Problem A. Poster

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are preparing a poster to attract up-and-coming coders for your programming competition. Naturally, you wish to make it as flashy as possible. The poster contains a text consisting of only English alphabet letters, digits, punctuation marks and spaces.

You are preparing the poster by hand and you wish to write the text using your three color pen. The pen can write in blue, yellow and red ink. However, there is only enough ink left in the pen that you can write at most b symbols in blue, y symbols in yellow and r symbols in red.

You've decided that the three types of symbols (letters, digits and punctuation marks) should each be colored in its own color. Determine whether it is possible to achieve this using the given pen!

Input

The first line of input contains 3 integers b, y and r ($0 \le b, y, r \le 1\,000$), the maximum number of symbols that can be written with the blue, yellow and red ink, respectively. The next line contains the text that needs to be written in the poster, which consists of only English alphabet letters, digits, punctuation marks and spaces. The only punctuation marks in the input can be "`,`", "`-`", "`;`", "`:`", "`.`", "`!`" and "`?`".

The length of the text is between 1 and 1 000 characters. The text begins and ends with characters different from the space character. There are no two consecutive space characters in the text.

Output

Output "`Yes`", if it is possible to choose a different color for each type of symbol to write the given text or "`No`" otherwise.

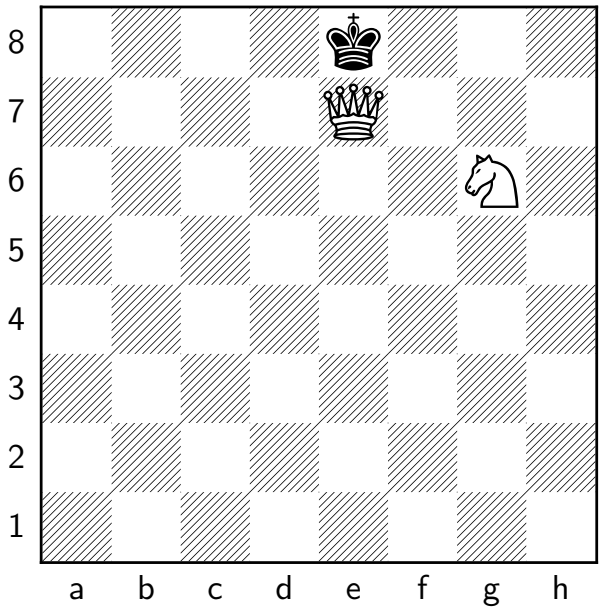
Examples

standard input	standard output
21 8 2 ICPC Programming Season 2023-2024!	Yes
15 5 10 Selection Contest: October 13, 2023.	No
5 20 5 13 problems in the contest!	Yes

Problem B. Checkmate

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

This is a chess problem. A single black king is located at some position on an otherwise empty chess board. You need to choose and place the minimum number of pieces from the standard set of 8 white main pieces (a king, a queen, two rooks, two bishops and two knights) so that the game becomes a checkmate for black. See the note below for an explanation of the relevant chess rules.



For example, if the black king is in the 5th square of the top row, then we can place a knight in the 7th square of the 6th row from the bottom and the queen in the 5th square of the 7th row from the bottom to achieve a checkmate for black. This is also the minimum possible number of pieces. In this problem it is not necessary to use the white king. However, if the white king is placed on the board, it must not be checked by the black king.

Chess notation. To describe the positions of the chess pieces on the board, we use the so-called *algebraic* chess notation. The positions of the cells of the 8×8 square chess board are described by the coordinate system where the columns are labeled with letters from `a` to `h` from left to right and the rows are numbered with integers from `1` to `8` from bottom to top. Each main piece type is denoted by a single capital letter: `K` for king, `Q` for queen, `R` for rook, `B` for bishop and `N` for knight. A piece on the board then is described by a string of three characters: the piece letter followed by the column and row coordinates of the location cell. For example, a rook in the 3rd square of the 5th row from the bottom is denoted by `Rc5`.

Input

The input contains the position of the black king in the algebraic notation.

Output

In the first line output an integer n, the minimum number of white pieces needed to achieve a checkmate. In the next n lines, describe the positions of these pieces on the board in the algebraic notation in any order. If there are multiple solutions, output any of those.

