Problem A. Natural Division

Time limit:	1 second
Memory limit:	512 megabytes

Vasya really loves natural numbers, especially dividing one natural number by another.

Vasya claims that he took two natural numbers a and b ($0 < a < b < 10^6$), divided a by b, and obtained a non-recurring decimal fraction with n digits after the decimal point ($n \le 17$). Unfortunately, he does not remember the numbers a and b. Check if this could have happened, and if so, provide such numbers a and b.

Input

The first line of input contains an integer n ($1 \le n \le 17$).

The second line of input contains n digits that followed the decimal point in the answer. It is guaranteed that the last digit is not zero.

Output

If there is no solution, output “`NO`” .

If a solution exists, output “`YES`”, followed by two integers a and b that Vasya used. The inequality $0 < a < b < 10^6$ must hold. If there are multiple suitable answers, output any of them.

Examples

standard input	standard output
1 2	YES 1 5
2 69	YES 69 100
3 001	YES 1 1000

Problem B. Area of a Triangle

Time limit:	1 second
Memory limit:	512 megabytes

Cheburashka told Gena that he drew a right-angled triangle and asked for help in finding its area. He provided Gena with the length of the hypotenuse and the height drawn to it.

Help Gena calculate the area of such a triangle, or tell him that Cheburashka made a mistake and that such a triangle does not exist.

Input

The first line contains one integer c ($1 \le c \le 10^4$) — the length of the hypotenuse.

The second line contains one integer h ($1 \le h \le 10^4$) — the length of the height drawn to the hypotenuse.

Output

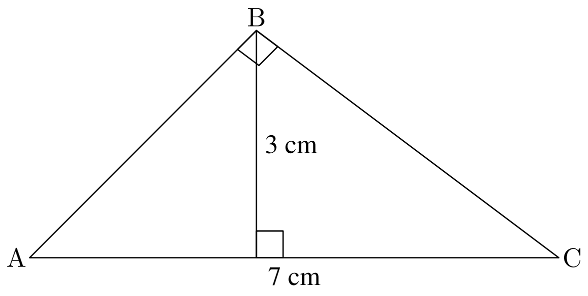
Print the area of the triangle. If such a triangle does not exist, print -1.

Examples

standard input	standard output
7 3	10.5
10 6	-1

Note

An example of the triangle from the first example.



Problem C. Cheburashka's Number

Time limit:	1 second
Memory limit:	512 megabytes

Cheburashka was given a basket containing n oranges. He decided to share them with his friend, the crocodile Gena. To divide the oranges, Cheburashka uses the following algorithm:

- Cheburashka takes one orange for himself and one orange for Gena;
- Cheburashka takes two oranges for himself, and Gena takes one;
- Cheburashka takes three oranges for himself, and Gena takes one;
- And so on..

A number n is called a *Cheburashka's number* if n oranges can be completely divided between the friends using the specified algorithm, and the last orange was taken by Cheburashka for Gena.

Find the k-th smallest Cheburashka's number.

Input

The input consists of a single integer k ($1 \le k \le 10^9$) — the index of the desired Cheburashka's number.

Output

Output the k-th Cheburashka's number.

Examples

standard input	standard output
1	2
2	5
3	9

Problem D. Lazy Cuts

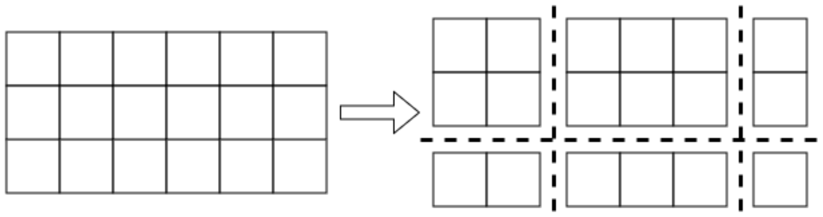
Time limit:	1 second
Memory limit:	512 megabytes

Anton needs to bring s squares to school for his craft lesson. Anton is very lazy, but he loves craft lessons, so he is willing to make the minimum necessary number of cuts.

