Problem A. Post-Increment

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

In the C++ programming language, a post-increment operation increases the value of an integer variable by 1. If the name of the variable is `x`, then a post-increment operation is applied using `x++`. However, the value of the expression `x++` is equal to the old value of `x` (the addition of 1 is performed "post"-operation).

You are given a valid C++ line which contains only characters "`x`" and "`+`", ending with a character "`;`". The task is to compute the value of the line. The only operations appearing in this line are the post-increment and sum operations. The sum operation is performed using `+` and takes two integer arguments (one to the left and one to the right of `+`). The resulting value is equal to the sum of the arguments. The only variable appearing in the line is `x`; its initial value is 0.

The operations are performed from left to right in a sequence. If a sum operation is applied to a variable with a post-increment operation afterwards, then the 1 is added to the variable after the sum operation.

To see the effect of the post-increment operation, we can consider a few examples. For example, the value of the line "`x++;`" is 0, as 1 is added only after the value of the line has been computed. For the line "`x+++x`", the resulting value is 1:

- first, `x++` is executed; the value of the expression `x++` is 0, but `x` is increased by 1;
- after the post-increment operation the value of the line becomes `0+x`; since the current value of `x` is 1, the value of the sum is 1;

Input

The only line of the input contains the described C++ line. The length of this line is between 2 and 10^4, inclusive. It is guaranteed that:

- all characters except the last one are either "`x`" or "`+`", and the last one is equal to "`;`";
- no two characters "`x`" are adjacent;
- no four characters "`+`" are adjacent;
- only the sum and post-increment operations appear in the line.

Output

Output a single integer, the value of the line.

Example

standard input	standard output
`x+++x+x++;`	2

Problem B. Tretris

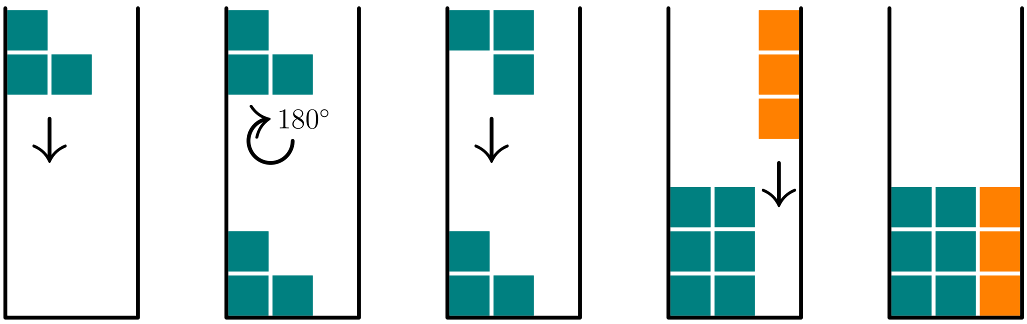
Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

After giving up on writing a Tetris solver, you've decided to start with the simpler game of Tretris. Unlike Tetris, in Tretris you are given only trominoes (either a 1×3 "I" piece or a corner "L" piece), the game board is only 3 columns wide (but infinitely tall) and completed lines do not get cleared.

Formally, you are given a sequence of n trominoes and need to choose how to rotate and drop them onto the game board one at a time so that they end up forming a $3 \times n$ rectangle. It is guaranteed that the given sequence of trominoes can form such a rectangle.

The initial orientation of the "I" and "L" pieces is shown below.

For example, if two "L" pieces are given first followed by a single "I" piece, they can be assembled into a 3×3 rectangle as follows.



Input

The only line of inputs contains a string of n characters "`I`" and "`L`" ($1 \leq n \leq 10^5$), with the i-th character denoting the shape of the i-th tromino to drop.

It is guaranteed that the given sequence of trominoes dropped in order can form a $3 \times n$ rectangle.

Output

For each tromino in order output two integers c and r denoting the position where it will be dropped. The first number c denotes the leftmost column the tromino occupies ($1 \leq c \leq 3$). The second number r denotes the number of 90° clockwise rotations performed on this tromino ($0 \leq r \leq 3$). If there are multiple solutions, output any of those.

Example

standard input	standard output

`LLI`	`1 0`
	`1 2`
	`3 0`

Problem C. Maximum Palindrome

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

You are given a string s of lowercase English characters, and q queries of the form l, r. For each such query, you need to find the length of the longest palindrome that is a substring of the string $s[l \dots r]$.

A palindrome is a string that reads the same backwards and forwards. For example, "`racecar`" and "`naan`" are palindromes, while "`car`" and "`nana`" are not.

