample

input	output
5 15 2.50	2.08

Problem Author: Igor Goldberg
Problem Source: Fifth High School Children Programming Contest, Ekaterinburg, March 02, 2002

1193. Queue at the Exam

Time limit: 0.5 second
Memory limit: 64 MB

A group of students are having an oral examination. At the beginning of the exam, all students simultaneously receive their questions and start preparing for the answer. Each student needs T_1 minutes for the preparation and T_2 minutes for the answer itself (these parameters can be different for different students). For each student, the time T_3 (in minutes from the planned beginning of the exam) is given when this student has to be free because she has other things to do (for example, other exams).

During the exam, a queue of students is formed as they are getting ready to speak. If a student is ready to answer and at that time moment the professor is free, then this student starts answering immediately. If the professor is busy with another person, then the student joins the queue and starts answering when the student before her in the queue finishes her examination.

It is possible that some students won't be free when they planned to be (i.e., at the time T_3). The professor is ready to cooperate and can shift the beginning of the exam to an earlier time. However, he doesn't want to come too early! You task is to write a program that will calculate the minimal period in minutes by which the exam should be shifted so that all the students will manage to complete the exam before their T_3 time.

Input

The first line contains the number of students N ($1 \le N \le 40$). Each of the next N lines contains the corresponding numbers T_1, T_2, and T_3. The numbers are separated with spaces and satisfy the constraints $0 \le T_1 \le T_3 \le 600$, $1 \le T_2 \le 240$. All the numbers T_1 are distinct.

Output

Output the nonnegative integer that is the answer to the problem. If there is no need to shift the beginning of the exam, output 0.

Samples

input	output
3 100 10 120 70 40 150 99 15 400	15
2 100 10 110 80 15 100	0

Problem Author: Anatoly Uglov
Problem Source: Fifth High School Children Programming Contest, Ekaterinburg, March 02, 2002

1194. Handshakes

Time limit: 1.0 second
Memory limit: 64 MB

A party at the "Prancing Pony" tavern was over and the hobbits were breaking up. At the first crossroad all the company fell into several groups, each of which went its own way. As a matter of courtesy, the hobbits shook each other's hands before parting (each hobbit shook the hand of each hobbit he was parting with). Each group fell at the next crossroad into several smaller groups (with handshakes of course) and so on. This process continued until single hobbits and married couples reached their homes. In other words, the groups were splitting up until there were only groups of two or one hobbits left. Your task is to calculate the number of the handshakes made.

Input

Let's numbers the groups of hobbits so that the first group (the one that left the tavern) gets number 1 and other groups get distinct positive integers greater than 1. In the first line of the input there are the total number of hobbits N and the number of married couples K. These numbers meet the following conditions: $2 < N \le 20000$; $0 \le 2K \le N$. Each of the next lines of the input starts with the number of the group and the number of groups it fell into. After that there are several pairs of numbers representing the number and size of each new group. It is guaranteed that if a group no. Y formed as a result of splitting of a group no. X, then the description of the group no. X occurs before the description of the group no. Y. In particular, this means that the description of the group no. 1 is in the second line of the input. If the group no. Y formed as a result of splitting of the group no. X and its description is absent, then the group no. Y didn't split further.

Output

Output the total number of the handshakes made.

Sample

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

Problem Author: Leonid Volkov
Problem Source: Fifth High School Children Programming Contest, Ekaterinburg, March 02, 2002

1195. Ouths and Crosses

Time limit: 1.0 second
Memory limit: 64 MB

The Olympic Winter Games. The Ouths and Crosses on the 3×3 Board Event. The Panel of Judges received an unfinished game with 3 moves made by each player. The players couldn't continue the game because they had lost too much blood at the doping control. The Panel of Judges had to determine the outcome in the case of the optimal play of both parties.

Since there were as many crosses as oughs, the Panel of Judges wanted to decide that it was a draw. However, Head Judge observed that one player was able to win the game in one move. Your task is to determine the outcome in the case of the optimal play of both rivals. According to the International Rules, Crosses move first in the game and the winner is he who puts three of his symbols in the same row, column, or diagonal.

Input

The input contains a 3×3 table consisting of the symbols X, O (the capital English letters, which mark the moves of Crosses and Ouths), and # (this symbol denotes an empty field). The table contains exactly three Crosses and three Ouths. It is guaranteed that the game is not finished, i.e. any row, column or diagonal contains neither three Crosses nor three Ouths.

Output

If Crosses win the game, output “Crosses win”. If Ouths win, output “Ouths win”. In the case of a draw, output “Draw”.

Samples

input	output
XXO #X# #OO	Ouths win
O#O #X# XOX	Draw

Problem Author: Leonid Volkov, Oleg Katz
Problem Source: Fifth High School Children Programming Contest, Ekaterinburg, March 02, 2002

1196. History Exam

Time limit: 1.5 second
Memory limit: 64 MB

Professor of history decided to simplify the examination process. At the exam, every student should write a list of historic dates she knows (she should write the years only and, of course, must be able to explain what event took place in this or that year). Professor has a list of dates that students must know. In order to decide upon the student's mark, Professor counts the number of dates in the student's list that are also present in his list. The student gets her mark according to the number of coincidences.

