Problem A. Avengers, The

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 second
Memory limit:	64 mebibytes

After Chitauri attack on New York the Avengers team led by Captain America decided to secure the city using power pillars. Having completed defence system construction, Tony Stark approaches programming of the pillars. This brand new security system is based on an algorithm that is quite simple and effective. Once enemy appears at a certain point, a set of pillars defining a convex polygon that contains the point is selected. After that field of force is generated that exterminates every living creature inside as well as at the frontier of the polygon. In order for such system to operate effectively it is required to place an extremely powerful source of energy inside each of the pillars. That is why if one of the pillars happens to be subject to direct effect of the field of force (falls strictly inside one of the polygon), a blast of a colossal power demolishing most of Earth will occur. The same will happen if two different fields (i.e. fields generated by different sets of pillars) of force happen to have non-zero area of intersection.

To provide robustness of the system the city should be contained in a rectangle with power pillars in its vertices.

In order to test the efficiency of the system the team decided to simulate a Chitauri attack on the city. The number of Chitauri attacks as well as the point of the first strike are defined before the test. Additionally, certain shifts $Shift^i$ for each of the following attacks are generated. After that all the attacks are defined in the following way:

1. A set of pillars is defined to exterminate current target.
2. Let point O be the arithmetic mean of the selected pillars coordinates, rounded towards zero, then the next attack point is calculated as follows:

$$x = R.x + (O.x + Shift_x^i) \mod Width$$
$$y = R.y + (O.y + Shift_y^i) \mod Height$$

   where R — coordinates of the lower left pillar, $Width$, $Height$ — width and height of the rectangle surrounding the city.
3. These actions are repeated until every attack is repelled.

Input

The first line contains integer N — number of power pillars ($4 \leqslant N \leqslant 10^4$).

Each of the next N lines contains two integers X, Y — coordinates of the corresponding pillar. The first four points make a rectangle limiting the attack simulation area, with its sides parallel to the coordinate axes. The rectangle points are given in a clockwise winding order, starting with the upper left. It is guaranteed that no three of those N points belong to a single straight line. The next line contains three integers: K, X_0, Y_0 — the number of Chitauri attacks and the coordinates of the first strike ($1 \leqslant K \leqslant 10^5$).

Next $K-1$ lines contain pairs of integers $Shift_x^i$, $Shift_y^i$ are given — shifts of the point of the next attack.

All coordinates and shifts are integer not exceeding 10^4 in absolute value.

Output

For each attack the program must output the number $P \leqslant 10$ — quantity of points in the convex polygon containing the strike point, then print P of numerals p_i — numbers of the pillars, which are vertices of this polygon, listed in the clockwise order.

If there are multiple answers, return any of them.

Example

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

Problem B. Black Widow

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 megabytes

Natalia Romanova a.k.a. Black Widow is the legendary agent of S.H.I.E.L.D. and the amazing spy. She knows dozens of languages and had mastered karate, aikido, jujitsu, kung fu and various Soviet self-defence techniques. Moreover, Natalia has outstanding mental abilities and remarkable acting skills which help her to retrieve information even when she is tortured.

The next mission for Natalia is to steal blueprints of the new Hydra's secret weapon. That was not a big problem to penetrate into the safe through a ventilation. But it is only possible to leave the safe through a passage. Unfortunately, where are pressure sensors on the floor placed in several points. They have a form of segments between walls and those segments are orthogonal to the passage.

Of course, Natalia has the plan of the passage so she knows the distance to each sensor. Agent Romanova wants to go through the passage using the fastest method, i.e. doing a series of backflips. Though, the disadvantage of this method is the fact that a distance between two consecutive touches of a floor (with hands or feet) is always the same.

Despite all her abilities Natalia is not a tall person. What is the minimal distance between two consecutive touches of a floor which guarantees Black Widow will not touch any sensor

Input

There is a number N ($1 \leqslant N \leqslant 1000$) in the first line which is the number of sensors in the passage. There are N integers in the second line which are the distances between the Natalia's starting point and each sensor. Each distance is between 1 and 10^9 inclusive.

Output

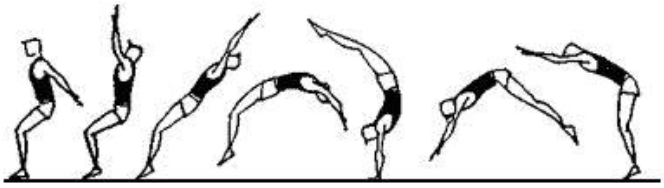
Print one integer — the minimal distance between two consecutive touches of a floor.

