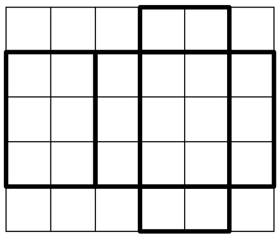
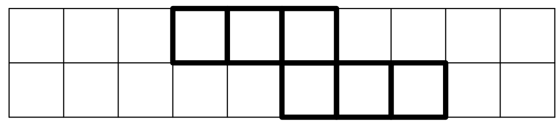
Problem A Box

Time limit: 3 seconds

Bella is working in a factory that produces boxes. All boxes are in a shape of rectangular parallelepipeds. A *net* of the corresponding parallelepiped is cut out of a flat rectangular piece of cardboard of size $w \times h$. This net is a polygon with sides parallel to the sides of the rectangle of the cardboard. The net is bent along several lines and is connected along the edges of the resulting parallelepiped to form a box. The net is bent only along the edges of the resulting box.



The first example



The third example

Bella is a software developer and her task is to check whether it is possible to make a box of size $a \times b \times c$ out of a cardboard of size $w \times h$. Bella did write a program and boxes are being produced. Can you do the same?

Input

The first line contains three integers a, b, and c — the dimensions of the box.

The second line contains two integers w and h — the width and the height of the cardboard.

All integers are positive and do not exceed 10^8.

Output

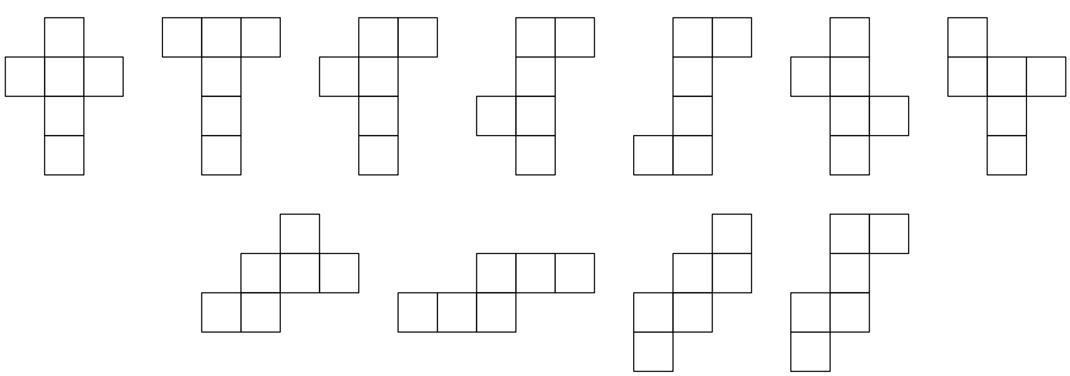
Print "Yes" if it is possible to cut a box $a \times b \times c$ out of a cardboard of size $w \times h$. Print "No" otherwise.

Examples

input	output
1 2 3 6 5	Yes
1 2 3 5 5	No
1 1 1 10 2	Yes

Note

There are 11 different nets of a cube, ignoring rotations and mirror images.



Problem B Easy Quest

Time limit: `3 seconds`

A young hero is starting his heroic life. The wise wizard suggested him an easy first quest. During this quest our young hero meets n magical creatures, in specific order. In order to help the young hero, the wizard gave him a clue — a list of n integers a_i.

- If a_i is positive, then the i-th magical creature is benevolent and gives to our hero one magical item of type a_i. The hero can keep several items of the same type.
- If a_i is negative, then the i-th magical creature is evil and in order to defeat it the young hero needs one magical item of type $-a_i$. All magical items are fragile and can be used only once.
- If a_i is zero, then the i-th creature is a unicorn. It gives the hero any magical item he asks for, but only one.

Your task is to help the young hero to finish the first quest, defeating all enemies on the way, or say that it is impossible.

Input

The first line of input contains one integer n ($1 \le n \le 1000$). The second line contains n integers a_i ($-1000 \le a_i \le 1000$).

Output

If it is impossible to defeat all enemies, then output one string "`No`". If it is possible, then output string "`Yes`", and in the next line output the types of items the hero should ask the unicorns for, in order they meet during the quest. Types must be integers in range from 1 to 1000 inclusive. If there are several solutions, output any of them.

Examples

input	output
10 1 0 -4 0 0 -1 -3 0 -1 -2	Yes 4 1 3 2
5 5 8 0 -6 -3	No
3 2 -2 -2	No

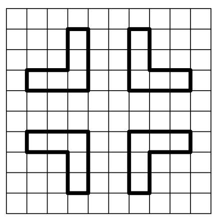
Problem C The Final Level

Time limit: `3 seconds`

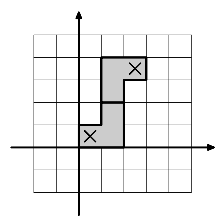
Fiora is a game designer. Now she is designing the final level for her new game.

A level for this game is a labyrinth on a rectangular grid with lots of enemies. Player starts her game at the square $(0, 0)$ and her purpose is to get to the square (a, b). Fiora has lots of ideas on how to put enemies, but she does not like designing labyrinths. She needs your help here.

