Problem A. Meeting Near the Fountain

Time limit: 1 second
Memory limit: 512 megabytes

Kira plans to meet with her friend Anna after T minutes. The meeting point is in the center of the park near the fountain. The distance between the meeting point from Kira's current location is n meters.

The path to the meeting point first goes through the city streets, and then through the park. Kira realized that she could be late for the meeting by getting there on foot, so she took an electric scooter.

Kira drove on an electric scooter through the city streets to the park entrance m meters with a speed of x m/s. Park is a pedestrian zone, where electric scooter speed is limited. Kira noticed that the electric scooter's speed reduced to y m/s immediately after she entered the park.

Determine whether Kira will be at the meeting point in time, and, if not, how many minutes Anna will be waiting for Kira.

Input

The first line of the input contains two integers n ($500 \le n \le 10000$) and T ($1 \le T \le 60$) — the distance in meters between Kira's start and the meeting point and the number of minutes until the meeting with Anna is planned.

The second line of the input contains a single integer m ($100 \le m \le 9000$, $m \le n$) — the distance in meters between Kira's start and the park entrance.

The third line of the input contains two integers x and y — ($3 \le y < x \le 9$) — the electric scooter's speed before the park entrance and in the park, respectively.

Output

If Kira gets to the meeting point on time, output a single integer — 0.

If Kira is going to be late, output a single integer L — how many minutes Anna will be waiting for Kira. This value should be rounded up.

Example

standard input	standard output
5000 10 2500 8 4	6

Problem B. GPS Hack

Time limit: 1 second
Memory limit: 512 megabytes

Denis is a student and he's failed his data security exam, so today he's having a retake. He woke up late and decided to take a taxi.

Denis lives in a city, which can be represented as n crossroads connected by m bidirectional roads. For every road it is known, how much it takes to drive along it. Denis lives near a crossroad with number s and the university is near crossroad t. As soon as the taxi is near the university, Denis immediately leaves it and rushes for his exam.

Unfortunately, everything is a bit more complicated — hackers had hacked a GPS in Denis's taxi. When the GPS works properly, it helps you to reach your destination as fast, as it's possible. On every crossroad it shows, which road should be taken in order to finish your journey in the fastest way. If there are more than one possible road, GPS can show you any. Hackers can make GPS show you any road they want, but no more than once.

Denis doesn't remember exactly, how did he get to the university, but he remembers the time L the road took. It turned out that his teacher had already known about the attack on GPS, so he asked Denis to calculate the number of different possible routes from his place to the university as a part of his retake.

Help Denis calculate the number of routes from crossroad s to crossroad t, that took L time and were possible considering the GPS hack. As this number might be very large, find it modulo $10^9 + 7$.

Input

The first line contains two integers n and m — number of crossroads and roads between them in the city, where Denis lives ($2 \le n \le 2 \cdot 10^5$, $1 \le m \le \min(2 \cdot 10^5, \frac{n(n-1)}{2})$).

The second line contains three integers s, t and L — crossroad number, where Denis lives, number of the crossroad near the university and the ride time ($1 \le s, t \le n$, $s \ne t$, $1 \le L \le 2 \cdot 10^{14}$).

The next m lines contain descriptions of roads between crossroads. The description contains three integers v, u, w ($1 \le v, u \le n$, $v \ne u$, $1 \le w \le 10^9$) — numbers of the two crossroads connected by this road and the time, needed to drive along it.

All the roads are bidirectional, any pair of crossroads is connected with no more than one road. No roads connect a crossroad with itself. It is guaranteed that it is possible to get from crossroad s to the crossroad t.

Output

Print the only integer — number of routes Denis could have modulo $10^9 + 7$.

Example

standard input	standard output
6 7 1 6 12 1 2 2 1 3 4 2 5 2 3 5 4 2 4 4 4 6 6 5 6 4	4

Problem C. Tournament

Time limit: 1 second
Memory limit: 512 megabytes

The linguistic game “Hat” is played by several pairs of players. Also there should be one host of the game.

A teacher plans to organize a “Hat” tournament in their class consisting of n students, where n is odd number. In order to do this he wants to split students into pairs and leave one student to be the host.

