

TASK	PET	KEMIJA	CROSS	MATRICA	BST	NAJKRACI
input	standard input					
output	standard output					
time limit	1 second	1 second	1 second	0.2 seconds	1 second	5 seconds
memory limit	32 MB	32 MB	32 MB	32 MB	32 MB	32 MB
points	30	40	70	110	120	130
	500					

In the popular show "Dinner for Five", five contestants compete in preparing culinary delights. Every evening one of them makes dinner and each of other four then grades it on a scale from 1 to 5.

The number of points a contestant gets is equal to the sum of grades they got. The winner of the show is of course the contestant that gets the most points.

Write a program that determines the winner and how many points they got.

INPUT

Five lines, each containing 4 integers, the grades a contestant got.

The contestants are numbered 1 to 5 in the order in which their grades were given.

The input data will guarantee that the solution is unique.

OUTPUT

Output on a single line the winner's number and their points, separated by a space.

EXAMPLES

input	input
5 4 4 5	4 4 3 3
5 4 4 4	5 4 3 5
5 5 4 4	5 5 2 4
5 5 5 4	5 5 5 1
4 4 4 5	4 4 4 4
output	output
4 19	2 17

Luka is fooling around in chemistry class again! Instead of balancing equations he is writing coded sentences on a piece of paper. Luka modifies every word in a sentence by adding, after each vowel (letters 'a', 'e', 'i', 'o' and 'u'), the letter 'p' and then that same vowel again.

For example, the word "kemija" becomes "kepemipijapa" and the word "paprika" becomes "papapripikapa". The teacher took Luka's paper with the coded sentences and wants to decode them.

Write a program that decodes Luka's sentence.

INPUT

The coded sentence will be given on a single line. The sentence consists only of lowercase letters of the English alphabet and spaces. The words will be separated by exactly one space and there will be no leading or trailing spaces. The total number of character will be at most 100.

OUTPUT

Output the decoded sentence on a single line.

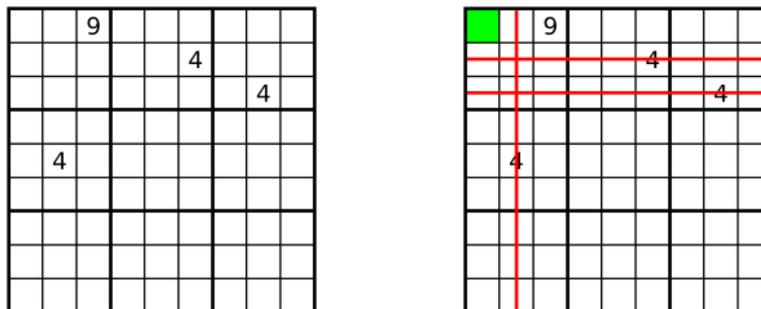
EXAMPLES

input zepelepenapa papapripikapa	input bapas jepe doposapadnapa opovapa kepemipijapa
output zelena paprika	output bas je dosadna ova kemija

In the game of Sudoku, the objective is to place integers between 1 and 9 (inclusive) into a 9x9 grid so that each row, each column, and each of the nine 3x3 boxes contains all nine numbers. The starting board is partially filled in so that it is possible to logically deduce the values of the other cells. Sudoku puzzles range in difficulty, and complex analysis methods are required to solve the hardest puzzles. In this problem, however, you will implement one of the simplest methods, cross-hatching.

In cross-hatching, we select one of the nine numbers and, for each of its occurrences in the grid, cross out the corresponding row, column and 3x3 box. Now look for any 3x3 boxes where there is only one possible placement for the number and place it there.

The first image below shows a very sparsely filled in Sudoku grid. However, even in this grid it is possible to deduce using cross-hatching that the number in the top left cell is 4, as illustrated in the second image.



You will be given a partially filled-in grid. Your task is to repeatedly apply the cross-hatching method for different numbers until no more deductions can be made about any number.

The initial placement of the numbers in the grid may be invalid. It is also possible that there will be no available cell for a number in a 3x3 box. In both cases, you are to report an error.

INPUT

Input will consist of 9 lines, each containing exactly 9 characters. Each character will either be a digit between 1 and 9, or a period (.) denoting an empty cell.

OUTPUT

If the input is valid and there is no contradiction while solving, you should output the grid in the same format it was given in, with cells filled in if their value can be deduced using cross-hatching. Otherwise, output "ERROR" (quotes for clarity).

EXAMPLES

<p>input</p> <pre>..9.....4...4.4.....</pre> <p>output</p> <pre>4.9.....4...4.4.....</pre>	<p>input</p> <pre>...1...6. 18...9... ..7.642.. 2.9..6.5. .43...72. .6.3..9.1 ..265.1.. ...2...97 .5...3...</pre> <p>output</p> <pre>524137869 186529473 397864215 219476358 843915726 765382941 972658134 638241597 451793682</pre>	<p>input</p> <pre>1..... ..1.....1.</pre> <p>output</p> <pre>ERROR</pre>	<p>input</p> <pre>.....21... 1.....1..1.</pre> <p>output</p> <pre>ERROR</pre>
--	--	--	--

A matrix is a rectangular table of letters. A square matrix is a matrix with an equal number of rows and columns. A square matrix M is called **symmetric** if its letters are symmetric with respect to the main diagonal ($M_{ij} = M_{ji}$ for all pairs of i and j).

The following figure shows two symmetric matrices and one which is not symmetric:

AAB	AAA	ABCD	AAB
ACC	ABA	ABCD	ACA
BCC	AAA	ABCD	DAA
		ABCD	
Two symmetric matrices.		Two matrices that are not symmetric.	

