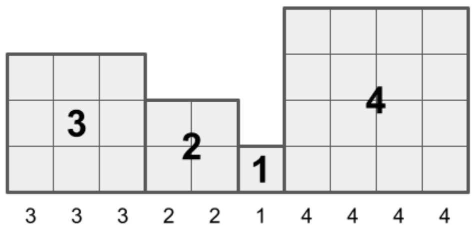
Problem A. Histogram Sequence 3

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Consider the histogram composed of n squares with side lengths $a_1, a_2, \cdots, a_n$. Let's call the sequence $(a_1, a_2, \cdots, a_n)$ the histogram sequence of this histogram.

Let's consider the height of each column in this histogram. The first a_1 columns will each have height a_1, the following a_2 columns will each have height a_2, ... and the last a_n columns will each have height a_n. Now, let us define the height sequence $(b_1, b_2, \cdots, b_{a_1+a_2+\cdots+a_n})$ where b_j $(1 \le j \le a_1 + a_2 + \cdots + a_n)$ is the height of the j-th column.

For example, the histogram with $(3, 2, 1, 4)$ as its histogram sequence has $(3, 3, 3, 2, 2, 1, 4, 4, 4, 4)$ as its height sequence.



Write a program to find the histogram sequence given the height sequence.

Input

The first line contains a single integer m $(1 \le m \le 10^6)$ representing the length of the height sequence $\{b_i\}$ is given.

The second line of the input contains m integers, the height sequence. Specifically, the i-th integer in the line is b_i $(1 \le b_i \le m)$.

The input is designed such that the provided height sequence corresponds to a valid histogram sequence.

Output

Output n integers on a single line, $a_1, a_2, \cdots, a_n$ where $(a_1, a_2, \cdots, a_n)$ is the histogram sequence corresponding to the given height sequence. If there are multiple answers, any one of them will be accepted.

Examples

standard input	standard output
10 3 3 3 2 2 1 4 4 4 4	3 2 1 4
5 2 2 2 2 1	2 2 1

Problem B. Automatic Sprayer 2

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

A farm is divided into $n \times n$ unit squares of n rows and n columns. Let's define (i, j) as the unit square in the i-th row and the j-th column ($1 \le i \le n$, $1 \le j \le n$).

The distance between two squares (i_1, j_1) and (i_2, j_2) is defined to be $d((i_1, j_1), (i_2, j_2)) = |i_1 - i_2| + |j_1 - j_2|$, the Manhattan distance between those two squares.

There are automatic sprayers on this farm that spray fertilizer solution or herbicide so that the owner can produce grain efficiently.

Each sprayer lies entirely in a unit square. The sprayer in (x, y) sprays $A_{x,y}$ liters of solution to all unit squares. $A_{x,y}$ can be any nonnegative integer.

The energy required for the sprayer in (x, y) to spray solution to (i, j) is exactly $d((x, y), (i, j)) \times A_{x,y}$. For each square (i, j), we compute $E_{i,j}$, the sum of energies needed for all sprayers to spray the square (i, j).

Given the matrix E, write a program that generates *any possible* matrix A that corresponds to matrix E. E will be given such that there exists such a matrix A of nonnegative integers whose sum is at most 10^{12}.

Input

The first line contains a single positive integer n ($2 \le n \le 1\,000$).

The next n lines each contain n integers. The j-th ($1 \le j \le n$) integer in the i-th ($1 \le i \le n$) line is $E_{i,j}$ ($0 \le E_{i,j} \le 10^{16}$).

The input is designed such that a matrix A consisting of only non-negative integers whose sum is at most 10^{12} exists which can yield E.

Output

Output n lines, each containing n integers. The y-th ($1 \le y \le n$) integer in the x-th ($1 \le x \le n$) line should be $A_{x,y}$.

Examples

standard input	standard output
5 4 3 2 3 4 3 2 1 2 3 2 1 0 1 2 3 2 1 2 3 4 3 2 3 4	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
6 43 34 25 24 33 42 42 33 24 23 32 41 41 32 23 22 31 40 40 31 22 21 30 39 39 30 21 20 29 38 48 39 30 29 38 47	0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0

Problem C. Beautiful Numbers

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Byteazar calls the positive integer beautiful if the sum of its digits is equal to 10. Byteazar got n cards, each card containing one digit between 1 and 9, inclusively, on the face.

Count the maximal number of the beautiful numbers that can be built from the given cards **simultaneously**, i.e. that any card can be used in no more than one integer.