Number students from 1 to n. Student number i is known to have a skill value of a_i in the “Hat” game. Skill of a pair of students is defined as the sum of their individual skills.

In order for the tournament to be as fair as possible the teacher wants the difference between the maximum and minimum skills of resulting pairs to be as small as possible. Help the teacher to choose the host and split other $n-1$ students into pairs in order to achieve the desired goal.

Input

The first line of input contains an integer n — the number of students in the class ($3 \le n \le 5 \cdot 10^5$, n is guaranteed to be odd).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

Output

Output one number — the smallest possible difference between maximum and minimum skills of pairs participating in the tournament.

Example

standard input	standard output
5 1 2 3 5 9	1

Problem D. Gas Stations

Time limit: 1 second
Memory limit: 512 megabytes

Michelle and Marie went to a journey by their car. During the journey they are going to drive along the long straight road. There are n gas stations on the road. Gas stations are numbered from 1 to n along the road. Initially the car is located in the city at the beginning of the road, at the same place there is a gas station number 1.

Gas station number i is located at the distance of x_i kilometers from the beginning of the road and sells gas for the price of p_i dollars per liter. It is only allowed to purchase an integer number of liters of gas at each station. One liter of gas is enough to drive 1 unit of distance along the road. The gas tank of the car has the capacity of C liters. Initially the tank is empty.

Michelle and Marie have B dollars in total to be spent on gas. How far along the road are they able to go if they don't bother about the way back yet?

Input

The first line of input contains three integers n, B and C — the number of gas stations, total budget in dollars and the capacity of the gas tank in liters correspondingly ($1 \le n \le 10^5$; $1 \le B \le 10^9$; $1 \le C \le 10^9$).

Each of the next n lines contains two integers x_i and p_i — the coordinate of i-th gas station and the gas price there correspondingly ($0 = x_1 < x_2 < \ldots < x_n \le 10^9$; $1 \le p_i \le 10^9$).

Output

Output one integer — maximal possible distance that may be driven along the road.

Example

standard input	standard output
3 10 5 0 3 1 1 4 2	7

Note

In the sample test it is possible to buy 1 liter of gas at the first station, drive 1 unit of distance, buy 5 liters of gas at the second station, drive another 3 units of distance to get to the third station. At this moment there are 2 liters of gas in the tank, the friends can buy another 1 liter of gas and drive another 3 units of distance, finishing their trip at a distance of 7 from the beginning.

Problem E. kex

Time limit: 2 seconds
Memory limit: 512 megabytes

Consider the set of non-negative integers A. The minimum non-negative integer that does not occur in this set is considered, for example, in game theory, and is denoted as $\text{mex}(A)$. For example, $\text{mex}(\{0, 1, 2, 4, 5, 9\}) = 3$.

Ann has decided to generalize the concept of mex. Consider a positive integer k and a set of non-negative integer A. Denote as $\text{kex}(A, k)$ a non-negative integer that is k-th in ascending order among all integers that are not in A. For example, $\text{kex}(\{0, 1, 2, 4, 5, 9\}, 2) = 6$.

You must find $\text{kex}(A, k_i)$ for the given set of integers A and q values of k_i.

Input

The first line of input contains two integers n and q ($1 \le n, q \le 10^5$) — number of elements in A and number of kex numbers, that you have to find.

In second line of input contains n different not negative integers, each of which is at most 10^9, — elements of A.

In third line of input contains q integers k_i ($1 \le k_i \le 10^9$).

Output

Print values: $\text{kex}(A, k_1), \text{kex}(A, k_2), \ldots, \text{kex}(A, k_q)$.

Example

standard input	standard output
4 10 1 2 6 7 1 2 3 4 5 6 7 8 10 11	0 3 4 5 8 9 10 11 13 14

Problem F. Bob's Average

Time limit:	3 seconds
Memory limit:	512 megabytes

In a course in mathematical statistics, Bob came up with a new way to calculate the average for arrays containing an odd number of elements.

As long as the length of the array is greater than one, the following operation is performed: an arbitrary segment of the array of length 3 with boundaries l and $r = l + 2$ is selected, the median of these three elements is calculated and then these elements are replaced by one element equal to the median.