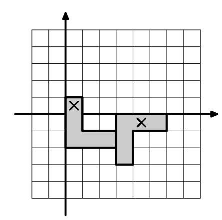
Fiora is drawing levels in a special level editor which supports one basic block to design a labyrinth. This block is an L-shaped corner, consisting of two perpendicular rectangles $1 \times n$ squares in size intersecting at 1×1 square. It is possible to rotate this block in four ways. Blocks cannot intersect, but they can touch each other. Player can move through all the squares lying in any block. She can move between two squares if they are sharing a side, even if they are in different blocks.



Blocks with $n = 3$



The first example



The second example

Fiora wants to design the final level with the minimal possible number of blocks. Of course, it should be possible to move from square $(0, 0)$ to square (a, b).

Input

The first line of the input consists of a single integer m ($1 \le m \le 100$) — the number of test cases. It is followed by m test cases. Each test case is on a separate line and consists of three integers a, b, and n ($-10^8 \le a, b \le 10^8$; $2 \le n \le 10^8$) — a is the coordinate of the final point along the horizontal axis, b is the coordinate of the final point along the vertical axis, and n is the size of the block. The final point is not same as the starting one (either $a \ne 0$ or $b \ne 0$).

Output

For each test case, in the first line print the minimal number k of blocks you need. In the following k lines print description of these blocks. Each L-shaped corner block is described by coordinates of two cells. Print coordinates of the end of its vertical rectangle, followed by coordinates of the end of its horizontal rectangle. Specify the coordinates of the ends that are opposite to the intersection of the rectangles. Note that the order of cells in the block description matters, since a change of the order results in a reflected block. Coordinates of each end should be printed with the coordinate along the horizontal axis first, followed by the coordinate along the vertical axis.

All coordinates in the output should not exceed 10^9 by absolute value.

It is guaranteed that the total number of blocks in the correct output does not exceed 10^5 for all test cases combined.

Examples

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

Problem D Interactive Sort

Time limit: `10 seconds`

Ivan wants to play a game with you. He took all integers from 1 to n inclusive, shuffled them and then put all even numbers into array e and all odd numbers into array o.

Your task is to find arrays e and o.

You can ask Ivan questions of certain kind. Each question consists of two integers i and j. For each question Ivan says whether $e[i] < o[j]$ or not.

You can ask at most 300 000 questions.

Interaction Protocol

First, the testing system writes the integer n ($1 \le n \le 10\,000$) — the number of integers Ivan used.

Your solution shall print requests of two types:

- "? i j". $1 \le i \le \lfloor \frac{n}{2} \rfloor, 1 \le j \le \lceil \frac{n}{2} \rceil$. The testing system responds with the symbol "<" if $e[i] < o[j]$ or with the symbol ">" otherwise.
- "! e_1 e_2 ... $e_{\lfloor \frac{n}{2} \rfloor}$ o_1 o_2 ... $o_{\lceil \frac{n}{2} \rceil}$" tells the values of e and o that your program has determined.

Don't forget to flush the output after each request!

Your solution must issue exactly one request of the second type, which must be the last request, and the solution must terminate gracefully after issuing it.

Your solution is allowed to issue at most 300 000 requests of the first type.

For each test case the number n is fixed and the numbers are shuffled using Java built-in shuffle function with fixed seed.

Examples

input	output
5	? 1 1
>	? 1 2
>	? 1 3
<	? 2 1
>	? 2 2
<	? 2 3
<	! 4 2 1 3 5

Problem E Laminar Family

Time limit: `3 seconds`

While studying combinatorial optimization, Lucas came across the notion of "laminar set family". A subset family $\mathcal{F}$ of some set Ω is called *laminar* if and only if it does not contain an empty set and for any two distinct sets $A, B \in \mathcal{F}$ it is correct that either $A \subset B$ or $B \subset A$ or $A \cap B = \varnothing$.

As an experienced problem setter Lucas always tries to apply each new piece of knowledge he gets as an idea for a programming competition problem. An area of his scientific interests covers *recognition problems* that usually sound like "Given some weird combinatorial property, check if the given structure satisfies it".

Lucas believes that the perfect programming competition problem should contain ~~a cactus~~ a tree in it. Trying to put together laminar sets and trees into a recognition problem, he finally came up with the following problem: given an undirected tree on n vertices and a family $\mathcal{F} = \{F_1, \ldots, F_k\}$ of sets, where F_i consists of all vertices belonging to the simple path between some two vertices a_i and b_i of the tree, check if the family $\mathcal{F}$ is a laminar family. Note that in this case $\Omega = V$, and each $F_i \subseteq V$.

As you can see, Lucas had succeeded in suggesting this problem to the programming contest. Now it is up to you to solve it.

Input

The first line of the input contains two integers n and f ($1 \le n, f \le 100\,000$) — the number of vertices in the tree and the number of elements in a family $\mathcal{F}$.

Next $n-1$ lines describe the tree structure. In the i-th line there are two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$) — the indices of the vertices that are connected by the i-th edge of the tree.

Next f lines describe the sets forming the family $\mathcal{F}$. In the i-th line there are two integers a_i and b_i ($1 \le a_i, b_i \le n$), such that F_i consists of all vertices belonging to the simple path between vertices a_i and b_i in the tree (including a_i and b_i).

Output

Output the only word "`Yes`" or "`No`" depending on whether or not the given set family is laminar.

Examples

input	output
4 2 1 2 2 3 2 4 1 2 4 2	No
6 5 1 2 2 3 3 4 5 6 5 2 2 1 6 6 1 4 3 4 4 1	Yes