Examples

standard input	standard output
5 1 3 4 8 10	6
10 2 3 5 10 12 15 20 30 44 63	8

Note

Backflip is the joining of two jumps: back jump on hands and then on feet.



Problem C. Chitauri

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

As you may know, Chitauri is a warlike race at the service of Thanos. Orders of one of the greatest supervillains can send them to the most remote corners of the galaxy. Thankless tasks, a long way, hostile local population... The only joy — marauding. But still there's a problem — how to divide the loot. After a long time fighting for the parts on one of the captured planets Chitauri heard about "democracy". After understanding this concept they came up with a pattern.

1. They accumulate all the loot. They consider that any two things in the loot have the same value.
2. The oldest and most experienced Chitauri, who is still alive, is chosen to divide the loot rightly.
3. He suggests the distribution of wealth. Other chitauri vote for that distribution.
4. If more than a half of chitauri not agree with the elder, than they kill him and the division starts from step 2.
5. In the other way they divide the loot just as the elder suggested.

You may assume that any two Chitauri differ in their age and/or experience.

Chitauri are not fools, so they wish to stay alive. But they are known as really passionate collectors — they like to collect souvenirs from captured planets. So the next factor in their choices is to maximize the amount of loot they get. Moreover, if the amount of loot doesn't depend of the choice, a Chitauri seeks to kill as many other chitauri as he can. Moreover, while making a decision on how to divide the loot, the Chitauri uses an additional principle: you'd better not anger an experienced chitauri. Thus he selects the variant in which more loot goes to older chitauri.

Unfortunately this process turned out to be terribly tedious. And when chitauri think about something for a long time, they become angry and everything ends with a bloody massacre. Help them to make a choice.

Input

In the only line you are given N and K — the number of chitauri and the amount of goods ($1 \leqslant N \leqslant 1000, 0 \leqslant K \leqslant 1000$).

Output

For every chitauri, starting from the oldest one, output one integer — the amount of goods he will receive or -1 if he will be killed.

Examples

standard input	standard output
2 1	1 0
3 0	-1 0 0

Problem D. Dr. Banner

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

Dr. Banner is an outstanding physicist. He knows gamma rays as no one in the world. Sometimes unfortunately he has ... ehmm... you know... "green" problems. But he came up with an idea that he can deal with the problems by focusing on building pyramids from children's blocks.

For building a series of pyramids Doctor Banner chooses the integer — side length of the block in the foundation. After that he starts to build the pyramid using next algorithm. If the side length of the highest block is K, then he can put a block with integer side length from 1 to $K/2$ on top of it. Or he can just stop building this pyramid and start building the next one.

Banner doesn't like to repeat, so every two pyramids must be different.

Because one of the Banner's best friends is Tony Stark, Doctor has an infinite amount of blocks he needs.

Friends love Banner, especially when he is not green. They know that it takes Doctor Banner one second to builds a pyramid. They want to know how long they can stand close to their friend.

Input

Input consist of a single positive integer N — the length of the block in the foundation of the pyramid ($1 \leqslant N \leqslant 10^5$).

Output

Output the number of seconds friends can stay with Doctor. You should print the answer modulo $10^9 + 7$.