Input

The first line of input contains the string s consisting of lowercase English alphabet letters ($1 \le |s| \le 10^5$). The next line contains one integer q, the number of queries ($1 \le q \le 10^5$). Each of the next q lines contains two integers l, r ($1 \le l \le r \le |s|$), the indices of the query substring.

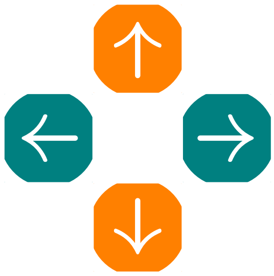
Example

standard input	standard output
abacaba 1 2 5	3

Problem D. DDR

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

You are playing a dancing game by performing a sequence of moves, where a move requires you to step on one of the four buttons with your foot. The buttons are labeled “`^`” (up), “`v`” (down), “`<`” (left) and “`>`” (right). During a single move, you can lift one of your feet and place it onto any button not occupied by your other foot. You are also allowed to place your foot back onto the button you just lifted it from.



In this problem, you are given a sequence of n buttons, and you need to find any sequence of moves that presses these buttons in the given order. Initially both of your feet are off the buttons.

Input

The first line contains a single integer n, the number of moves ($1 \le n \le 10^5$). The second line contains a string of length n consisting of the characters “`^`”, “`v`”, “`<`” and “`>`”.

Output

Output a string of length n consisting of the characters “`L`” and “`R`”. A character at the i-th position should be “`L`” if you use your left foot to perform the i-th move, or “`R`” if you use your right foot. If there are multiple solutions, output any of those.

Example

standard input	standard output
`7` `>>^v^<<`	`LLRRLRR`

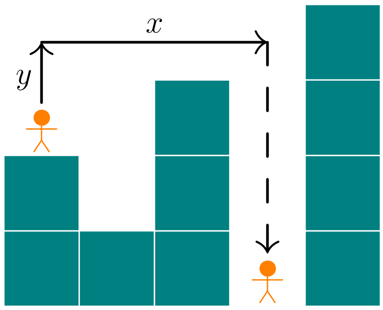
Problem E. Jetpack

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

You are traveling across a steep mountain ridge, which can be described as a sequence of n consecutive mountains with heights $a_1, \ldots, a_n$. The i-th mountain is a $1 \times a_i$ rectangle at coordinate i. You start at coordinate 0 and you wish to get to the coordinate $n + 1$.

You move using your jetpack, characterized by an integer parameter k. With a single burst, the jetpack can transport you y units straight up and then x units right (x and y being non-negative integers), provided that $x + y \leq k$. After each burst, you use your parachute to gently fall onto the mountain below you. Naturally, you cannot move through mountains. Note that you do not walk, but only use your jetpack to move around.

For example, if $k = 4$, then you can make a burst as depicted below, where $x = 1$ and $y = 3$.



Find the minimum number of bursts needed for your journey, or determine that it is impossible.

Input

The first line contains two integers n and k, the number of mountains and the parameter of the jetpack, respectively ($1 \leq n, k \leq 5 \cdot 10^5$). The next line contains n integers $a_1, \ldots, a_n$, the heights of the mountains ($0 \leq a_i \leq 10^9$).

Output

If the desired journey is not possible, output -1. Otherwise, output a single integer, the minimum number of bursts required for the journey.

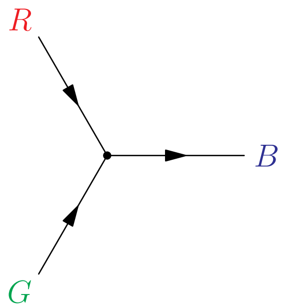
Examples

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

Problem F. Color Dynamics

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

You live in a strange universe where the smallest possible elements are particles of three types: red (R), green (G) and blue (B). It is known that when two particles of different colors interact, they annihilate and a particle of the third color is created. Formally, the particle system evolves by the rules $R + G \to B$, $G + B \to R$ and $B + R \to G$.



You are studying a system with r red, g green and b blue particles. Is it possible for it to evolve to a state with r', g' and b' particles with respective colors, evolving only according to the given rules?

Input

The input consists of multiple tests. The first line contains a single integer t, the number of tests ($1 \le t \le 10^5$).

Each of the tests is described by two lines. The first line contains three integers r, g and b, the numbers of red, green and blue particles in the initial state of the system, respectively ($0 \le r, g, b \le 10^9$). The next line contains three integers r', g' and b', the numbers of particles in the final state of the system ($0 \le r', g', b' \le 10^9$).

Output

For each of the tests, if the required evolution is possible, output "`Yes`", otherwise output "`No`" in a separate line.

