Problem A. A + B

Time limit: 2 seconds
Memory limit: 512 megabytes

You are given two real numbers a and b. Write a program to calculate $a + b$.

Input

The first line contains two real numbers — a and b ($-1\,000 \le a, b \le 1\,000$).

Output

Print the value of $a+b$ on the first line of the output. The answer is considered to be correct if its absolute or relative error does not exceed 10^{-4}.

Scoring

Subtask 1 (50 points)

Both a and b are integers

Subtask 2 (50 points)

No additional constraint

Examples

input	output
1.1 2.2	3.3
1 -1	0

Problem B. Zero-complexity Transposition

Time limit:	2 seconds
Memory limit:	512 megabytes

You are given a sequence of integer numbers. Zero-complexity transposition of the sequence is the reverse of this sequence. Your task is to write a program that prints zero-complexity transposition of the given sequence.

Input

The first line of the input contains one integer n — length of the sequence ($0 < n \leq 10^4$). The second line contains n integer numbers — $a_1, a_2, \ldots, a_n$ ($-10^{15} \leq a_i \leq 10^{15}$).

Output

On the first line of the output print the sequence in the reverse order.

Scoring

Subtask 1 (50 points)

$-1\,000 \leq a_i \leq 1\,000$

Subtask 2 (50 points)

No additional constraint

Examples

input	output
3 1 2 3	3 2 1
5 -3 4 6 -8 9	9 -8 6 4 -3

Problem C. Wedding cake

Time limit: 2 seconds
Memory limit: 512 megabytes

Julia's wedding is going to have a huge one ton cake. All n guests want to taste the cake, so it's going to be cut in n pieces. But this task is not that easy, because all guests are on a special mathematical diet. Guest i is only willing to eat the cake if the weight of his piece in tons w_i has exactly a_i significant digits after the decimal point. In decimal representation all digits up to the last non-zero digit after the decimal point are significant. For example, number `0.007` contains three significant digits after the decimal point, number `1.45` — two, and number `17.0` has no significant digits after the decimal point.

Your task is to cut the cake for Julia's wedding so that every guest could taste it.

Input

First line contains single integer n — number of guests ($1 \le n \le 10^5$).

Next line contains n integers a_i — constraint for the weight of i-th piece ($1 \le a_i \le 10^5$).

Total value of all a_i doesn't exceed 10^5.

Output

Output should contain "`NO`", if there is no way to cut the cake.

Otherwise, output "`YES`" on the first line. Each of the next n lines should contain one single real number w_i — weight of the piece for the i-th guest with exactly a_i digits after the decimal point. All a_i digits after the decimal point have to be significant.

Scoring

Subtask 1 (17 points)

$n \le 100$, $a_i \le 10$

Subtask 2 (21 points)

$n \le 10^5$, all a_i are equal

Subtask 3 (25 points)

$n \le 1\,000$, total value of all a_i doesn't exceed $1\,000$

Subtask 4 (37 points)

No additional constraint

Example

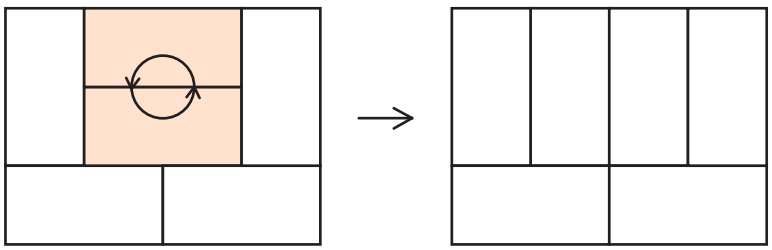
input	output
5 2 4 4 3 2	YES 0.47 0.1234 0.1326 0.024 0.25

Problem D. Parquet Re-laying

Time limit:	2 seconds
Memory limit:	512 megabytes

Peter decided to lay a parquet in the room of size $n \times m$, the parquet consists of tiles of size 1×2. When the workers laid the parquet, it became clear that the tiles pattern looks not like Peter likes, and workers will have to re-lay it.

The workers decided that removing entire parquet and then laying it again is very difficult task, so they decided to make such an operation every hour: remove two tiles, which form a 2×2 square, rotate them 90 degrees and put them back on the same place.



They have no idea how to obtain the desired configuration using these operations, and whether it is possible at all.