For example, if we have the cards with digits 1,9,9,7,9,3, we can build two beautiful numbers: first with the digits 1 and 9 and second with the digits 7 and 3. From cards 1,1,1,1,1,1,1,1,1,2,8 we can build only one beautiful number (use 8 and 2, or two ones and 8, or eight ones and 2). From the set of cards 4, 5, 7 we cannot build any beautiful number.

Input

First line of the input contains one integer n ($1 \le n \le 100$) — number of cards. Second line contains n integers a_i ($1 \le a_i \le 9$) — the numbers that are written on the faces of the cards.

Output

Print one integer — the maximal number of the beautiful numbers that Byteazar can build simultaneously from the given cards.

Examples

standard input	standard output
6 1 9 9 7 9 3	2
11 1 1 1 1 1 1 1 1 1 2 8	1
3 4 5 7	0

Problem D. Equivalent Pipelines

Input file:	standard input
Output file:	standard output
Time limit:	3 seconds
Memory limit:	1024 megabytes

You are planning to construct a water pipeline network, connecting n buildings in KAIST. Due to budget problems, you can only use $n-1$ pipes. Each pipe is undirected and connects two different buildings, and all n buildings must be pairwise connected through some sequence of pipes. These pipes form a network.

As a careful planner, you designed d different networks and want to compare them. One can describe each pipe in the network with a durability, which is a single positive integer. Given a network T, define the **vulnerability** $v_T(i,j)$ of two distinct buildings i and j to be the minimum durability of a pipe whose removal separates buildings i and j. In other words, $v_T(i,j)$ is the minimum durability over all pipes on the path connecting i to j.

If two networks T_1 and T_2 satisfy $v_{T_1}(i,j) = v_{T_2}(i,j)$ for all $1 \le i < j \le n$, we say T_1 and T_2 are **equivalent**. To filter out unnecessary plans, group the d designs up to equivalency.

Input

The first line contains two integers d and n ($d \ge 1$, $n \ge 2$, $d \cdot n \le 500\,000$), separated by a space.

From the second line, the descriptions for the d designs are given. Each design is described over $n-1$ lines, each line consisting of three integers a, b and c ($1 \le a, b \le n$, $a \ne b$, $1 \le c \le 10^9$), indicating there is a pipe connecting buildings a and b directly, whose durability is equal to c.

Output

Output d integers in a line. For $1 \le i \le d$, the i-th number should be the minimum index j, where the j-th network in the input is equivalent to the i-th network in the input.

Examples

standard input	standard output
3 3 1 2 1 1 3 1 1 2 1 2 3 1 1 2 1 2 3 2	1 1 3
3 4 1 2 2 2 3 1 3 4 2 1 3 2 2 4 2 2 3 1 1 2 2 1 3 1 3 4 2	1 2 1

Problem E. Flowerbed Redecoration

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Joon-Pyo decorated a flowerbed in front of his home. The flowerbed is in the shape of an $n \times m$ grid, and one flower is planted in each cell. There are 26 colors, one corresponding to each uppercase letter from A to Z. Suddenly, he wanted to redecorate the flowerbed.

The flowerbed is too large to adjust the flowers one by one. He rented some equipment that can lift and rotate a square plot of land with a side length of d. He planned the construction in the following order, expecting the flowerbed to be properly redecorated.

1. Place the equipment so that exactly the flowers in the first d rows and the first d columns are inside.
2. Rotate the $d \times d$ square inside the equipment 90° clockwise. If this square contains flowers from the last d rows and the last d columns, then the construction is finished. Otherwise, if this square does not contain flowers in the last d columns, move the equipment x squares to the right. Otherwise, move the equipment down by y squares and all the way to the left so it contains flowers from the first d columns.
3. Repeat step 2 until construction is finished.

Note that the equipment will never go out of the flowerbed, as x, y, and d are carefully determined before construction begins.

He cannot start construction without knowing the outcome. Write a program that outputs the result.

Input

On the first line, five integers n, m, y, x, and d are given. ($1 \le n \times m \le 10^6$, $1 \le y \le n$, $1 \le x \le m$, $1 \le d \le \min(n, m)$, $\boldsymbol{n \equiv d \pmod{y}}$, $\boldsymbol{m \equiv d \pmod{x}}$).

Each of the next n lines contains exactly m uppercase letters, the current flowerbed.

Output

Output n lines, each containing m uppercase letters, the flowerbed after the planned construction.