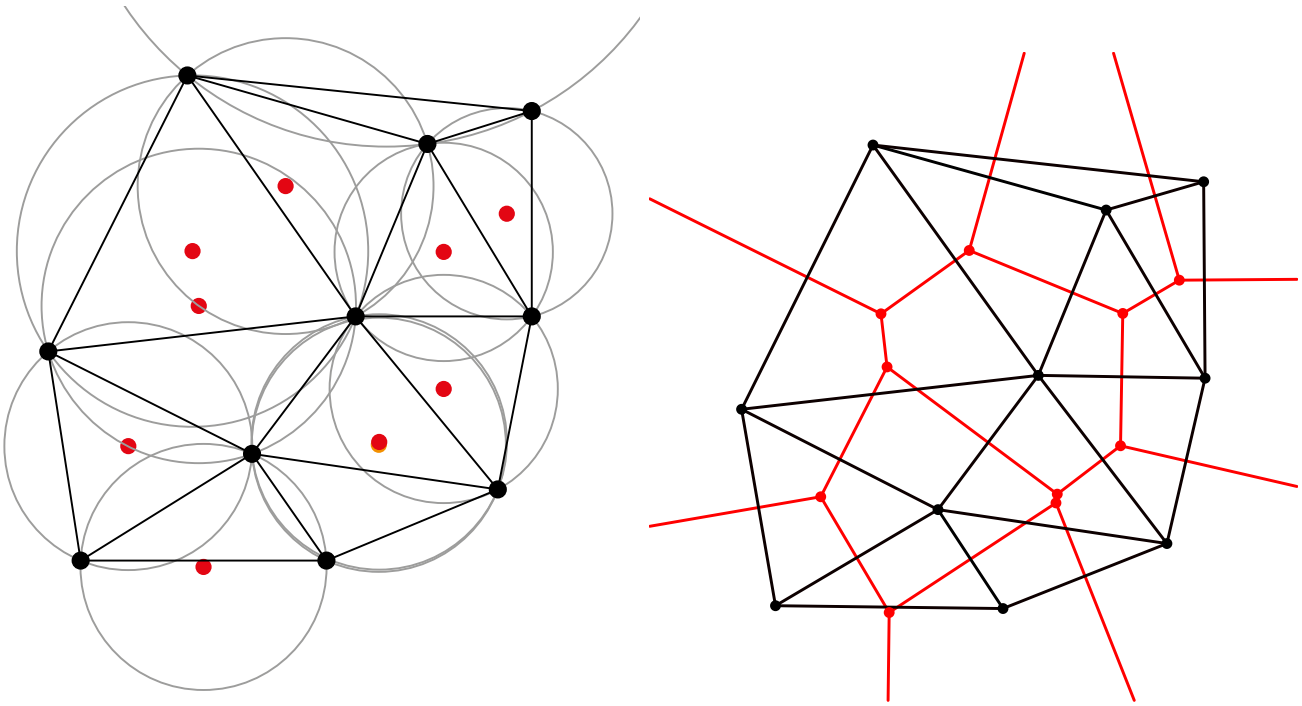
Examples

standard input	standard output
1	1
2	2

Problem E. Egocentric Loki

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	64 megabytes

Loki was growing as a frail Asgardian. He couldn't be compared with Thor in strength and bravery. Lack of these qualities he compensated with guile, agility and intellect. That's why he is a rather proud god. From a child he loves to solve different geometric problems and especially the triangulation problem. After Odin noticed it, he taught Loki how to construct a triangulation with some useful properties. This kind of triangulation was called "Odin's triangulation". If you did not study at Asgardian school, we remind you that Odin's triangulation for a given set of points is a partition into triangles with vertices in these points. Moreover, for each triangle there are no other points from a set (except its vertices) that lie strictly inside its circumcircle. The dual problem for Odin's triangulation is the Allfather's diagram. For the given set of points it is a division of the plane into Allfather's cells in the way that every cell matches some element from the given set and consists of every points for which this element is the nearest among all the elements from the set. It is known that there's one correspondence between Odin's triangulation and Allfather's diagram. Easy to see that in order to build Odin's triangulation one must just connect with the edges points, which cells borders with each other.



For many years Loki used the next iterative algorithm to construct a triangulation from a set of N points with time complexity of $O(N \log N)$:

1. There is only one triangulation possible for a set of three points: there must be edges between every two of them.
2. There are 2 variants for a set of 4 points:
   (a) If these points form a non-convex polygon, we have to put edges between every two of them.
   (b) If these points form a convex polygon, we have to choose any 3 of them and check the fourth point's position with respect to the curcumcircle of the chosen 3 points. There are 3 variants:
      i. The fourth point lies outside the circle. In that case our triangulation is optimal. We have to connect into a triangle first 3 points and add the edges between the fourth point and 2 nearest points from a triangle.

ii. The fourth point lies inside the circle. In this case the Odin's triangulation's condition is failed. We have to add edges between the fourth point and other 3. And in addition, we have to add edges between any two points if there will not be any intersections between edges.

iii. The point lies on the circle. The triangulation is optimal.

3. It there are more than 4 points, the set must be divided into 2 sets with less amount of points. To do this we must draw a vertical or horizontal line in the middle of the set and with respect to these lines we must divide the points into two sets with about the same numbers of points. After that for every groups of points start a recursive algorithm for their division:

   (a) If the number of points $N \geq 12$, this set is divided using straight lines.

