Information

Source code limit

The size of each solution source code can't exceed 256 KiB.

Submissions limit

You can submit at most 50 solutions for each problem.

You can submit a solution to each task at most once per 30 seconds. This restriction does not apply in the last 15 minutes of the contest round.

Testing

Notice, that each subtask has a list of required subtasks. Subtask will be tested only if all tests of all required subtasks are passed. Be careful, some subtasks might not be tested, if your solution doesn't pass sample tests.

Scoring

We have two types of subtask scoring: "test" and "subtask".

"Test" means that points are awarded for each test in a subtask independently of other tests in this subtask.

"Subtask" means that points are awarded only if all tests in this subtask are passed.

For more information on subtask scoring read "Scoring" section of each problem.

The number of points scored for the problem is the total number of points scored on each of its subtasks. The score for the subtask is the maximum number of points earned for this subtask among all the solutions submitted.

Feedback

To get feedback for your solution, go to "Runs" tab in PCMS2 Web Client and use "View Feedback" link. In each problem of the contest you will see the score for each subtask, or the verdict for the first failed test.

Scoreboard

The contestants' scoreboard is available during the contest. Use "Monitor" link in PCMS2 Web Client to access the scoreboard. The standings provided in PCMS2 Web Client are not final.

Problem A. Innohorse

Time limit: 1 second
Memory limit: 512 megabytes

In Innokingdom there are special horses for chess knights, innohorses. Each innohorse is represented by a pair of integers (x, y), $0 \leq x \leq y$. Innohorse moves in the following way: first it moves x cells in one of the four general directions, then turns 90 degrees to the left or to the right, then finally moves y more cells. For instance, an ordinary chess knight horse is an innohorse of type $(1, 2)$.

Alex and Jane has just seen one innohorse, it jumped from cell A to cell B. They wonder what is the type of this innohorse. Help them answer this question.

Input

Chessboard consists of 8 rows and 8 columns. Rows are assigned integers from 1 to 8, and columns are assigned letters from '`a`' to '`h`'. So every cell is represented by a pair of a letter and a digit.

Two lines consist of descriptions of cells A and B respectively.

Output

Print two integers x and y ($0 \leq x \leq y$), representing the type of innohorse.

Scoring

Solution is tested on 10 testcases, each of them giving 10 points. Solution scores the number of points equal to the total number of points for each test.

Feedback

The testing verdict is shown for each testcase.

Examples

standard input	standard output
a1 b3	1 2
g5 d3	2 3

Problem B. Squarow

Time limit:	1 second
Memory limit:	512 megabytes

Moris's favorite computer game is called «Squarow». The game starts with a row of colored squares appearing on the screen. The player can choose a color and remove all the squares of that color (you cannot choose a color that is not among the colors of the squares). After removal, all remaining squares are shifted to the left so that there are no empty spaces in the row between adjacent squares. The order of the squares in the row does not change. All squares of the same color, standing in a row one after another, form a *block*. When you remove a selected color, you get the number of points equal to the number of blocks left in the row.

Help Moris find out what is the largest number of blocks he can get after removing the squares of some color, and what color to choose. If there are several such colors, output any of them.

Input

The first line contains a single integer n ($1 \le n \le 2 \cdot 10^5$), the number of squares in the row.

The second line contains n integers a_i ($1 \le i \le n$, $1 \le a_i \le 2 \cdot 10^5$), the color of the i-th square in the row.

Output

In a single line output two integers: the largest number of blocks that can remain after removing all the squares of one chosen color, and the number corresponding to the color that must be removed to achieve such a number of blocks.

Scoring

Subtask	Score	Constraints		Scoring	Required subtasks
		n	a_i		
1	19	$n \le 100$	$a_i \le 100$	subtask	—
2	43	$n \le 10^4$	$a_i \le 10^3$	subtask	1
3	38	$n \le 2 \cdot 10^5$	$a_i \le 2 \cdot 10^5$	subtask	1, 2

Feedback

The verdict of each test in each subtask is given.

