

Problem A. Forks

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 256 megabytes

A shipment of 5 sets of forks has been delivered to Viktor's store. Each set consists of a box containing exactly 5 forks.

Before displaying the sets for sale, it is necessary to find out the exact number of different sets of forks in order to free up space for them on the shelf. Each fork can be characterized by a single integer k — the number of tines.

Forks are considered identical if they have the same number of tines. Two sets a and b are considered the same if for every k , the number of forks with k tines in set a matches the number of forks with k tines in set b .

Help Viktor determine the number of different sets.

Input

Each of the 5 lines contains 5 integers — the number of tines. Each line represents a separate set of forks. All numbers in the input are positive integers not exceeding 100.

Output

Output a single integer — the number of different sets.

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

Note

In the first example, the sets are grouped as follows: (1), (2, 3, 5), (4). In the second example: (1, 5), (2, 3, 4).

Problem B. Producer

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 256 megabytes

Through the hands of producers pass many talented musicians, each absorbed in their craft, but lately there have been too many musicians. To solve this problem, one of the producers decided to group the musicians into ensembles of three people (a guitarist, a drummer, and a bassist). Each musician is characterized by their skill level X . The profit that a group brings to the producer is determined by the maximum skill level among all its members. Help the producer form groups of musicians to maximize profit.

Input

In the first line, there is an integer N ($1 \leq N \leq 10^5$) — the number of groups that need to be formed. In the following three lines for guitarists, bassists, and drummers respectively, there are N integers X_{ij} — their skill levels ($1 \leq X_{ij} \leq 1000$).

Output

Output one integer — the maximum possible profit.

Examples

standard input	standard output
4 1 2 3 4 1 2 5 6 1 2 9 3	24
5 1 2 3 4 5 1 2 5 6 4 1 2 9 3 10	35

Note

In the first example, the musicians can be grouped as follows: (1, 2, 9), (2, 6, 1), (3, 5, 2), (4, 1, 3).

Problem C. Mosaic

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 256 megabytes

Artists can create their works in many unusual ways, but many of them are lazy, so they ask for your help.

You need to find a “beautiful” pattern in a mosaic, where each element is colored in one of 26 colors, represented by lowercase letters of the Latin alphabet (from “a” to “z” inclusive). A “beautiful” pattern has the following characteristics:

- The elements are colored in K different colors;
- The quantity of elements of each color is unique, meaning that elements of different colors cannot appear the same number of times;
- The quantities of elements of each color form a sequence of natural numbers from 1 to K , where each number appears exactly once.

Help the artists find a “beautiful” pattern.

Input

In the first line, there is a number K ($2 \leq K \leq 26$) — the number of different colors in the “beautiful” pattern. In the second line, there are two numbers N , M ($2 \leq N \leq 400$, $2 \leq M \leq 400$) — the number of rows and columns in the mosaic. In each of the following N lines, there are M lowercase Latin letters representing the elements of the mosaic. It is guaranteed that a “beautiful” pattern exists.

Output

In the first line, output R_1 , C_1 — the row and column of the top-left corner of the “beautiful” pattern. In the second line, output R_2 , C_2 — the row and column of the bottom-right corner of the “beautiful” pattern. The rows and columns of the pattern are numbered starting from one. If there are multiple answers, output any.

Examples

standard input	standard output
2 3 4 abcz bbcy ccdx	1 2 3 2
3 3 4 abcz bbcy ccdx	1 1 3 2

Note

In the first example the character “c” appears 1 time, while the character “b” appears 2 times.

<i>a</i>	<i>b</i>	<i>c</i>	<i>z</i>
<i>b</i>	<i>b</i>	<i>c</i>	<i>y</i>
<i>c</i>	<i>c</i>	<i>d</i>	<i>x</i>

In the second example the character “a” appears 1 time, while “c” and “b” appear 2 and 3 times, respectively.

<i>a</i>	<i>b</i>	<i>c</i>	<i>z</i>
<i>b</i>	<i>b</i>	<i>c</i>	<i>y</i>
<i>c</i>	<i>c</i>	<i>d</i>	<i>x</i>

Problem D. Cakes

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 256 megabytes

Everyone loves sweets, so it's not surprising that Charlie's chocolate factory is expanding again. This time, it was decided to install a conveyor belt for making cakes. For this purpose, the conveyor will have N molds for cakes. Above the conveyor are dispensers, each filled with one of three types of filling. Each cake must consist of three different layers, arranged in a specific order. Each layer can be applied using one of the dispensers.