At home, Anton found a sheet of grid paper sized $n \times m$, with $s \le n \cdot m$. Anton also has a laser cutter that works as follows:

1. The original sheet of paper is placed in the cutter.
2. The cutter makes x straight vertical cuts along the edges of the squares.
3. The cutter makes y straight horizontal cuts along the edges of the squares.
4. The cutter produces $(x + 1) \cdot (y + 1)$ new sheets of paper.

For example, from a 3×6 sheet, three cuts can produce six sheets with areas 4, 2, 6, 2, 1, and 3.



Since Anton is lazy, he wants to make the minimum possible number of cuts so that from the sheets produced by the cutter, he can collect a set with a total area of exactly s squares to bring to school for his craft lesson.

Anton is afraid of overworking while using the cutter, so he asks you to help determine the minimum possible number of cuts he will have to make.

Input

The input consists of a single line containing three integers n, m, and s — the dimensions of the original sheet of grid paper and the required number of squares for the class ($1 \le n, m \le 10^5$, $1 \le s \le n \cdot m$).

Output

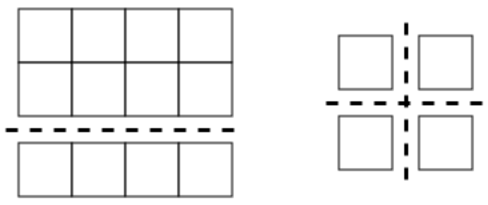
Output a single integer — the minimum possible number of cuts needed so that it is possible to collect a set with a total area of exactly s squares from the resulting sheets of paper.

Examples

standard input	standard output
3 4 8	1
2 2 3	2
10 9 90	0

Note

The image below shows examples of cuts for the first two tests from the example.



In the third test from the example, $s = n \cdot m$, so no cuts are needed.

Problem E. Decart the Grasshopper

Time limit:	1 second
Memory limit:	512 megabytes

A grasshopper named Decart lives on the Cartesian plane. Today, he urgently needs to get from point (x_0, y_0) to point (x_1, y_1).

Unfortunately, the grasshopper doesn't have time to think, so he strictly decided to jump n times, with the length of the i-th jump being exactly a_i, as he just dreamed of such a sequence of numbers.

Help the grasshopper to devise a route from point (x_0, y_0) to point (x_1, y_1) using jumps of lengths a_i, or confirm that this is impossible.

Input

The first line contains two integers x_0 and y_0 — the coordinates of the starting point $(-1000 \le x_0, y_0 \le 1000)$.

The second line contains two integers x_1 and y_1 — the coordinates of the ending point $(-1000 \le x_1, y_1 \le 1000)$. Note that the starting and ending points of the grasshopper's route may coincide.

The third line contains a single integer n — the number of jumps the grasshopper makes $(1 \le n \le 10^5)$.

The fourth line contains n integers $a_1, a_2 \ldots a_n$ — the desired lengths of the jumps $(1 \le a_i \le 1000)$.

Output

If it is impossible to jump from the starting point to the ending point using the given sequence of jumps, output "`Impossible`" on the first line.

Otherwise, output "`Possible`" on the first line, and in the following n lines, output the coordinates of the points. In the i-th line, two real numbers should be output—the coordinates of the point where the grasshopper will land after the i-th jump.

If there are multiple solutions, any of them is allowed. The following quantities must be equal with a relative or absolute error of no more than 10^{-4}:

- The distance between the $(i-1)$-th and i-th points in the answer must be equal to a_i;
- The distance between the starting point and the first point in the answer must be equal to a_1;
- The coordinates of the n-th point in the answer must be equal to the coordinates of the ending point.

