

Problem A. Treasure Island

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

Pirate John Silver has found a map depicting exactly one island in a sea. The map is a piece of cloth divided into cells: n cells in height and m cells in width. John Silver knows that every cell denotes either land or water, but some of the cells are erased, and now it's absolutely impossible to say what these cells represent.

Help John Silver to restore the map of the island. An island is a non-empty set of land cells connected in four directions (up, down, left and right).

Input

The first line contains two integers n and m ($1 \leq n, m \leq 50$) — the sizes of the map.

Each of the next n lines contains m characters and describes the map. A character is equal to «#» if this cell is a water cell, «.» if it's a land cell, and «?» if this cell is erased.

It's guaranteed that the input contains at least one character «.» and at least one character «?».

Output

If it's not possible to restore the map so that it would depict exactly one island, output «Impossible».

If the map can be restored in a unique way, output n lines of m characters in the same format as they are in the input, but replacing «?» with «.» or «#».

And if there are several correct ways to restore the map, output «Ambiguous».

Examples

standard input	standard output
5 7 ##### #.#.# #..?..# #.#.# #####	##### #.#.# #....# #.#.# #####
5 7 ##### #...#.# #???.# #.#...# #####	Ambiguous
5 7 ##### #.#.#.# #.#?#.# #.#.#.# #####	Impossible

Problem B. Derangement

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

A permutation of n numbers is a sequence of integers from 1 to n where each number is occurred exactly once. If a permutation p_1, p_2, \dots, p_n has an index i such that $p_i = i$, this index is called a fixed point.

A derangement is a permutation without any fixed points.

Let's denote the operation $swap(a, b)$ as swapping elements on positions a and b .

For the given permutation find the minimal number of $swap$ operations needed to turn it into derangement.

Input

The first line contains an integer n ($2 \leq n \leq 200000$) — the number of elements in a permutation.

The second line contains the elements of the permutation — n distinct integers from 1 to n .

Output

In the first line output a single integer k — the minimal number of $swap$ operations needed to transform the permutation into derangement.

In each of the next k lines output two integers a_i and b_i ($1 \leq a_i, b_i \leq n$) — the arguments of $swap$ operations.

If there are multiple possible solutions, output any of them.

Examples

standard input	standard output
6	1
6 2 4 3 5 1	2 5

Problem C. Triangles

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

There is a set of n segments with the lengths l_i . Find a segment with an integer length so that it could form a non-degenerate triangle with any two segments from the set, or tell that such segment doesn't exist.

Input

The first line contains a single integer n ($2 \leq n \leq 200000$) — the number of segments in the set.

The second line contains n integers l_i separated by spaces ($1 \leq l_i \leq 10^9$) — the lengths of the segments in the set.

Output

If the required segment exists, in the first line output «YES» (without quotes). In this case in the second line output a single integer x — the length of the needed segment. If there are many such segments, output any of them.

If the required segment doesn't exist, output «NO» (without quotes).

Examples

standard input	standard output
2 3 4	YES 2
3 3 4 8	YES 6
3 3 4 9	NO

Problem D. Laying Cables

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

One-dimensional country has n cities, the i -th of which is located at the point x_i and has population p_i , and all x_i , as well as all p_i , are distinct. When one-dimensional country got the Internet, it was decided to place the main server in the largest city, and to connect any other city j to the city k that has bigger population than j and is the closest to it (if there are many such cities, the largest one should be chosen). City k is called a parent of city j in this case.

Unfortunately, the Ministry of Communications got stuck in determining from where and to where the Internet cables should be laid, and the population of the country is suffering. So you should solve the problem. For every city, find its parent city.

Input

The first line contains a single integer n ($1 \leq n \leq 200000$) — the number of cities.

Each of the next n lines contains two space-separated integers x_i and p_i ($1 \leq x_i, p_i \leq 10^9$) — the coordinate and the population of the i -th city.

Output

Output n space-separated integers. The i -th number should be the parent of the i -th city, or -1 , if the i -th city doesn't have a parent. The cities are numbered from 1 by their order in the input.

Examples

standard input	standard output
4 1 1000 7 10 9 1 12 100	-1 4 2 1
3 1 100 2 1 3 10	-1 1 1
3 1 10 3 100 2 1	2 -1 2

Problem E. Bisection

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

Pavel has developed the best game in the world — «Bisection». In this game the square grid of size $n \times n$ is given to a player. Some of the cells on the grid are selected and form a figure. A player should paint every cell of this figure in one of two colors so that the resulting figures of both colors would be equal. The figures are equal if one of them could be transformed into the other by translations, reflections relative to horizontal and vertical axes and turns by right angles.

Pavel has created an extra level for this game, but he can't understand if it has any solution. You have to find it out.