Examples

standard input	standard output
4 4 1 1 2 AAAA BBBB AAAA BBBB	BAAA ABBB BAAA BBBA
6 5 1 2 3 RBRCY YBPBR PBRCY CYPBR PBRCY CYPBR	PYRBR CRCBB PPBPY CRCYB YRBCY PYRBR

Problem F. Inspection

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

There are n cities in Byteland and $n-1$ bidirectional highways connecting those cities such as each highway connects exactly two distinct cities and that it is possible to travel between any two cities using one or several highways. For each highway, its length l_i is known.

Byteazar is the road inspector. He plans to check all highways in the country. Byteazar is very experienced, so she can inspect the highway just by visiting any of the two cities connected by that highway. Of course, Byteazar moves between cities using the highway network.

Byteazar starts in city 1. Calculate the minimum summary distance for Byteazar to check all the highways in the Byteland.

Input

The first line of the input contains one integer n ($2 \le n \le 20$). i-th of the following $n-1$ lines contains three integers a_i, b_i and l_i — the cities connected by i-th highway and length of this highway, respectively.

Output

Print one integer — minimal movement distance for Byteasar to check all highways.

Example

standard input	standard output
7 3 1 6 3 4 3 4 5 4 3 6 4 6 7 4 6 2 9	16

Problem G. Or Machine

Input file: `standard input`
Output file: `standard output`
Time limit: 4 seconds
Memory limit: 1024 megabytes

We are developing the Or Machine, a computer heavily optimized solely for one kind of operation: the `|=` operator in C++'s term.

The Or Machine has n registers, each containing a nonnegative integer less than 2^8. We label them $x_1, x_2, \cdots, x_n$. A program is represented by a list of l operations. Each operation is represented by a pair of integers (a, b), meaning that the machine should update x_a with the bitwise OR of x_a's and x_b's values.

The Or Machine takes a program, the initial values of the registers, and a positive integer t. When run, the program performs each operation in the program one by one. When the last operation is performed, it goes back to the first operation and repeats the process. The machine stops after performing exactly t operations.

We want our machine to be much faster than general-purpose computers, and hardware optimization is probably not enough. Can you help us with some software optimization?

Input

The first line contains three integers, n, l, and t ($1 \leq n, l \leq 2^{18}$, $1 \leq t \leq 10^{18}$). l is the length of the program.

The program is given on the next l lines. Each line contains two integers a and b ($1 \leq a, b \leq n$) representing the pair of registers that participate in the given operation.

The final line contains n integers, the initial values of the registers $x_1, \cdots, x_n$ ($0 \leq x_i < 2^8$).

Output

Output n integers on a single line, the values of the registers $x_1, \cdots, x_n$ after t operations.

Example

standard input	standard output
5 4 5 1 2 2 3 2 4 4 4 8 0 5 3 10	15 7 5 3 10

Problem H. Periodic Ruler

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 1024 megabytes

Hitagi has a ruler of infinite length. It has a mark on every integer, where the mark on integer i has color c_i. Each color is represented by an integer from 1 to 100.

She noticed that the ruler's color pattern repeats with a period of t. The period t is defined by the **smallest** positive integer that satisfies $c_i = c_{i+t}$ for all integers i.

Hitagi told Koyomi the colors of n marks of her choice. Koyomi wants to find all positive integers that **cannot** be a period of the ruler, regardless of the colors of unchosen marks. Write a program to find all such numbers, and output their count and sum.

Input

The first line contains a single integer n ($1 \le n \le 50$).

The following n lines each contain two integers x_i ($|x_i| \le 10^9$) and a_i ($1 \le a_i \le 100$). This indicates that the integer x_i is marked with the color a_i.

If $i \neq j$, then $x_i \neq x_j$.

Output

Output two integers on one line. The first integer is the number of positive integers that cannot be the period of the ruler. The second integer is their sum.

Examples

standard input	standard output
3 -1 1 1 2 2 1	2 3
5 1 1 2 1 3 1 4 1 5 1	4 14
1 1000000000 100	0 0

Problem I. Cities Solitaire

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Bytica is playing the solitaire game based on the well-known Cities game.

In the solitaire version, you have the list of n city names, and your goal is to arrange them in sequence such as for any $1 \le i < n$ the last letter of i-th word is the first letter of $i+1$-st word, and the last letter of the n-th word is the first letter of the first word.

Given the list of city names, prepared by Bytica, check if she can reach the goal of the solitaire game.