The median of an array of three elements is the second-highest element of this array. For example, the median of the array $[1, 5, 4]$ is 4, and the median the array $[2, 2, 2]$ is 2.

Let's consider one of the ways to calculate the Bob's average on an array $[4, 1, 3, 2, 5]$. In the first step, we select a sub-section with boundaries $l = 2$, $r = 4$. Since 2 is the median of the subarray $[1, 3, 2]$, then the array is converted as follows: $[4, \mathbf{1},\ \mathbf{3},\ \mathbf{2}, 5] \to [4, \mathbf{2}, 5]$. In the second step, the only possible segment of length 3 has boundaries $l = 1$, $r = 3$. Since 4 is the median of the subarray $[4, 2, 5]$, then $[\mathbf{4},\ \mathbf{2},\ \mathbf{5}] \to [4]$.

Bob noticed that his method of calculating the average is not quite correctly defined: depending on the choice of segments, the result of the calculation may be different. To fix this situation, Bob decided to choose segments to replace with the median in such a way that the only number remaining at the end was the maximum possible. It is this number that Bob calls *Bob's average* for this array.

For an array a of n elements, you need to answer q queries, The j-th of those queries is characterized by the boundaries of L_j and R_j.

The answer to the i-th query is the Bob's average of the segment of the original array $[a_{L_j}, a_{L_j+1}, \ldots a_{R_j}]$. For all queries, it is guaranteed that the length of the query segment is odd.

Input

The first line contains an integer n ($3 \le n \le 5 \cdot 10^4$) — array size.

The second line contains n integers a_i ($0 \le a_i \le 10^9$) — array elements.

The third line contains an integer q ($1 \le q \le 10^5$) — number of requests.

The following q lines contain integers L_j, R_j ($1 \le L_j \le R_j \le n$) — the boundaries of the sub-section of the j-th query, it is guaranteed that $R_j - L_j + 1$ is not divisible by 2.

Output

In the j-th line of the output, output a single integer — the answer to the j-th request.

Example

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

Problem G. Loop around Lake

Time limit: 2 seconds
Memory limit: 512 megabytes

Lesya is working as a landscape designer. Her task is to create a road around the lake in the park using as few of construction materials as possible. She can only create the road on the lawn cells.

THe park is a grid of $h \times w$ cells, each cell can be either ground, or lawn, or lake. You are given the park map where lawn cells are shown. It is guaranteed that:

1) the set of lawn cells is 4-connected, that is, it is possible to get from each cell to any other cell by walking between cells that have a common side;
2) the set of lawn cells has only one internal area which is 4-connected — the lake;
3) it is possible to start from some lawn cell, walk around the lake, and get back to the same cell, and the lake will be inside the walk loop. Let us call such set of cells *4-looping* the lake.

Lesya must design the road by replacing some lawn cells with road cells, so that the road was 4-looping the lake, and the number of cells in the road was as small as possible.

Help her to come up with the optimal road plan.

Input

The first line contains two integers h and w ($3 \le h, w \le 1000$).

Each of the next h lines contains w characters "." and "#" denoting land/lake and lawn, respectively.

It is guaranteed that the lawn in 4-connected and contains exactly one inner 4-connected area.

Output

Output h lines with w characters each — the optimal plan of the road. The character '#' denotes the road cell, the character '.' denotes any other cell. Note that the road can only be built on lawn cells.

If there are several optimal plans to build the road, you can output any of them.

Example

standard input	standard output
4 6 ###### #.#.## #...#. #####.	#####. #...#. #...#. #####.

Problem H. One-dimensional Game

Time limit: 2 seconds
Memory limit: 512 megabytes

Bogdan is playing a game. The game is one-dimensional — there are n platforms located on the horizontal line. Bogdan can move between platforms. The i-th platform is a horizontal segment $[l_i, r_i]$. All segments are distinct.

Let's say that the platform j is *inside* the platform i if $l_i \le l_j$ and $r_j \le r_i$.

At the beginning of the game, Bogdan chooses the platform from which he will start his journey. Moving between platforms is dangerous and Bogdan tries to stay away from the danger, so he can move from the platform i to the platform j only if the platform j is inside the platform i and there is no other platform k, which is inside the platform i and the platform j is inside the platform k.