The conveyor can only move to the right. When the molds for the cakes are positioned under the dispensers, layers can be applied to the corresponding molds using the specified dispensers. When all three required layers have been applied to a cake, the cake is considered made. Your task is to load the dispensers so that all cakes can be made in the minimum number of movements of the conveyor.

More formally, there are N cakes, each defined by a triplet of numbers a, b, c . A cake must sequentially (but not necessarily consecutively) be positioned under the dispensers filled with fillings numbered a , then b , then c in that order to be considered made. The conveyor can be represented as a number line. At points 0, 1, etc., above it are the dispensers (you can assume an unlimited number of them). The first cake is located at point -1 , the second at -2 , and so on, with the N -th cake at point $-N$. Note that the dispensers are located at non-negative coordinates, while the cakes are at negative ones. You can shift all cakes one unit to the right (increase their coordinates by 1). A cake is under a dispenser if it is at the same point.

Your task is to load the dispensers and then make all the cakes with the minimum number of movements of the conveyor, and print this number on the screen.

Input

The first line of the input contains the number N ($1 \leq N \leq 10^5$) — the number of cakes. The next N lines describe the i -th cake with three numbers a, b, c ($1 \leq a, b, c \leq 3, a \neq b \neq c, a \neq c$).

Output

The output should contain a single integer — the minimum necessary number of movements of the conveyor.

Examples

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

Note

In the first test case, it is necessary to load the dispensers (1, 2, 3). In the second example, the sequence would be (2, 1, 3, 2, 1).

Problem E. Millionaire

Input file: **standard input**
Output file: **standard output**
Time limit: 2.5 seconds
Memory limit: 256 megabytes

This is an interactive problem.

A millionaire decided to open N accounts in the bank to store his money. The accounts are numbered from 1 to N . Each account can hold any integer amount of money (including negative values). The bank supports three types of operations:

1. $l\ r\ c$ — Add the value c to the accounts numbered $l, l + 1, \dots, r$;
2. $d\ c$ — Add the value c to the accounts numbered $d, 2d, 3d, \dots$;
3. $l\ r$ — Get the total amount of money in the accounts $l, l + 1, \dots, r$.

The millionaire has many accounts and plans to use them actively, but he fears that the banking system may not handle his requests efficiently. Help the bank by writing a program that efficiently executes all the required operations.

Input

The first line of input contains a single integer N ($1 \leq N \leq 10^5$) — the number of accounts. The second line contains N integers $a_1, a_2, \dots, a_N$ ($|a_i| \leq 10^9, 1 \leq i \leq N$) — the initial amounts of money in each account. The third line contains the integer M ($1 \leq M \leq 10^5$) — the number of operations. The next M lines describe the operations. The first number indicates the type of operation.

If the type is 1, it is followed by three numbers l, r, c ($1 \leq l \leq r \leq N, |c| \leq 10^4$).

If the type is 2, it is followed by two numbers d, c ($1 \leq d \leq N, |c| \leq 10^4$).

If the type is 3, it is followed by two numbers l, r ($1 \leq l \leq r \leq N$).

Output

For each operation of the third type, output the total amount of money in the specified accounts.

Interaction Protocol

Initially, the program will output the number of accounts and the initial amounts of money in them. After that, the program will output a number representing the number of operations and the operations themselves. After each operation of the third type, you need to output the answer to it, and only after that will you have access to information about the next operations.

Example

standard input	standard output
10	10
1 2 3 4 5 6 -2 -3 -4 -5	-6
9	15
3 4 8	20
1 7 10 2	
3 7 10	
2 2 -2	
3 1 6	
2 3 2	
2 5 -7	
1 2 8 4	
3 3 10	

Note

In the example, after the second operation, the amounts in the accounts will be as follows:

1 2 3 4 5 6 0 -1 -2 -3

After the fourth operation:

1 0 3 2 5 4 0 -3 -2 -5

After the sixth operation:

1 0 5 2 5 6 0 -3 0 -5

After the seventh operation:

1 0 5 2 -2 6 0 -3 0 -12

After the eighth operation:

1 4 9 6 2 10 4 1 0 -12

Problem F. Volume

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 256 megabytes

Igor is an audiophile, and he is very sensitive to sound volume. He has speakers that output sound with a volume of M , but he wants to reduce the volume exactly to V . To achieve this, he has N volume regulators. The special feature of these regulators is that they can only be set to values that are prime numbers or 1. Initially, all the regulators are set to their maximum values, which are different for each regulator.