Input

The first line of the input contains one integer n ($1 \le n \le 10^4$) — the number of cities in the list. Each of the following n lines contains one city name — the non-empty string s_i, composed from lowercase English letters ($1 \le |s_i| \le 30$).

Output

Print "`YES`" if Bytica can reach the goal of the game, or "`NO`" otherwise.

Examples

standard input	standard output
4 kyiv minsk vinnytsia amsterdam	YES
3 paris stockholm miami	NO

Problem J. Three Competitions

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

Last month, n people participated in three competitions. The people are labeled with distinct integers from 1 to n. In each competition, the people were sorted by performance and got ranked from 1 to n. The lower the rank, the better the player. There were no ties in any of the rankings.

Today, instead of n people participating at once, two people compete head-to-head. The winner of the match is the person who wins at least two out of three competitions. The winner then proceeds to compete with another person. This turned out to be quite interesting: even if a person a cannot directly win against another person b, it's possible that another person c wins against b and then a wins against c. That way, we can say that a "indirectly" wins against b. It's also possible that two people can indirectly win against each other!

Formally, a person a is said to **directly win against** another person b if a has a lower rank than b in at least two competitions. Also, a is said to **indirectly win against** b if there exists a sequence of people $p_1, p_2, \cdots, p_k$ ($k \geq 2$) such that p_i directly wins against p_{i+1} for all $i = 1, \cdots, k-1$, $p_1 = a$ and $p_k = b$.

Given the ranks of the people in each competition, answer q questions asking whether person a indirectly wins against another person b.

Input

The first line contains a single integer n ($2 \leq n \leq 2 \cdot 10^5$), the number of people.

Each of the next n lines contains three integers, which represent the ranks of each person in each of the three competitions, in order from person 1 to person n. For each competition, each integer rank from 1 to n appears exactly once.

The next line contains a single integer q ($1 \leq q \leq 2 \cdot 10^5$), the number of questions.

Each of the next q lines contains two integers a and b ($1 \leq a, b \leq n$, $a \neq b$), asking whether person a indirectly wins against person b.

Output

Output Q lines. The i-th line should be either `YES` or `NO`. If person a indirectly wins against person b, output `YES`, otherwise output `NO`.

Example

standard input	standard output
4 2 4 3 3 1 4 4 3 2 1 2 1 3 1 2 2 1 3 4	YES YES NO

Note

Person 1 directly (and indirectly) wins against 2. Person 2 doesn't directly win against 1, but 2 directly wins against 3 and 3 directly wins against 1, so 2 indirectly wins against 1.

Problem K. Organizing Beads

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Hyunuk has a long barrel of n $(2 \le n \le 2 \cdot 10^5)$ cells. Each cell is either empty or contains a bead. When storing beads, it doesn't look good if they are scattered here and there, so Hyunuk wants to gather all the beads at one end. Specifically, if the barrel has k beads, the beads must be in cells 1 through k or in cells $n-k+1$ through k.

Hyunuk can move the bead in the i-th cell of the barrel to the $(i-1)$-th cell or the $(i+1)$-th cell by lightly pushing it. If there is a bead in the cell to be moved, that bead is also pushed in the same direction. For example, let there be beads in the 2nd, 3rd, and 5th cells. If Hyunuk pushes the bead in the 2nd cell to the 3rd cell, the bead in the 3rd cell is also pushed to the 4th cell. The bead in the 5th cell remains in place.

Hyunuk wonders how the minimum number of moves required to organize all beads will change when he adds or removes beads from the barrel. Write a program that calculates the minimum number of moves required to organize the bead barrel as Hyunuk inserts or removes beads from the barrel.

Input

The first line contains a single integer n $(2 \le n \le 2 \cdot 10^5)$, the length of the bead barrel.

The second line contains a string of length n consisting of only `O`'s and `X`'s representing the state of the bead barrel. If the i-th character of the string is `O`, then the cell has a bead in it. Otherwise, the i-th character of the string is `X` and the cell is empty.

The third line contains a single integer q $(1 \le q \le 2 \cdot 10^5)$, the number of actions performed by Hyunuk.

The next q lines each contain a single integer k $(1 \le k \le n)$ representing Hyunuk's action. This means that if there is a bead in the k-th cell of the barrel, the bead is removed, and if there is no bead, it is inserted at the corresponding position.

The input will be designed such that the barrel will never have zero beads.

Output

Output q lines, the i-th of which contains a single integer, the minimum number of moves required to organize all beads after Hyunuk's first i actions.

Example

standard input	standard output
6 OXXOXO 4 3 1 6 3	2 1 2 2