For each platform Bogdan wants to know how many different paths he can take, starting from this platform and ending on any other. To paths are different if there is a platform, which is in the one of the paths but not in the other one.

Help Bogdan count the number of paths from each platform. Since the number of paths can be very large, find it modulo $10^9 + 7$.

Input

The first line contains the only integer n ($1 \le n \le 3 \cdot 10^5$) — the number of platforms.

The next n lines contains the platforms' data, the i-th of these lines contains two integers l_i and r_i ($1 \le l_i \le r_i \le 10^9$) — the ends of the i-th segment of the platform.

Output

Print n integers, the i-th integer must be equal to the number of different paths from the platform i, modulo $10^9 + 7$.

Examples

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

Problem I. Soviet Kindergarden

Time limit: 1 second
Memory limit: 512 megabytes

Mikhail Abramovich is a member of the CPSU since 1961, a doctor of sciences, a scientific atheist, loving dad and husband. He has a 10-year grandson Maxim. Maxim plays on his phone all day long instead of making science like his grandfather.

Maxim downloaded a new game "Snake 2022". The playing area of the "Snake 2022" is a rectangular table $n \times m$. Rows are enumerated from 1 to n, columns are enumerated from 1 to m. Cell (r, c) is in the intersection of a row r and a column c.

There is an apple in each cell of the table. When the head of the snake gets to the cell, the snake immediately eats an apple from this cell. The player gets w_{ij} points when the snake eats an apple from the cell (i, j).

The snake has a length of 1 at the beginning of the game. The snake's head starts at cell (a_r, a_c). The snake immediately eats an apple from a cell (a_r, a_c). The game ends when the snake's head gets to the cell (b_r, b_c).

The move in the game is moving the snake's head to any of the neighboring cells, in which there is no snake yet. On each move the snake eats an apple and increases its length by 1. The set of cells occupied by the snake remains the same, plus the cell in which the snake's head appears in the current move. The move of the snake is described by one symbol: "U" to move up, from (r, c) to $(r-1, c)$; "D" to move down, from (r, c) to $(r+1, c)$; "L" to move left, from (r, c) to $(r, c-1)$; "R" to move right, from (r, c) to $(r, c+1)$.

Let W be the total cost of apples on the whole table. The player wins if he gets strictly more than $\frac{1}{2}W$ points. To simplify the game, it is guaranteed that any apple brings strictly fewer points than the total cost of apples in cells (a_r, a_c) and (b_r, b_c).

"Snake 2022" is too difficult for Maxim, he can't win. He asked his grandfather for help. Mikhail Abramovich told Maxim the story of how in his youth the same problem was solved by an ordinary Soviet kindergartner.

You play the role of this kindergartner. Your task is to present a winning strategy for each configuration of the playing field from the tests.

Input

The first line of the input contains a single integer t — a number of the tests.

Each test is described in the following format. The first line contains six integers n, m, a_r, a_c, b_r, b_c — the size of the table, coordinates of the start and finish cell ($2 \le n, m \le 5000$, $1 \le a_r, b_r \le n$, $1 \le a_c, b_c \le m$, the start and the finish cells are different). The sum $n \cdot m$ for all tests in one set of input data does not exceed 10^6.

The next n lines contain the costs of the apples in the table cells. The line i contains integers $w_{i1}, w_{i2}, \ldots, w_{im}$ ($1 \le w_{ij} \le 10^9$. It is guaranteed that for any $1 \le i \le n$ and $1 \le j \le m$ inequality $w_{ij} < w_{a_r a_c} + w_{b_r b_c}$ is satisfied).

Output

Output a line containing symbols "U", "D", "L", "R" for each test case — the sequence of the snake's moves, in which its head starts in the cell (a_r, a_c), ends in the cell (b_r, b_c), does not visit cells already occupied by the snake, and gets more than $\frac{1}{2}W$ points.

Example

standard input	standard output
2 2 2 1 1 2 2 1 9 1 9 2 4 1 2 2 3 2 1 5 6 3 4 8 7	RD RRDL

Problem J. Pyramid Construction

Time limit: 2 seconds
Memory limit: 512 megabytes

As you might know, the Egyptian pyramids are one of the most famous wonders of the world. They were built by aliens in ancient times. The alien Ar'glh flies past Earth to work and back every day and he often makes a stop to admire these magnificent structures and find inspiration.