Examples

standard input	standard output
Ke8	2 Ng6 Qe7
Kc1	2 Ra2 Rf1
Kh4	2 Qg4 Kf4

Note

Pieces. There are 5 types of main pieces in chess, *kings*, *queens*, *rooks*, *bishops* and *knights*. Each of these pieces is able to move in the following squares in a single move:

- a king can move to any of the squares sharing either a side or a corner with its current square;
- a queen can move to any square horizontally, vertically or diagonally in any direction;
- a rook can move to any distance horizontally or vertically;
- a bishop can move to any distance diagonally in any direction;
- a knight can move to any of the closest squares that are not in the same row, column, or a diagonal as its current square.

The pieces cannot jump over other pieces, with the exception of the knights. A piece cannot move to a square which is already occupied by a piece of the same color. If a square contains some piece, a piece of the other color can move there, removing the other piece from the board (which is called *capturing* this piece). A piece is said to be *attacking* a square if it can move there in a single move. If a king's square is attacked by some other piece, it is said that the king is *in check*.

Checkmate. In this problem, a checkmate is a configuration of pieces on the board such that it is turn for black to move and the following conditions are true:

- the black king is in check;
- after any possible move, the black king is still in check.

Problem C. Flipping

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given an $n \times m$ ($n, m \leq 400$) cell grid where each cell contains either a 0 or a 1. You wish to make all cells contain 0. Initially you are located in the upper left cell of the grid and in one move you can move into an adjacent cell (one sharing a side with your current cell). When you move into a new cell, its value flips to the opposite (0 to 1 and vice versa).

Find a journey of at most $4 \cdot 10^5$ moves after which all grid cells contain a 0. You can finish in any cell and you can visit each cell any number of times.

Input

The first line contains two integers n and m ($2 \leq n, m \leq 400$), the height and width of the grid. The next n lines each contains a string of m characters (either `0` or `1`), the values of the corresponding row.

Output

Output a single string of length at most $4 \cdot 10^5$ which describes your journey. The i-th character should be one of "`U`", "`R`", "`D`" or "`L`", meaning that your i-th move is to the adjacent cell up, right, down or left, respectively. It is forbidden to go outside of the grid. If there are multiple solutions, output any of those. The output string can be empty.

Example

standard input	standard output
2 3 010 001	RDRL

Problem D. Railroads

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

There are n train stations numbered from 1 to n. For each station there are two one-way outgoing railroad connections. From the i-th station each outgoing railroad connection can go to one of these destinations:

- the 1-st station;
- the $(i+1)$-th station, if $i < n$.

You are responsible for evaluating the route of an automated train on these railroads. During one run the automated train keeps count of how many times it has visited each station. If the train has visited the current station an odd number of times, it will use the first outgoing railroad connection, otherwise it will use the second outgoing railroad connection.

You have been given a list of q possible runs of the train, for each run you have been given the start station and how many times the train goes to the next station. The train considers the start station visited for its routing. For each run you need to find the station number where the train will stop.

Input

The first line contains an integer n ($1 \le n \le 2 \cdot 10^5$), the number of stations. The next two lines each contains a length n symbol sequence consisting of the symbols "<" and ">". The first line represents the first outgoing connection from each station, the second line represents the second outgoing connection from each station. The i-th symbol in these sequences describes the outgoing connection from the i-th station, with ">" representing a connection to the $(i+1)$-th station, and "<" to the 1-st station. It is guaranteed that the last symbol in both sequences will be "<".

The following line contains an integer q ($1 \le q \le 2 \cdot 10^5$), the number of runs to be queried. Then follow q lines each representing a run of the automated train. In each line there are two space separated integers: the starting station number s ($1 \le s \le n$) and the number of times the train will go to the next station l ($1 \le l \le 10^{18}$).

Output

Output q lines with a single integer each – for each queried run of the automated train output the station number where the train will stop.

Examples

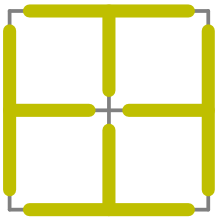
standard input	standard output
12 >>>><><><>< <>>>>><>><< 3 1 1 4 7 8 14	2 6 6
2 << << 2 1 3 2 3	1 1

Problem E. T-shirt

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are a fashion designer and you are excited about your fresh idea for a T-shirt. A T-shirt of size n for your purposes is an $n \times n$ grid of unit squares. Your idea is to make such a T-shirt by sewing together *T-patches*. A T-patch consists of 3 near-unit length segments connecting in a single point as shown:

Your design idea requires placing T-patches so that each side of each square of the T-shirt is covered by exactly one T-patch. Each T-patch can be rotated in any of the four orthogonal directions. For example, a T-shirt of size 2 can be sewn as follows:



Given the size of the T-shirt n, find a correct way to sew it from T-patches or determine that it is impossible!