Examples

standard input	standard output

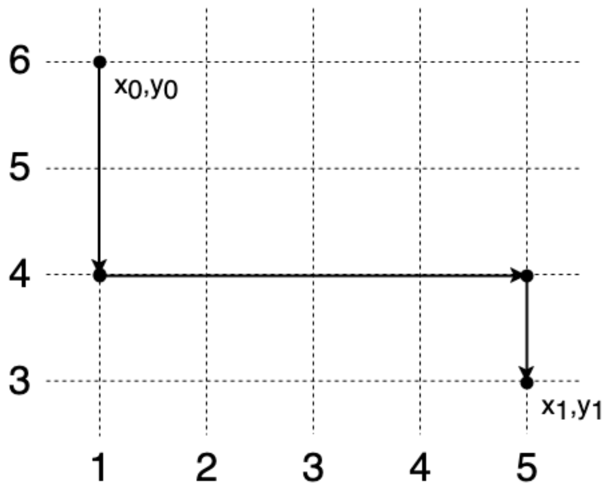
`1 6`
`5 3`
`3`
`2 4 1`
	`Possible`
	`1 4`
	`5 4`
	`5 3`

`0 0`
`6 8`
`5`
`2 3 4 5 6`
	`Possible`
	`-1.20000000 -1.60000000`
	`-3.00000000 -4.00000000`
	`-0.60000000 -0.80000000`
	`2.40000000 3.20000000`
	`6.00000000 8.00000000`

`-10 10`
`10 -10`
`2`
`5 5`
	`Impossible`

Note

One possible route for the grasshopper for the first test from the example is shown in the figure.



Problem F. Delivery Robot

Time limit:	1 second
Memory limit:	512 megabytes

The progressive burger joint "ITBurger" plans to launch the delivery of its burgers using autonomous delivery robots that can independently plan the optimal delivery route.

To train the robot, engineer Albina built a testing ground, which is a coordinate plane. On the nodes of the coordinate plane, there are $n \times n$ light markers in the form of a square, with neighboring markers in columns and rows located at a unit distance from each other. The bottom-left marker is located at the node with coordinates $(0, 0)$, and the top-right marker is at the node with coordinates $(n - 1, n - 1)$.

The robot's goal is to create a route that visits all the light markers. The robot can start moving from any point on the coordinate plane with integer coordinates from -1000 to 1000. It can only move in a straight line, so its trajectory will be a polyline. Albina believes that the robot can be programmed to traverse the light markers along a polyline consisting of exactly $2n - 2$ segments, where the start and end of the route, as well as all turns, will be at points with integer coordinates from -1000 to 1000. Help the robot find such a route.

Input

The first line of input contains a single integer n ($3 \le n \le 100$) — the number of light markers in one row of the square.

Output

Output $2n - 1$ pairs of integers — the robot's route. The first pair of numbers is the coordinates of the starting point of the route, the next $2n - 3$ pairs of numbers are the points where the robot makes turns, and the last pair of numbers is the coordinates of the end of the route.

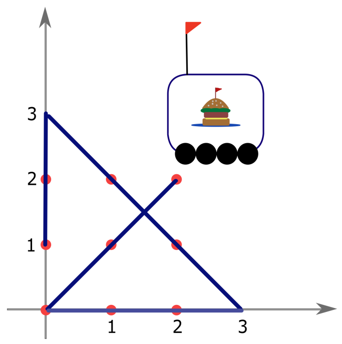
Example

standard input	standard output

3	2 2
	0 0
	3 0
	0 3
	0 1

Note

The robot's route for $n = 3$. The robot starts at point $(2, 2)$ and ends at point $(0, 1)$. The trajectory is shown in the figure.



Problem G. Trip to Moscow

Time limit:	1 second
Memory limit:	512 megabytes

Andrew lives in Saint Petersburg, but he often has to travel to Moscow for work. For traveling between the cities, Andrew prefers to use his personal car and the modern high-speed highway connecting the two cities. Since Andrew has a lot of work, he does not want to spend much time on the road to Moscow. Therefore, he is thinking about how to minimize the travel time.

The distance from Moscow to Saint Petersburg is m units. For convenience, let's represent the highway connecting the cities as a number line. Saint Petersburg is located at point 0, and Moscow is at point m. There are also n gas stations built along the highway, with the i-th station located at a distance of x_i units from Saint Petersburg. Thus, the i-th gas station will be located on the number line at point x_i.

