Problem A. Queries

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2.25 seconds
Memory limit:	128 megabytes

XORin discovers an interesting function called Elf. XORina has given XORin an array A of N integers and Q queries. The queries are of 2 types:

1 p x - the element on the position p changes his value to x (A_p=x)

2 a b - find the sum of Elf(i,j) for each $a \leq i \leq j \leq b$, where Elf(i,j)=A_i xor A_{i+1} xor ... xor A_j.

Your task is to print the responses to the 2nd type of queries.

Input

The first line of the input contains n, the size of the array. The second line of the input contains m, the number of queries ($1 \leq n, m \leq 100000$). The third line of the input contains n numbers, the initial elements of the array (each of them is between 1 and 1000). On the next m lines each line contains a query. The first type of query is 1 p x ($1 \leq p \leq n$ and $0 \leq x \leq 1000$). The second type of query is 2 a b ($1 \leq a \leq b \leq n$).

Output

The output contains the answer to the second type of queries, each answer on a line. Print the answer modulo 4001.

Examples

standard input	standard output
4 8 1 2 3 4 2 1 2 1 1 2 2 1 3 2 1 4 1 3 7 2 1 3 1 4 5 2 1 4	6 11 34 23 32

Problem B. Yet another vector problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.25 seconds
Memory limit:	64 megabytes

You're given a 0-based array a of length n. Answer Q queries of form: what is sum of elements a[i], such as i modulo p = q, for given p and q.

Input

The first line of the input contains numbers n and Q ($1 \le n, Q \le 100000$). Next n lines contain the array a. Each element of a is between 1 and 100. Next Q lines describe the queries by two integers p and q ($0 \le q \le p \le n - 1$)

Output

The output contains Q lines, corresponding to answers of the queries.

Examples

standard input	standard output
3 3 1 2 3 0 1 0 2 1 2	6 4 2

Problem C. Broken robot

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.25 seconds
Memory limit:	64 megabytes

Most probably, when you need to move some things from one room to another you just carry those things by hand. However, this is not the case. For such things Informikas uses robots!

The usual task is to take a box from one corner of the room and move it to the opposite one. The room is divided into squares so the robot could easily navigate around (robots, which can navigate in the rooms not divided into squares, are very expensive!). However, in some of those squares there are unmovable obstacles. The robot can not move into those squares, the best he can do is go around.

When the robot was new, he could move stuff around the room in the matter of seconds. However, currently it's a bit broken. It takes one minute to move from one square to the adjacent one. To turn 90 degrees it also takes one minute. And even better, robot can't turn right anymore – only left. Also, robot can only move forward, not sideways – if it is looking at the bottom , but need to move to the square to the right, he first need to turn 90 degrees left (it takes one minute) and then move forward to the target square (one more minute).

You will be given the map of the room. The width and height of the room are always the same and described by the number N. At the beginning the robot is always in the top left square of the room looking south. The target square is one at the bottom right square of the room. When finished, robot should be standing on the bottom right square and also looking south. Your task is to find how long would it take for robot to move from starting position to the target one.

Input

On the first line of the input there is integer N ($1 \leq N \leq 200$) – the size of the field. In the following N lines there are N characters for the each of the squares of the room: `'.'` for unoccupied square and `'#'` for occupied one. It is guaranteed that the top left and bottom right squares are unoccupied and that there always is a path between two of those squares.

Output

Output a single integer – the minimal number of minutes it would take the robot to move from the top left to the bottom right square of the room.

Examples

standard input	standard output
1 .	0
2 .. #.	6
2 .# ..	6

Note

In the first test case robot is already standing in the bottom right square and looking south.

In the second test case robot will turn left once, move to other square, turn left three times and then move forward. This will take $1 + 1 + 3 + 1 = 6$ minutes.

In the third test case robot will go forward, turn left, go forward and then turn left three times. $1 + 1 + 1 + 3 = 6$ minutes in total.

Problem D. Board

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2.5 seconds
Memory limit:	64 megabytes

You are given a matrix A of size $M * N$, where $0 \leq A_{ij} < K$. The rows are numbered from 1 to M, the columns are numbered from 1 to N.

You are allowed to apply 2 types of operations:

- Choose a value R ($1 \leq R \leq M$). For each j where $1 \leq j \leq N$, apply $A(R, j) = (A(R, j) + 1) \,\%\, K$. (In other words, for each cell in the R-th row, add 1 modulo K).
- Choose a value C ($1 \leq C \leq N$). For each i where $1 \leq i \leq M$, apply $A(i, C) = (A(i, C) + 1) \,\%\, K$. (In other words, for each cell in the C-th column, add 1 modulo K). Find the minimum number of operations, to transform the given matrix into all 0.