Input

The input contains a single integer n ($1 \le n \le 1\,000$), the size of the T-shirt.

Output

The first line of output should contain "`Yes`" if there exists a required T-shirt of the given size, and "`No`" otherwise. In the first case you should then output $n + 1$ lines of $n + 1$ characters describing the design. The j-th character in the i-th line describes the intersection of the i-th horizontal and j-th vertical grid lines. For a T-patch, denote the shortest segment as its *stem*. If a center of a T-patch is located at that point output "`D`", "`R`", "`U`" or "`L`, if the stem points downward, rightward, upward or leftward, respectively. Otherwise, output the point character "`.`". If there are multiple solutions, output any of those.

Examples

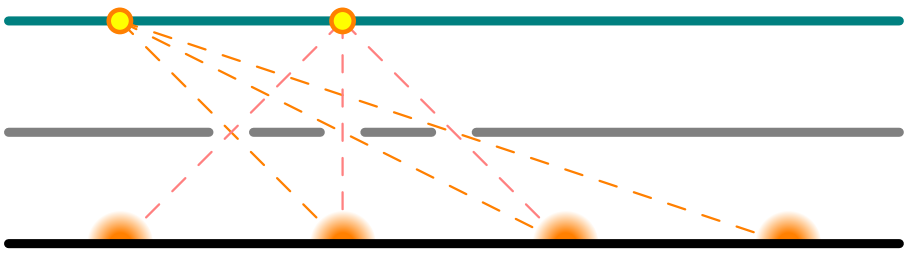
standard input	standard output
2	Yes .D. R.L .U.
3	No

Problem F. Pinholes

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are planning lighting in a warehouse room, which in the cross section consists of the ceiling (line $y = 2$) and the floor (line $y = 0$). The room is infinite and extends to left and right indefinitely. To illuminate the room, some lamps must be installed in the ceiling, where each lamp shines in all possible downwards directions.

The ceiling is ugly (you can see all of the pipes and wires), hence an aesthetic drop ceiling will be installed (at the line $y = 1$). To let the lamps shine through, n pinpoint holes (pinholes) were made at distinct positions. For each pinhole, each lamp will shine through in a single straight line. The result is that there are some points on the floor which light shines on.



For the desired lighting, it is necessary to illuminate m distinct points on the floor (such points may be illuminated by one or multiple lamps). No other points on the floor should be illuminated. Your task is, given the list of the coordinates of n pinholes and m points on the floor required to be illuminated, determine whether it is possible to install some lamps in the ceiling so that the required points (and only such points) are illuminated.

Input

The first line of input contains two integers n and m ($1 \le n, m \le 2\,000$), the numbers of pinholes and points to be illuminated. The second line contains n distinct integers $a_1, \ldots, a_n$ ($-10^8 \le a_i \le 10^8$) which mean that the i-th pinhole is located at the coordinates $(a_i, 1)$. The third line contains m integers $b_1, \ldots, b_m$ ($-10^8 \le a_i \le 10^8$) which mean that the i-th point on the floor that needs to be illuminated is located at the coordinates $(b_i, 0)$.

Output

If the required is impossible, output "`No`". Otherwise output "`Yes`" and an integer k ($1 \le k \le m$) in the next line, the number of lamps. In the third line output k distinct integers $c_1, \ldots, c_k$ ($-10^9 \le c_i \le 10^9$) which mean that the i-th lamp should be installed at the coordinates $(c_i, 2)$.

If there are multiple solutions, output any of those. It can be proven that if the solution exists, then there is a solution that uses at most m lamps.

Examples

standard input	standard output
3 4 0 -1 1 4 0 -2 2	Yes 2 -2 0
2 2 3 4 2 6	No

Problem G. Respect

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	256 megabytes

Let us call a subset S of $\{1, 2, \ldots, n\}$ *respectable* with respect to n if:

- there is a way to assign $+$ and $-$ signs to each element of S so that the their total sum is divisible by n;
- no non-empty proper subset of S is respectable with respect to n.