Let t_i denote the amount of time required to refuel at the i-th station. For simplicity, we will assume that the refueling time does not depend on the amount of fuel being filled in the car.

Andrew's personal car has a fuel tank capacity of c liters. Naturally, the fuel consumption while driving on the highway depends on the speed at which the car is moving. Andrew knows that when driving at a speed of v units of distance per unit of time, his car will consume v liters of fuel per unit distance. There are no speed limits on the highway; one can drive at any speed.

Initially, Andrew is in Saint Petersburg, and his car's tank is full (that is, it contains c liters of fuel). Knowing the information about all the gas stations on the highway, help Andrew calculate the minimum amount of time required to reach Moscow. Note that Andrew does not care how many liters of fuel remain in his car when he arrives in Moscow.

Input

The first line contains three integers n, m, and c ($1 \le n \le 250\,000$, $n+1 \le m \le 10^9$, $1 \le c \le 10^9$) — the number of gas stations, the distance from Saint Petersburg to Moscow, and the capacity of Andrew's car's fuel tank.

Each of the following n lines contains two integers x_i and t_i ($0 < x_i < m$, $1 \le t_i \le 1\,000$) — the distance from Saint Petersburg to the i-th gas station, and the time required to refuel the car at that station.

It is guaranteed that $0 < x_1 < x_2 < \ldots < x_n < m$.

Output

Output a single real number — the minimum amount of time required to reach Moscow.

Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Formally, let your answer be a, and the jury's answer be b. Your answer will be accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-6}$.

Example

standard input	standard output

4 100 25
10 50
15 30
50 100
80 60
	284.0000000000

Note

In the first example, Andrew's optimal route is arranged as follows:

- Drive from Saint Petersburg to the second gas station, covering 15 units of distance. This segment can be traveled at a speed of $\frac{25}{15}$ to consume all the fuel. Thus, this leg will take 9 units of time;
- Refuel at the second station, taking 30 units of time;
- Drive from the second gas station to the fourth gas station, covering 65 units of distance. This segment can be traveled at a speed of $\frac{25}{65}$ to consume all the fuel. Thus, this leg will take 169 units of time;
- Refuel at the fourth station, taking 60 units of time;
- Drive from the fourth gas station to Moscow, covering 20 units of distance. This segment can be traveled at a speed of $\frac{25}{20}$ to consume all the fuel. Thus, this leg will take 16 units of time.

Problem H. Colorful Graph

Time limit:	1 second
Memory limit:	512 megabytes

Recently, Matvey studied graphs in his computer science classes. A graph consists of n vertices, some of which may be connected by edges. Matvey is an artist, and he loves to draw. When he draws a graph, he represents its edges in different colors. We will denote colors by integers from 1 to 10^5.

Matvey came up with the following task for himself. Initially, the graph has n vertices and no edges, and then q queries are executed.

- $+\ v\ u\ c$ — add an edge (v, u) of color c to the graph. It is guaranteed that there was no edge of color c between vertices v and u.
- $-\ v\ u\ c$ — remove the edge (v, u) of color c from the graph. It is guaranteed that there was an edge of that color between vertices u and v.

Since Matvey is an artist, he considers a color c in the graph to be *excellent* if no more than one edge of that color is connected to each vertex. We define the *beauty* of color c as the number of edges of that color.

After each query, he became interested in what the total beauty of the graph is for all excellent colors.

Input

The first line contains two integers n and q — the number of vertices in the graph and the number of queries, respectively ($2 \le n \le 10^5$, $1 \le q \le 10^5$).

In the following q lines, the descriptions of the queries are given, one per line ($1 \le v, u \le n$; $v \ne u$; $1 \le c \le 10^5$).

Output

After each query, you need to output the total beauty of the graph for all excellent colors.

Examples

standard input	standard output
4 5 + 1 2 1 + 3 4 2 + 2 4 1 - 2 1 1 + 2 4 4	1 2 1 2 3
3 6 + 1 2 1 + 2 3 1 + 1 3 1 - 1 3 1 - 1 2 1 + 1 2 1	1 0 0 0 1 0

Note

In the first example, after the first query, all colors are excellent.