Input

The first line contains a single integer n ($1 \leq n \leq 50$) — the size of the grid.

Each of the next n lines contains n characters. The character is «.» if this cell doesn't belong to the figure, and «#» if it does.

It's guaranteed that the input contains positive even number of «#» characters.

Output

If there is no solution, output «NO» (without quotes).

Otherwise in the first line output «YES» (without quotes), and then output n lines of n characters describing the coloring of the figure. Characters «.» from the input shouldn't be touched, and characters «#» should be replaced with «A» or «B», depending on the color that should be used to paint this cell.

If there are many possible solutions, output any of them.

Examples

standard input	standard output
3 .## ### ###	YES .AA BAB BBA
3 ### .## ###	NO

Problem F. Two Points

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

There are two points (x_1, y_1) and (x_2, y_2) on the plane. They move with the velocities (v_{x_1}, v_{y_1}) and (v_{x_2}, v_{y_2}) . Find the minimal distance between them ever in future.

Input

The first line contains four space-separated integers x_1, y_1, x_2, y_2 ($-10^4 \leq x_1, y_1, x_2, y_2 \leq 10^4$) — the coordinates of the points.

The second line contains four space-separated integers $v_{x_1}, v_{y_1}, v_{x_2}, v_{y_2}$ ($-10^4 \leq v_{x_1}, v_{y_1}, v_{x_2}, v_{y_2} \leq 10^4$) — the velocities of the points.

Output

Output a real number d — the minimal distance between the points. Absolute or relative error of the answer should be less than 10^{-6} .

Examples

standard input	standard output
1 1 2 2 0 0 -1 0	1.0000000000000000
1 1 2 2 0 0 1 0	1.414213562373095

Problem G. Repair

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

Alex is repairing his country house. He has a rectangular metal sheet of size $a \times b$. He has to cut two rectangular sheets of sizes $a_1 \times b_1$ and $a_2 \times b_2$ from it. All cuts must be parallel to the sides of the initial sheet. Determine if he can do it.

Input

The first line contains two space-separated integers a and b ($1 \leq a, b \leq 10^9$) — the sizes of the initial sheet.

The second line contains two space-separated integers a_1 and b_1 ($1 \leq a_1, b_1 \leq 10^9$) — the sizes of the first sheet to cut out.

The third line contains two space-separated integers a_2 and b_2 ($1 \leq a_2, b_2 \leq 10^9$) — the sizes of the second sheet to cut out.

Output

Output «YES» (without quotes), if it's possible to cut two described sheets from the initial sheet, or «NO» (without quotes), if it's not possible.

Examples

standard input	standard output
12 20 14 7 5 6	YES
12 20 11 9 20 7	NO

Problem H. Pavel's Party

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

Pavel is going to have a party, and he wants to invite exactly k people.

Pavel has n friends and he has already decided in what order he is going to call and invite them. When Pavel calls the i -th friend, he tells he will come if from a_i to b_i people come to the party.

Once the required number of people is assembled, Pavel invites them and the party is arranged. Pavel won't call the rest friends.

For every $k = 1, \dots, n$ find the number of people whom Pavel will call.

Input

The first line contains a single integer n ($1 \leq n \leq 200000$) — the number of Pavel's friends.

Each of the next n lines contains two integers a_i and b_i ($1 \leq a_i \leq b_i \leq n$) — the lower and the upper bound of the invited people required for the i -th Pavel's friend to come.

The data is listed in the order Pavel will get to know it.

Output

Output n space-separated integers. The k -th number should be equal to the number of friends called by Pavel if he wants to invite exactly k people to the party. If for some k the party cannot be arranged, output -1 for this k .

Examples

standard input	standard output
6 3 3 1 2 3 6 3 4 1 4 4 6	2 5 4 6 -1 -1
3 3 3 1 3 3 3	2 -1 3

Problem I. Deadline

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

Alex works as a developer, and he has hard times now. He has to complete n tasks. Every task i has the latest time d_i when it can be completed (starting from the moment of time 0), the time c_i needed to complete it, and the tasks that must be completed before task i to get the possibility to start it. We'll say that the i -th task depends on these tasks.

What should be the order of doing tasks to complete all of them in time?

Input

The first line contains a single integer n ($1 \leq n \leq 200000$) — the number of tasks.

Next n lines describe tasks. The i -th line starts with three space-separated integers d_i , c_i and r_i ($1 \leq d_i, c_i \leq 10^9$, $0 \leq r_i \leq n - 1$) — the latest time to complete the i -th task, the time needed to complete it and the number of tasks which it depends on. Then in the same line r_i space-separated integers — the numbers of these tasks — are written. It's guaranteed that there is no number i among these r_i numbers.