You are given n and a subset A of $\{1, 2, \ldots, n\}$. Your task is to find the total number of subsets of A that are respectable with respect to n.

For example, if $n = 7$ and $A = \{1, 2, 3, 5, 6\}$, the subset $\{2, 3, 5\}$ is respectable, as $+2 + 3 - 5 = 0$ is divisible by 7. The subset $\{1, 6\}$ is also respectable, as $+1 + 6 = 7$ is also divisible by 7. On the other hand, the subset $\{1, 2, 3, 6\}$ is not respectable, as even though $+1 + 2 + 3 - 6 = 0$ is divisible by 7, it contains the proper respectable subset $\{1, 6\}$.

Input

The first line contains two integers n and m ($1 \le m \le n \le 40$), where m is the size of A. The next line contains m distinct integers a_1, ..., a_m ($1 \le a_i \le n$), the elements of A.

Output

Output a single integer, the total number of subsets of A that are respectable with respect to n.

Example

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

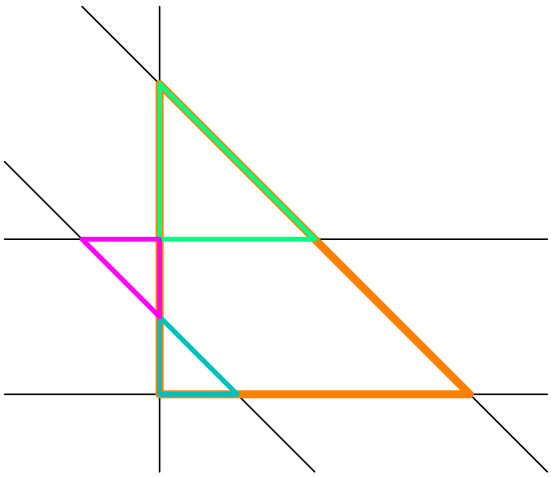
Note

A set P is called a *proper* subset of S if it is a subset of S and is not equal to S.

Problem H. Triangles

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.25 seconds
Memory limit:	256 megabytes

You are given n distinct lines in the plane. Each of them is either horizontal ($y = c$), vertical ($x = c$) or diagonal ($x + y = c$). You need to find the number of non-degenerate triangles such that each of their sides lies on some of the given lines. For example, there are 4 such triangles in the following configuration:



Input

The first line contains an integer n ($1 \le n \le 3\,000$), the number of lines. Each of the next n lines describes a single line as a character d and an integer c in the following way:

- d is equal to "`H`" if the line is horizontal; c is an integer such that the line is described by the equation $y = c$;
- d is equal to "`V`" if the line is vertical; c is an integer such that the line is described by the equation $x = c$;
- d is equal to "`D`" if the line is diagonal; c is an integer such that the line is described by the equation $x + y = c$;

In each of these cases $0 \le c \le 3\,000$. All of the given lines are guaranteed to be distinct.

Output

Output a single integer, the total number of non-degenerate triangles with sides on the given lines.

Examples

standard input	standard output
5 V 0 D 1 H 0 D 4 H 2	4
3 V 1 H 2 D 3	0

Problem I. Loginess

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given a directed graph with n vertices and m edges. The vertices are numbered with integers from 1 to n. Each vertex v is labeled with a positive integer a_v.

A path is a sequence of $\ell \geq 2$ distinct vertices v_1, v_2, ..., v_ℓ such that for each i from 1 to $\ell - 1$ there is an edge from v_i to v_{i+1}. The *loginess* of an edge from vertex u to vertex v is defined as $\log_{a_u}(a_v)$. The loginess of such a path is equal to the product of the loginess of the edges.

Your task is to find the maximum loginess over all possible paths in the graph!

Input

The first line contains two integers n and m ($2 \leq n \leq 2 \cdot 10^5$, $1 \leq m \leq 2 \cdot 10^5$), the numbers of vertices and edges in the graph. The next line contains n integers a_1, ..., a_n ($2 \leq a_v \leq 10^9$), the labels of the vertices. The next m lines contain the description of the edges. The i-th of these line contains two integers u_i and v_i, the endpoints of the i-th edge of the graph, meaning there is a directed edge from u_i to v_i.