Input

The first line contains three integers, M, N and K, ($1 \leq M, N \leq 1000$, $1 \leq K \leq 10^9$).

The following M lines describe the matrix. Each line consists of N numbers A_{ij} ($0 \leq A_{ij} < K$).

Output

The minimal number of operations is displayed in the first line.

In the following line you should output M numbers describing how many operations were applied for each row (starting from the first row).

In the following line you should output N numbers describing how many operations were applied for each column (starting from the first column).

Examples

standard input	standard output
3 3 2 0 0 0 1 1 1 1 1 1	2 0 1 1 0 0 0

Problem E. Life as a Monster

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	64 megabytes

In a kingdom there are N people. The i-th person lives on a point (x_i, y_i). You are a monster that needs to go around the kingdom and capture all N people.

- You start at some point (u, v)
- In 1 second, you can move from point (u, v) to point (u', v') where $|u' - u| \leq 1$ and $|v' - v| \leq 1$. This means in 1 second you can move to 8 neighboring points. You can also stay at the same position (which is pointless).

Your journey will be like this:

- You start from (u, v)
- You go to the 1st person at (x_1, y_1), instantly capture him.
- Then you need to go back to your home, and rest for 2 seconds.
- Then you go to the 2nd person at (x_2, y_2), instantly capture him.
- Then you need to go back to your home, and rest for 2 seconds.
- This repeats until you capture all N people.

You have to process Q queries of 2 types:

- 0 – The i-th person move to some new point (x'_i, y'_i).
- 1 – If you start at point (u, v), what is the minimum time to complete your journey?

In this problem, you will be given integer BASE and multiple queries. You will have to set the initial value $T = 0$.

Query of type 0 means that the specified person moves to the point $((a_1 * T + b_1)\%BASE, (a_2 * T + b_2)\%BASE)$.

For query of type 1 you will need to output the minimum time of your journey of capturing all the people, if you would start at the coordinate $((a_1 * T + b_1)\%BASE, (a_2 * T + b_2)\%BASE)$. You should update the value of T with the calculated time.

Input

In the first line of input there are three integers N, Q and $BASE$ ($0 \leq N, Q \leq 10^5$, $0 \leq BASE \leq 10^9$. In the following N lines there are two integers x_i and y_i each ($0 \leq x_i, y_i \leq BASE$) – coordinates of the people.

In the following Q lines there are queries of the following types:

- 0 i a_1 b_1 a_2 b_2
- 1 a_1 b_1 a_2 b_2

Where i is the index of the person ($1 \leq i \leq N$) and a_1, b_1, a_2, b_2 are the values mentioned before ($0 \leq a_1, b_1, a_2, b_2 \leq 10^9$).

Output

For the every query of type 1 output single integer – the minimal time it would take to complete a journey of capturing all the people – each on separate line.

Examples

standard input	standard output
3 4 1000000000 2 2 3 1 7 0 1 2 4 3 5 0 3 1 1 7 3 1 4 3 2 6 0 2 2 7 4 1	28 728

Note

In the sample test case, you process four queries:

1. You calculate the minimal time of journey starting at (4, 5). The answer is 28. You output the answer and update the value T = 28;
2. The 3rd person moves to (29, 199);
3. You calculate the minimal time of journey starting at (115, 62). The answer is 728. You output the answer and update the value T = 728;
4. The 2nd person moves to (1463, 2913);

Problem F. What were those numbers?

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.25 seconds
Memory limit:	64 megabytes

Informikas likes to play a simple game with his friends. First he takes a list of numbers from 1 to N. Later he takes some of those numbers and calculates the average of them. When done, Informikas writes down number N and the average value as a fraction P/Q on a piece of paper.

The next part of the game is easy one. Informikas puts the piece of paper with values N and P/Q in a drawer and doesn't look at it for a couple of days. During this time he usually eats three times a day, solves couple or more programming problems and does other things.

Then there comes the tricky part. After some time, when Informikas is completely sure that he has no memory of the first step of the game, he takes out the piece of paper, looks at the values of N and P/Q and tries to come up with numbers from interval $[1; N]$ so that the average of those numbers would be equal to P/Q. However, he many times fails figuring out those numbers and he ends up wondering if he might have made a mistake in the first step and there are no solutions at all.

Informikas would like to have a program which would find the solution in case he is not able to do that himself. Could you help him?

