Problem J. Jimi Hendrix

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

You are given a tree T consisting of n vertices and $n-1$ edges. Each edge of the tree is associated with a lowercase English letter c_i.

You are given a string s consisting of lowercase English letters. Your task is to find a simple path in the tree such that the string formed by concatenation of letters associated with edges of this path contains string s as a subsequence, or determine that there exists no such simple path.

Input

The first line of input contains two positive integers n and m ($2 \le n \le 5 \cdot 10^5$, $1 \le m \le n-1$), the number of vertices in the tree and the length of the string s.

The following $n-1$ lines contain triples u_i, v_i, c_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$, c_i is a lowercase English letter), denoting an edge (u_i, v_i) associated with letter c_i.

The last line contains a string s ($|s| = m$) consisting of lowercase English letters.

Output

If the desired path exists, output its endpoints a and b. Otherwise, output “`-1 -1`”. If there are several possible answers, you are allowed to output any of them.

Example

standard input	standard output
9 3 1 2 a 2 3 b 2 4 a 4 5 b 4 6 c 6 7 d 6 8 a 8 9 b acb	8 3

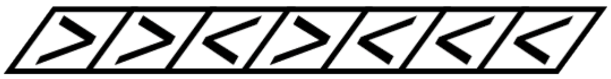
Problem D. Linear Gimmick

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 mebibytes

You are in front of a linear gimmick of a game. It consists of N panels in a row, and each of them displays a right or a left arrow.

You can step in this gimmick from any panel. Once you get on a panel, you are forced to move following the direction of the arrow shown on the panel and the panel will be removed immediately. You keep moving in the same direction until you get on another panel, and if you reach a panel, you turn in (or keep) the direction of the arrow on the panel. The panels you passed are also removed. You repeat this procedure, and when there is no panel in your current direction, you get out of the gimmick.

For example, when the gimmick is the following image



and you first get on the 2nd panel from the left, your moves are as follows.

- Move right and remove the 2nd panel.
- Move left and remove the 3rd panel.
- Move right and remove the 1st panel.
- Move right and remove the 4th panel.
- Move left and remove the 5th panel.
- Get out of the gimmick.

You are given a gimmick with N panels. Compute the maximum number of removed panels after you get out of the gimmick.

Input

The input consists of two lines. The first line contains an integer N ($1 \leq N \leq 10^5$) which represents the number of the panels in the gimmick. The second line contains a character string S of length N, which consists of '>' or '<'. The i-th character of S corresponds to the direction of the arrow on the i-th panel from the left. '<' and '>' denote left and right directions respectively.

Output

Output the maximum number of removed panels after you get out of the gimmick.

Example

standard input	standard output
7 `>><><<<`	7
5 `>><<<`	5
6 `><<><<`	6
7 `<<><<>>`	5

Problem I. Archaeological Research

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

Professor Tupids has found a mysterious manuscript during his recent archaeological expedition.

The manuscript, in fact, looks like a series of symbols with meaning yet to be discovered. In order to simplify studying of the manuscript, professor Tupids created an alphabet of all symbols occurred (let's denote their number as c) and replaced each symbol with its position in the alphabet (positions are numbered from one). So, the manuscript became represented as a list of n integers from the segment $[1; c]$.

Professor's intuition told him that the key to understanding the manuscript is to find some regularities in locations of the symbols. So he wrote down a huge table with n rows and c columns, where cell at i-th row and j-th column contained the position of the next occurrence of symbol number j after the position i (the cell was left empty if there were no appropriate occurrences).

But then a disaster happened: a fire in the laboratory completely destroyed the manuscript! Fortunately, the table built by professor was salvaged, though it was damaged to some extent. Not only some of the cells were lost in the fire, but, thanks to the careless assistants, cells in each of the rows were reordered arbitrarily!

Professor Tupids doesn't want to lose face in the scientific community, so he asks you to help him with restoring the original manuscript, given the remaining information from the table. As there may be infinitely many different solutions (even the size c of the alphabet was lost!), professor wants you to restore the lexicographically smallest solution. It is not guaranteed that the solution exists, though, as the table could have been completely spoiled by the assistants.

Input

The first line contains a single integer n ($1 \le n \le 3 \cdot 10^5$), the length of the original manuscript. After that, n lines follow, the i-th of which contains an integer c_i ($0 \le c_i \le n-i$) followed by c_i distinct integers from the segment $[i+1; n]$: the contents of the survived non-empty cells of the i-th row of the table in arbitrary order.

It is guaranteed that the sum of all c_i ($1 \le i \le n$) does not exceed $3 \cdot 10^5$.

Output

If the solution exists, print a single line with n positive integers: the representation of the lexicographically smallest manuscript. Otherwise, print "`No solution`" (without quotes).

Examples

standard input	standard output

4
3 2 3 4
2 4 3
1 4
0
	1 1 2 3

5
1 2
1 4
1 4
1 5
0
	1 1 1 2 1

Problem K. Korn

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

Consider a connected graph $G = (V, E)$ without loops and parallel edges. Let's define a *complete walk* starting in vertex v as a sequence of visited vertices $v = p_0, p_1, p_2, \ldots, p_k = u$ such that the following conditions are satisfied:

1. For each $i = 1, 2, \ldots, k$, the unordered pair $(p_i, p_{i-1}) \in E$, that is, each two consecutive vertices are connected by an edge.
2. Each edge (a, b) appears at most once among all (p_i, p_{i-1}), that is, the walk p doesn't pass through the same edge twice.
3. There exists no vertex p_{k+1} that can be appended at the end of the walk such that the previous two conditions are still satisfied.

The vertex u is called the *terminal* vertex.

The vertex v is called *unavoidable* if any complete walk starting in v visits all edges in the graph (that is, $k = |E|$), and its terminal vertex is also v.

Your task is to find all *unavoidable* vertices in a given graph.

Input

The first line of input contains two integers n and m ($3 \le n \le 2 \cdot 10^5$, $n - 1 \le m \le 5 \cdot 10^5$), the number of vertices and the number of edges in the graph respectively.

The following m lines contain pairs of integers a_i, b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$), denoting endpoints of i-th edge.

It is guaranteed that the graph contains no loops and no parallel edges, and also that it is connected.

Output

Print the number of *unavoidable* vertices on the first line of output, and 1-based indices of all *unavoidable* vertices on the second line in ascending order.

Example

standard input	standard output

6 8
3 5
5 1
3 4
4 1
6 3
1 6
2 3
1 2
	2
	1 3

Note

In the sample, for example, vertex 4 is not unavoidable because there exists a complete walk 4, 5, 2, 1, 4 that terminates in 4 but that doesn't visit all edges in the graph.