After the second query, all colors are excellent.

After the third query, color 1 is not excellent, as there are two edges of color 1 emanating from vertex 2. Color 2 is excellent, and there is one edge of that color in the graph.

After the fourth query, all colors are excellent.

After the fifth query, all colors are excellent.

Problem I. “Galactic Timeline“

Time limit:	1 second
Memory limit:	512 megabytes

In a Distant Galaxy in the Far Future, old board games are still popular. The humanoid Hyomul is playing “Galactic Timeline“.

As is known, in the game “Timeline“, there are cards that indicate a year and an event that occurred in that year. The cards must be laid out in sequence so that the order of events is correct: each subsequent event cannot be earlier than the previous one.

In the Distant Galaxy, they save paper, so in “Galactic Timeline“, the cards for the game are double-sided — each card has two years and events written on it — one on each side of the cards.

Hyomul has n cards that he lays out on a glass table. Hyomul noticed that when laying out double-sided cards on the glass table, the timeline is formed from both sides: from the top of the table with the face sides of the cards and from the bottom of the table with the reverse sides.

Initially, all the cards are laid out in arbitrary order from left to right. Hyomul can choose any card on the table and flip it. Hyomul can also choose any two cards on the table and swap them without flipping either of them.

Hyomul became curious whether it is possible to arrange the cards such that the events go in the correct order from left to right on both the top and bottom sides of the table. Help him do this or determine that it is impossible to arrange the cards in the correct order on both sides.

Input

The first line contains the number n ($1 \le n \le 10^5$). The second line contains n numbers a_i ($1 \le a_i \le 10^9$) — the years written on the face side of the i-th card. The third line contains n numbers b_i ($1 \le b_i \le 10^9$) — the years written on the reverse side of the i-th card.

Output

Output the information about the cards in the order in which they need to be laid out.

In the first line, output n years from the face side of the cards, and in the second line—n years from the reverse side of the cards. If it is impossible to achieve the goal of the game, output -1.

Examples

standard input	standard output

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

3
30 10 20
10 30 20
	-1

Note

In the first example, Hyomul can act, for example, as follows. First, he flips the first card, then he flips the second card. The cards laid out from left to right (with the face and reverse sides indicated by |) are: 2|6, 4|8, 1|5, and 3|7. Then Hyomul will swap the first and third cards, the second and third cards, and the third and fourth cards to achieve the correct order on both sides.

Problem J. Chess Bishop

Time limit:	1 second
Memory limit:	512 megabytes

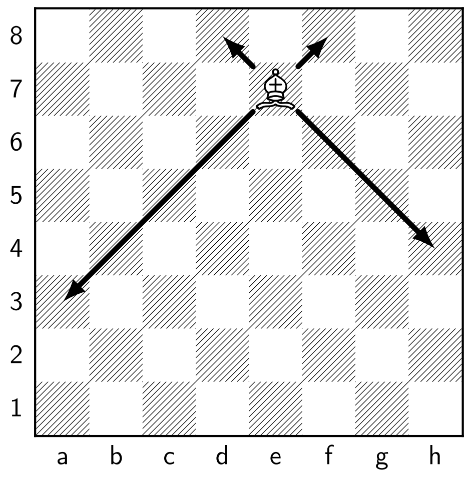
This is an interactive problem.

The 8×8 chessboard is numbered in the standard way: the ranks are numbered from 1 to 8, and the files are labeled with lowercase Latin letters from 'a' to 'h', such that the bottom left corner is `a1` and the top right corner is `h8`.

On the chessboard, there is a chess bishop whose initial position is unknown. The bishop can move any number of squares diagonally while remaining within the chessboard. A bishop's move is specified as follows: first, the horizontal offset x is given (positive values indicate right), followed by the vertical offset y (positive values indicate up).

If the move is successful, the bishop moves to the new square, and the jury program outputs a string consisting of a single character '+'. If the destination square is outside the board, the bishop remains in place, and the jury program outputs a string consisting of a single character '-'.