   (b) If the number of points $N \geq 12$, then this set is divided into the set of 3 and $N - 3$ points.

   (c) If there are $N = 8$ points, then the set is divided into 2 sets of 4 points. The division ends when there are 3 or 4 points in the set.

4. Combining optimal triangulations. To start with, we choose 2 pairs of points which segments form a convex polygon with the constructed triangulations. They are then connected with an edge and one of these edges is chosen for an upcoming walk. The walk is done as follows: on the current segment we "inflate a bubble" to the inside, until it reaches any point. That point is then connected to the point of the segment hasn't connected to it yet. By this operation we acquire a different segment which is check for the intersection with already existing segments of triangulation, and if there is an intersection, those segments are deleted from triangulation. After that, this new segment is considered as base for the next "bubble". This whole cycle repeats until the segment matches the second segment of the convex hull.

Loki's ego was growing up and he decided to invent an algorithm which constructs the Odin's triangulation with $O(N)$ time complexity. He is terribly proud of it and demonstrates his work whenever possible hoping that Odin's triangulation will be renamed to the Loki's triangulation.

You have a unique opportunity to wound Loki's pride.

Input

In the first line you have a number N — the number of triangles in Loki's triangulation. In the next N lines you have 6 numbers x_1, y_1, x_2, y_2, x_3, y_3 — the coordinates of the triangles.

It is guaranteed that these triangles form a triangulation of a set of points with a size not bigger than 1000 and the absolute value of any coordinate doesn't exceed 10^4.

Output

Print "`YES`" if Loki's triangulation is Odin's triangulation or "`NO`" otherwise.

Examples

standard input	standard output
1 0 0 1 0 0 1	YES
2 1 1 3 0 0 3 3 3 3 0 0 3	NO

Problem F. Fury

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

S.H.I.E.L.D is a huge international law-enforcement and counter-terrorism agency. It have a big impact not only in the Solar system but also out of its borders. This powerful organization, that opposes itself to the army of Thanos, has complicated pseudo-hierarchical structure. According to the rules of this structure every agent can give orders to his direct subordinates (structure is complicated, so even cases when a is direct subordinate of b, and in same time b is direct subordinate of a, are legal). Moreover, to avoid unnecessary bureaucracy, if one agent can give task to another through any chain of agents, he can do it directly.

Unfortunately, after just another crash of the Helicarrier, all databases of the agency were destroyed. Of course backups were never made. Therefore all the information concerning organization's structure was lost. Luckily, however, the Journal remained. By the order of Nick Fury, every agent wrote all his orders in the Journal, which made it possible to determine for every agent, a list of all agents, who receive tasks from him.

Now Fury wants you to restore structure of the organization that most suits the initial one. Namely, in new structure every agent must be able to assign only those tasks, which are correspond to the Journal data or can be derived from it by the given rules.

Moreover due to extremely large size of S.H.I.E.L.D you are asked to build a structure with the minimal count of direct subordination links between agents.

Input

In the first line of input, there are two integer numbers N and M — the number of agents in the organization and the number of the records int the Journal ($1 \leqslant N \leqslant 300, 0 \leqslant M < N \cdot (N-1)$).

Next M lines describe records in the Journal. Every record is a pair of integers J and K, and describes an order given by agent J to agent K ($1 \leqslant J, K \leqslant N$, $J \neq K$).

Output

In the first line, output two positive integer numbers $P(P = N), Q(Q \leqslant M)$, — the number of agents and the number of direct subordination links in your structure.

In the next Q lines output pairs of agents J and K, describing that agent J can give direct orders to agent K.

Example

standard input	standard output
7 11 1 2 2 1 3 6 6 3 4 5 5 4 1 3 1 4 2 5 2 6 3 7	7 9 1 2 2 1 3 6 6 3 4 5 5 4 2 3 1 5 6 7

Problem G. Groot

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

I am Groot.

Input

I am Groot.

Output

I am Groot.

Examples

standard input	standard output
I am Groot	Pfff
I am Groot!	Wow
I am Groot!!!!!!	Wooooooow

Problem I. Iron Man

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1.5 seconds
Memory limit:	64 mebibytes

Even geniuses, billionaires and philanthropists have bad days. Weekend came, and Tony was going to finally prove that P = NP when his creature Ultron, artificial intelligence, took control over Jarvis. But the operations of Stark Industries should go on! Tony will have to lease some cloud servers.