However, the values set on the regulators can be changed, and the overall volume will change proportionally. That is, if the value of one regulator is changed by a factor of X , while the values of the other regulators remain unchanged, the overall volume will also change by a factor of X , regardless of the values set on the other regulators. Of course, the regulator cannot be set to a value higher than its maximum.

Help Igor set the values on the regulators such that the volume from the speakers becomes equal to V .

Input

The first line of the input contains three positive integers M , V , and N — the maximum volume, desired volume, and number of regulators ($0 < V < M < 10^9$; $N \leq 50$). The second line contains N positive integers m_i — the maximum values of the volume regulators ($2 \leq m_i \leq 10^9$).

Output

You need to output N prime numbers — the values that need to be set on the regulators so that the volume becomes equal to V . The i -th number must not exceed the maximum value of the i -th regulator m_i .

If there are multiple solutions, output any.

If it is impossible to achieve a volume of V , output -1 .

Examples

standard input	standard output
50 15 3 5 2 3	3 1 3
35 21 2 7 5	7 3
21 12 4 3 3 5 7	-1

Note

In the first example, by setting the values of the regulators to 3, 1, and 3, the volume is $\frac{3}{5} \cdot \frac{1}{2} \cdot \frac{3}{3} = \frac{3}{10}$ of the maximum, which corresponds to the required volume of 15.

In the second example, the volume is set to $\frac{3}{5}$ of the maximum 35, which equals 21.

In the third example, it is impossible to achieve a volume of 12.

Problem G. Metro

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **256 megabytes**

A metro map is considered beautiful if it can be drawn by following these steps:

- At least three distinct straight lines must be drawn, all intersecting at a single point referred to as the **central station**. This intersection divides each line into two distinct rays.
- Each ray must contain a finite, non-zero number of stations, and the count of stations can vary from ray to ray.
- Identify the ray that has the smallest number of stations among all rays.
- From the identified ray, select at least one station that is distinct from the central station.
- A ring is defined as a set of stations selected from each ray that are equidistant (in terms of the number of stations) from the central station. For instance, if the ring includes the fifth station from the central station on one ray, it must also include the fifth station from the central station on each of the remaining rays. The stations in the ring are then connected in a clockwise manner.
- It is permissible to place additional stations on the rings, though this is not mandatory.

Help determine whether the metro map is beautiful, that is, whether it can be obtained using the algorithm described above.

Input

The first line contains two integers n and m ($1 \leq n \leq 10^5$, $1 \leq m \leq 2 \cdot 10^5$) — the number of stations and the number of metro segments connecting the stations, respectively.

In the next m lines, the segments are defined using two integers a_i and b_i ($1 \leq a_i, b_i \leq n$) — the numbers of the stations connected by the segment.

It is guaranteed that there are no duplicate segments and no segments connecting a station to itself.

Output

If the metro map is beautiful, print “Yes” without quotes. Otherwise, print “No” without quotes.

Examples

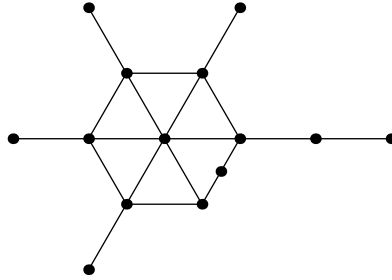
standard input	standard output
14 19 1 2 7 6 7 12 8 1 8 7 9 14 10 3 10 9 11 4 11 10 12 5 12 11 13 7 13 8 13 9 13 10 13 11 13 12 14 8	Yes
7 6 1 4 2 4 4 3 5 4 6 4 7 4	No
6 10 2 1 2 6 3 1 3 2 4 1 4 3 5 1 5 4 6 1 6 5	No

standard input	standard output
8 13 2 1 3 1 3 2 4 1 4 3 5 1 5 4 6 1 6 5 7 2 7 6 8 1 8 7	No
13 24 1 2 1 3 2 3 2 7 4 1 4 3 5 3 5 4 6 3 6 5 7 3 7 6 8 2 9 1 9 8 10 4 10 8 11 5 11 12 12 6 12 10 13 7 13 9 13 11	No

standard input	standard output
26 36 2 1 2 7 3 1 3 2 4 1 4 3 5 1 5 4 6 1 6 5 7 1 7 6 8 2 9 3 10 4 11 5 12 6 14 13 14 19 15 13 15 14 16 13 16 15 17 13 17 16 18 13 18 17 18 19 19 13 20 14 21 15 22 16 23 17 24 18 25 19 26 7	No