Your task is to determine the initial position of the bishop by making no more than 7 attempts to move (successful or unsuccessful).



In the illustration, the chessboard is shown with chess notation. The bishop is located on square `e7`. The directions of possible moves are indicated by arrows.

Interaction Protocol

The interaction begins with the participant's program outputting a query. Each query has the form «`?` x y», where x and y are integers and $|x| = |y|$; $1 \le |x|, |y| \le 7$. In response to the query, the jury program outputs a string consisting of the character '+' if the move was successful, and '-' if the move was unsuccessful.

When the participant's program is ready to output an answer, it should output «`!` *pos*», where *pos* is the initial position of the bishop given in standard chess notation. After that, the program should terminate.

Example

standard input	standard output
? 2 2 ? 1 -1 ? 2 2 ? -1 1 ? 1 1 ? 1 -1 ? -1 -1 ! e7	 - + + - - - +

Note

In the example input/output, the queries from the participant's program and the responses from the jury program are formatted with empty lines to show which response corresponds to which query. In actual interaction between the participant's program and the jury, there will be no empty lines.

Problem K. Magic Spell

Time limit:	1 second
Memory limit:	512 megabytes

The great wizard plans to compose a magic spell. The spell consists of runes, which we will denote by integers. The spell that the wizard plans to create is a permutation p consisting of n distinct integers from 1 to n.

To compose the spell, the wizard can use a special magical reference book containing m pages, numbered from 1 to m, each of which contains one rune, an integer from 1 to n. Unfortunately, the wizard is already old and cannot rewrite the runes from the reference book himself; fortunately, he has k students whom he can summon in some order and ask to transcribe fragments of the reference book onto one common long scroll. If the i-th student is summoned, he will sequentially transcribe the runes from the reference book from the l_i-th to the r_i-th page, inclusive. Students can be summoned in any order, but each student can be summoned no more than once.

For example, suppose the reference book has $m = 6$ runes written in the following order: $[2, 1, 3, 1, 2, 4]$, and the wizard has $k = 3$ students with $l_1 = 5$, $r_1 = 6$; $l_2 = 2$, $r_2 = 6$; and $l_3 = 1$, $r_3 = 3$.

If the third student is summoned first and then the first student, they will write the runes onto the scroll in the following order: $[2, 1, 3, 2, 4]$.

However, if the second student is summoned first, then the first, and then the third student, the resulting sequence of runes on the scroll will be $[1, 3, 1, 2, 4, 2, 4, 2, 1, 3]$.

After the runes have been transcribed onto the scroll, the wizard can cut out any continuous segment of the scroll. He wants to summon the minimum number of students so that he can obtain the desired magic spell by cutting out a continuous segment from the resulting scroll.

In the example above, if the spell is the permutation $[1, 3, 2, 4]$, it can be obtained by summoning the third student first and then the first student, and cutting out from the resulting scroll $[2, 1, 3, 2, 4]$ the runes from the 2nd to the 5th.

Help the wizard determine the minimum number of students he must summon and in what order they should be summoned to create the desired spell, if possible.

Input

Each test consists of several sets of input data. The first line contains one integer t ($1 \le t \le 10^5$) — the number of sets of input data. The following describes the sets of input data.

The first line of each set of input data contains three integers n, m, and k ($1 \le n \le 3 \cdot 10^5$, $1 \le m \le 3 \cdot 10^5$, $1 \le k \le 3 \cdot 10^5$) — the length of the permutation p, the number of pages in the book, and the number of students.

The next line contains n distinct integers p_i ($1 \le p_i \le n$) — the permutation p. It is guaranteed that all p_i are pairwise distinct.

The next line contains m integers a_i ($1 \le a_i \le n$) — the runes contained in the book.

The following k lines each contain two integers; the i-th of them contains the numbers l_i and r_i ($1 \le l_i \le r_i \le m$), defining the segment of pages that the i-th student transcribes.

It is guaranteed that the sums of n, m, and k across all sets of input data do not exceed $3 \cdot 10^5$ respectively.

