Problem A. Streets of Working Lanterns - 2

Time limit: 2 seconds
Memory limit: 256 megabytes

Policeman Anatoliy again monitors a lair of unorganized criminal group spreading prohibited Asian drawings. Presently the criminals are also sharing wireless Internet which can be used anonymously by whoever wants to. The lair still has only one entrance, which is also an exit. When someone enters into the lair, Anatoliy writes an opening round bracket in his notepad, and when someone comes out, he writes a closing round bracket.

Anatoliy decided to refrain from eating donuts in order not to spoil the records just like the previous time, but nevertheless, when he tore a sheet out of a notepad to file it to a criminal case, accidentally tore it into pieces. He doesn't want his boss to shout on him, so he must restore his records by connecting pieces in the right order. It's good for him that the layout of notepad sheets allows to determine where top and bottom sides of these pieces are. Anatoliy ensured that the lair of criminals was empty before he started the surveillance and after he ended it.

Input

The first line contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the number of pieces of paper with opening and closing round brackets written on them.

The next n lines contain only characters «`(`» and «`)`» — these are the strings written on pieces. The total number of brackets does not exceed $2 \cdot 10^5$.

Output

If the given pieces can be connected so that the resulting string does not contradict the statement, output in the first line «`YES`» without quotes. In the second line output n integers separated by spaces — the numbers of pieces in the order they must be connected. The pieces are numbered from one in the order they are mentioned in the input.

If Anatoliy messed up something and the given pieces cannot form the correct record, output in the only line «`NO`» without quotes.

Examples

standard input	standard output
3 (())	YES 1 2 3
3))) (((	YES 2 3 1
3))) (( )	NO

Problem B. Pursuing the Happiness

Time limit: 2 seconds
Memory limit: 256 megabytes

Mike wants to find a substring «`happiness`» in the string s, but Constantine cannot allow this and decided to hinder him. He is planning to swap two characters on two different positions in the string s so that Mike wouldn't be able to find what he looks for. Which two characters Constantine should swap?

Input

The only line contains from 2 to $2 \cdot 10^5$ lowercase Latin letters — the string s, in which Mike wants to find a substring «`happiness`».

Output

If Constantine succeeds in achieving his goal, in the first line output «`YES`» without quotes. In the second line output two distinct integers separated by a space — the positions of characters in the string s, which Constantine should swap. Positions in the string are numbered from one. If there are several possible answers, output any of them.

If for any choice of Constantine Mike still would be able to find a substring «`happiness`», in the only line output «`NO`» without quotes.

Examples

standard input
pursuingthehappiness
standard output
YES 15 18

standard input
happinessformehappinessforyouhappinessforeverybodyfreeandletnoonebeleftbehind
standard output
NO

Problem C. Urn with Balls

Time limit: 2 seconds
Memory limit: 256 megabytes

Megabrain has usual problems with the occupiers: he is captured again and is forced to solve logical puzzles about urns with balls. In the urn staying in front of Megabrain there are a red balls, b green balls and also c balls, colors of which are unknown to Megabrain. The occupiers demand an answer to a question: what is the maximal number of balls one can take out of this urn, so that there would be no more than n red and no more than m green balls among them for sure?

Input

The first line contains three integers a, b and c separated by a space ($0 \le a, b, c \le 10^9$) — the number of red balls in the urn, the number of green balls in the urn, and the number of balls of unknown color in the urn, correspondingly.

The second line contains two integers n and m separated by a space ($0 \le n, m \le 10^9$) — the maximal number of red balls allowed to be taken out of the urn and the maximal number of green balls allowed to be taken out of the urn, correspondingly.

Output

Output a single integer — the maximal number of balls that can be taken out of the urn so that no restrictions are violated.

Examples

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

Problem D. Jumps

Time limit: 2 seconds
Memory limit: 256 megabytes

A frog lives in a one-dimensional world in the point with the coordinate 0. He needs to get to the point with the coordinate x. For some reason he cannot make jumps of arbitrary length, and can jump only by $a_1, \ldots, a_n$ in any direction. Is he able to reach x?

Input

The first line contains two integers n and x separated by a space ($1 \le n \le 200000$, $-10^9 \le x \le 10^9$) — the number of variants of jump length and the coordinate of the point to reach.

The second line contains n integers a_i separated by spaces ($1 \le a_i \le 10^9$) — the lengths of jumps the frog can make.

Output

Output «YES» (without quotes), if the frog can reach the point x, otherwise output «NO» (without quotes).