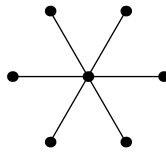
Note

1. In the first example, the metro map looks as follows:



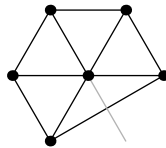
It is beautiful.

2. In the second example, the metro map looks as follows:



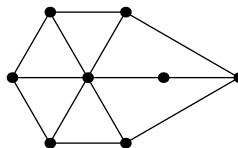
It is not beautiful because there are no rings.

3. In the third example, the metro map looks as follows:



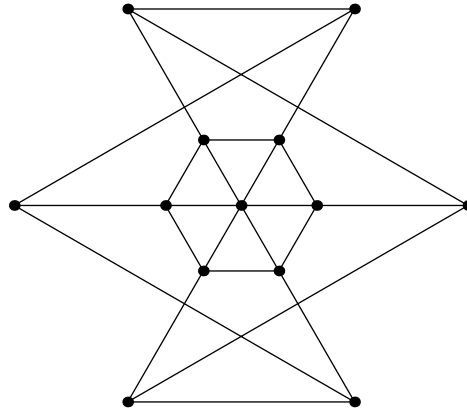
It is not beautiful because there is a ring that does not pass through all rays.

4. In the fourth example, the metro map looks as follows:



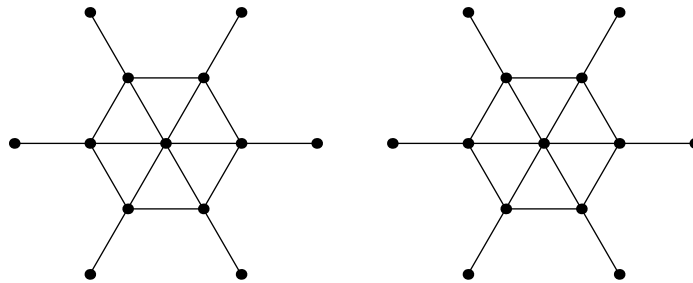
It is not beautiful because the number of stations from the central station to the ring is different.

5. In the fifth example, the metro map looks as follows:



It is not beautiful because the second ring intersects the rays in a different order than the first ring.

6. In the sixth example, the metro map looks as follows:



It is not beautiful because there are pairs of stations between which there is no path.

Problem H. Domino

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **256 megabytes**

Alexey has a set of domino pieces of different heights. Deciding to study domino principle in practice, he placed the pieces at different integer points on a coordinate line. Then he chose a particular coordinate. He decided to push the closest domino piece to the left of that coordinate to fall left, and the closest piece to the right of it to fall right. If there is no domino piece to push in some direction, Alexey simply pushed nothing in that direction.

Now he wants to find out how many domino pieces will fall down.

Each domino piece, when falling to the right, will knock down all domino pieces located to the right of it within a distance equal to the height of that piece. Conversely, when falling to the left, it will knock down all domino pieces to the left of it within the same distance.

Thus, if a piece with height H , located at coordinate X , is knocked down to the right, it will knock down all pieces located in the segment $[X + 1; X + H]$ to the right. If it is knocked down to the left, it will knock down all pieces in the segment $[X - H; X - 1]$ to the left.

Help Alexey to find out how many domino pieces will fall.

Input

The first line contains two positive integers N and K — the number of domino pieces and the coordinate chosen by Alexey respectively ($N \leq 10^5$; $K \leq 10^7$; K is odd). Each of the following N lines contains two positive integers X_i and H_i — the coordinate and height of the i -th domino piece respectively ($X_i, H_i \leq 10^7$; X_i is even; H_i is odd). It is guaranteed that the coordinates of all domino pieces are distinct.

Output

Output one number — the total number of domino pieces that will fall down.

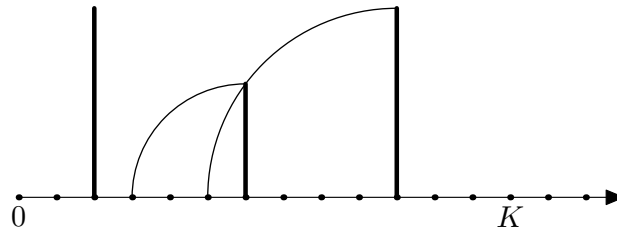
Examples

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

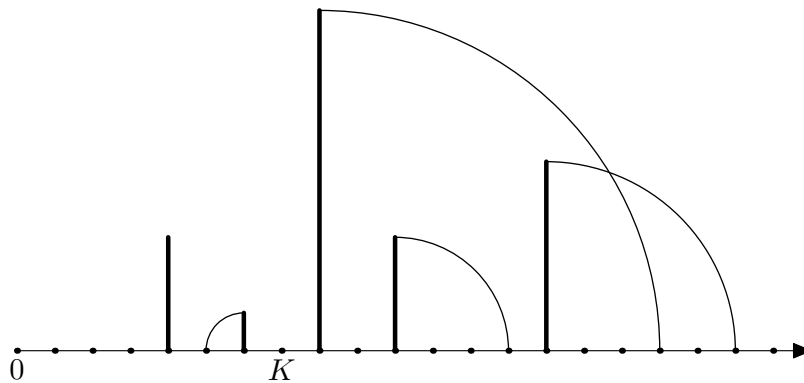
Note

In the first example, there are no domino pieces to the right of coordinate K . On the left, Alexey will

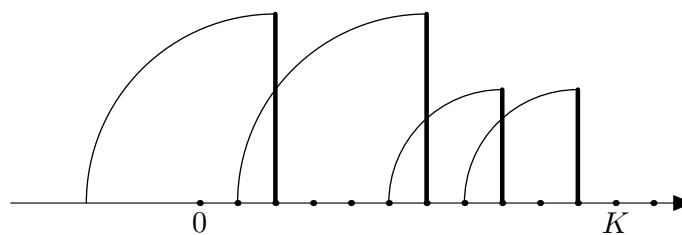
push the piece at coordinate 10. When it falls, it will knock down another piece at coordinate 6, but none of the fallen pieces will reach the domino piece at coordinate 2.



In the second example, Alexey will push the pieces at coordinates 6 and 8. The piece at 6 does not reach the domino at 4 when falling left. However, the piece at 8 will knock down two more pieces when falling right.



In the third example, there are no domino pieces to the right of coordinate K . On the left, Alexey will push the piece at coordinate 10. When it falls, it will knock down the piece at coordinate 8, which will knock down the piece at coordinate 6, and that one, in turn, will knock down the piece at coordinate 2.



Problem I. Satellite Internet

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 256 megabytes

Pasha is traveling by train from Moscow to Izhevsk — he has another contest tomorrow. Suddenly, while passing through Vekovka, he realized that he made a mistake in the interactor for one of the problems, and he urgently needs to send a new interactor. However, the interactor is huge, and for a guaranteed upload, Pasha needs each satellite located above Russia to be available.

Unfortunately, a lot of clouds have gathered directly above the train's route today. Fortunately, in this problem, the cloud accumulation can be considered as a segment, and the train's route can be represented as a horizontal segment. Unfortunately, the cloud accumulation is significant, which means this accumulation must be treated as impenetrable for the satellite signal. The endpoints of the segment are also considered impenetrable.

More formally: each satellite can be thought of as a point. A satellite is considered available from a point on the train's route if there are no clouds between them. Help Pasha find the nearest point on the route to Moscow from which all satellites are available.

Input

The first line contains a single integer N ($1 \leq N \leq 10^5$) — the number of satellites. The next N lines each contain exactly two integers x_i and y_i ($-10^4 \leq x_i \leq 10^4; 1 \leq y_i \leq 10^4$), which are the coordinates of the i -th satellite.

The following line contains two integers $startx$ and $endx$ ($-10^4 \leq startx < endx \leq 10^4$). The train's route is represented by the segment connecting the points $(startx, 0)$ and $(endx, 0)$. The start of the route is at the point $(startx, 0)$.

The next line contains 4 integers $cloud_{x1}$, $cloud_{y1}$, $cloud_{x2}$, $cloud_{y2}$ ($-10^4 \leq cloud_{x1}, cloud_{x2} \leq 10^4; 1 \leq cloud_{y1}, cloud_{y2} \leq 10^4$) — the coordinates of the endpoints of the cloud segment. It is guaranteed that the cloud segment has a non-zero length.