Example

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

Problem G. Match Labyrinth

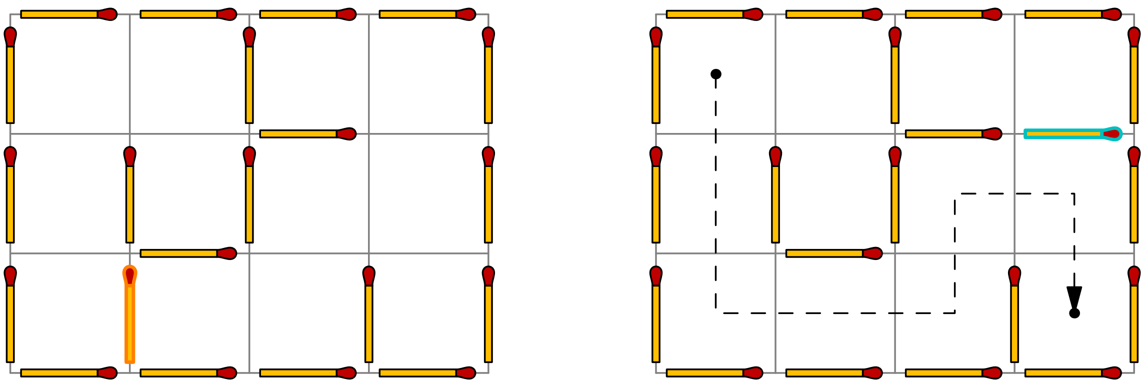
Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given a rectangular cell grid of size $n \times m$. Some of the borders of the cells have matches placed on them, thus forming a labyrinth. All of the borders on the perimeter of the grid have matches placed on them.

Now you are playing a game on this grid. You start with a token placed in the upper left cell of the labyrinth. You can then repeatedly move the token one cell in any of the four directions (up, down, left or right). However, you cannot move your token over any match. Your goal is to move the token to the bottom right cell of the labyrinth.

After some time, you have realized that such a journey is not possible in your labyrinth! In other words, there is no path connecting the starting and final cells that does not cross any match. However, you are a master of puzzles. You want to move exactly one match from its border to some free border on the grid so that there will be a path from the start to the finish. Your task is determine whether it is possible.

In the following example, one of the possible ways to free up a path is to move the highlighted match:



Note that this is not the only way to achieve the goal in this example.

Input

The first line contains two integers n and m, the dimensions of the grid ($2 \le n, m \le 1000$). Then $2n + 1$ lines follow describing the grid from top to bottom. The lines alternate between describing the matches on the horizontal borders and vertical borders of the current row of the grid. A line describing horizontal borders contains m characters which are either "`-`" or "`.`": "`-`" denotes a match at the corresponding cell border, and "`.`" means that there is no match there. Similarly, a line describing vertical borders contains $m + 1$ characters which are either "`|`" or "`.`": "`|`" denotes a match at the corresponding cell border, and "`.`" means that this border is free.

It is guaranteed that initially there is no path between the upper left cell and the bottom right cell that does not cross any of the matches. It is also guaranteed that all borders on the perimeter of the labyrinth are occupied by matches.

Output

If it is possible to move a single match to some free border and free a required path, output "`Yes`"; otherwise, output "`No`".

Examples

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>
3 4
 - - - -
| . | . |
 . . - .
| | | . |
 . - . .
| | . | |
 - - - -
</pre></td><td><pre>
Yes
</pre></td></tr>
<tr><td><pre>
2 2
 - -
| | |
 - -
| | |
 - -
</pre></td><td><pre>
No
</pre></td></tr>
</table>

Problem H. Reminders

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

You have n reminder stickers on your fridge. The i-th sticker reminds you of some task you need to perform no later than day d_i. Each task takes exactly one day to perform. Some of the stickers are glued over other stickers; formally, you have k stacks of stickers, where the k-th stack has s_k stickers glued over each other in sequence.

You wish to perform all of the n tasks in n consecutive days. You start at day 1. Then each day you remove a single sticker visible to you (that is, a sticker on top of some stack), revealing the next sticker of that stack, and perform the task of the removed sticker that day.

Find a schedule for performing all of the tasks in n days, or determine that it is impossible!

Input

The first line contains two integers n and k, the total number of stickers and the number of stacks, respectively ($1 \leq k \leq n \leq 10^5$). The i-th of the next k lines describes the i-th stack. The line starts with an integer s_i, the number of stickers in this stack ($1 \leq s_i \leq n$). Then s_i integers $d_1, \ldots, d_{s_i}$ follow; the number d_j is the last day when the j-th task from the top of the stack can be performed ($1 \leq d_j \leq n$). It is guaranteed that $s_1 + \ldots + s_k = n$.