The tasks are numbered from 1 in order of their appearance in the input. The sum of all r_i in the input doesn't exceed 200000.

Output

In the first line output «YES» (without quotes) if Alex will be able to complete all the tasks without exceeding any deadlines, or «NO» (without quotes) otherwise.

If the answer is «YES», in the second line output n space-separated integers — the numbers of tasks in the order they must be done to fit into all deadlines. If there are many solutions, output any of them.

Examples

standard input	standard output
3 10 5 0 2 2 0 5 3 0	YES 2 3 1
2 100 1 0 10 10 1 1	NO
2 100 1 1 2 200 2 1 1	NO

Problem J. Robots at Warehouse

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

Vitaly works at the warehouse. The warehouse can be represented as a grid of $n \times m$ cells, each of which either is free or is occupied by a container. From every free cell it's possible to reach every other free cell by moving only through the cells sharing a side. Besides that, there are two robots in the warehouse. The robots are located in different free cells.

Vitaly wants to swap the robots. Robots can move only through free cells sharing a side, moreover, they can't be in the same cell at the same time or move through each other. Find out if the swap can be done.

Input

The first line contains two positive integers n and m ($2 \leq n \cdot m \leq 200000$) — the sizes of the warehouse.

Each of the next n lines contains m characters. The j -th character of the i -th line is «.» if the corresponding cell is free, «#» if there is a container on it, «1» if it's occupied by the first robot, and «2» if it's occupied by the second robot. The characters «1» and «2» appear exactly once in these lines.

Output

Output «YES» (without quotes) if the robots can be swapped, and «NO» (without quotes) if that can't be done.

Examples

standard input	standard output
5 3 ### #1# #.# #2# ###	NO
3 5 #...# #1.2# #####	YES

Problem K. Palindromization

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

Mihahim has a string s . He wants to delete exactly one character from it so that the resulting string would be a palindrome. Determine if he can do it, and if he can, what character should be deleted.

Input

The input contains a string s of length ($2 \leq |s| \leq 200000$), consisting of lowercase Latin letters.

Output

If the solution exists, output «YES» (without quotes) in the first line. Then in the second line output a single integer x — the number of the character that should be removed from s so that the resulting string would be a palindrome. The characters in the string are numbered from 1. If there are several possible solutions, output any of them.

If the solution doesn't exist, output «NO» (without quotes).

Examples

standard input	standard output
evertree	YES 2
emerald	NO
aa	YES 2

Problem L. Chess Match

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

Two best chess teams in the world are preparing to the match against each other. Both teams have n players. Each player has a rating, and the player with the higher rating always wins. Ratings of the players in the first team are a_1, a_2, \dots, a_n , and in the second team — b_1, b_2, \dots, b_n . There are no players with equal ratings in both teams.

The chess match between two teams consists of n games, called the games on the first, second, ..., n -th board. Before the match captains of both teams must decide which player will play on which board. When they do that, they don't know the opponent's choice.

After that a single game is played on each board. The team with the higher score wins.

For both teams, determine if it can win the match.

Input

The first line contains a single integer n ($1 \leq n \leq 200000$) — the number of players in each team.

The second line contains n space-separated integers a_i ($1 \leq a_i \leq 10^9$) — the ratings of the players in the first team.

The third line contains n space-separated integers b_i ($1 \leq b_i \leq 10^9$) — the ratings of the players in the second team.

All a_i and b_i are pairwise distinct.

Output

Output «First» if only first team can win, «Second» if only second team can win, «Both» if both teams can win, and «None» if there will be a draw in any case. Don't output quotes.

Examples

standard input	standard output
5 1 3 5 7 9 2 4 6 8 10	Both
5 1 2 3 4 5 6 7 8 9 10	Second
4 9001 9002 1 3 8 6 4 2	First
2 1 1000000000 10000 100000	None

Problem M. Decomposition into Good Strings

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

Sometimes, in string problems you have to find a string satisfying some conditions. The authors again were too lazy to think of a tricky term for such string, so they called it *good*.

A string is called good if it contains exactly k different characters. You are given a string s . Find the minimal number of good strings so that their concatenation is s .

Input

The first line contains an integer k ($1 \leq k \leq 26$).

The second line contains a string s ($1 \leq |s| \leq 200000$), consisting of lowercase Latin letters.

Output

For every prefix of s , starting from the prefix consisting of the first character of s , and ending with s itself, output the minimal number of good strings so that their concatenation is this prefix. Solutions will be tested more carefully this way, won't they?

If for some prefix the decomposition into good strings isn't possible, output «-1» for it.

Examples

standard input	standard output
2 abacaba	-1 1 1 2 2 3 3