Input

The first line of input consist of three integer values N, P and Q ($1 \leq N \leq 10^5$, $1 \leq P \leq 10^{10}$, $1 \leq Q \leq 10^5$). It is guaranteed that the fraction can't be simplified (i.e. the greatest common divisor $GCD(P, Q) = 1$).

Output

Output either a set of integers from the range $[1; N]$ having an average value of P/Q or write "IMPOSSIBLE"if there is no way to make this kind of set. If there are multiple solutions, output any of them.

Examples

standard input	standard output
1 1 1	1
2 3 2	1 2
3 7 3	IMPOSSIBLE

Problem G. Old town

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.25 seconds
Memory limit:	64 megabytes

The city of Kaunas has an amazing history, and its old town reveals a lot of secrets. However, this does not amaze Informikas at most, who is keen on the structure of the city roads. The ancient roads connect all parts of the city, and the removal of any of them would make some parts of Kaunas unreachable from other parts (by the old roads). As Informikas is quite smart (he is a software developer), he understands that roads in past were much more reliable, which were built by orcs with huge hammers, with dragons flying around them... As a consequence, the old roads can not be removed or damaged in any way.

In the present times, Informikas is sure that everyone knows the Dijkstra's algorithm, and that is why a lot of new roads are constructed or removed. Those new roads are constructed by ordinary humans, and because of this they can not stand the test of time...

In Informikas's opinion, it is interesting to know the number of important roads after each construction or demolition. A road is called important if after the theoretical removal of it, the city would become disconnected (there would exist two parts of the city that do not have any path between them by the old or new roads).

Input

The first line of the input contains n and m ($1 \le n, m \le 10^5$) - the number of the city parts and the initial number of roads (either old or new).

The following m lines contain 3 integers each, a, b and c ($1 \le a, b \le n$, $a \ne b$, $0 \le c \le 1$), showing the numbers of connected city parts by this road, and $c = 1$ if the road is new, and $c = 0$ if it is old. Two cities can not have more than one road in-between.

The following line contains an integer number of updates (constructions or demolitions), q ($1 \le q \le 10^5$).

The following q lines contain 2 integers each, u and v ($1 \le u, v \le n$, $u \ne v$). If the road between u and v exists, the line corresponds to the deletion of the road, and if not, then to the construction of the road between those cities.

Output

On the first line print the number of important roads of the initial network.

On the following q lines print the numbers of important roads after each modification.

Examples

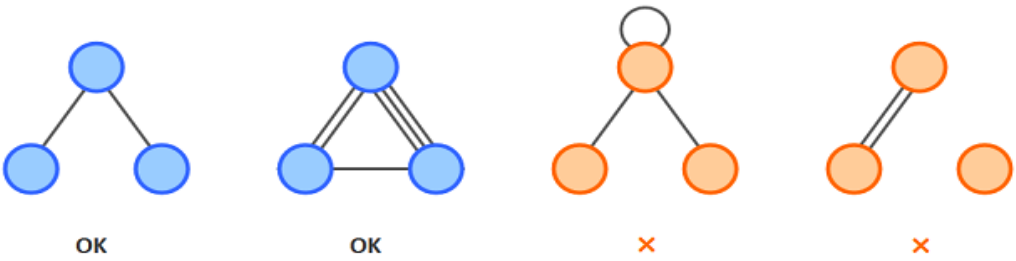
standard input	standard output
6 6 1 2 0 1 5 1 1 4 0 5 4 0 6 4 0 2 3 0 4 6 5 1 5 5 6 3 6	3 2 3 5 1

Problem H. Hard Molecules

Input file:	`stdin`
Output file:	`stdout`
Time limit:	0.5 seconds
Memory limit:	256 megabytes

The mad scientist Dexter performs experiments in molecular chemistry. Today he will try to build a single molecule out of n atoms.

It is well-known that a molecule consists of individual atoms, some pairs of which are connected by atomic bonds. Each atom has its value of valence — the number of the bonds the atom can form with other atoms. An atom can form **one or several** bonds with another atom, but not with itself. The number of the bonds the atom has must be equal to its valency. A molecule must be **connected**, i.e. for each pair of atoms there must be a path of the bonds that connects the two atoms.



Dexter knows the valences of each of the n atoms. Help him find a molecule that can be built out of these atoms, or determine that it is impossible.

Input

The first line of input contains a single integer n ($1 \le n \le 5000$), the number of atoms. The second line contains n space-separated integers d_i ($1 \le d_i \le 10^9$), where d_i is equal to the valence of the i-th atom.