Output

For each set of input data, if it is impossible to summon students in some order such that the permutation p is a subsegment of the sequence of runes written on the scroll, output -1.

Otherwise, output the number c — the minimum number of summoned students.

In the next line, output c numbers — the indices of the students in the order they should be summoned.

The students are numbered with integers from 1 to k in the order they are listed in the input. All student indices must be distinct.

If there are multiple optimal answers, output any of them.

Example

standard input	standard output

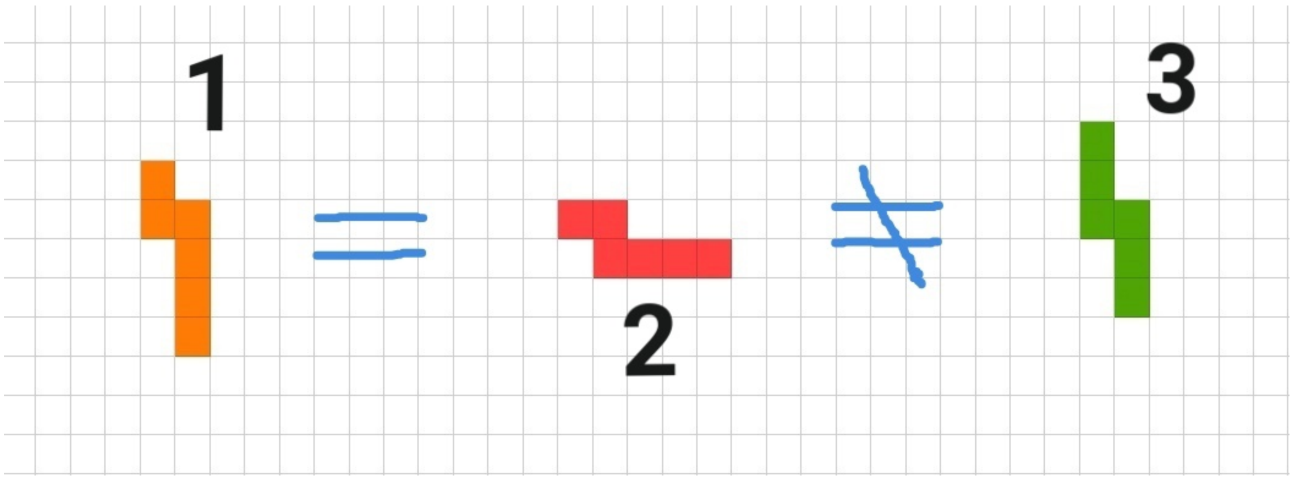
2
4 6 3
1 3 2 4
2 1 3 1 2 4
5 6
2 6
1 3
3 4 1
3 1 2
2 1 3 1
1 4
	2
	3 1
	-1

Problem L. Multimino

Time limit:	1 second
Memory limit:	512 megabytes

Let's define an n-mino as a connected figure composed of n squares 1×1. Let's say that two n-minoes are equal if one of them can be translated into another using the operations of parallel transport, rotation and reflection.

For example, in the illustration below, the red hexamino (number 2) is equal to the orange hexamino (number 1), and they both aren't equal to the green hexamino (number 3).



You are given a number n. Cut a checkered board of size $n \times n$ squares into n pairwise distinct non-intersecting n-minoes or report that no such splitting exists.

Input

The only line of the input contains a single integer n ($1 \leqslant n \leqslant 100$).

Output

Output a single integer -1 if the required splitting does not exist.

Otherwise, let's enumerate n-minoes with natural numbers from 1 to n. Output n rows with n integers in each row corresponding to the splitting of the board into n-minoes according to the principle that a cell in the i-th column and j-th row belongs to n-mino number a_{ij} ($1 \leqslant a_{ij} \leqslant n$).

All n-minoes in the splitting must be connected and pairwise distinct from each other. If there are several correct splittings for a board, output any one of them.