Tony has split all existing tasks into K types and took N servers. The cost of performing a task of each type is known for each server, which is able to perform task of this type.

As far as tasks of various types differ significantly, each server can be set up to perform tasks of only one type. Provider should be contacted in order to alter server configuration settings. Any number of servers can be configured per single request. Each request has a cost and takes some time, that is why reconfiguring servers can be done not more frequently than daily. For technical reasons, configuration of servers can only be done before performing all calculations — during maintenance works in the morning.

Tony has planned the computing for several days. It is planned how many tasks of each type must be performed during each day. Tony was just about to sign a contract wit the provider when Pepper intruded and demanded that the plan that minimizes costs should be selected.

Don't you stand rooted to the ground! Help the Iron Man!

Input

The first line contains three integers N, K, C — quantity of servers, number of task types and cost of server reconfiguration request ($1 \leqslant K \leqslant N \leqslant 15$, $0 \leqslant C \leqslant 10^5$).

The next line contains integer M ($1 \leqslant M \leqslant N \cdot K$); next M lines contain three integers each: S, T, W, where W stands for cost of performing one task of type T on server S ($1 \leqslant S \leqslant N$, $1 \leqslant T \leqslant K$, $1 \leqslant W \leqslant 1000$, for any two different lines of M atleast one of S and T differs).

Next goes integer Q — scope of computing plan in days ($1 \leqslant Q \leqslant 100$).

Each of next Q lines consists of K integers $a_1, a_2, \ldots a_k$, where a_i is quantity of tasks of type i that must be performed this day. $\left(0 \leqslant a_i \leqslant 100, \sum_{i=1}^{K} a_i > 0\right)$

It is guaranteed that solution for all the given test cases exists.

Output

Print one number — minimal possible cost of performing all tasks including cost of server configuration.

Example

standard input	standard output
2 2 6 4 1 1 2 1 2 1 2 1 4 2 2 2 2 1 1 1 10	25

Problem J. Jarvis

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

Established the work of a Stark Industries cluster, Tony proceeded to fix Jarvis. At first, Tony was thinking that Ultron's attack was focused on the source code of his assistant, Jarvis. But after a whole night debugging Tony concluded that the program remained intact. But Jarvis, though it speaks with a human voice, totally depends on the cluster of servers, on which he is launched.

While checking the servers, Tony found that Jarvis in an attempt to self-improvement had complicated exchange and synchronisation algorithms between the servers. But he didn't take into account one important characteristic — the server load.

Finally Tony had understood that he has to change the latency between some pairs of servers in order to balance the load. But due to very complicated routing protocol used by Jarvis, Tony can't predict what route will be used for data transmission. So Tony came to a simple solution — to make changes to the network in such way, that the latency is changed by a given value for *any* possible route (not necessarily simple!) between some pairs of servers.

Genius, billionaire, playboy, philanthropist had defined a list of pairs of servers for which he has to correct the latency. Help Tony to calculate the new network map that satisfied all the constraints. Note that, since Jarvis' network is not a plain network, but a tachyon one, it allows negative latencies.

Input

The first line contains a single positive integer $N (1 \leqslant N \leqslant 150)$ — the number of servers in Jarvis' cluster.

The next N lines consist of N integers $a_{ij} (0 \leqslant a_{ij} \leqslant 1000)$ — initial latencies between servers. Note that a_{ij} doesn't always equal to a_{ji}. It is guaranteed that $a_{ii} = 0$.

The next line contains a single positive integer $M (1 \leqslant M \leqslant 150)$ — the number of pairs of the key servers.

Next M describe each of Tony's requests: change the latency between servers S and T by value D. $(1 \leqslant S, T \leqslant N, -1000 \leqslant W \leqslant 1000)$

All requests should be done at the same time.

It is guaranteed that no pair S, T appears twice in the input data.

Output

Output N lines with N integers b_{ij} — the final latency matrix. Values of $|b_{ij}|$ should not exceed 10^9.

If there is no latency matrix, satisfying all Tony's requests, output "`Impossible`".

Examples

standard input	standard output
2 0 1 1 0 1 1 2 2	0 3 -1 0
2 0 1 1 0 2 1 2 1 2 1 1	Impossible

Note

In the first sample, if Jarvis use $1 \to 2$ route, then we have latency 1 before the correction, and latency 3 after. If it wants to use $1 \to 2 \to 1 \to 2$ route, then we have latency $1 + 1 + 1 = 3$ before the correction, and latency $3 - 1 + 3 = 5$ after — difference is still 2, and so on.

Problem K. KSON

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

S.H.I.E.L.D. agency is a huge organization with offices all over the world. Like any organization of such size it became very bureaucratic. Salaries, Spy's traveling expenses, purchases of uranium, cookies for employees — these are only a few examples of its expenditures.

To get rid of all the paperwork, the decision was made to move on to spreadsheets. These table are generated by Killing Spree Object Notation. KSON-object is a set of key-value pairs, where key is a string and value can also be a string or another KSON-object. Needless to say, that all keys in KSON-object are different.

For technical reasons, all strings in KSON are enclosed in double quotes. String that are keys, do not have any trailing whitespace characters and are limited in length to ten symbols. All key-value pairs are separated by commas. Keys and values are separated by colon. Moreover, to give KSON better appearance and make it more readable, any number of whitespace characters can be inserted between any elements of KSON or between an element and a separator.

KSON object can be viewed as a tree, where internals nodes are non-empty KSON-objects and leaves are string-values or empty objects. Edges in this tree go from internal nodes to the keys of all key-value pairs, stored in corresponding KSON-objects. Thereof, the root of the tree is main KSON-object.

Spreadsheet is generated by the table-template, where each cell represents either a text or a reference to some string stored in KSON. The reference is a string with a prefix "`Value:`", followed by a list of dot separated keys. This list of keys specifies a path from a root KSON-object to some leaf, storing string-value. Inside a cell of table-template, reference can be preceded by any number of whitespace characters. If a reference is valid, it is then substituted by the corresponding value, otherwise text in the cell remains intact.

Generated spreadsheet should be printed in the same format as the table-template. Columns of the resulting table should have appropriate size for their new contents: resulting column width should match the length of the longest string-value in that column.

Now, to check how you understood specifications of S.H.I.E.L.D. spreadsheets, you are asked to generate some spreadsheets for given KSON and table-template.

Input

In the first lines of the input there is KSON-object. Then, starting from a new line, table-template is given.

All strings and all table cells consist only of English letters, digits, spaces, dots and columns. Besides that, all string-keys of KSON-object doesn't contain dots and columns.

Total size of the input not exceeds 100 000 characters.

For a format of table-template refer the sample.

Output

Output spreadsheet, constructed using given KSON-object and table-template.

Example

standard input

```
{
    "Ural" : {
        "State" : " University ",
        "is" : "forever",
        "in" : "our"
    },
    "hearts" : " "
}
+-------------+-------------------------+----------------------+--+
| Static      |Value:Ural.State         |Value:hearts          |  |
+-------------+-------------------------+----------------------+--+
|   text      |Value:type               | Value:Ural.is        |  |
+-------------+-------------------------+----------------------+--+
|Value:Ural.in|                         |Value:Ural.in         |  |
+-------------+-------------------------+----------------------+--+
```

standard output

```
+-------------+-------------------------+-------+--+
| Static      | University              |       |  |
+-------------+-------------------------+-------+--+
|   text      |Value:type               |forever|  |
+-------------+-------------------------+-------+--+
|our          |                         |our    |  |
+-------------+-------------------------+-------+--+
```

Problem L. Loki and Forks

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

Loki received 5 sets of forks as the birthday gift. Each set is the box, containing exactly 5 forks.

Before putting those sets in the shell, Loki wants to knov precise number of different fork sets, to show only one shell of each type and save the space in the shell. Each fork is characterised by one integer k — number of the nibs, i.e. two forks are equivalent, if they have same number of nibs. Two sets a and b are considered to be similar, if for any k number of forks with k nibs is the same for the both sets, otherwise they considered different.

Help Loki to count number of different sets of forks.

Input

Each of 5 lines of the input contains exactly 5 integers — number of nibs; each line represents one fork set. Integers in the input are in range between 1 and 100.

Output

Print one integer — number of different sets of forks.

Examples

standard input	standard output
1 1 7 1 1 1 7 1 7 1 7 1 1 1 7 7 7 7 7 7 7 1 1 1 7	3
1 1 2 2 1 1 2 3 1 1 2 3 1 1 1 1 2 3 1 1 1 1 2 2 1	2

Explanation

In the first sample, fork sets in three groups (1), (2, 3, 5), (4) are identical, in second one we have two groups (1, 5) and (2, 3, 4).