Output

If the required schedule does not exist, simply output "`No`". Otherwise in the first line output "`Yes`" and in the next line output a permutation of the numbers from 1 to n, the corresponding schedule. The stickers are numbered from 1 to n as their corresponding deadline appears in the input. If there are multiple solutions, output any of those.

Examples

standard input	standard output
7 3 3 3 6 5 2 5 7 2 1 7	Yes 6 4 1 2 3 7 5
5 1 5 5 2 4 5 1	No

Problem I. Product Sum

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

A sum of n numbers $x_1 + \ldots + x_n$ can be abbreviated using the notation $\sum_{i=1}^{n} x_i$. In this problem, you are given an expression which consists of k sum signs $\sum_{i_j=1}^{n}$ and k terms a_{i_j}. The overall expression is formed by writing them in some order. The only condition is that for any j, the term a_{i_j} must appear after $\sum_{i_j=1}^{n}$. If multiple a_{i_j} follow each other, they are multiplied together. The following is an example of such an expression for $k = 4$:

$$\sum_{i_1=1}^{n} a_{i_1} \sum_{i_2=1}^{n} a_{i_2} \sum_{i_3=1}^{n} \sum_{i_4=1}^{n} a_{i_4} a_{i_3}.$$

Your task is to find the value of the given expression, given the values of the elements $a_1, \ldots, a_n$. Since this value can be large, output it modulo $10^9 + 7$.

Input

The first line contains two integers n and k, the number of elements a_i and the number of sum signs in the expression ($1 \le n, k \le 10^5$). The second line contains n integers $a_1, \ldots, a_n$ ($1 \le a_i \le 10^4$).

The next line contains $2k$ space-separated tokens and describes the expression. A "+" token denotes a sum sign. The index of the j-th sum sign in this line is equal to i_j. An integer token with value j denotes the term a_j. It is guaranteed that all of the integers from 1 to k appear in this line exactly once, and each integer j appears after the j-th sum sign.

Output

Output a single integer, the value of the expression modulo $10^9 + 7$.

Examples

standard input	standard output
3 4 1 3 2 + 1 + 2 + + 4 3	1296
1 3 10000 + + + 1 2 3	999993007

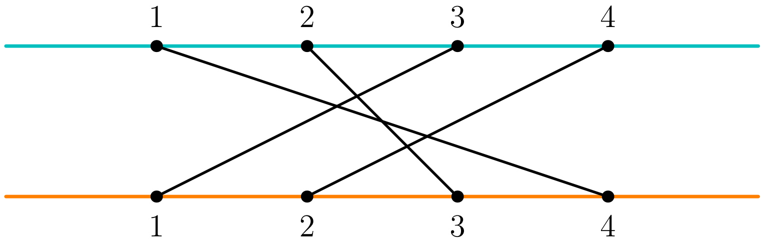
Problem J. Segment Crossings

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

You are given two parallel lines with n points selected on each of the lines, numbered by 1, 2, ..., n in order on each line. You need to draw n segments such that the endpoints of each segment are two of the given points on separate lines, and each of the points is an endpoint of exactly one of these segments.

Find a way to draw the segments in such a way that the total number of pairs of segments that intersect is exactly k, or determine that this is not possible!

For example, there are 5 pairs of segments that intersect in the following arrangement for $n = 4$:



Note that in this problem we are interested in the number of segment pairs that intersect, not in the number of distinct intersection points (which can happen to coincide for different pairs of segments).

Input

The input contains two integers n and k, the number of points on each line and the required number of segment pairs that intersect ($1 \le n \le 10^5$, $0 \le k \le 10^{18}$).

Output

If the required arrangement is not possible, simply output "`No`". Otherwise first output "`Yes`" and then output n lines. Each of these lines must contain two integers a_i and b_i describing the endpoints of the i-th segment on the first and second lines, respectively ($1 \le a_i, b_i \le n$). Each integer from 1 to n should appear once throughout a_i and once throughout b_i.

If there are multiple solutions, output any of those. You can output the description of the segments in any order.