Examples

standard input	standard output

`6`
`1 1 1 2 2 1`
	`1 1`

`12`
`1 1 2 3 3 3 2 1 1 1 2 2`
	`4 3`

`15`
`4 5 5 4 4 4 5 9 9 5 5 5 5 9 9`
	`4 4`

Explanations

In the first example, if you remove color 1, one block of color 2 will remain, and if you remove color 2, there will be one block of color 1 (since the blocks on opposite sides of the block of color 2 will merge into one block).

In the second example, if you remove color 1, there will be three blocks, if you remove color 2, there will be also three blocks, and if you remove color 3, then 4 blocks will remain.

In the third example: for color 4 : 4 blocks, for color 5 : 2 blocks, for color 9 : 4 blocks.

Problem C. Innoforest

Time limit:	2 seconds
Memory limit:	512 megabytes

Trees in Innopolis are exceptional, if you water an innotree (Innopolis Tree), it will grow by number of litres you have watered it. In other words if you water innotree of height h by x litres of water then it will have new height of $h + x$.

Innoforest (Innopolis Forest) is a $n \times m$ grid, each cell of the grid contains one innotree. Irrigation system in innoforest contains $n + m$ canals: one for each row and column. The irrigation system in one operation can water all trees along one of canals by the same amount of water.

The mayor of Innopolis wants to transform the innoforest by performing some operations on the irrigation system. For each tree in innoforest you know its current height and the desired height. Your task is to find the sequence of operations that transforms the innoforest to the desired shape.

Input

First line contains two numbers n and m ($1 \le n, m \le 1000$).

Then n lines follow, each line contains m numbers $a_{i,j}$, current heights of the trees in innoforest ($1 \le a_{i,j} \le 10^9$).

Then n more lines follow, each line contains m numbers $b_{i,j}$, desired heights of the trees ($1 \le b_{i,j} \le 10^9$).

Output

The first line should contain the number of operations k ($0 \le k \le 10^6$), then k lines contain the description of operations.

- "`R r x`" The system will water the r-th row by x litres. ($1 \le r \le n$, $1 \le x \le 10^9$).
- "`C c x`" The system will water the c-th column by x litres. ($1 \le c \le m$, $1 \le x \le 10^9$).

If it is impossible to transform the innoforest to the desired shape, output only one integer `-1`.

Notice that it is not required to minimize the number of operations, only make sure that it does not exceed 10^6.

Scoring

Subtask	Score	Constraints		Scoring	Required subtasks
		n and m	$a_{i,j}$ and $b_{i,j}$		
1	11	$n = 1$ and $m \le 1000$	$a_{i,j}, b_{i,j} \le 10^9$	subtask	—
2	23	$n, m \le 50$	$a_{i,j}, b_{i,j} \le 50$	subtask	—
3	21	$n, m \le 300$	$a_{i,j}, b_{i,j} \le 300$	subtask	2
4	45	$n, m \le 1000$	$a_{i,j}, b_{i,j} \le 10^9$	subtask	1, 2, 3

Feedback

The score for each subtask and the verdict for the first failed testcase is shown.

Examples

standard input	standard output

1 1
4
9
	2
	C 1 2
	R 1 3

3 3
2 2 2
2 2 2
2 2 2
1 1 1
1 1 1
1 1 1
	-1

3 3
1 2 3
4 5 6
7 8 9
2 4 4
5 7 7
7 9 9
	3
	R 1 1
	R 2 1
	C 2 1

Problem D. Stones Distribution

Time limit:	1 second
Memory limit:	512 megabytes

There is an amazingly equipped hi-tech sauna in Innopolis sport center. However, due to complex techniques were used while building it people are not sure how to maintain it properly.

There are $n-1$ consecutive compartments in the sauna. Between each pair of adjacent compartments, there is a stove. There are two more stoves: one connected only to the first compartment, and one connected only to the last, which makes exactly n stoves.

