

Problem A. Little Brackets

Input file: **brackets.in**
Output file: **brackets.out**
Time limit: 1 second
Memory limit: 64 megabytes

Consider all regular bracket sequences with one type of brackets. Let us call the *depth* of the sequence the maximal difference between the number of opening and the number of closing brackets in a sequence prefix. For example, the depth of the sequence “()()()” is 2, and the depth of “(((()())())” is 4. Find out the number of regular bracket sequences with n opening brackets that have the depth equal to k . For example, for $n = 3$ and $k = 2$ there are three such sequences: “()()()”, “(())()”, “(()())”.

Input

Input file contains several test cases. Each test case is described with n and k ($1 \leq k \leq n \leq 50$).

Last testcase is followed by two zeroes. They should not be processed.

Output

For each testcase output the number of regular bracket sequences with n opening brackets that have the depth equal to k .

Separate output for different testcases by a blank line. Adhere to the format of the sample output.

Example

brackets.in	brackets.out
3 2	Case 1: 3
37 23	
0 0	Case 2: 203685956218528

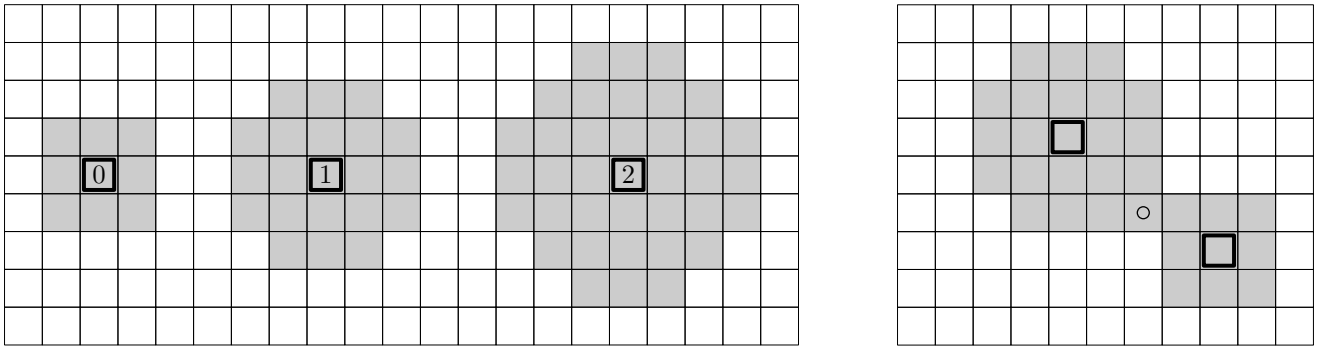
Problem B. Under Control

Input file: control.in
 Output file: control.out
 Time limit: 1 second
 Memory limit: 64 megabytes

In a game of *Civilization III* the area controlled by a city is defined by its culture level. The game proceeds on a rectangular grid. A city occupies one grid square. Each city has a *culture level* which is a non-negative integer number.

A city with a culture level 0 controls its own square and eight adjacent squares. A city with a culture level 1 additionally controls all squares that share a side with those squares (a total of $9 + 12 = 21$ squares). Generally, if a city with a culture level i controls the set A of squares, a city with the same location and a culture level $i + 1$ would control all these squares and also squares that share a side with at least one square from A .

The picture on the left shows the sets of squares controlled by cities with culture levels of 0, 1 and 2.



The area controlled by the civilization is defined as follows. Consider the total area controlled by all its cities. The civilization area is the smallest set of squares, such that it contains all the squares controlled by some city, and its complement contains no hanging squares. A square x of a set B is called *hanging* if there is no 2×2 square in B that contains x .

Calculate the total area controlled by a civilization, given the locations of all its cities on a map. You may consider that the map is infinite and that there are no other civilizations.

Input

The first line of the input file contains an integer number n — the number of the cities of a civilization ($1 \leq n \leq 50$). Next n lines describe cities. Each city is described with its integer coordinates (x_i, y_i) and its culture level c_i . Coordinates do not exceed 10^9 by their absolute value, culture level does not exceed 10.

Output

Output the total number of squares controlled by a civilization.

Example

control.in	control.out
2 0 0 1 4 -3 0	31

The squares controlled by the civilization in the example are shown on the right picture. The square marked by a small circle is not controlled by any city, however it belongs to the area controlled by the civilization because otherwise it would be hanging.