Given a collection of available letters, you are to output a **subset of columns** in the **lexicographically smallest symmetric** matrix which can be composed using **all** the letters.

If no such matrix exists, output "IMPOSSIBLE".

To determine if matrix A is lexicographically smaller than matrix B , consider their elements in row-major order (as if you concatenated all rows to form a long string). If the first element in which the matrices differ is smaller in A , then A is lexicographically smaller than B .

INPUT

The first line of input contains two integers N ($1 \leq N \leq 30000$) and K ($1 \leq K \leq 26$). N is the dimension of the matrix, while K is the number of distinct letters that will appear.

Each of the following K lines contains an uppercase letter and a positive integer, separated by a space. The integer denotes how many corresponding letters are to be used. For example, if a line says "A 3", then the letter A must appear three times in the output matrix.

The total number of letters will be exactly N^2 . No letter will appear more than once in the input.

The next line contains an integer P ($1 \leq P \leq 50$), the number of columns that must be output.

The last line contains P integers, the indices of columns that must be output. The indices will be between 1 and N inclusive, given in increasing order and without duplicates.

OUTPUT

If it is possible to compose a symmetric matrix from the given collection of letters, output the required columns on N lines, each containing P character, without spaces. Otherwise, output "IMPOSSIBLE" (quotes for clarity).

SCORING

In test cases worth 60% of points, N will be at most 300.

In test cases worth 80% of points, N will be at most 3000.

EXAMPLES

input 3 3 A 3 B 2 C 4 3 1 2 3 output AAB ACC BCC	input 4 4 A 4 B 4 C 4 D 4 4 1 2 3 4 output AABB AACC BCDD BCDD	input 4 5 E 4 A 3 B 3 C 3 D 3 2 2 4 output AC BE DE ED	input 4 6 F 1 E 3 A 3 B 3 C 3 D 3 4 1 2 3 4 output IMPOSSIBLE
--	--	---	--

A binary search tree is a tree in which every node has **at most** two children nodes (a left and a right child). Each node has an integer written inside it. If the number X is written inside a node, then the numbers in its left subtree are less than X and the numbers in its right subtree are greater than X .

You will be given a sequence of integers between 1 and N (inclusive) such that each number appears in the sequence exactly once. You are to create a binary search tree from the sequence, putting the first number in the root node and inserting every other number in order. In other words, run $\text{insert}(X, \text{root})$ for every other number:

```
insert( number X, node N )
    increase the counter C by 1
    if X is less than the number in node N
        if N has no left child
            create a new node with the number X and set it to be the left child of node N
        else
            insert(X, left child of node N)
    else (X is greater than the number in node N)
        if N has no right child
            create a new node with the number X and set it to be the right child of node N
        else
            insert(X, right child of node N)
```

Write a program that calculates the value of the counter C after every number is inserted. The counter is initially 0.

INPUT

The first line contains the integer N ($1 \leq N \leq 300000$), the length of the sequence.

The remaining N lines contain the numbers in the sequence, integers in the interval $[1, N]$. The numbers will be distinct.

OUTPUT

Output N integers each on its own line, the values of the counter C after each number is inserted into the tree.

SCORING

In test cases worth 50% of points, N will be at most 1000.

EXAMPLES

<p>input</p> <p>4 1 2 3 4</p> <p>output</p> <p>0 1 3 6</p>	<p>input</p> <p>5 3 2 4 1 5</p> <p>output</p> <p>0 1 2 4 6</p>	<p>input</p> <p>8 3 5 1 6 8 7 2 4</p> <p>output</p> <p>0 1 2 4 7 11 13 15</p>
--	--	---

A road network in a country consists of N cities and M one-way roads. The cities are numbered 1 through N . For each road we know the origin and destination cities, as well as its length.

We say that the road F is a **continuation** of road E if the destination city of road E is the same as the origin city of road F . A **path** from city A to city B is a sequence of road such that origin of the first road is city A , each other road is a continuation of the one before it, and the destination of the last road is city B . The length of the path is the sum of lengths of all roads in it.

A path from A to B is a **shortest** path if there is no other path from A to B that is shorter in length.

Your task is to, for each road, output **how many different** shortest paths containing that road, modulo 1 000 000 007.

INPUT

The first line contains two integers N and M ($1 \leq N \leq 1500$, $1 \leq M \leq 5000$), the number of cities and roads.

Each of the following M lines contains three positive integers O , D and L . These represent a one-way road from city O to city D of length L . The numbers O and D will be different and L will be at most 10000.

OUTPUT

Output M integers each on its own line – for each road, the number of different shortest paths containing it, modulo 1 000 000 007. The order of these numbers should match the order of roads in the input.

SCORING

In test cases worth 30% of points, N will be at most 15 and M will be at most 30.

In test cases worth 60% of points, N will be at most 300 and M will be at most 1000.

EXAMPLES

<p>input</p> <p>4 3 1 2 5 2 3 5 3 4 5</p> <p>output</p> <p>3 4 3</p>	<p>input</p> <p>4 4 1 2 5 2 3 5 3 4 5 1 4 8</p> <p>output</p> <p>2 3 2 1</p>	<p>input</p> <p>5 8 1 2 20 1 3 2 2 3 2 4 2 3 4 2 3 3 4 5 4 3 5 5 4 20</p> <p>output</p> <p>0 4 6 6 6 7 2 6</p>
--	--	--