The 2021 ICPC Caribbean Finals Qualifier

Real Contest Problem Set

Problem set developers

Alberto González Rosales

Alfonso^2 Peterssen

Aurora Gil Pons

Carlos Joa

Daniel Cordovés Borroto

Dennis Gómez Cruz

Dovier Ripoll

Ernesto David Peña Herrera

José Carlos Gutiérrez

Leandro Castillo

Marcelo Fornet Fornés

Rodrigo Chaves

Rubén Alcolea Núñez

March 5th, 2022

Problem A. Alternating Function

Let h be a function defined on the set of all positive integers such that $h(n) = (-1)^n \cdot n \cdot p + k$, $(1 \le p, k \le 10^{17})$.
Find the minimum positive integer x such that $h(x) \ge m$, $(1 \le m \le 10^{17})$. It can be proven that such x exists.

Input

The first line of the input contains a single integer t $(1 \le t \le 10^5)$, representing the number of test cases. The following t lines contain three spaced separated integers p, k, m, describing each test case.

Output

For each test case, print a line with one integer, the minimum positive integer x such that $h(x) \ge m$.

Example

standard input	standard output
3 1 3 200 99 1 100 1 70 90	198 2 20
2 567 23 900000000 7 3 100000000000000000	1587302 14285714285714286

Problem B. Board Game

Alice and Bob decide to try a new board game. The game has a board of dimensions $n \cdot m$, c coins located in specific cells on the board and an integer r ($1 \le r \le m$), chosen by Alice and Bob at the beginning of each match. The board game is a turn-based game, and courtesy of Bob, Alice is the first player in every match. On its turn, the player must choose a coin located in a cell (i, j) such that $(1 \le i \le n, 1 \le j \le r)$ and move it to another cell (i, k) of the same row of the board that is to the left of (i, j), it does not matter if this new cell already contains another coin. The player who cannot make a move loses. Alice and Bob decide to play q matches, it is wanted to determine the winner of each match, knowing that both players play optimally.

Input

First line contains four integers n, m, c and q ($1 \le n, m, c, q \le 10^5$), which represent the dimensions of the board, the number of coins and number of matches, respectively. Next c lines contain two integers x, y ($1 \le x \le n$, $1 \le y \le m$) each, the cell where the i-th coin is located. The next q lines contain an integer r ($1 \le r \le m$), the integer chosen by Alice and Bob for each match.

Output

For each of the q matches, print the winner of the game,"Alice" or "Bob"(without quotes).

Example

standard input	standard output
4 4 3 3 1 1 1 3 2 4 3 4 2	Alice Alice Bob

Problem C. Counting Products

You are given the integers n and k. You want to calculate how many different integers x ($1 \leq x \leq n$) can be obtained as a product of x_i such that:

- $x_1 + x_2 + \cdots + x_m = k$
- $x_i \geq 1$ and $m \geq 1$

As an example, assume that $n = 20$ and $k = 8$, then:

- As $1 + 2 + 2 + 3 = 8$, product will be $1 \cdot 2 \cdot 2 \cdot 3 = 12$
- As $4 + 2 + 2 = 8$, product will be $4 \cdot 2 \cdot 2 = 16$
- As $8 = 8$, product will be $8 = 8$
- As $1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 8$, product will be $1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 1$

So 1, 12, 16, 8 and others can be obtained.

Input

First line contains integers n and k ($1 \leq n, k \leq 1000$).

Output

On a single line print the answer to the problem.

Example

standard input	standard output
20 8	14

Problem D. Decorating Trees

You are given a tree with n vertices numbered from 1 to n and rooted at vertex 1. Initially, each vertex has a color c_i.
It is needed to perform q queries of the following types:

1. Update the colors of all vertices in the subtree of vertex v. For each vertex of the subtree replace its color by the formula: $c_i = (c_i + 1)\ mod\ 64$. Where $x\ mod\ y$ stands for the remainder that results of dividing x by y.
2. Count the number of vertices in the subtree of vertex v with color c.