Problem C. Holidays

Input file: holidays.in
Output file: holidays.out
Time limit: 1 second
Memory limit: 64 megabytes

You must know that our country is well known for its strange holidays system. We celebrate anything we can, and often new public holidays are announced, though sometimes some are cancelled.

In year 3141 some archeologists have discovered the document that they consider to be the log of changes in the system of public holidays in our country from year Y_s till year Y_f , inclusively. Each record of the document has the form

<date-record>, added <date-holiday>
 or
<date-record>, removed <date-holiday>

The record of the first type means that the public holiday on <date-holiday> was announced on day <date-record>, and the record of the second type means that the public holiday on <date-holiday> was cancelled on day <date-record>.

Unfortunately, all dates of records only include day and month, but not the year. Now the archeologists wonder, what maximal number of public holidays could have been there during the years described. Your task in this problem is to answer this question and provide the version of the document with years inserted, that would guarantee this number of the holidays.

You must assume that all records in the document are in the chronological order and that there were no day when two different events took place, that is, all dates of records in the document must be different.

Note that if the holiday was announced on its own day, the year it is announced there is no public holiday on this date. Analogously, if the holiday is cancelled on its own day, there is still the holiday this year (so people do not have to go to work after listening to morning radio programs).

Also recall, that the day of February 29 exists only in leap years, that is, years that are divisible by 4, except those divisible by 100, except those divisible by 400. For example, years 1996, 2000 and 2004 are leap, while 1999 or 2100 are not.

Input

The first line of the input file contains Y_s and Y_f ($1800 \leq Y_s \leq Y_f \leq 2200$, $Y_f - Y_s \leq 100$). Next line contains n — the number of records in the document ($1 \leq n \leq 100$). Next n lines contain the the document records, one on a line. See sample input for more detailed information.

You must not consider any other holidays except those explicitly specified in the document. You may assume that no holiday is removed before it is announced.

Output

On the first line of the output file print the maximal number of public holidays for the given period. After that print n lines — the version of the document with years inserted that provides the specified number of holidays. Adhere to the format of the sample output.

If it is impossible to interpret the document in the specified way, print -1 on the first and the only line of the output file.

Example

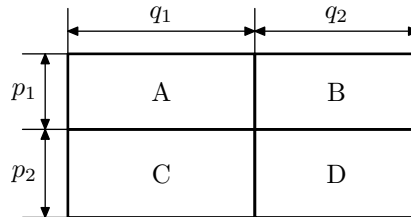
holidays.in	holidays.out
1900 1999	406
9	January 1 1900, added January 1
January 1, added January 1	January 1 1901, added January 7
January 1, added January 7	February 29 1904, added February 29
February 29, added February 29	November 7 1904, added November 7
November 7, added November 7	November 7 1905, removed January 7
November 7, removed January 7	September 1 1906, added May 1
September 1, added May 1	August 21 1907, removed November 7
August 21, removed November 7	September 1 1907, added June 12
September 1, added June 12	September 1 1908, added December 12
September 1, added December 12	

Problem D. Laboratory

Input file: `lab.in`
Output file: `lab.out`
Time limit: 1 second
Memory limit: 64 megabytes

John is working in a secret laboratory. Now he is preparing a special experiment involving four dangerous liquid substances with code names 'A', 'B', 'C', and 'D'.

A special container is used to keep the substances. The container is made of a special strong glass. It consists of four sections: 'A', 'B', 'C', and 'D' as shown on the picture.



Each substance is to be kept in its own part, marked with the letter that coincides with the code name of the substance. Since the container must be transported using standard carts, the height of the container is fixed and must be equal to one inch.

John knows, that he needs a , b , c , and d cubic inches of substances 'A', 'B', 'C', and 'D' respectively. Now he wonders what can be the minimal possible total square of the container so that the required volume of each substance would fit its section. Help him to find it out.

Input

Input file contains four positive integer numbers a , b , c , and d , not exceeding 10^6 .

Output

On the first line of the output file print S — the minimal possible square of the container. On the next line print four real numbers p_1 , p_2 , q_1 , and q_2 — the dimensions of the container sections as shown on the picture. Print at least six digits after the decimal point.