The i-th compartment has volume k_i. Each stove can have from 0 to v stones. Let p_i be the number of stones in the i-th stove, then the i-th compartment receives $k_i \cdot p_i \cdot p_{i+1}$ units of heat.

There are s stove stones in sport center. Sport center management wants to minimize the sum of heat received by all compartments, so the rest of the building would not be heated up, but all stones have to be used as it is a waste to buy them otherwise. Help them solve the problem.

Input

The first line contains three integers n, s and v, number of stoves, number of stones and stove capacity, respectively ($2 \le n \le 1000$, $1 \le v \le 10^5$, $s \le n \cdot v$).

The second line contains $n-1$ integers k_i, the volume of the i-th compartment ($1 \le k_i \le 10^5$).

Output

Print the minimum possible total heat received by all compartments.

Scoring

Subtask	Score	Constraints		Scoring	Required subtasks
		n	v		
1	15	$n \le 1000$	$v = 1$	subtask	—
2	25	$n \le 50$	$v \le 50$	subtask	—
3	30	$n \le 200$	$v \le 100$	subtask	2
4	30	$n \le 1000$	$v \le 10^5$	subtask	1, 2, 3

Feedback

The verdict of each test in each subtask is given.

Example

standard input	standard output
4 10 4 1 2 3	8

Explanation

Correct answer for the sample is achieved by putting four stones in the first and the last stove and by putting two stones in the second. After that, the heat in every compartment except second is equal to zero, while the heat in the second compartment is equal to 8.

Problem E. XOR sum

Time limit: 1 second
Memory limit: 512 megabytes

You have an array of n k-bit numbers $a_1, a_2, \ldots, a_n$.

You need to calculate $\sum_{i=1}^{n} \sum_{j=i+1}^{n} (a_i \oplus a_j)^x$.

Operation $a \oplus b$ is bitwise exclusive OR of two numbers a and b.

Since the answer can be very large, output it modulo 998 244 353.

Input

The first line of the input contains three integers n, k, x ($1 \le n, k, n \cdot k \le 300\,000$, $1 \le x \le 3$) — length of an array, number of bits in each number and the power of exclusive OR operations results in the sum.

Next n lines contain array elements.

The i-th of them contains string $s_0, s_1, \ldots, s_{k-1}$, consisting of '0' and '1'.

Then $a_i = \sum_{i=0}^{k-1} s_i \cdot 2^i$.

Output

Print one number — the remainder of division of the sum of x-th powers of exclusive OR results of all pairs of numbers in the array by 998 244 353.

Scoring

Subtask	Score	Additional constraints	Scoring	Required subtasks
1	5	$n \le 5000, k \le 30$	subtask	—
2	5	$n \cdot k \le 5000$	subtask	—
3	5	$k \le 30$, $a_i = 2^{b_i}$ for some b_i	subtask	—
4	10	$k \le 20$	subtask	—
5	10	$x = 1, n \cdot k \le 300\,000$	subtask	—
6	15	$x \le 2, n \cdot k \le 100\,000$	subtask	—
7	10	$x \le 2, n \cdot k \le 300\,000$	subtask	6
8	15	$n \cdot k \le 100\,000$	subtask	2,6
9	10	$n \cdot k \le 200\,000$	subtask	1,2,6,8
10	15	$n \cdot k \le 300\,000$	subtask	1,2,3,4,5,6,7,8,9

Examples

standard input	standard output

`3 3 1`
`101`
`001`
`001`
	`2`

`2 6 3`
`101111`
`011001`
	`19683`

Explanations

In the first sample array contains integers $[5, 4, 4]$, and the answer is $(5 \oplus 4) + (5 \oplus 4) + (4 \oplus 4) = 1 + 1 + 0 = 2$.

In the second sample array contains integers $[61, 38]$, and the answer is $(61 \oplus 38)^3 = 27^3 = 19683$.