Help Peter to make a plan for the workers or tell that it is impossible. The plan should contain at most 100 000 commands.

Input

The first line contains integer n and m, size of the room ($1 \le n, m \le 50$).

The following n lines contain m characters each, the description of the current configuration of the parquet tiles. Each character represents the position of the half-tile. Characters '`L`', '`R`', '`U`' and '`D`' correspond to the left, right, upper and lower halves, respectively.

The following n lines contain m characters each, describing the desired configuration in the same format.

Output

In the first line output integer k, the number of operations. In the next k lines output description of operations. The operation is specified by coordinates (row and column) of the left upper half-tile on which the operation is performed.

If there is no solution, output `-1` in the first line.

Scoring

Subtask 1 (13 points)

$n, m \le 4$

Subtask 2 (28 points)

$n, m \le 20$

Subtask 3 (59 points)

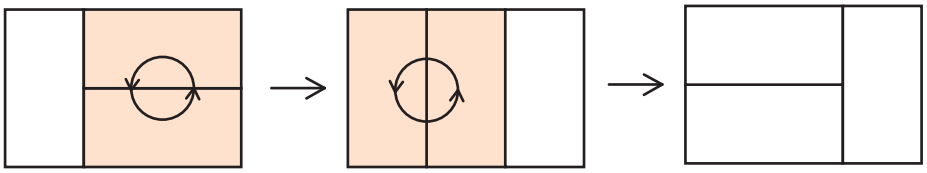
$n, m \le 50$

Example

input	output
2 3 ULR DLR LRU LRD	2 1 2 1 1

Explanation

First operation is to rotate two rightmost tiles, after this all tiles lie vertically. Second operation is to rotate two leftmost tiles, after this we will get desired configuration.



Problem E. Lock Puzzle

Time limit:	2 seconds
Memory limit:	256 megabytes

Welcome to another task about breaking the code lock! Explorers Whitfield and Martin came across an unusual safe, inside of which, according to rumors, there are untold riches, among which one can find the solution of the problem of discrete logarithm!

Of course, there is a code lock is installed on the safe. The lock has a screen that displays a string of n lowercase Latin letters. Initially, the screen displays string s. Whitfield and Martin found out that the safe will open when string t will be displayed on the screen.

The string on the screen can be changed using the operation «`shift` x». In order to apply this operation, explorers choose an integer x from 0 to n inclusive. After that, the current string $p = \alpha\beta$ changes to $\beta^R\alpha$, where the length of β is x, and the length of α is $n - x$. In other words, the suffix of the length x of string p is reversed and moved to the beginning of the string. For example, after the operation «`shift` 4» the string «`abcacb`» will be changed with string «`bcacab` », since $\alpha =$ `ab`, $\beta =$ `cacb`, $\beta^R =$ `bcac`.

Explorers are afraid that if they apply too many operations «`shift`», the lock will be locked forever. They ask you to find a way to get the string t on the screen, using no more than m operations.

Input

The first line contains two integers numbers n and m, the length of the strings s and t, and the maximum number of operations.

After that, there are two strings s and t, consisting of n lowercase Latin letters each.

Output

If it is impossible to get string t from string s using no more than m operations «`shift`», print a single number -1.

Otherwise, in the first line output the number of operations k ($0 \le k \le m$). In the next line output k numbers x_i corresponding to the operations «`shift` x_i» ($0 \le x_i \le n$) in the order in which they should be applied.

Scoring

Subtask 1 (9 points)

$n \le 8$, $m = 10\,000$

Subtask 2 (21 points)

$n \le 100$, $m = 10\,000$

Subtask 3 (24 points)

$n \le 1\,000$, $m = 10\,000$

Subtask 4 (12 points)

$n \le 2\,000$, $m = 10\,000$

Subtask 5 (12 points)

$n \le 2\,000$, $8\,100 \le m \le 10\,000$

Subtask 6 (11 points)

$n \le 2\,000$, $6\,100 \le m \le 10\,000$

Subtask 7 (11 points)

$n \leq 2\,000$, $5\,100 \leq m \leq 10\,000$

Examples

input	output
6 10000 abacbb babcba	4 6 3 2 3
3 10000 aba bba	-1

Explanation

In the first example, after applying the operations, the string on the screen will change as follows:

1. abacbb → bbcaba
2. bbcaba → ababbc
3. ababbc → cbabab
4. cbabab → babcba