Example

lab.in	lab.out
5 6 8 7	27.8571428571 1.6641005886 1.9414506867 4.1206300292 3.6055512754

Problem E. Maps

Input file: `maps.in`
Output file: `maps.out`
Time limit: 1 second
Memory limit: 64 megabytes

The well known *fixed point theorem* states that if you take two maps of the same region that have different scales and put the smaller one atop the larger so that it is located completely within it, there is the point on the plane that represents the same actual point on both maps. Of course, there may be no common point for three or more maps — pairwise fixed points of different maps may be different.

Consider that you are given three rectangular maps M_1 , M_2 and M_3 of the same region but with different scales $s_1 > s_2 > s_3$ (the scale of the map is the ratio of linear sizes of objects on the map to linear sizes of the actual objects, for any given region the larger is the scale of the map, the larger are the linear sizes of the map). Let the smallest map M_3 be placed atop the largest one M_1 in such a way that it is completely located within it. You have find the way to place the medium map M_2 , in such a way that the following conditions are satisfied:

- M_2 is completely located within M_1 ;
- M_3 is completely located within M_2 ;
- there is the point that represents the same actual place on all three maps.

Input

The first line of the input file contains three real numbers — the scale s_1 and the linear sizes x_1 , y_1 of the largest map ($1 \leq x_1, y_1 \leq 1000$). Let the map be placed on the cartesian plane (with OX axis directed rightwards and OY axis directed upwards) in such a way that its corners are in the points with coordinates $(0, 0)$, $(x_1, 0)$, (x_1, y_1) and $(0, y_1)$.

The second line of the input file contains one real number s_2 — the scale of the medium map.

The third line of the input file contains nine real numbers — the scale s_3 of the smallest map and the coordinates of its four corners, given in counterclockwise order, starting from the corner corresponding to the lower-left one of the largest map. It is guaranteed that the input data is correct, that is, the map is rectangular, linear sizes of the map are all in compliance with its scale and linear sizes of the largest map, and the map is completely within the largest map.

You may also assume that the smallest map has no common points with the border of the largest map (although you may put the medium map in such a way that it has common border points with either of the other maps).

The scales of the maps satisfy $10^2 \geq s_1 > s_2 > s_3 \geq 10^{-2}$, it is guaranteed that the difference between any two scales is at least 10^{-2} .

Output

If it is impossible to find the requested place for the medium map, print “impossible” on the first line of the output file. In the other case print eight real numbers — the coordinates of the corners of the second map in counterclockwise order, starting from the corner corresponding to the lower-left one of the largest map. Making calculations with the precision of 10^{-6} is satisfactory.

Example

<code>maps.in</code>	<code>maps.out</code>
1.0 10.0 5.0	3.2500000000 2.7083333333
0.3	0.2500000000 2.7083333333
0.2 3.0 2.5 1.0 2.5 1.0 1.5 3.0 1.5	0.2500000000 1.2083333333
	3.2500000000 1.2083333333

Problem F. Crazy Painter

Input file: `painter.in`
Output file: `painter.out`
Time limit: 1 second
Memory limit: 64 megabytes

Crazy painter Henry Daub is planning to draw his new masterpiece. Like all paintings of Henry, the new masterpiece is going to be a rectangle of size $m \times n$ inches, each unit square of which will be painted with some color.

Henry would like to minimize the time needed to paint his masterpiece. He has already created the painting sketch and now he is planning the process of drawing. There are three technical tricks that Henry uses in creating his paintings.

In a turn he can paint either a horizontal line, a vertical line or a single unit square. Painting a horizontal line colors a number of horizontally adjacent unit squares to some color, painting a vertical line colors a number of vertically adjacent squares, and painting a square just colors this square. Painting a horizontal line takes Henry h seconds, a vertical line can be painted in v seconds, and a single square is painted in s seconds.

Since the resulting picture must be accurate, it is not allowed to change the color of the square, that is, a single square may be painted several times, but it must be painted the same color each time.

Help Henry to determine the time needed to paint the picture. Initially the canvas is empty, so each square must be colored.

Input

The first line of the input file contains m , n , h , v and s ($1 \leq m, n \leq 30$, $1 \leq h, v, s \leq 10^5$). Next m lines contain n characters each and describe the future masterpiece. The colors of the squares are identified using small letters of English alphabet.