Input

First line contains two integers n and q ($1 \leq n, q \leq 10^5$), the number of vertices in the tree and the number of queries to perform.
Second line contains n integers c_i ($0 \leq c_i \leq 63$), the initial colors of the vertices.
Next $N - 1$ lines contain two integers a and b ($1 \leq a, b \leq n$) representing the edges of the tree.
Last q lines contain the description of queries in the format described below:

$1\ v$: Update the colors of all vertices in the subtree of vertex v ($1 \leq v \leq n$).

$2\ v\ c$: Count the number of vertices in the subtree of vertex v ($1 \leq v \leq n$) with color c ($0 \leq c \leq 63$).

Output

Print the solution for each query of type 2. All solutions should be printed on a separate line following the same order given in the input.

Example

standard input	standard output
7 8 1 2 3 1 1 1 1 1 2 1 3 1 4 3 7 3 5 3 6 2 1 1 1 3 2 3 2 1 2 1 1 2 3 5 2 1 2 2 1 3	5 3 1 2 3

Problem E. Encoding Matrices

Farmer John (FJ) is teaching binary numbers to his cows and they learned quickly that binary numbers have only digits **0** and **1**. FJ was very happy with the obtained results, and then he decided to teach them how to create square binary matrices. However, the cows got bored after the second class. FJ was a bit sad and he thought, what if I teach my cows to encode binary matrices with other symbols?
FJ knows that his cows are not very clever. That's why he defined two simple rules for encoding binary matrices:

1. The most frequent bit will be encoded with the symbol **'*'** and the least frequent bit will be encoded with the symbol **'o'**.
2. In case of tie, the bit located at the top-left corner of the matrix will be encoded with the symbol **'*'** and the complementary bit will be encoded with the symbol **'o'**.

Apparently, the cows understood these rules. However, FJ is not sure and wants to evaluate cows' skill. Write a program to encode a square binary matrix using the rules proposed by FJ.

Input

The first line of the input contains an integer n ($1 \leq n \leq 100$) representing the dimension of the matrix. The following n lines contains n binary symbols **'0'** or **'1'** without spaces.

Output

The output contains the matrix obtained with the encoding mechanism proposed by FJ.

Example

standard input	standard output
6 111000 001010 110010 001101 001110 111111	***ooo oo*o*o **oo*o oo**o* oo***o ******
2 00 11	** oo

Problem F. Fading Polygon

There are n points in a plane. Each point will be removed with probability 0.5 independently. Determine the expected value of the area of the *convex hull* of the remaining points.

Input

The first line contains an integer n ($1 \le n \le 2000$).
The next n lines contain integer pairs x_i, y_i ($-10^9 \le x_i, y_i \le 10^9$), the coordinates of the i-th point. No three points are collinear.

Output

Print the expected value of the area of the *convex hull*. Print the value $P \cdot Q^{-1} \mod (10^9 + 7)$, where P and Q are coprime and $\frac{P}{Q}$ is the answer to the problem.

Example

standard input	standard output
4 0 0 0 1 1 0 1 1	687500005
4 -9 0 0 8 7 0 -1 1	12

Problem G. Guided by XOR

You are given an array A of n integers and an integer k. You start on the first position and you can only move to the right. Formally if you are in the position i you can go to all the positions j such that $(i < j \leq n$ and $A_i \oplus A_j < k)$, where $\oplus$ denote the *xor* operation (exclusive or).
Count how many different ways there are to get to the n-*th* position. Two ways are different if there exists at least one position that you visit in one way and not in the other. As the answer can be very large print it modulo $10^9 + 7$.

Input

First line of the input contains two integers n, k $(1 \leq n \leq 2 \cdot 10^5, 0 \leq k \leq 2^{20})$.
Second line of the input contains n integers $a_1, a_2, \cdots, a_n$ $(0 \leq a_i \leq 2^{20})$, where a_i is the value of the i-*th* position of the array.