It is guaranteed that a solution exists. It is guaranteed that if all satellites are accessible from point X , there is a segment of length 10^{-5} that contains point X , from which all satellites are accessible. It is guaranteed that satellites do not lie on the straight line containing the cloud segment.

Output

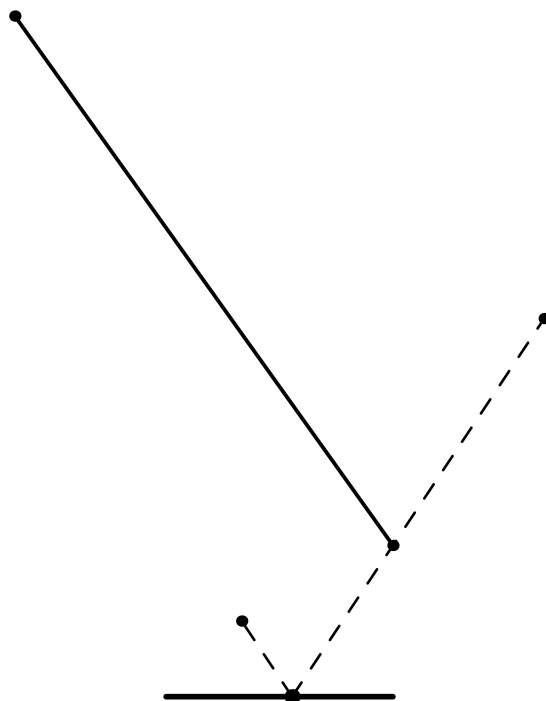
Output the X -coordinate of the sought point on the train's route. The answer must differ from the correct answer by no more than 10^{-5} in absolute or relative error.

Example

standard input	standard output
2 1 1 5 5 0 3 -2 9 3 2	1.666666666666667

Note

The left satellite is available from any point on the route. The right satellite is available on the segment from $1\frac{2}{3}$ to the end of the route.



Problem J. Maze

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 512 megabytes

This is an interactive problem.

You are given a maze that consists of a grid of $N \times M$ ($3 \leq N, M \leq 30$) rooms. Any two rooms that are adjacent by a side are connected by a bidirectional colored corridor. There are only four colors that the corridors can be painted: red, green, blue, and yellow.

One of the rooms in the maze is the starting room, and one is the ending room. You can only move between adjacent rooms through the connecting corridors. You are allowed to move through a corridor into another room and then return. As you move, a sequence of your moves is formed. When you move forward, the color of the corridor you are passing through is added to the sequence, and when you move back, the last element is removed from the sequence. However, there is one restriction: you cannot pass through a corridor of color C if among the last three elements of the sequence there is C .

Neither the maze, the starting position, nor the ending position are known to the participant. You need to reach the ending room from the starting room. It is guaranteed that the ending room is reachable.

Interaction Protocol

At the beginning, the jury program outputs two numbers N and M . Then for each move the participant makes, the jury program outputs the success of that step or asks to end the program if the participant has reached the ending room.

Each move must be one of these commands:

- “MOVE LEFT” — if you want to go to the room located to the left of you;
- “MOVE RIGHT” — if you want to go to the room located to the right of you;
- “MOVE UP” — if you want to go to the room located above you;
- “MOVE DOWN” — if you want to go to the room located below you;
- “BACK” — to move back.

The jury program will reply with one of these:

- “OK” — if your move was successful and you moved to another room;
- “FAIL” — if your move was unsuccessful, meaning you tried to pass through a corridor of an invalid color or tried to go back with an empty sequence;
- “EXIT” — if you have reached the ending room.

Starting from test 22, the jury’s program is adaptive to the participant’s program output.

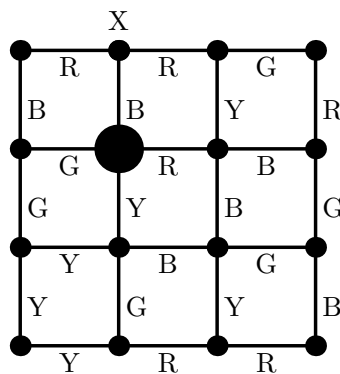
The total number of moves must not exceed 100 000.