You see, Ar'glh is an architect and is designing his own pyramid. Fortunately, with the new technology, he doesn't need huge blocks of stone or the countless Egyptian workers' labor. But that still doesn't mean that his task is easy.

Over the past centuries, Ar'glh has assembled n carbon triangular facets, the i-th of them has sides of length a_i, b_i, and c_i, respectively. He wants to choose four facets and assemble a pyramid in the shape of a non-degenerate tetrahedron with them. Of course, the facets must fit together perfectly without any deformations or gaps and must not protrude beyond the tetrahedron. The volume of the tetrahedron must be strictly positive.

Help the alien architect determine which four facets he can choose from his set to assemble a pyramid in the shape of a tetrahedron.

Input

The first line containts an integer n — the number of triangular facets available to Ar'glh ($4 \le n \le 1500$).

The i-th of the following n lines contains three integers a_i, b_i, and c_i — the lengths of the sides of the i-th facet ($1 \le a_i, b_i, c_i \le 10\,000$, it is guaranteed that each facet is a valid triangle, i.e. $a_i + b_i > c_i$, $a_i + c_i > b_i$, $b_i + c_i > a_i$).

Output

If Ar'glh can choose four facets from which a tetrahedron can be assembled, the first line must contain the word "`Yes`". In this case the second line must contain four different integers from 1 to n — the numbers of facets that can be used to build. If there are several possible answers, print any of them.

Otherwise, print the word "`No`".

Examples

standard input	standard output
7 3 3 3 3 4 5 4 5 6 3 4 3 3 3 5 6 5 7 3 5 5	Yes 2 4 5 7
5 2 3 4 3 4 5 4 5 6 5 6 7 6 7 8	No
4 2 3 4 3 4 2 3 2 4 4 2 3	No

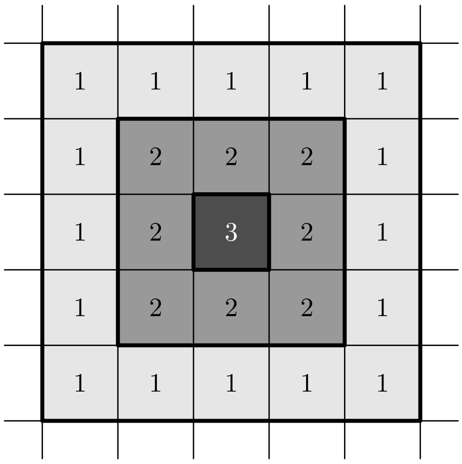
Problem K. The Fortress Defense

Time limit: 2 seconds
Memory limit: 512 megabytes

The Flatlanders are building a fortress on a grid of square cells. The fortress will have several contours of defense. Each contour is a rectangle with sides along the borders of the cells.

The first, outer, defense contour is a rectangle of size $h \times w$. Each next contour is strictly inside all previous ones. Defense contours should not have common points.

The level of cell defense is the number of defense contours inside which this cell lies. The defense level of a fortress is equal to the sum of the defense levels of all its cells. For example, the fortress defense level in the picture is $16 \cdot 1 + 8 \cdot 2 + 3 = 35$.



The Flatlanders are interested in all possible ways to build a fortress. They calculate the defense level of the fortress for each possible way, then calculate the sum of values of all ways. You should help the Flatlanders to calculate this sum. The answer can be large, output its modulo $10^9 + 7$.

Input

In the first line of input there are two integers h and w ($1 \le h, w \le 400$).

Output

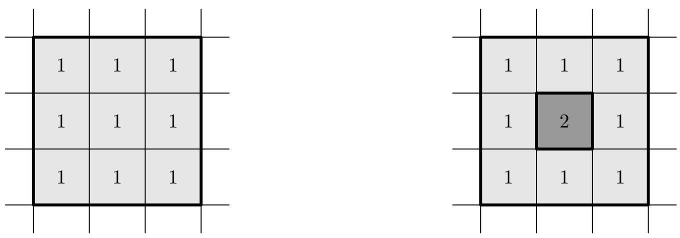
Output single integer: the sum of defense levels for all ways to build a fortress modulo $10^9 + 7$.