Output

Print one integer, the numbers of ways to get to the n-*th* position modulo $10^9 + 7$.

Example

standard input	standard output
3 4 1 3 3	2
4 4 4 3 4 1	0
8 4 2 3 5 6 3 1 4 1	8

Problem H. Heavy Graph

You are given a weighted and undirected graph with V vertices and E edges. Each vertex has an associated weight $p_1, p_2, \ldots, p_V$.

Find a subset S of vertices with maximum score, where the score is defined as the sum of the weights of all nodes in S and the weights of all the edges between nodes in S, divided by the number of vertices in S:

$$score(S) = \frac{\sum\limits_{u \in S} p_u + \sum\limits_{\substack{(u,v,w) \in E \\ u,v \in S}} w}{|S|}$$

Input

The first line contains two integers V ($1 \leq V \leq 100$) and E ($1 \leq E \leq 1000$).

The next line consist of V integers $p_1, p_2, \ldots, p_V$ ($0 \leq p_i \leq 1000$).

Then E lines, each containing three integers u_i, v_i, w_i ($1 \leq u_i \neq v_i \leq V$, $1 \leq w_i \leq 1000$, $1 \leq i \leq E$) denoting an edge between nodes u_i and v_i with weight w_i.

There will be at most one undirected edge between any pair of vertices.

Output

Print a subset of vertices S with maximum score as follows:

A line with an integer $|S|$, the number of vertices in S.

A line with $|S|$ integers separated by spaces: the vertices of S in any order.

If there's more than one subgraph with the maximum score, print any of them.

Example

standard input	standard output
3 3 10 5 8 1 2 10 1 3 1 2 3 2	2 1 2

Problem I. Isorectangle Triangle

You are given the coordinates of n points and a set of m isosceles right triangles. The legs (shortest sides) of the triangles are parallel to the coordinate axes.
For each triangle, you have to find how many of the given points it contains.

Input

First line contains two positive integers n and m ($1 \leq n, m \leq 10^5$), the numbers of points and the number of triangles respectively.
Next n lines contain two integers x y ($-10^9 \leq x, y \leq 10^9$), the coordinates of each of the given points.
Next m lines contain six integers x_1 y_1 x_2 y_2 x_3 y_3 ($-10^9 \leq x_1, y_1, x_2, y_2, x_3, y_3 \leq 10^9$), the coordinates of the vertices of the given triangles.

Output

Print m numbers in separated lines, the number of points on each triangle.

Example

standard input	standard output
3 2 1 1 -8 -4 -8 -4 0 0 -100 0 0 -100 2 3 3 3 2 4	2 0
9 5 0 0 10 0 20 0 0 10 10 10 20 10 0 20 10 20 20 20 20 0 20 20 0 0 0 20 0 -20 40 20 0 20 0 -19 39 20 9 9 12 9 9 12 -10 -10 -15 -10 -10 -15	6 9 8 1 0
5 4 1000000000 1000000000 -1000000000 1000000000 1000000000 -1000000000 -1000000000 -1000000000 0 0 1000000000 1000000000 -1000000000 1000000000 1000000000 -1000000000 1000000000 -1000000000 1000000000 1000000000 -1000000000 -1000000000 -1000000000 1000000000 -1000000000 -1000000000 1000000000 1000000000 -1000000000 -1000000000 1000000000 -1000000000 -1000000000 1000000000	4 4 4 4

Problem J. Joining Cities

A new city has been founded! The city has n houses, but no roads yet. The city council hired a construction company to build roads between the houses, so that one can go from any house to another by a road path. The cost of building a road is calculated depending on the value a of each house. The cost of building the road between house i and house j, is the absolute value of the difference between their values: $|a_i - a_j|$. The council needs to minimize the total cost of building the roads.

