Codeforces Round #234 (Div. 2)

A. Inna and Choose Options

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

There always is something to choose from! And now, instead of "Noughts and Crosses", Inna choose a very unusual upgrade of this game. The rules of the game are given below:

There is one person playing the game. Before the beginning of the game he puts 12 cards in a row on the table. Each card contains a character: "X" or "0". Then the player chooses two positive integers a and b $(a \cdot b = 12)$, after that he makes a table of size $a \times b$ from the cards he put on the table as follows: the first b cards form the first row of the table, the second b cards form the second row of the table and so on, the last b cards form the last (number a) row of the table. The player wins if some column of the table contain characters "X" on all cards. Otherwise, the player loses.

Inna has already put 12 cards on the table in a row. But unfortunately, she doesn't know what numbers a and b to choose. Help her win the game: print to her all the possible ways of numbers a, b that she can choose and win.

Input

The first line of the input contains integer t $(1 \leq t \leq 100)$. This value shows the number of sets of test data in the input. Next follows the description of each of the t tests on a separate line.

The description of each test is a string consisting of 12 characters, each character is either "X", or "0". The i-th character of the string shows the character that is written on the i-th card from the start.

Output

For each test, print the answer to the test on a single line. The first number in the line must represent the number of distinct ways to choose the pair a, b. Next, print on this line the pairs in the format axb. Print the pairs in the order of increasing first parameter (a). Separate the pairs in the line by whitespaces.

Examples

input
4 OXXXOXOOXOOX OXOXOXOXOXOX XXXXXXXXXXXX OOOOOOOOOOOO
output
3 1x12 2x6 4x3 4 1x12 2x6 3x4 6x2 6 1x12 2x6 3x4 4x3 6x2 12x1 0

B. Inna and New Matrix of Candies

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Inna likes sweets and a game called the "Candy Matrix". Today, she came up with the new game "Candy Matrix 2: Reload".

The field for the new game is a rectangle table of size $n \times m$. Each line of the table contains one cell with a dwarf figurine, one cell with a candy, the other cells of the line are empty. The game lasts for several moves. During each move the player should choose **all lines of the matrix where dwarf is not on the cell with candy** and shout "Let's go!". After that, all the dwarves from the chosen lines start to **simultaneously** move to the right. During each second, each dwarf goes to the adjacent cell that is located to the right of its current cell. The movement continues until one of the following events occurs:

- some dwarf in one of the chosen lines is located in the rightmost cell of his row;
- some dwarf in the chosen lines is located in the cell with the candy.

The point of the game is to transport all the dwarves to the candy cells.

Inna is fabulous, as she came up with such an interesting game. But what about you? Your task is to play this game optimally well. Specifically, you should say by the given game field what minimum number of moves the player needs to reach the goal of the game.

Input

The first line of the input contains two integers n and m ($1 \le n \le 1000$; $2 \le m \le 1000$).

Next n lines each contain m characters — the game field for the "Candy Martix 2: Reload". Character "*" represents an empty cell of the field, character "G" represents a dwarf and character "S" represents a candy. The matrix doesn't contain other characters. It is guaranteed that each line contains exactly one character "G" and one character "S".

Output

In a single line print a single integer — either the minimum number of moves needed to achieve the aim of the game, or -1, if the aim cannot be achieved on the given game field.

Examples

input
3 4 *G*S G**S *G*S
output
2

input
1 3 S*G
output
-1

C. Inna and Huge Candy Matrix

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Inna and Dima decided to surprise Sereja. They brought a really huge candy matrix, it's big even for Sereja! Let's number the rows of the giant matrix from 1 to n from top to bottom and the columns — from 1 to m, from left to right. We'll represent the cell on the intersection of the i-th row and j-th column as (i, j). Just as is expected, some cells of the giant candy matrix contain candies. Overall the matrix has p candies: the k-th candy is at cell (x_k, y_k).

The time moved closer to dinner and Inna was already going to eat p of her favourite sweets from the matrix, when suddenly Sereja (for the reason he didn't share with anyone) rotated the matrix x times clockwise by 90 degrees. Then he performed the horizontal rotate of the matrix y times. And then he rotated the matrix z times counterclockwise by 90 degrees. The figure below shows how the rotates of the matrix looks like.

Inna got really upset, but Duma suddenly understood two things: the candies didn't get damaged and he remembered which cells contained Inna's favourite sweets before Sereja's strange actions. Help guys to find the new coordinates in the candy matrix after the transformation Sereja made!

Input

The first line of the input contains fix integers n, m, x, y, z, p ($1 \le n, m \le 10^9$; $0 \le x, y, z \le 10^9$; $1 \le p \le 10^5$).