The graph contains no self-loops or duplicate edges. Between any two vertices, there can be edges in both directions.

Output

Output a single real number, the maximum loginess over all paths in the graph. The relative or absolute error cannot exceed 10^{-6}.

Examples

standard input	standard output
3 4 10 20 30 2 1 3 1 2 3 3 2	1.13534758
4 2 10 30 20 10 2 3 4 1	1

Problem J. Arcology

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

You've recently moved to an arcology – a single megastructure city, which consists of an $n \times m$ rectangular grid of towers. The tower in the i-th row and the j-th column has height h_{ij} and consists of h_{ij} size $1 \times 1 \times 1$ blocks, one block at each level. But getting around in your new city has proven surprisingly challenging.

You can move in two ways:

- you can move a single block up or down in the tower you're currently in, this move costs you 1 stress as you're afraid of elevators;
- you can walk to any block at the same level as your current one in any tower adjacent to your current one, this doesn't cause you any stress.

To plan your commute, you want to answer queries of the following form. For each query, suppose you start at the top block of the tower at the position (i_a, j_a) and you need to get to the top block of the tower at (i_b, j_b). Calculate the smallest possible stress of a such a journey.

Input

The first line of input contains two integers n and m ($1 \le n \cdot m \le 10^6$), the size of the arcology. Then follow n lines of m integers h_{ij} ($1 \le h_{ij} \le 10^9$), where the j-th integer in the i-th row specifies the height of the tower at the same position in grid. The next line contains a single integer, the number of queries q ($1 \le q \le 5 \cdot 10^5$). Then follow q lines of four integers i_a, j_a, i_b, j_b ($1 \le i_a, i_b \le n$, $1 \le j_a, j_b \le m$). These represent a query where you start at top block of the tower at the position (i_a, j_a) and you need to get to the top block of the tower at (i_b, j_b).

Output

Output q lines. In the i-th line output a single integer, the smallest possible stress of a journey from start to finish for the i-th query.

Example

standard input	standard output
2 3 7 4 1 5 3 9 3 1 1 2 3 2 2 1 3 2 3 2 3	10 2 0

Problem K. Quadroku

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

In this problem you need to solve the *quadroku* puzzle. In quadroku, you are given a 4×4 square grid with digits between 1 to 4 written in some of the squares. To solve it, you need to write digits in the empty squares so that the following conditions hold:

- each row contains all the digits from 1 to 4;
- each column contains all the digits from 1 to 4;
- each 2×2 *block* with two sides on the sides of the grid contains all the digits from 1 to 4.

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

In this problem, you are initially given all digits in the upper left and lower right blocks such that each of those blocks contain all the digits from 1 to 4. Find the solution to the given quadroku or determine that it is impossible!

Input

The four input lines contain four integers each, the description of the initial grid. The upper left and the lower right blocks each contain all the digits from 1 to 4. The digits in the upper right and lower left blocks will always be 0 to denote that the squares are empty.

Output

If there exists a solution, first output "`Yes`" in the first line. In the next four lines output four integers in each, the solution of the puzzle. It can be shown that if the solution exists, it is unique.

If there is no solution, simply output "`No`".

Examples

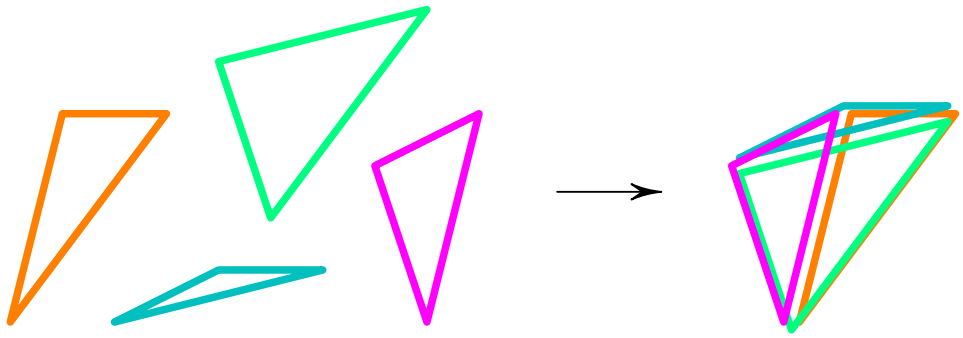
standard input	standard output
1 2 0 0 3 4 0 0 0 0 4 3 0 0 1 2	Yes 1 2 3 4 3 4 2 1 2 1 4 3 4 3 1 2
1 3 0 0 4 2 0 0 0 0 4 1 0 0 3 2	No

Problem L. Reconstruction

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

In this problem you need to reconstruct a convex polygon with n vertices on the plane given a description of all triangles with vertices coinciding with the vertices of the polygon. The triangles can be given in a translated form, but not rotated.