Examples

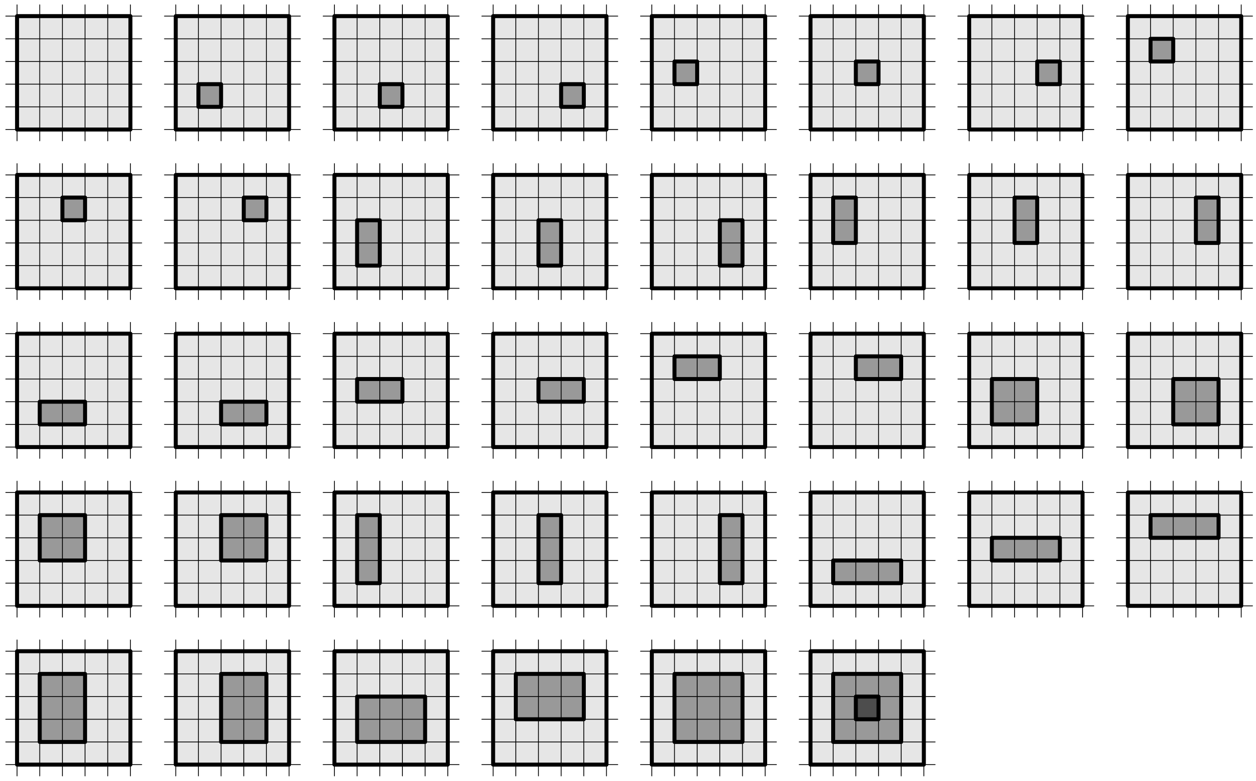
standard input	standard output
3 3	19
5 5	1060

Note

All possible ways to build a fortress from the first sample are in the picture.



All possible ways to build a fortress from the second sample are in the picture.



Problem L. Wires Puzzle

Time limit: 1 second
Memory limit: 512 megabytes

This is interactive problem. Your program will interact with judges' program using standard input and standard output.

Friends are planing a puzzle which has the following structure. There are n similar wires put through the opaque pipe. One of the pipe ends is the left one, another is the right one. The friends see n wires at each end of the pipe, the wires are labeled by distinct integers from 1 to n at each end, but a wire can have different labels at different ends of the pipe.

The goal of the puzzle is to make correspondence between wire ends, that is — for each i from 1 to n find the number a_i such that the wire with i on the right end is marked with a_i on the left end.