Each of the following p lines contains two integers x_k, y_k ($1 \le x_k \le n$; $1 \le y_k \le m$) — the initial coordinates of the k-th candy. Two candies can lie on the same cell.

Output

For each of the p candies, print on a single line its space-separated new coordinates.

Examples

input
3 3 3 1 1 9 1 1 1 2 1 3 2 1 2 2 2 3 3 1 3 2 3 3

output
1 3 1 2 1 1 2 3 2 2 2 1 3 3 3 2 3 1

Note

Just for clarity. Horizontal rotating is like a mirroring of the matrix. For matrix:

QWER REWQ
ASDF -> FDSA
ZXCV VCXZ

D. Dima and Bacteria

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Dima took up the biology of bacteria, as a result of his experiments, he invented k types of bacteria. Overall, there are n bacteria at his laboratory right now, and the number of bacteria of type i equals c_i. For convenience, we will assume that all the bacteria are numbered from 1 to n. The bacteria of type c_i are numbered from $\sum_{j=1}^{i-1} c_j + 1$ to $\sum_{j=1}^{i} c_j$.

With the help of special equipment Dima can move energy from some bacteria into some other one. Of course, the use of such equipment is not free. Dima knows m ways to move energy from some bacteria to another one. The way with number i can be described with integers u_i, v_i and x_i mean that this way allows moving energy from bacteria with number u_i to bacteria with number v_i or vice versa for x_i dollars.

Dima's Chef (Inna) calls the type-distribution *correct* if there is a way (may be non-direct) to move energy from any bacteria of the particular type to any other bacteria of the same type (between any two bacteria of the same type) for zero cost.

As for correct type-distribution the cost of moving the energy depends only on the types of bacteria help Inna to determine is the type-distribution correct? If it is, print the matrix d with size $k \times k$. Cell $d[i][j]$ of this matrix must be equal to the minimal possible cost of energy-moving from bacteria with type i to bacteria with type j.

Input

The first line contains three integers n, m, k $(1 \le n \le 10^5; 0 \le m \le 10^5; 1 \le k \le 500)$. The next line contains k integers $c_1, c_2, \ldots, c_k$ $(1 \le c_i \le n)$. Each of the next m lines contains three integers u_i, v_i, x_i $(1 \le u_i, v_i \le 10^5; 0 \le x_i \le 10^4)$. It is guaranteed that $\sum_{i=1}^{k} c_i = n$.

Output

If Dima's type-distribution is correct, print string «Yes», and then k lines: in the i-th line print integers $d[i][1], d[i][2], \ldots, d[i][k]$ $(d[i][i] = 0)$. If there is no way to move energy from bacteria i to bacteria j appropriate $d[i][j]$ must equal to -1. If the type-distribution isn't correct print «No».

Examples

input
4 4 2 1 3 2 3 0 3 4 0 2 4 1 2 1 2
output
Yes 0 2 2 0

input
3 1 2 2 1 1 2 0
output
Yes 0 -1 -1 0

input
3 2 2 2 1 1 2 0 2 3 1
output
Yes 0 1 1 0

input
3 0 2 1 2
output

No

E. Inna and Binary Logic

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Inna is fed up with jokes about female logic. So she started using binary logic instead.

Inna has an array of n elements $a_1[1], a_1[2], \ldots, a_1[n]$. Girl likes to train in her binary logic, so she does an exercise consisting of n stages: on the first stage Inna writes out all numbers from array a_1, on the i-th $(i \geq 2)$ stage girl writes all elements of array a_i, which consists of $n - i + 1$ integers; the k-th integer of array a_i is defined as follows: $a_i[k] = a_{i-1}[k]$ *AND* $a_{i-1}[k+1]$. Here AND is bit-wise binary logical operation.

Dima decided to check Inna's skill. He asks Inna to change array, perform the exercise and say the sum of all $\frac{n(n+1)}{2}$ elements she wrote out during the current exercise.

Help Inna to answer the questions!

Input

The first line contains two integers n and m $(1 \leq n, m \leq 10^5)$ — size of array a_1 and number of Dima's questions. Next line contains n integers $a_1[1], a_1[2], \ldots, a_1[n]$ $(0 \leq a_i \leq 10^5)$ — initial array elements.

Each of next m lines contains two integers — Dima's question description. Each question consists of two integers p_i, v_i $(1 \leq p_i \leq n; 0 \leq v_i \leq 10^5)$. For this question Inna should make $a_1[p_i]$ equals v_i, and then perform the exercise. Please, note that changes are saved from question to question.

Output

For each question print Inna's answer on a single line.

Examples

input
3 4 1 1 1 1 1 2 2 3 2 1 2
output
6 4 7 12

Codeforces (c) Copyright 2010-2016 Mike Mirzayanov
The only programming contests Web 2.0 platform