Examples

standard input	standard output
3 17 3 5 4	YES
4 5 10 20 30 40	NO

Problem E. Bonuses and Teleports

Time limit: 2 seconds
Memory limit: 256 megabytes

On a number axis n teleports are located in the points t_i and m bonuses are located in the points b_j. Being in the same point with a teleport, one can instantly relocate to another point with the teleport. Being in the same point with a bonus, one can instantly get this bonus.

You are in the point t_1 and must collect all the bonuses, and then return back to the point t_1. You can move along the number axis in any direction at the speed 1. How much time will it take to collect all the bonuses?

Input

The first line contans two integers n and m separated by a space ($1 \le n, m \le 200000$) — the number of teleports and the number of bonuses, correspondingly.

The second line contains n integers t_i separated by a space ($-10^9 \le t_i \le 10^9$, $t_i \le t_{i+1}$) — coordinates of the teleports in non-decreasing order.

The third line contains m integers b_j separated by a space ($-10^9 \le b_j \le 10^9$, $b_j \le b_{j+1}$) — coordinates of the bonuses in non-decreasing order.

Output

Output a single integer — the minimal time required to collect all the bonuses.

Examples

standard input	standard output
2 4 0 10 -1 1 9 11	8
2 2 0 10 4 6	10
1 1 1000000000 -1000000000	4000000000

Problem F. Circuits

Time limit: 2 seconds
Memory limit: 256 megabytes

The world famous scientist Innokentiy created n nuclear circuits. But some of them appeared to be defective, and he forgot which exactly. Luckily, only strictly less than a half of circuits were defective, and now the scientist wants to find out which circuits are operational and which are not.

The scientist can connect two circuits with each other, and each of them will tell whether the other one is operational or not. An operational circuit always gives the right answer, while a defective one can give any answer. Defective circuits can give different answers for the same checks. You should help the scientist to determine which circuits are operational, and do it quickly, because the circuits are, well, nuclear, and if the scientist will not finish in $4n$ checks, a radiation will cause him irretrievable damage.

Interaction Protocol

This is an interactive problem. Your program should communicate with the jury's program, using standard input and output for that.

At the very beginning, you program gets a single integer n ($1 \le n \le 5000$) — the number of circuits.

After that you can make at most $4n$ checks. To do that, output character «?» and two distinct integers — the numbers of circuits a and b you want to connect and check. Circuits are numbered from one.

As a response you will get two characters, each of which is «+» or «-». The first character is the output of circuit a about condition of circuit b, and the second character is the output of circuit b about condition of circuit a. The character «+» means a circuit is operational, and «-» means it is defective.

As soon as you determine conditions of all circuits, output character «!», then the integer k — the number of operational circuits, and then k distinct integers — their numbers. After that your program must terminate.

Example

standard input	standard output
5	? 1 2
+ +	? 1 3
- +	? 1 4
+ +	? 1 5
- +	? 2 3
- +	? 2 4
+ +	? 2 5
- +	? 3 4
- -	? 3 5
- +	? 4 5
- -	! 3 1 2 4

Note

Please note that each your message must end with a line break. Also after outputting each message your program must flush the stream buffer, so that the outputted information could reach jury's program: for instance, this can be done by calling «`fflush(stdout)`» or «`cout.flush()`» in C++, «`System.out.flush()`» in Java, «`Console.Out.Flush()`» in C#, «`flush(output)`» in Pascal, «`sys.stdout.flush()`» in Python.

Problem G. I love Codeforces

Time limit: 2 seconds
Memory limit: 256 megabytes

Among participants of programming contests at codeforces.com the following tradition is common: whenever a participant falls in love with another participant, he changes his nickname to «`I_love_<nickname of the participant he fell in love with>`». Unfortunately, for technical reasons site administration has lost the current nicknames of participants who had changed their nicknames recently, but the information about who of them fell in love with whom remains. Now the administration needs to write a program for determining the current nickname of the given participant. It should be noted that due to the small bug in the database structure different participants can have the same nicknames on this site.

Input

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

In the next n lines initial nicknames of participants are given. Each nickname is a non-empty string of length at most 24 characters, consisting only of uppercase and lowercase Latin letters and also underscores. Some nicknames can coincide.

The next line contains a single integer m ($1 \leq m \leq 200000$) — the number of records that one of the participants fell in love with another participant.

In the next m lines pairs of integers a_j and b_j, separated by a space, are given ($1 \leq a_j, b_j \leq n$, $a_j \neq b_j$) — a record that the a_j-th participant fell in love with the b_j-th participant and changed his nickname correspondingly. Participants are numbered from one in the order their initial nicknames are given. Renamings occur in the order they are given.

Output

In the only line output the final nickname of the first participant after all renamings.

Examples

standard input	standard output
5 anonymous natalia LeBron Tanya_Romanova MikeMirzayanov 6 1 2 3 4 2 1 4 3 1 4 3 2	I_love_I_love_I_love_Tanya_Romanova
5 anonymous natalia LeBron Tanya_Romanova MikeMirzayanov 3 2 1 1 2 1 5	I_love_MikeMirzayanov
2 MikhailRubinchik evol_I 1 1 2	I_love_evol_I

Problem H. Perfect Ban

Time limit: 2 seconds
Memory limit: 256 megabytes

Constantine and Mike are playing the board game «Wrath of Elves». There are n races and m classes of characters in this game. Each character is described by his race and class. For each race and each class there is exactly one character of this race and this class. The power of the character of the i-th race and the j-th class equals to a_{ij}, and both players know it perfectly.

Now Constantine will choose a character for himself. Before that Mike can ban one race **and** one class so that Constantine would not be able to choose characters of this race **or** of this class. Of course, Mike does his best to leave Constantine the weakest possible character, while Constantine, on the contrary, chooses the strongest character. Which race and class Mike should ban?

Input

The first line contains two integers n and m ($2 \le n, m \le 1000$) separated by a space — the number of races and classes in the game «Wrath of Elves», correspondingly.

The next n lines contain m integers each, separated by a space. The j-th number in the i-th of these lines is a_{ij} ($1 \le a_{ij} \le 10^9$).

Output

In the only line output two integers separated by a space — the number of race and the number of class Mike should ban. Races and classes are numbered from one. If there are several possible answers, output any of them.

Examples

standard input	standard output
2 2 1 2 3 4	2 2
3 4 1 3 5 7 9 11 2 4 6 8 10 12	3 2

Problem I. Matrix God

Time limit: 2 seconds
Memory limit: 256 megabytes

Matrices are square tables, containing integers in rows and columns. Do you know how to multiply matrices? The product of matrices A and B of size $n \times n$ is a matrix $C = AB$, such that

$$C(i, j) = \sum_{k=1}^{n} A(i, k) \cdot B(k, j).$$

Here the brackets contain numbers of row and column, where the corresponding element of matrix is located.

As it is known, multiplying matrices is not the computationally simplest task. But not for Matrix god! He can easily multiply matrices using the number of operations of order $O(n^2)$. Here and now for some incomprehensible for mortals reason he is trying to determine, is it true that $AB = C$ for three given matrices A, B and C. For him solving this problem has no difficulties. Can you compare with him? The Matrix god gives you an advantage: determine, at least, whether or not reminders of division of each $C(i, j)$ by $10^9 + 7$ are found correctly.

Input

The first line contains a single integer n ($1 \le n \le 1000$) — the size of all matrices.

Then in n lines n integers a_{ij}, separated by a space, are given ($0 \le a_{i,j} \le 10^9 + 6$) — the elements of matrix A. In the next n lines the elements of matrix B are given in the same way, and after that in n more lines — the reminders of division of elements of matrix C by $10^9 + 7$.

Output

Output «`YES`» without quotes, if the reminders of division by $10^9 + 7$ of each element of matrix C are found correctly, otherwise output «`NO`» without quotes.

Examples

standard input	standard output
2 1 2 3 4 5 6 7 8 19 22 43 50	YES
3 1 2 3 4 5 6 7 8 9 1 4 7 2 5 8 3 6 9 14 32 50 32 76 122 50 122 194	NO
1 1000000006 1000000006 1	YES

Problem J. Catch the Monster

Time limit: 2 seconds
Memory limit: 256 megabytes

Manse and Eckels are special agents of time patrol. Now they are trying to catch Faceless Chrono Monster, whose existence threatens the integrity of space-time continuum. The monster dwells between worlds on the Time Tree, consisting of n time nodes, connected with time corridors. Each character can move from one node to another through time corridors. There are $(n-1)$ time corridors, and any time node can be reached from any other one by moving through the time corridors.

Manse and Eckels can initially appear in two arbitrary (not necessarily distinct) time nodes, wherever they want, after which they would be able to move between time nodes through time corridors independently of each other. As soon as one of them meets the Monster, the Monster will be instantly destroyed, no matter whether it happened in the node or in one of the corridors.

The problem is that the Faceless Chrono Monster moves along the Time Tree many times faster than patrolmen, and, most importantly, foresees the future. Thereby, he knows in advance, in which time nodes the heroes will appear and how they will move, so he can use this, planning his escape. The Monster is unbelievably clever, so he will use his abilities optimally to survive. Will Mance and Eckels be able to destroy the Monster?

Input

The first line contains a single integer n ($1 \le n \le 200000$) — the number of time nodes.