Your task is to automatize this process. Write a program that would count the number of dates in the student's list that also occur in Professor's list.

Input

The first line contains the number N of dates in Professor's list, $1 \le N \le 15000$. The following N lines contain this list, one number per line. Each date is a positive integer not exceeding 10^9. Professor's list is sorted in non-descending order. The following line contains the number M of dates in the student's list, $1 \le M \le 10^6$. Then there is the list itself; it is unsorted. The dates here satisfy the same restriction. Both in Professor's and in the student's lists dates can appear more than once.

Output

Output the number of dates in the student's that are also contained in Professor's list.

Sample

input	output
2 1054 1492 4 1492 65536 1492 100	2

Problem Author: folklore
Problem Source: Fifth High School Children Programming Contest, Ekaterinburg, March 02, 2002

1197. Lonesome Knight

Time limit: 1.0 second
Memory limit: 64 MB

The statement of this problem is very simple: you are to determine how many squares of the chessboard can be attacked by a knight standing alone on the board. Recall that a knight moves two squares forward (horizontally or vertically in any direction) and then one square sideways (perpedicularly to the first direction).

Input

The first line contains the number N of test cases, $1 \le N \le 64$. Each of the following N lines contains a test: two characters. The first character is a lowercase English letter from 'a' to 'h' and the second character is an integer from 1 to 8; they specify the rank and file of the square at which the knight is standing.

Output

Output N lines. Each line should contain the number of the squares of the chessboard that are under attack by the knight.

Sample

input	output
3 a1 d4 g6	2 8 6

Problem Author: folklore
Problem Source: Fifth High School Children Programming Contest, Ekaterinburg, March 02, 2002

1198. Jobbery

Time limit: 1.5 second
Memory limit: 96 MB

Hard times came for Martian senate. Even this pride of Martian democracy can not oppose the almighty jobbery. Let's consider the procedure of typical decision making. A member of Martian senate, who needs a certain law, submits it to senate. To improve his chances that law is accepted, he makes a phone call to each of senate members on whom he has the goods in his safe. Then he kindly suggests to those senators to support the new law. Moreover, to avoid occasional rejection of this important law, he asks each of them to make the same procedure with their safes. And each of them having no choice makes a similar range of phone calls to those on whome, in turn, he has the goods. If every senator supports the new law the president has nothing to do but to approve it. Otherwise he can reject it and send back to senate for law improvements.

It is evident, that just elected president Honestman dislikes such situation. So he starts to struggle against the jobbery. And first of all he wants to put to jail the most dangerous senate members. And definitely, if senator is able to make even the harmful law approved, he is a dangerous one. So secret service of Martian president has already checked safes of each senate member, and found out on whom each of them has the goods. Martian president knows about your achievements in programming and he asks you personally for a help.

Input

The first line of input contains single integer N — the number of senate members ($1 \le N \le 2000$). Each senate member is uniquely identified with a number from 1 to N. Each of subsequent N lines contains information about senate members. The i-th line contains list of senate members (given by numeric identifiers) on whome he has the goods. List is terminated by number 0.

Output

Print the list of identifiers of all dangerous senate members in a single line. The numbers in the list must be present in increasing order. The list must be terminated by number 0.

Sample

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

Problem Author: Nikita Rybak

1199. Mouse

Time limit: 2.0 second
Memory limit: 64 MB

In the kitchen lives a mouse. There are also a cat and a piece of cheese in the kitchen. The coordinates of the cheese and the mouse are known, and the cat is sleeping. Finally, there is some furniture in the kitchen. The furniture is a set of convex polygons. The mouse wants to get to the cheese unnoticed. A point of the route is called dangerous if the distance to the nearest piece of furniture is greater than 10 cm. It is required to find the least dangerous route for the mouse, i.e., the route in which the sum of the lengths of dangerous segments is minimal.

Input

In the first line there are four numbers x_m, y_m, x_c, y_c separated with a space. They are the coordinates of the mouse (x_m, y_m) and of the cheese (x_c, y_c). In the second line there is the number of pieces of furniture N ($0 \le N \le 100$). The next N lines describe these pieces. Each description starts with the number of vertices of the corresponding polygon K ($3 \le K \le 10$), given in a separate line. Each of the next K lines contains two numbers, which are the coordinates of the corresponding vertex. It is known that the distance between any two points of different polygons is greater than 20 cm (so that it would be easier for the cat to catch the mouse). Neither the mouse nor the cheese are inside any of the polygons. All the coordinates are given in meters and have no more than three fractional digits. The absolute values of coordinates do not exceed 10^5.

Output

You should give the mouse's route in the form of a broken line. In the first line output the number of its vertices (including the initial and final ones). Then give the coordinates of the vertices, two numbers per line, accurate to 10^{-4}. Each segment of the broken line must be either entirely dangerous or entirely safe (with the possible exception of its endpoints). The broken line must contain no more than 1000 vertices.

Sample

input	output
1.0 1.5 0.0 1.5 1 4 0.0 0.0 0.0 1.0 1.0 1.0 1.0 0.0	4 1.0 1.5 1.0 1.1 0.0 1.1 0.0 1.5

Problem Author: Nikita Rybak