Output

On the first line of the output file print t — the time needed to paint the masterpiece and k — the number of turns needed to do this. Next k lines must contain the description of the turns. Each turn is first described with one of the letters 'h', 'v' or 's'. In the first two cases four integers numbers must follow — the coordinates of the leftmost topmost and the rightmost bottommost square of the line painted; in the last case just two integers are needed — the coordinates of the square painted. The last character of each line must be the color used in the turn.

Example

<code>painter.in</code>	<code>painter.out</code>
4 4 3 5 2	23 8
aabc	h 1 1 1 2 a
aabc	h 2 1 2 2 a
bbbb	h 3 1 3 4 b
ccbc	v 1 3 4 3 b
	h 4 1 4 2 c
	s 1 4 c
	s 2 4 c
	s 4 4 c

Problem G. Puzzle

Input file: puzzle.in
 Output file: puzzle.out
 Time limit: 1 second
 Memory limit: 64 megabytes

Little Georgie likes puzzles very much. Recently he has found a wooden triangle in the box with old toys. The side of the triangle is n inches long. The triangle is divided into n^2 unit triangles with lines drawn on his surface.

The interesting fact about that triangle is that it is not solid — it consists of two parts. Each of the parts is a connected set of unit triangles. Georgie has put his triangle onto the table and now wonders whether he can separate the parts. He wants to separate them without taking any part of the triangle off the table, just moving the parts by the table surface. The triangle has a small but non-zero thickness, so while being moved the parts must not intersect.

For example, if the triangle is divided into parts as it is shown on the top picture below, Georgie can separate the parts the way he wants. However in the case displayed on the bottom picture, he cannot separate the parts without lifting one of them.

Help Georgie to determine whether he can separate the parts moving them by the surface of the table.

Input

Input file contains one or more testcases. The first line of each testcase contains n ($2 \leq n \leq 50$). Next n lines contain the description of the triangle, i -th of these lines contains $2i - 1$ characters, describing unit triangles in the i -th row, from left to right. Character '0' means that the triangle belongs to the first part of the main triangle, '1' means that it belongs to the second one.

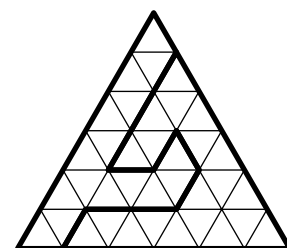
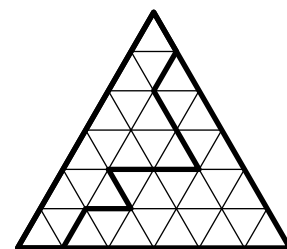
Testcases are separated with blank lines. Testcase with $n = 0$ designates the end of the test data, this testcase must not be processed.

Output

For each puzzle output the line with its number followed by the line that states whether the parts can be separated. Separate output for different cases with a blank line.

Example

puzzle.in	puzzle.out
6	Puzzle 1
0	Parts can be separated
001	
00011	Puzzle 2
0000011	Parts cannot be separated
000111111	
00111111111	
6	
0	
001	
00111	
0011011	
000000111	
00111111111	
0	



Problem H. Quest

Input file: `quest.in`
Output file: `quest.out`
Time limit: 1 second
Memory limit: 64 megabytes

Association of Challenge Masters (ACM) sponsored by International Battle Merchandisers (IBM) is planning to create a series of new quest games. Brotherhood of Tired Virtual Adventurers (BoTVA) has stolen the scripts of the quests and now is planning to publish the solution guide for all the games.

In all quests the adventures take part in some building that has several rooms. Rooms are connected to each other by the doors. The doors are opened using the keys. Also there are characters, objects and items in the game. Keys are considered to be items.

The player in the quest can talk to any character. After he does, the character names several items that he needs to give the player some other items. After the player bring the requested items to him, he get the items promised. Some items can be found on the floor. Also the player may use items on objects. After doing so, he may get some other items.

Given the description of the quest world, create the solution guide for it — the sequence of actions one must do to complete the quest — save the princess.

Input

The first line of the input file contains n — the number of the rooms in the game. Next n lines contain the names of the rooms. Next line contains m — the number of the doors. Next m lines describe doors — each line contains the numbers of the rooms connected by the door and the name of the key to open the door.