Examples

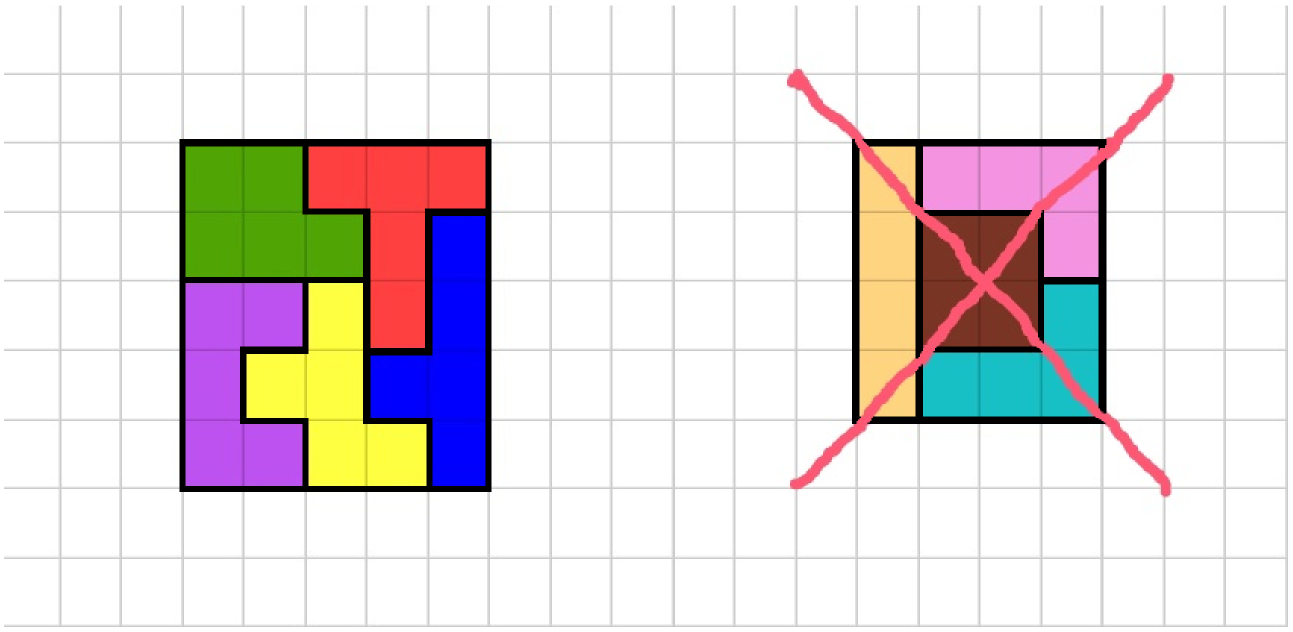
standard input	standard output

5	4 4 2 2 2
	4 4 4 2 3
	5 5 1 2 3
	5 1 1 3 3
	5 5 1 1 3

4	-1

Note

In the first example, the splitting looks like the picture on the left.



In the second example, the picture on the right would not be a valid splitting because the lilac (top) and turquoise (bottom) n-minoes can be translated into each other by parallel transport, rotation, and reflection. Therefore, they are equal.

Problem M. Three Points

Time limit:	1 second
Memory limit:	512 megabytes

Cheburashka, Crocodile Gena, and Old Lady Shapoklyak decided to move to the desert. They found three identical houses located at coordinates x_1, y_1, x_2, y_2, x_3, and y_3 respectively. The rat Lariska was assigned the task of calculating the coordinates and digging a straight trench of infinite length for the water supply in such a way as to minimize the total distance from the houses to this trench.

Help her determine the coordinates of two points that lie on this line. The trench may pass under one or more houses.

Input

A single line contains six integers x_1, y_1, x_2, y_2, x_3, and y_3 — the coordinates of the three houses, respectively ($-10^9 \le x_1, y_1, x_2, y_2, x_3, y_3 \le 10^9$). It is guaranteed that the houses are located at different points.

Output

Output the integer coordinates of two points through which the trench will pass. The points must not coincide. The coordinates must not exceed 10^9 in absolute value.

Example

standard input	standard output
`0 0 10 10 5 6`	`1 1 3 3`