Problem A. Cheaterius's Problem

Input file:	`stdin`
Output file:	`stdout`
Time limit:	2 seconds
Memory limit:	256 megabytes

Cheaterius is a famous in all the Berland astrologist, magician and wizard, and he also is a liar and a cheater. One of his latest inventions is Cheaterius' amulets! They bring luck and wealth, but are rather expensive. Cheaterius makes them himself. The technology of their making is kept secret. But we know that throughout long nights Cheaterius glues together domino pairs with super glue to get squares 2×2 which are the Cheaterius' magic amulets!

That's what one of Cheaterius's amulets looks like

After a hard night Cheaterius made n amulets. Everyone of them represents a square 2×2, every quarter contains 1 to 6 dots. Now he wants sort them into piles, every pile must contain similar amulets. Two amulets are called similar if they can be rotated by 90, 180 or 270 degrees so that the following condition is met: the numbers of dots in the corresponding quarters should be the same. It is forbidden to turn over the amulets.

Write a program that by the given amulets will find the number of piles on Cheaterius' desk.

Input

The first line contains an integer n ($1 \leq n \leq 1000$), where n is the number of amulets. Then the amulet's descriptions are contained. Every description occupies two lines and contains two numbers (from 1 to 6) in each line. Between every pair of amulets the line "`**`" is located.

Output

Print the required number of piles.

Examples

stdin	stdout
4 31 23 ** 31 23 ** 13 32 ** 32 13	1
4 51 26 ** 54 35 ** 25 61 ** 45 53	2

Problem B. bHTML Tables Analisys

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

In this problem is used an extremely simplified version of HTML table markup. Please use the statement as a formal document and read it carefully.

A string is a bHTML table, if it satisfies the grammar:

```
TABLE ::= <table>ROWS</table>
ROWS ::= ROW | ROW ROWS
ROW ::= <tr>CELLS</tr>
CELLS ::= CELL | CELL CELLS
CELL ::= <td></td> | <td>TABLE</td>
```

Blanks in the grammar are only for purposes of illustration, in the given data there will be no spaces. The bHTML table is very similar to a simple regular HTML table in which meet only the following tags : "`table`", "`tr`", "`td`", all the tags are paired and the table contains at least one row and at least one cell in each row. Have a look at the sample tests as examples of tables.

As can be seen, the tables may be nested. You are given a table (which may contain other(s)). You need to write a program that analyzes all the tables and finds the number of cells in each of them. The tables are not required to be rectangular.

Input

For convenience, input data can be separated into non-empty lines in an arbitrary manner. The input data consist of no more than 10 lines. Combine (concatenate) all the input lines into one, to get a text representation s of the specified table. String s corresponds to the given grammar (the root element of grammar is TABLE), its length does not exceed 5000. Only lower case letters are used to write tags. There are no spaces in the given string s.

Output

Print the sizes of all the tables in the non-decreasing order.

Examples

stdin
<table><tr><td></td></tr></table>
stdout
1

stdin
<table> <tr> <td> <table><tr><td></td></tr><tr><td></ td ></tr><tr ><td></td></tr><tr><td></td></tr></table> </td> </tr> </table>
stdout
1 4

stdin
<table><tr><td> <table><tr><td> <table><tr><td> <table><tr><td></td><td></td> </tr><tr><td></td></tr></table> </td></tr></table> </td></tr></table> </td></tr></table>
stdout
1 1 1 3

Problem C. Three Base Stations

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

The New Vasjuki village is stretched along the motorway and that's why every city on it is characterized by its shift relative to some fixed point — the x_i coordinate. The village consists of n houses, the i-th house is located in the point with coordinates of x_i.

TELE3, a cellular communication provider planned to locate three base stations so as to provide every house in the village with cellular communication. The base station having power d located in the point t provides with communication all the houses on the segment $[t - d, t + d]$ (including boundaries).

To simplify the integration (and simply not to mix anything up) all the three stations are planned to possess the equal power of d. Which minimal value of d is enough to provide all the houses in the village with cellular communication.

Input

The first line contains an integer n ($1 \le n \le 2 \cdot 10^5$) which represents the number of houses in the village. The second line contains the coordinates of houses — the sequence $x_1, x_2, \ldots, x_n$ of integer numbers ($1 \le x_i \le 10^9$). It is possible that two or more houses are located on one point. The coordinates are given in a arbitrary order.

Output

Print the required minimal power d. In the second line print three numbers — the possible coordinates of the base stations' location. Print the coordinates with 6 digits after the decimal point. The positions of the stations can be any from 0 to $2 \cdot 10^9$ inclusively. It is accepted for the base stations to have matching coordinates. If there are many solutions, print any of them.

Examples

stdin	stdout
4 1 2 3 4	0.500000 1.500000 2.500000 3.500000
3 10 20 30	0 10.000000 20.000000 30.000000
5 10003 10004 10001 10002 1	0.500000 1.000000 10001.500000 10003.500000

Problem D. Geometrical problem

Input file:	stdin
Output file:	stdout
Time limit:	1 second
Memory limit:	256 megabytes

Polycarp loves geometric progressions — he collects them. However, as such progressions occur very rarely, he also loves the sequences of numbers where it is enough to delete a single element to get a geometric progression.

