Codeforces Round #517 (Div. 1, based on Technocup 2019 Elimination Round 2)

A. Cram Time

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

In a galaxy far, far away Lesha the student has just got to know that he has an exam in two days. As always, he hasn't attended any single class during the previous year, so he decided to spend the remaining time wisely.

Lesha knows that today he can study for at most hours, and he will have hours to study tomorrow. Note that it is possible that on his planet there are more hours in a day than on Earth. Lesha knows that the quality of his knowledge will only depend on the number of lecture notes he will read. He has access to an infinite number of notes that are enumerated with positive integers, but he knows that he can read the first note in one hour, the second note in two hours and so on. In other words, Lesha can read the note with number in hours. Lesha can read the notes in arbitrary order, however, he can't start reading a note in the first day and finish its reading in the second day.

Thus, the student has to fully read several lecture notes today, spending at most hours in total, and fully read several lecture notes tomorrow, spending at most hours in total. What is the maximum number of notes Lesha can read in the remaining time? Which notes should he read in the first day, and which — in the second?

Input

The only line of input contains two integers and ( ) — the number of hours Lesha has today and the number of hours Lesha has tomorrow.

Output

In the first line print a single integer ( ) — the number of lecture notes Lesha has to read in the first day. In the second line print distinct integers ( ), the sum of all should not exceed .

In the third line print a single integer ( ) — the number of lecture notes Lesha has to read in the second day. In the fourth line print distinct integers ( ), the sum of all should not exceed .

All integers and should be distinct. The sum should be largest possible.

Examples

input
3 3

output
1 3 2 2 1

input
9 12

output
2 3 6 4 1 2 4 5

Note

In the first example Lesha can read the third note in hours in the first day, and the first and the second notes in one and two hours correspondingly in the second day, spending hours as well. Note that Lesha can make it the other way round, reading the first and the second notes in the first day and the third note in the second day.

In the second example Lesha should read the third and the sixth notes in the first day, spending hours in total. In the second day Lesha should read the first, second fourth and fifth notes, spending hours in total.

B. Minimum path

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a matrix of size filled with lowercase English letters. You can change no more than letters in this matrix.

Consider all paths from the upper left corner to the lower right corner that move from a cell to its neighboring cell to the right or down. Each path is associated with the string that is formed by all the letters in the cells the path visits. Thus, the length of each string is .

Find the lexicographically smallest string that can be associated with a path after changing letters in at most cells of the matrix.

A string is lexicographically smaller than a string , if the first different letter in and is smaller in .

Input

The first line contains two integers and (,) — the size of the matrix and the number of letters you can change.

Each of the next lines contains a string of lowercase English letters denoting one row of the matrix.

Output

Output the lexicographically smallest string that can be associated with some valid path after changing no more than letters in the matrix.

Examples

input
4 2 abcd bcde bcad bcde

output
aaabcde

input
5 3 bwwwz hrhdh sepsp sqfaf ajbvw
output
aaaepfafw

input
7 6 ypnxnnp pnxonpm nxanpou xnnpmud nhtdudu npmuduh pmutsnz
output
aaaaaaadudsnz

Note

In the first sample test case it is possible to change letters 'b' in cells and to 'a', then the minimum path contains cells . The first coordinate corresponds to the row and the second coordinate corresponds to the column.

C. Triple Flips

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given an array of length that consists of zeros and ones.

You can perform the following operation multiple times. The operation consists of two steps:

1. Choose three integers , that form an arithmetic progression ( ).
2. Flip the values (i.e. change to , change to).

Determine if it is possible to make all elements of the array equal to zero. If yes, print the operations that lead the the all-zero state. Your solution should not contain more than — operations. Here denotes the number rounded down. We can show that it is possible to make all elements equal to zero in no more than this number of operations whenever it is possible to do so at all.

Input

The first line contains a single integer ( ) — the length of the array.

The second line contains integers ( ) — the elements of the array.

Output

Print "YES" (without quotes) if the answer exists, otherwise print "NO" (without quotes). You can print each letter in any case (upper or lower).

If there is an answer, in the second line print an integer (—) — the number of operations in your answer.

After that in ( )-th line print the -th operations — the integers . You can print them in arbitrary order.

Examples

input
5 1 1 0 1 1
output
YES 2 1 3 5 2 3 4

input
3 0 1 0
output
NO

Note

In the first sample the shown output corresponds to the following solution:

- `1 1 0 1 1` (initial state);
- `0 1 1 1 0` (the flipped positions are the first, the third and the fifth elements);
- `0 0 0 0 0` (the flipped positions are the second, the third and the fourth elements).

Other answers are also possible. In this test the number of operations should not exceed — .

In the second sample the only available operation is to flip all the elements. This way it is only possible to obtain the arrays `0 1 0` and `1 0 1`, but it is impossible to make all elements equal to zero.

D. Familiar Operations

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given two positive integers and . There are two possible operations:

1. multiply one of the numbers by some prime ;
2. divide one of the numbers on its prime factor .

What is the minimum number of operations required to obtain two integers having the same number of divisors? You are given several such pairs, you need to find the answer for each of them.