Examples

standard input	standard output

`4 5`	`Yes`
	`2 3`
	`1 4`
	`4 2`
	`3 1`

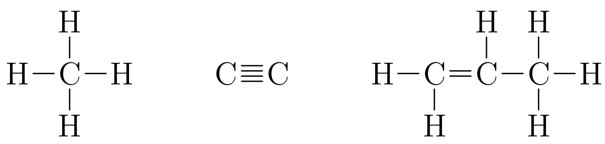
`4 32`	`No`

Problem K. Hydrocarbons

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

In this problem you will investigate molecules consisting only of hydrogen (H) and carbon (C), known as hydrocarbons. A molecule is formed when atoms share their covalent bonds. The number of covalent bonds that an atom forms is called its covalence; for hydrogen, it is equal to 1; for carbon it is 4. Any two atoms in the molecule can share any number of bonds, but the total number of bonds the atom shares with all the atoms in the molecule is exactly its covalence. An atom cannot form covalent bonds with itself. A molecule is connected, meaning it does not consist of separate parts not connected by bonds.

For example, the following structures are hydrocarbon molecules:



Given the number of hydrogen and carbon atoms, determine whether it is possible to construct a valid hydrocarbon molecule using all of these atoms!

Input

The input contains two integers h and c ($0 \leq h, c \leq 10^9$, $h + c \geq 2$), the numbers of hydrogen and carbon atoms, respectively.

Output

If a valid hydrocarbon molecule with the given numbers of atoms exists, output "`Yes`", otherwise output "`No`".

Examples

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

Problem L. Spam Control

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

You work in a company with n employees. The company has a subordinate structure described by a rooted tree with n nodes, each representing an employee. The nodes are numbered with integers from 1 to n, and the root is the vertex with number 1.

Now k of the employees have to be chosen to be managers. A person can be a manager only if all of their superiors are also managers. If a person is chosen to be a manager, they will send an e-mail to all their subordinates in the tree (including themselves) notifying the people below about their new managing position.

You need to find the smallest total number of e-mails that will be sent if the k managers are chosen optimally. Moreover, you need to do this for all values of k from 1 to n.

Input

The first line contains a single integer n, the number of nodes ($1 \le n \le 10^4$). Then $n-1$ lines follow describing the tree. The i-th line contains two integers a_i and b_i, the numbers of the endpoints of the i-th edge ($1 \le a_i, b_i \le n$, $a_i \ne b_i$). It is guaranteed that the given graph is a tree.

Output

Output n integers $s_1, \ldots, s_n$, where s_k is the smallest number of e-mails that will be sent if k managers are chosen optimally.

Example

standard input	standard output
4 1 2 2 4 3 2	4 7 8 9

Note

A graph is a set of objects with some pairs of objects connected by links, where objects are called *vertices* and links are called *edges*. A tree is a graph such that there is a unique path of edges connecting any two vertices. A *rooted* tree is a tree where one vertex r is chosen to be the *root*.

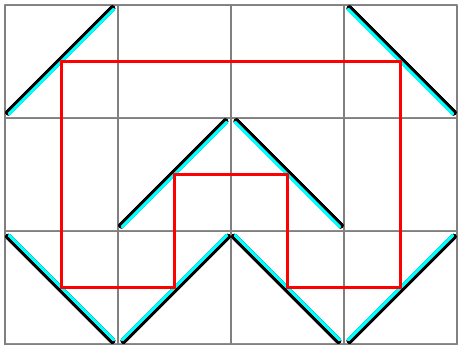
Problem M. Laser Maze

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

You are setting up an experiment involving light and mirrors. Today, you are given an $n \times m$ cell grid, which is initially empty. You can place mirrors in the cells; each cell can contain at most one mirror. Each mirror is positioned at a 45° angle; therefore, horizontal light gets reflected into vertical light and vice versa. The mirrors are one-sided, meaning that light hitting the back side of a mirror gets absorbed.

After placing the mirrors in some cells, you shine a laser beam in the grid. You do that by choosing any cell in the grid and shine the laser from the center of that cell in any of the four directions (up, down, left or right). Then the laser travels through the grid, reflecting off the mirrors. Your task is to place the mirrors in such a way that the beam visits every cell of the grid at least once and returns to its original starting cell traveling in the same direction as in the beginning.

For example, the following is such a placement of mirrors in the 3×4 grid:



Find any such placement of mirrors or determine that it is impossible!

Input

The input contains two integers n and m, the dimensions of the grid ($1 \leq n, m \leq 1000$).

Output

If the required placement of mirrors is not possible, simply output "`No`". Otherwise first output "`Yes`", and then output n lines of m digits each. The i-th of these lines should describe the i-th row of the grid. The j-th digit in this line describes the j-th cell of the i-th row:

- 0, if the cell remains empty;
- 1, if the mirror in this cell is facing north-east;
- 2, if the mirror in this cell is facing south-east;
- 3, if the mirror in this cell is facing south-west;
- 4, if the mirror in this cell is facing north-west.

If there are multiple solutions, output any of those.

Examples

standard input	standard output
3 4	Yes 2003 0230 1414
6 1	No