Example

standard input	standard output
4 4	
FAIL	BACK
OK	MOVE DOWN
FAIL	MOVE LEFT
OK	MOVE RIGHT
FAIL	MOVE LEFT
FAIL	MOVE UP
FAIL	MOVE DOWN
OK	BACK
OK	BACK
EXIT	MOVE UP

Note

In the first test, the jury has the following maze:

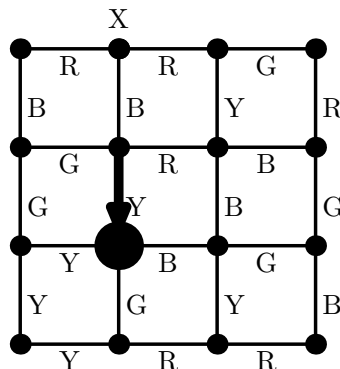


Black dots represent rooms. The corridors between rooms are represented by lines. The letter next to the corridor indicates its color (“Y” — yellow, “R” — red, “G” — green, “B” — blue). Initially, the participant is in the room marked with a bold dot. Next to the room that is the final destination for the participant is the letter “X”. Neither the maze, nor the starting position, nor the ending position are known to the participant.

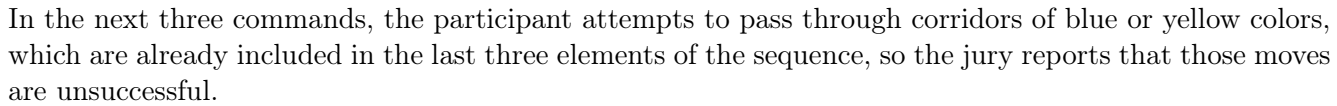
The first command is BACK, but since there have been no moves yet, nothing happens. The jury program reports that the move was unsuccessful (“FAIL”).

The second command is MOVE DOWN. After this, the sequence of colors of the passed corridors will be: “yellow”.

The movement is shown in the picture below.



The next command is **MOVE RIGHT**. The sequence of colors now becomes: “yellow, blue”. The movement is shown in the picture below.



The next command is **BACK** again. The sequence of moves is: “” (empty).

Page 21 of 23

Problem K. Knights

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 256 megabytes

We all know that the Middle Ages were a time of lords, vassals, and of course, knights. In this task, you need to solve a very important question: which of the N courtiers should become knights, the defenders of the kingdom.

For convenience, let's assign numbers to the courtiers from 1 to N . Each courtier, except for the king, is subordinate to exactly one courtier. The king is assigned the number one and does not obey anyone. For each courtier, the subordinate courtiers (vassals) are known, as well as two numbers L and R . These numbers imply that the retinue of this courtier must contain at least L and at most R knights. The retinue of a courtier includes all knights who are directly or indirectly subordinate to him, and of course, he himself if he is a knight.

More formally, a knight Y is subordinate to a courtier X if there exists a sequence $a_1, a_2, \dots, a_j$ such that a_1 is subordinate to X , a_2 is subordinate to a_1 , and so on, with Y being subordinate to a_j . It is guaranteed that it is possible to select knights in such a way that the specified condition is satisfied.

Input

The first line of the input contains the number N ($1 \leq N \leq 2 \cdot 10^5$) — the number of courtiers.

Each of the following N lines describes the i -th courtier: L_i, R_i (the minimum and maximum size of the retinue), K_i ($0 \leq K_i \leq 2 \cdot 10^5, 0 \leq L \leq R \leq 10^9$). After that, there are K_i numbers — the list of vassals of the i -th courtier (vassals can be numbered from 2 to N , different from i , and each number appears exactly once in the input file). It is guaranteed that an answer exists and each courtier is directly or indirectly subordinate to the king, who has the number 1.

Output

Print in the first line the number M ($0 \leq M \leq N$) — the number of knights. In the next line, print M distinct numbers ($1 \leq M_i \leq N$) — the numbers of the courtiers who should become knights. If there are multiple answers, you may print any.

Examples

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

Note

In the first test, the king is forced to become a knight.

In the second test, courtiers 2 and 4 can become knights. Then the king and courtier 2 will have 2 knights in their retinue. Courtiers 3 and 5 will have no knights at all. Courtier 4 will have exactly 1 knight — himself.

In the third test, the king requires 4 knights, and thus all courtiers will have to become knights and join his retinue.