Input

The first line contains a single integer ( ) — the number of pairs of integers for which you are to find the answer.

Each of the next lines contain two integers and ( ).

Output
Output lines — the -th of them should contain the answer for the pair , .

Example

input
8 9 10 100 17 220 70 17 19 4 18 32 20 100 32 224 385
output
1 3 1 0 1 0 1 1

Note
These are the numbers with equal number of divisors, which are optimal to obtain in the sample test case:

- , 4 divisors
- , 9 divisors
- , 12 divisors
- , 2 divisors
- , 6 divisors
- , 6 divisors
- , 12 divisors

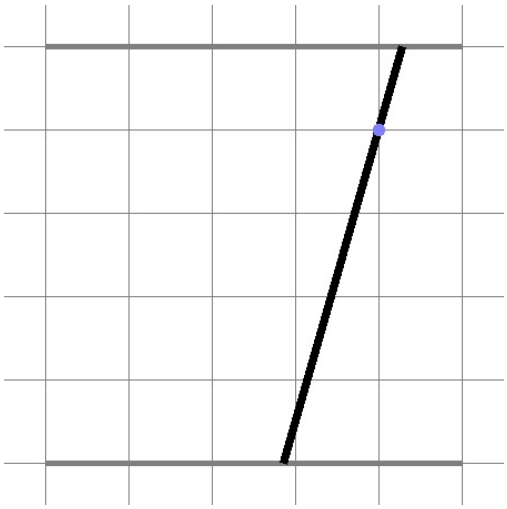
Note that there can be several optimal pairs of numbers.

E. Rain Protection

time limit per test: 7 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A lot of people dream of convertibles (also often called cabriolets). Some of convertibles, however, don't have roof at all, and are vulnerable to rain. This is why Melon Ask, the famous inventor, decided to create a rain protection mechanism for convertibles.

The workplace of the mechanism is a part of plane just above the driver. Its functional part consists of two rails with sliding endpoints of a piece of stretching rope. For the sake of simplicity we can consider this as a pair of parallel segments in a plane with the rope segment, whose endpoints we are free to choose as any points on these rails segments.



The algorithmic part of the mechanism detects each particular raindrop and predicts when and where it reaches the plane. At this exact moment the rope segment must contain the raindrop point (so the rope adsorbs the raindrop).

You are given the initial position of the rope endpoints and all information about raindrops. You are to choose the minimal possible speed of the endpoints sliding (both endpoints can slide in any direction along their segments independently of each other) in such a way that it is possible to catch all raindrops moving both endpoints with speed not greater than , or find out that it's impossible no matter how high the speed is.

Input
The first line contains three integers , and (,), meaning that there are raindrops, and two rails are represented as segments connecting and and connecting and .

The second line contains two integers and , meaning that the initial (that is, at the moment) positions of the endpoints are and ( ).

The -th of the following lines contains three integers , and (, ,) meaning that the -th raindrop touches the plane at the point at the time moment . It is guaranteed that for all valid .

Output
If it is impossible to catch all raindrops, print .

Otherwise, print the least possible maximum speed of the rope endpoints for which it is possible to catch them all. Your answer is considered correct if the absolute or relative error doesn't exceed .

Formally, let your answer be , and the jury's answer be . Your answer is considered correct if .

Examples

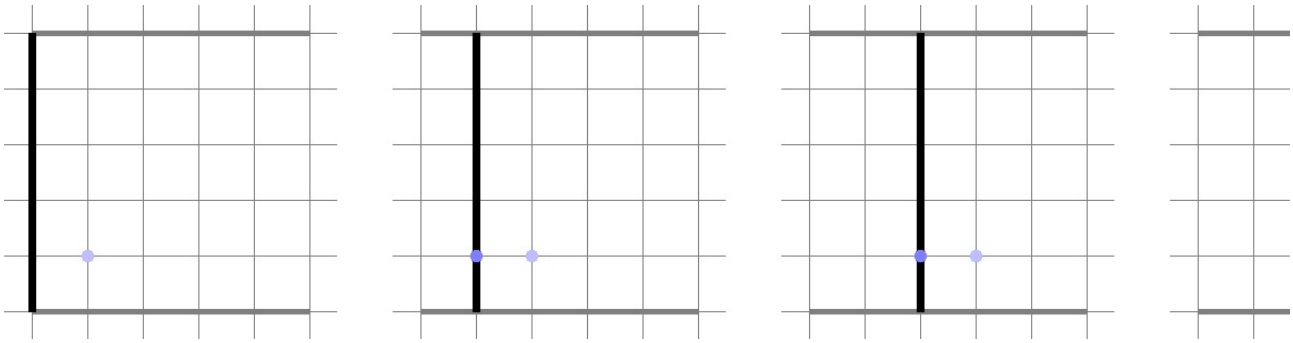
input
3 5 5 0 0 1 1 4 2 2 4 3 3 4
output
1.0000000019

input
2 5 5 0 0 2 4 1 3 1 4
output
2.1428571437

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

Note

That is how one can act in the first sample test:



Here is the same for the second: