TASK	PUŽ	NAPOR	IGRA	KNJIGE	LUNAPARK	CRNI
input	standard input (*stdin*)					
output	standard output (*stdout*)					
time limit	0.1 second	1 second	1 second	1 second	2 seconds	1 second
memory limit	32 MB	32 MB	32 MB	32 MB	64 MB	128 MB
total points	**30**	**60**	**80**	**80**	**120**	**130**
	500					

There is a snail on the ground. It wants to climb to the top of a wooden pole with the height of **V** meters, measuring from the ground level. In one day it can climb **A** meters upwards, however during each night it sleeps, sliding **B** meters back down. Determine the number of days it needs to climb to the top.

INPUT

The first and only line of input contains three integers separated by a single space: **A**, **B**, and **V** ($1 \leq \mathbf{B} < \mathbf{A} \leq \mathbf{V} \leq 1\,000\,000\,000$), with meanings described above.

OUTPUT

The first and only line of output must contain the number of days that the snail needs to reach the top.

SAMPLE TESTS

input	**input**	**input**
2 1 5	5 1 6	100 99 1000000000
output	**output**	**output**
4	2	999999901

Little Mirko wasn't paying attention in math class, so the teacher has decided to give him a tedious assignment to solve during the weekend.

The teacher has given him a text consisting of **N** lines, containing only digits and lower case letters of the English alphabet. Mirko has to find all numbers in the text and print them out in a **nondecreasing sequence**. He also has to **omit** any **leading zeros** that the numbers may have in the text.

The numbers can be uniquely determined by scanning through the text and always taking the largest possible number, i.e. delimited only by letters or line beginnings/ends. For example, the solution of **01a2b3456cde478** is **1**, **2**, **478**, **3456**.

Since Mirko is as slow as the snail from the previous task, he has asked you to write him a program to quickly solve his assignment, so that he can go play with Slavko as soon as possible.

INPUT

The first line of input contains the integer **N** ($1 \leq \mathbf{N} \leq 100$), the number of lines of the text.

The next **N** lines contain the text, consisting exclusively of lowercase English letters and decimal digits. Each line of the text is at most 100 characters long.

OUTPUT

The output must contain **M** lines, where **M** is the number of numbers found in the provided text. Each line must contain a single number from the text. The numbers must be arranged in a nondecreasing sequence.

Note: The test data will ensure that **M** will never exceed 500.

SAMPLE TESTS

input	**input**	**input**
2 lo3za4 01 **output** 1 3 4	4 43silos0 zita002 le2sim 231233 **output** 0 2 2 43 231233	4 01bond 02james007 03bond 04austinpowers000 **output** 0 1 2 3 4 7

Having solved the tedious assignment, Mirko decided to play a game with his good friend Slavko.

They have written a sequence of **N** letters on a piece of paper. Each one of them is trying to put together a word using letters from the sequence. They alternate taking turns consisting of removing a single letter from the sequence and appending it to the end of their word. Mirko has the first turn. The game ends when no letters are remaining in the sequence.

We define a word to be more beautiful than another word if it comes first alphabetically. The player who has the more beautiful word at the end of the game wins. If both players have equal words, they both lose.

Mirko is a much better player than Slavko, so he has decided to make it easier for Slavko by always selecting the rightmost remaining letter in the sequence. Knowing this, Slavko wants to find out if it is possible for him to win and which is the most beautiful word he can end the game with.

INPUT

The first line of input contains an **even** positive integer **N** ($2 \leq \mathbf{N} \leq 100\,000$).

The second line of input contains **N** characters, the starting letter sequence. All characters are lower case letters from the English alphabet.

OUTPUT

The first line of output must contain "DA" if it is possible for Slavko to win, and "NE" otherwise.

The second line of output must contain the most beautiful word that Slavko can have at the end of the game.

SCORING

In test cases worth 50% of total points the number **N** will not exceed 1000.

SAMPLE TESTS

input	**input**	**input**
2 ne	4 kava	8 cokolada
output	**output**	**output**
NE n	DA ak	DA acko

Mirko has a home library consisting of **N** books arranged one on top of the other in a narrow cabinet. Since being well trained in the secrets of alphabet in the previous task, he now wishes to arrange the books alphabetically, so that the book whose title comes first alphabetically ends up on top, and the alphabetically last one at the bottom of the pile.

Mirko can easily **pull** a book **out** of the cabinet, but it is difficult to push it back into the pile, so the book can only be returned to the **top** of the pile. Thus, the only available method of sorting the books is repeatedly pulling a book out of the pile and placing it on top of the pile.

The books are labelled with integers from 1 to **N**, in alphabetical order. Therefore, Mirko wants them to be ordered as (1, 2, ..., **N**), counting from the top. For example, if **N** = 3 and the starting order is (3, 2, 1), two moves are sufficient. First, he pulls out the book number 2 and places it on top, so the pile becomes (2, 3, 1). After that, he does the same with book number 1, thus the pile becomes (1, 2, 3).

Help Mirko by calculating the minimum number of moves needed to sort a given starting order.

INPUT

The first line of input contains the integer **N** (**N** ≤ 300 000).

Each of the next **N** lines contains a single positive integer. These **N** integers represent the order of Mirko's books from top to bottom of the cabinet. Each of the integers 1, 2, ..., **N** appears exactly once.

OUTPUT

The first and only line of output must contain the required minimum number of moves.

SAMPLE TESTS

input	**input**
3 3 2 1 **output** 2	4 1 3 4 2 **output** 2

Mirko has grown tired of all the books, so he decided to go to the amusement park with his friends, despite not liking roller coasters. While his friends are having the time of their lives riding the coasters, Mirko is sitting on a bench, waiting and thinking about the possible paths of the coasters.

The amusement park's area can be represented as a table of **R** rows by **C** columns. A roller coaster has to start from the upper left corner and end at the lower right corner of the table. Each cell can be visited at most once, but not all cells need to be visited. It can continue its path from the current cell to the adjacent one above, below, to the left, or to the right of it.

Each cell has a positive integer value associated with it, specifying how amusing that cell is to visitors. The total amusement value of the coaster is the sum of amusement values of all cells that the coaster visits. Help Mirko determine any one of the most amusing coasters (ones with the maximum sum).

INPUT

The first line of input contains two integers **R** and **C** ($2 \leq \mathbf{R}, \mathbf{C} \leq 1000$), the dimensions of the table.

Each of the next **R** lines contains **C** positive integers smaller than 1000, specifying the amusement values of the respective table cells.

OUTPUT

The first and only line of output must contain a sequence of letters with no blanks. The letters specify the sequence of directions that the coaster follows, starting from the upper left and ending at the lower right corner. The directions up, right, down, left are marked by letters 'U', 'R', 'D', 'L', respectively.

Note: The solution isn't guaranteed to be unique.

SCORING

Test cases worth 70% of total points the numbers **R** and **C** will not exceed 30.

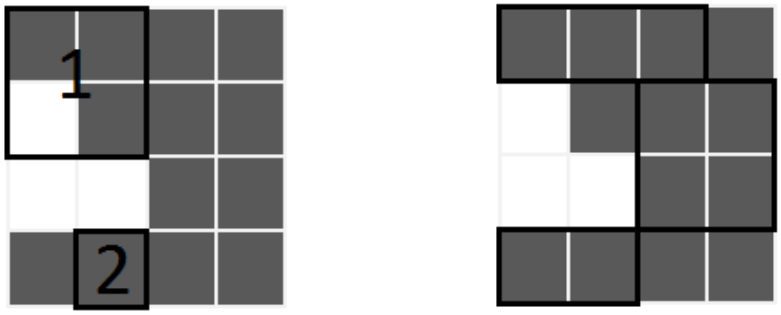
SAMPLE TESTS

input	**input**
3 3 5 1 3 2 4 8 1 1 2 **output** RRDLLDRR	2 2 2 1 3 4 **output** DR

Even though he has found all the most amusing rides, Mirko's enthusiasm still isn't fading. He opened his graph paper notebook and started colouring squares, and a new, even harder problem dawned on him.

You are given a square table consisting of **N** rows by **N** columns. Each cell is either black or white.

A set of cells forming a rectangle, with horizontal and vertical edges following cell borders, shall be called a **black rectangle** if all cells inside the rectangle are black and it consists of **at least two** cells.



The left image shows two rectangles which **are not** black rectangles. The rectangle labelled 1 is not a black rectangle because it contains a white cell, and the rectangle labelled 2 is not a black rectangle because it consists of only one cell. On the other hand, the right image shows three valid black rectangles.

Calculate the number of possible selections of two black rectangles that have **no common cells**. As the required number can be extremely large, you should output the remainder of dividing that number by 10 007.

INPUT

The first line of input contains the integer **N** ($2 \leq N \leq 1000$).

Each of the next **N** lines contains a single row of the table, consisting of **N** symbols. The symbol 'C' represents a black cell, while 'B' represents a white cell.

OUTPUT

The first and only line of output must contain the remainder of dividing the required number by 10007.

SAMPLE TESTS

input	input	input
2 CC CC output 2	3 CCB CCB CBB output 5	5 BCCBB BBCBB BCCBB BBBBB CCBBB output 8