In this task we shall define geometric progressions as finite sequences of numbers $a_1, a_2, \ldots, a_k$, where $a_i = c \cdot b^{i-1}$ for some real numbers c и b. For example, the sequences [2, -4, 8], [0, 0, 0, 0], [199] are geometric progressions and [0, 1, 2, 3] is not.

Recently Polycarp has found a sequence and he can't classify it. Help him to do it. Determine whether it is a geometric progression. If it is not, check if it can become a geometric progression if an element is deleted from it.

Input

The first line contains an integer n ($1 \le n \le 10^5$) — the number of elements in the given sequence. The second line contains the given sequence. The numbers are space-separated. All the elements of the given sequence are integers and their absolute value does not exceed 10^4.

Output

Print `0`, if the given sequence is a geometric progression. Otherwise, check if it is possible to make the sequence a geometric progression by deleting a single element. If it is possible, print `1`. If the position is not unique, output any of them. If it is impossible, print `-1`.

Examples

stdin	stdout
4 3 6 12 24	0
4 -8 -16 24 -32	1
4 0 1 2 3	2

Problem E. Pentagon

Input file:	stdin
Output file:	stdout
Time limit:	10 seconds
Memory limit:	256 megabytes

According to the last order issued by the president of Berland every city of the country must have its own Ministry Defense building (their own Pentagon). A megapolis Berbourg was not an exception. This city has n junctions, some pairs of which are connected by two-way roads. Overall there are m roads in the city, no more than one between each pair of junctions.

At the moment choosing a location place for Pentagon in Berbourg is being discussed. It has been decided that Pentagon should cover the territory of five different junctions which are joined into a cycle by roads. In the order to build Pentagon a special wall will be built along the roads (with high-tension razor, high-voltage wire and other attributes). Thus, the number of possible ways of building Pentagon in the city is equal to the number of different cycles at lengths of 5, composed of junctions and roads.

Your task is to prints the number of ways of building Pentagon in Berbourg. Only well-optimized solutions will be accepted. Please, test your code on the maximal testcase.

Input

The first line contains two integers n and m ($1 \le n \le 700; 0 \le m \le n \cdot (n-1)/2$), where n represents the number of junctions and m is the number of roads in the city. Then follow m lines containing the road descriptions, one in each line. Every road is set by a number of integers a_i, b_i ($1 \le a_i, b_i \le n; a_i \ne b_i$), where a_i and b_i represent the numbers of junctions, connected by the road. The junctions are numbered from 1 to n. It is not guaranteed that from any junction one can get to any other one moving along the roads.

Output

Print the single number which represents the required number of ways. Please, do not use `%lld` specificator to read or write 64-bit integers in C++. It is preffered to use `cout` (also you may use `%I64d`).

Examples

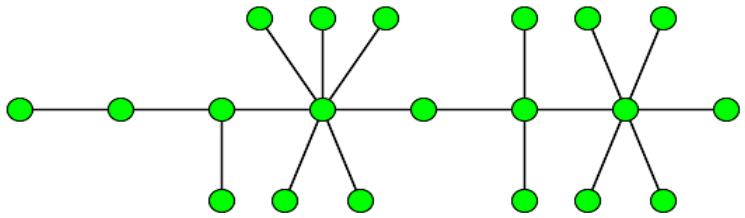
stdin	stdout
5 5 1 2 2 3 3 4 4 5 5 1	1
5 10 1 2 1 3 1 4 1 5 2 3 2 4 2 5 3 4 3 5 4 5	12

Problem F. Caterpillar

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

An undirected graph is called a *caterpillar* if it is a connected graph without cycles and it has such a path p that any vertex is located at a distance of at most 1 from the path p. The caterpillar can contain loops (edges from a vertex to itself) but cannot contain multiple (parallel) edges.

The picture contains an example of a caterpillar:



You are given an undirected graph G. You are allowed to do a *merging operations*, each such operation merges two vertices into one vertex. For that two any vertices a and b ($a \neq b$) are chosen. These verteces are deleted together with their edges (which are incident to at least one of the vertices a or b) but a new vertex w is added together with edges (x, w) for each edge (a, w) and/or (b, w). If there was the edge (a, b) it transforms to the loop (w, w). The resulting graph (after the merging operation) may contain multiple (parallel) edges between pairs of vertices and loops. Let us note that this operation decreases the number of vertices of graph by 1 but leaves the number of edges in the graph unchanged.

The merging operation can be informally described as a unity of two vertices of the graph into one with the natural transformation of the graph edges.

You may apply this operation consecutively and make the given graph to be a caterpillar. Write a program that will print the minimal number of merging operations required to make the given graph a caterpillar.

Input

The first line contains a pair of integers n, m ($1 \leq n \leq 2000; 0 \leq m \leq 10^5$), where n represents the number of vertices in the graph and m is the number of edges in it. Then the following m lines contain edge descriptions, one edge description per line. Every line contains a pair of integers a_i, b_i ($1 \leq a_i, b_i \leq n; a_i \neq b_i$), a_i, b_i which represent the indices of the vertices connected by the edge. The vertices are numbered from 1 to n. In the given graph it will be no more than one edge between any pair of vertices. The given graph is not necessarily connected.

Output

Print the minimal required number of operations.

Examples

stdin	stdout
4 4 1 2 2 3 3 4 4 2	2
6 3 1 2 3 4 5 6	2
7 6 1 2 2 3 1 4 4 5 1 6 6 7	1