The descriptions of the contents of the rooms follow. Each description starts with three integers: c — the number of characters in the room and o — the number of objects in the room and i — the number of items on the floor of the room.

Next lines contain descriptions of the characters. Each description consists of three lines. The first line contains the name of the character. Next line contains one or more item names — the items the character wants the player to bring to him. Last line also contains one or more item names — the names of the items the character would give to the player after he brings him the requested items. Item names are separated by commas. The descriptions of the objects in the same format follow. Note that the difference between objects and characters is that after the player gives some item to the character, it no longer belongs to him, while after using some item on object he keeps the item. Also one cannot talk to objects. Last i lines in the description of the room contain the names of the items on its floor.

Two last lines of the input file contain the name of the room the player starts the game in, and the name of the room where the princess is locked.

All names consist only of letters of English alphabet, digits and spaces. You must ignore leading and trailing spaces in all names. All names are case sensitive. All names are at most 100 characters long. All names are different. The number of rooms is at least two and does not exceed eight. There is at most one door between any two rooms. All doors are two-way. The total number of characters, objects and items in the game does not exceed 200. You always start in the room other then the room where the princess is. No item is requested by more than one character, no item is simultaneously requested by some character and needed by some object. No item is provided by more than one character or object, or both, no item is both on the floor of some room and provided by some character or object. No key opens more than one door, no key is required by any object or character.

Output

Output the sequence of the actions.

Remember that the player can: talk to character, give items to character (after talking to him), take items from character (immediately after giving him items he requests), use items on objects, take items from objects (immediately after using items needed on it), pick items from the floor, open the doors using the keys, walk from room to room (after opening the door that connects them), save the princess (if the player is in the room where the princess is).

Adhere to the format of the sample output. Print one action on a line. The player must always give all requested items to the character, use all needed items on the object, and take items from the character or the object in one action.

If it is impossible to win the game, output “dead princess” on the first line of the output file.

Example

quest.in	quest.out
4 white room black room blue room green room 3 1 3 crystal key 1 4 esmerald key 2 3 strange key 0 0 2 crystal key oranges 0 0 0 2 1 0 Wild Joe rat, red pepper, coin strange key Dead man orange juice red pepper, esmerald key squeezer oranges orange juice 0 1 1 monkey oranges coin rat white room black room	pick crystal key pick oranges open door to blue room go to blue room use oranges on squeezer take orange juice from squeezer talk to Dead man give orange juice to Dead man take red pepper and esmerald key from Dead man go to white room open door to green room go to green room pick rat use oranges on monkey take coin from monkey go to white room go to blue room talk to Wild Joe give red pepper, coin and rat to Wild Joe take strange key from Wild Joe open door to black room go to black room save princess
2 white room black room 0 0 0 0 0 0 0 white room black room	dead princess

Problem I. Stable Sets

Input file: **stable.in**
Output file: **stable.out**
Time limit: 1 second
Memory limit: 64 megabytes

Consider integer numbers in range from 0 to $r - 1$ for some r . They form a ring, denoted by \mathbb{Z}_r . That means that you can perform addition, subtraction and multiplication of the numbers, each time taking the result modulo r .

Let us call some subset $A \subset \{0, 1, \dots, r - 1\}$ *stable* with respect to a polynomial $p(x)$ if $A = \{p(x) | x \in A\}$. Note that the empty set is always stable with respect to any polynomial.

Consider two polynomials

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

and

$$q(x) = b_m x^m + b_{m-1} x^{m-1} + \dots + b_1 x + b_0.$$

Your task is to calculate the number of subsets of \mathbb{Z}_r that are stable with respect to both $p(x)$ and $q(x)$.

Input

The first line of the input file contains r ($2 \leq r \leq 100$). Next line describes $p(x)$, its degree n ($0 \leq n \leq 100$) is followed by $n+1$ integer numbers ranging from 0 to $r-1$, they define a_n, a_{n-1}, \dots, a_0 . Next line describes $q(x)$ in a similar way. A constant zero polynomial has the degree of 0 for the purpose of this problem.

Output

Output the only number — the number of subsets of \mathbb{Z}_r that are stable with respect to both $p(x)$ and $q(x)$.

Example

stable.in	stable.out
6 1 2 0 1 1 2	2

The stable sets in the example are \emptyset and $\{0, 2, 4\}$.