Output

If it is possible to build a single connected molecule out of the given atoms, in the first line output `"Yes"`. Then output $n - 1$ lines, where i-th line must contain $n - i$ space-separated integer numbers. The j-th number in the i-th line must denote the number of the bonds between the atoms with numbers i and $i + j$. The atoms are numbered from 1 to n in the order as they appear in the input. If there are several solutions, output any of them.

If such molecule does not exist, output "`No`" in a single line (without quotes).

Examples

stdin	stdout
2 200 200	Yes 200
3 3 5 4	Yes 2 1 3
3 1 1 1	No

Problem I. Red and yellow

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

You can have one red circle with radius in the range $[A; B]$ and unlimited number of yellow circles with radius R. Task is to put all circles in a way that each yellow circle touches 2 other yellow circles and 1 red circle. A circle cannot be inside another circle. Calculate how many different real sizes (X) of the red circle radius can satisfy this condition.

Input

A single line contains 3 integers A, B and R. ($1 \leq A, B, R \leq 10^7, A \leq B$)

Output

Print a single integer X.

Examples

standard input	standard output
2 2 2	1
1 2 2	2

Problem J. Longest cheap palindrome

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.25 seconds
Memory limit:	64 megabytes

Given a string S of length N, you must find the longest palindromic substring of even length. This sounds like a simple problem, but wait, there's more to it. You also have a budget of B coins and R restrictions that may affect it. For example, a restriction of shape (a, b, c) means that given the case you decided to include in your substring all characters from position a to b in your string S, you must pay c coins. Given the budget and a set of restrictions, output the size of the longest palindromic substring of even length that can be formed.

Input

On the first line there are 3 integers, N ($1 \leq N \leq 33$), the length of the string, R ($1 \leq R \leq 5000$), the number of restrictions and B ($1 \leq B \leq 10^9$), the budget. On the second line you can find string S. Next R lines contain three numbers: a, b, and c describing a restriction.

Output

Output a single number, the maximal length of the palindromic substring.

Examples

standard input	standard output
5 2 9 acbba 1 2 10 4 5 11	2
4 2 5 xyyx 3 4 2 2 3 3	4

Problem K. Easy vector

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.75 seconds
Memory limit:	64 megabytes

You're given a vector with n elements. Piro starts from cell 1. He walks x seconds. In each second, he can move to an adjacent cell: for a cell i with $2 \leq i \leq n$, he can move to $i-1$ and $i+1$. From cell 1 he can only move to cell 2 and from cell n he can only move to cell n-1.

Initially, his sum is what's written on cell 1, from where he starts. Next x seconds, his sum increases by what's written on each cell he visits during those x seconds. What's the maximum sum he can get? Formally, given vector A, choose i_1, i_2, ..., i_x such as you maximize the sum A[1] + A[i1] + A[i2] + ... + A[ix], where $1 \leq i_k \leq n$ and $|i_k - i_{k-1}| = 1$.

Given an array and q queries, each query containing a value x, please answer all of them!

Input

The first line contains numbers n and q ($1 \leq n, q \leq 10^5$, n is at least 2). Next n lines contain the values of vector: each element of vector is between 1 and 10^5. Next q lines contain values of queries: each lines contains a new value for x ($1 \leq x \leq 10^9$).

Output

Output q lines, containing the answers for each query.

Examples

standard input	standard output
3 2 5 2 1 1 2	7 12

Problem L. Many recursions

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.25 seconds
Memory limit:	64 megabytes

It was a long time since Informikas has had something to do with recursions and this had to be fixed. For that reason he decided to come up with some recursion and later solve it all by himself.

The first recursion Informikas managed to come up was

$$f(k) = \begin{cases} 1 + f(k+1) & ,"when" k < 10; \\ 10 & ,"otherwise"; \end{cases}$$

"Nah, too easy he told to himself, crumpled the paper and threw it to a trash can.

$$g(k) = \begin{cases} g(k-1) + g^2(k-2) & ,"when" k > 0; \\ 1 & ,"otherwise"; \end{cases}$$

"Squaring is overrated. Still too easy he thought while the paper was flying towards a trash can.

$h(k) = 1 + \frac{a-k}{1+k} \times h(1+k)$;

"Oh no, what have I done! This recursion has no terminating condition! It's too hard even for me! Informikas shouted out loud.

Could you be so kind and, given the value of a, help out Informikas by finding the value of $h(0)$.

Input

Single integer a ($0 \leq a \leq 10^18$).

Output

Calculate the integer part of $h(0)$. As the answer can be quite a large number, output it modulo $10^9 + 7$ (1000000007). In cases when the recursion goes infinitely long, output "Nobody got time for that"(without quotation marks).

Examples

standard input	standard output
0	1
1	2