To solve the puzzle it is possible to make the following requests. You can choose an integer k ($1 \le k \le n$) and divide the wires at the right end to k non-empty groups, connecting all wires inside the same group. After that using a special device it is possible to detect for each pair of wires at their left end, whether their right ends are connected. Therefore for the left ends of the wires it is now known which labels correspond to the wires from the same group.

It is required to find the correspondence between left and right ends of the wires, using exactly three such queries.

To better understand the actions in the problem, it is recommended to study the example description below.

Interaction Protocol

First your program must read the integer n from standard input ($3 \le n \le 200$). Then it must make exactly three queries.

To make a query first print a single integer k on a line by itself — the number of groups. It must be followed by n integers $g_1, g_2, \ldots, g_n$. Each integer must be from 1 to k, each integer from 1 to k must occur at least once. The number g_i gives the number of the group that the right end of the wire labeled with i goes to. All right ends that are in the same group get connected.

The judges' program responds with the information which labels on the left ends of the wires correspond to the same group of wires. The format is the following: the information about k groups will be printed. The information about the j-th group starts with an integer s_j on a line by itself — the size of the group. It is followed by s_j integers — the labels on the left ends of the wires that are in this group. The order of groups and the order of wires inside the groups can be arbitrary, at the judges' program's discretion. Specially, it can be different from the numbering of groups in the query.

After performing exactly three queries, your program must print n integers $a_1, a_2, \ldots, a_n$, where a_i is the label of the left end of the wire with the label i at its right end.

Example

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

Note that input and output are formatted with empty lines to give a better view which line is a response to which query. In real interaction there are no empty lines in input and should be no empty lines in output.

Note

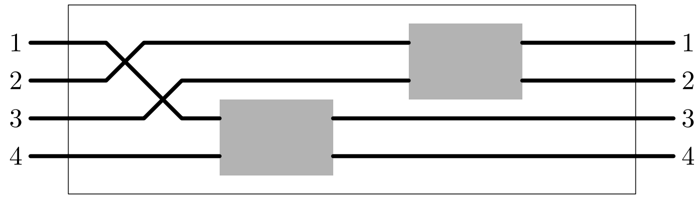
Consider the example above.

There are 4 wires.

The first query: there are three groups, the wires with right labels 1 and 2 are connected, the wire with right label 3 is in a group by itself, so is the wire with right label 4.

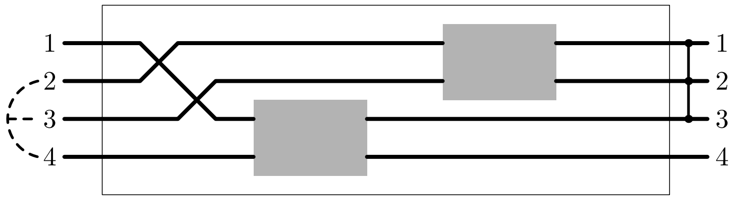
The judges' program responds that the wires with left labels 2 and 3 are in the same group, the wire with the left label 1 is in a group by itself, and so is the wire with the left label 4.

Analyzing the response, the contestant's program understands that the wires with left labels 2 and 3 are in some order the wires with right labels 1 and 4, and the wires with left labels 1 and 4 are in some order the wires with right labels 3 and 4.

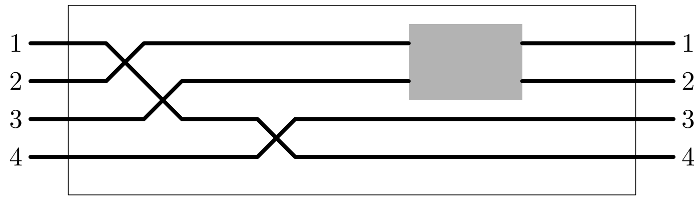


The second query has two groups of wires, wires with right labels 1, 2 and 3 are connected, the wire with right label 4 is in a group by itself.

The judges' program responds that the wires with left labels 2, 3 and 4 have the right ends connected.



The contestant's program understands that the wire with the right label 4 is has its left label 1, therefore the wire with the right label 3 has its left label 4.



The third query similarly allows to distinguish between wires with right labels 1 and 2. Now the correspondence is clear.