For example, the following 4 triangles can be reconstructed into the following rectangle:



Input

The first line of input contains an integer n ($3 \le n \le 50$), the number of vertices of the polygon. The next $\frac{n(n-1)(n-2)}{6}$ lines contain the description of the triangles. The i-th of these lines describes a single triangle with six integers x_1, y_1, x_2, y_2, x_3 and y_3 ($-10^5 \le x_i, y_i \le 10^5$). The points (x_1, y_1), (x_2, y_2) and (x_3, y_3) are the coordinates of its vertices.

Output

Output n lines, the coordinates of the polygon; in the i-th line, output two integers p_i, q_i ($-10^6 \le p_i, q_i \le 10^6$), with (p_i, q_i) being the coordinates of the i-th vertex of the polygon. You can output the vertices **in any order**. You can output the polygon in any position on the plane, but it must be a translation of the original polygon. It is guaranteed that the solution is unique (up to translation).

Example

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

Problem M. Hexagons

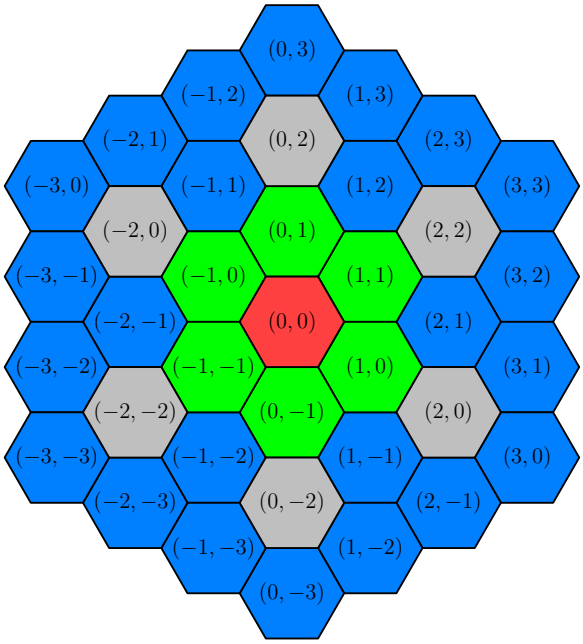
Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

The plane is partitioned into regular hexagons. A grasshopper is sitting in one of the hexagons (the origin). The grasshopper begins to jump around; at time i, it makes a jump covering i hexagons positioned in a straight line. That is, at time 1 it jumps 1 cell, at time 2 it jumps 2 cells, at time 3 it jumps 3 cells and so on.

Let the distance between two hexagons be the length of the shortest path between them that consists of hexagons and where each two adjacent hexagons are connected by an edge. There are two restrictions on the jumps of the grasshopper:

- the grasshopper always jumps so that the distance from the origin after the jump is larger that before;
- the grasshopper never jumps to a cell it could also have reached with fewer jumps.

Your task, given the coordinates of a goal hexagon, is to determine whether the grasshopper can ever reach it by performing the jumping described above. The coordinate system is defined so that the first coordinate defines the column, and the second defines the north-west to south-east diagonal as in the figure.



For example, the picture shows the cells reachable in at most 2 jumps. The red cell is the origin, the green cells are reachable with the first jump, the blue cells are reachable with the second jump, the gray cells are unreachable.

Input

The first line contains a single integer t ($1 \le t \le 10^5$), the number of test cases. Each of the next t lines contains two integers x and y ($-10^9 \le x, y \le 10^9$), the coordinates of the goal vertex.

Output

For each test case, output "`Yes`" if the given cell is reachable and "`No`" otherwise.

Example

standard input	standard output
7 0 -1 -2 0 1 2 -3 0 2 2 0 0 -1 -2	Yes No Yes Yes No Yes Yes