The next $(n-1)$ lines describe time corridors. They contain two integers each, x_i and y_i, separated by a space ($1 \le x_i, y_i \le n$, $x_i \ne y_i$) — numbers of time nodes connected by i-th time corridor.

Output

If the heroes will be able to destroy the Monster, output «`YES`» (without quotes), otherwise output «`NO`» (without quotes).

Examples

standard input	standard output
4 1 4 2 4 3 4	YES
7 1 2 1 3 2 4 2 5 3 6 3 7	YES
21 1 2 1 3 1 4 1 5 2 6 2 7 2 8 2 9 3 10 3 11 3 12 3 13 4 14 4 15 4 16 4 17 5 18 5 19 5 20 5 21	NO

Problem K. Competitions

Time limit: 2 seconds
Memory limit: 256 megabytes

Max is preparing hard for the most important event in his life — the ACM ICPC finals. He knows that in the nearest future n programming competitions are going to be held, and that the i-th of them starts at the moment of time a_i, ends at the moment of time b_i and has usefulness c_i. To prepare better, he wants to choose for participating such set of competitions that their total usefulness will be as large as possible, and in case of a tie — that their total duration will be as small as possible. Of course, Max cannot participate in several competitions simultaneously, and also never starts competition after its moment of start and never gives up a competition before its end.

Input

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

The next n lines contain three integers each a_i, b_i, c_i separated by a space ($0 \le a_i < b_i \le 10^9$, $1 \le c_i \le 10^9$) — times of the start and the end of the i-th competition and its usefulness.

Output

In the first line output three integers — the number of competitions k, in which Max should participate, and then their total usefulness and total duration.

In the second line output k integers separated by a space — the numbers of competitions Max should participate in. Competitions are numbered from one in the order they are listed in the input.

If there are several correct answers, output any of them.

Examples

standard input	standard output
5 1 6 7 2 3 2 3 8 6 7 10 3 8 9 3	3 11 7 2 3 5
5 1 6 7 2 3 2 3 8 5 7 10 3 8 9 3	2 10 6 1 5

Problem L. High Probability Cast

Time limit: 2 seconds
Memory limit: 256 megabytes

A mage knows n spells, the i-th of which, when casted, deals random damage (not necessarily integer), uniformly distributed from a_i to b_i. There are m monsters dwelling in the world, and to kill the j-th of them it is required to deal him at least x_j damage. Unfortunately, monsters are so fast that the mage has time to cast only a single spell while fighting each of the monsters, before being killed by that monster. Which spell should be chosen to destroy each of the monsters, so that the probability of killing that monster would be maximized?

Input

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

The next n lines describe spells. Each of them contains two integers a_i and b_i ($1 \leq a_i \leq b_i \leq 10^9$) — the minimal and maximal damage the i-th spell can deal.

The next line contains a single integer m ($1 \leq m \leq 200000$) — the number of monsters.

The next line contains m integers x_j separated by a space ($1 \leq x_j \leq 10^9$) — the minimal amount of damage required to destroy the j-th monster.

Output

Output m integers separated by a space, the j-th of which should be equal to the number of spell which should be used to destroy the j-th monster. The spells are numbered from one in the order they are listed in the input. If there are many spells which give the maximal probability of destroying some monster, you can output any of these spells.

Examples

standard input	standard output
2 1 10 4 8 4 3 6 7 11	2 2 1 1
2 2 5 7 9 5 10 8 6 3 1	2 2 2 2 2

Problem M. Last Man Standing

Time limit: 2 seconds
Memory limit: 256 megabytes

A company of n friends decided to play a multiplayer shooter in the Last Man Standing mode. The distinctive feature of this mode is when a player is killed, he does not respawn, but waits a round to finish. A round finishes when there is only one player alive.

You are given a notarized screenshot of the current fight results, made during the first round of the game. It provides an information about how many kills each player has made. You want to check if it's a fake or not, i.e. whether such situation could occur during the game.

Input

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

The second line contains n integers a_i separated by a space ($0 \le a_i \le 10^9$) — the numbers of kills made by each player. These numbers are given in non-increasing order.

Output

If such situation was not possible in the game, output «NO» (without quotes).

Otherwise, in the first line output «YES» (without quotes), and then output a log of kills in the round, consisting of k pairs of integers, where k is the total number of kills made at the moment of taking the screenshot. Each pair of integers in the log must consist of the number of the player who made the kill, and the number of the killed player, exactly in this order. Records in the log must be ordered chronologically. If there are several correct logs, output any of them.

Examples

standard input	standard output
5 2 1 1 0 0	YES 3 5 2 4 1 3 1 2
7 3 2 2 0 0 0 0	NO
1 0	YES
3 1 0 0	YES 1 3