Input

First line contains the integer n ($1 \leq n \leq 10^5$), the amount of houses in the city. Second line contains n integers separated by spaces $a_1, a_2, ..., a_n$ ($1 \leq a_i \leq 10^9$), the value of each house.

Output

Print on a single line the minimum total cost to connect the houses in the city.

Example

standard input	standard output
2 1 1	0
5 10 10 9 2 3	8
9 8 12 4 10 11 1 2 5 5	11

Problem K. Koa the Koala

There are n cats and n dogs, each numbered from 1 to n. From their locations each of them can look other cats and dogs. In particular the following is happening:

1. Every **cat** looks exactly a **dogs** (ie. a distinct integers from 1 to n).
2. Every **dog** looks exactly b **cats** (ie. b distinct integers from 1 to n).

Something really bad will happen if a cat and a dog are looking to each other at the same time, so Koa the Koala will help to sort out this situation. She has to determine whether it is possible to arrange which are the target animals each animal is looking to in such a way that:

- (1.) and (2.) are satisfied
- there isn't a pair of **cat** and **dog** looking to each other, that is: there can't exists integers i and j $(1 \leq i, j \leq n)$ such that cat i is looking to dog j and dog j is looking to cat i.

Help her!

Input

First line of the input contains one integer t $(1 \leq t \leq 100)$, the number of test cases. Then t test cases follow.
The only line of each test case contains three integers n, a and b $(1 \leq n \leq 100\,; 1 \leq a, b \leq n)$.
It is guaranteed that the sum of n over all test cases does not exceed 100 $(\sum n \leq 100)$.

Output

For each test case: Print "Yes" or "No" (without quotes), depending on whether such arrangement exists.
If the answer is "Yes":

- Then exactly n lines must follow, which dogs is looking each cat.
- The i-th $(1 \leq i \leq n)$ line must consist of exactly a distinct integers $w_1, w_2, \ldots, w_a$ $(1 \leq w_i \leq n)$ indicating that cat i looks dogs $w_1, w_2, \ldots, w_a$. These integers can be in any order.
- Then exactly n lines must follow, which cats is looking each dog.
- The i-th $(1 \leq i \leq n)$ line must consist of exactly b distinct integers $m_1, m_2, \ldots, m_b$ $(1 \leq m_i \leq n)$, indicating that dog i looks cats $m_1, m_2, \ldots, m_b$. These integers can be in any order.

If there are many possible arrangements print any.

Example

standard input	standard output
4 3 1 2 5 4 4 7 7 6 2 1 1	Yes 1 2 3 2 3 1 3 1 2 No No Yes 1 2 2 1

Problem L. LCS Recovery

The *LCS* (Longest Common Subsequence) algorithm of two binary strings A and B returns a matrix as follows:

```
LCS(A, B):
    n = len(A)
    m = len(B)
    M = Array[n, m] // the matrix is full of zeroes
    for i = 1 to n
        for j = 1 to m
            if A[i] == B[j]
                M[i][j] = M[i-1][j-1] + 1
            else
                M[i][j] = max(M[i-1][j], M[i][j-1])
    return M
```

Given a matrix M, find the smallest among all possible pairs of binary strings A and B such that *LCS(A, B)* returns M. A pair (A, B) is smaller than (C, D) if $(A + B)$ is lexicographically smaller than $(C + D)$ where the operator $+$ denotes string concatenation.

Input

First line of the input contains two integers n and m ($1 \leq n, m \leq 2 \cdot 10^3$), the number of rows and the number of columns in the matrix.

Next n lines contain m integers each. The j-*th* element in the i-*th* line is $M[i][j]$.

It is guaranteed that there exists at least two binary strings such that the *LCS* algorithm returns the given matrix.

Please note that the given matrix differs from the one at the pseudocode by not having the row number 0 and column number 0 for simplicity.

Output

Print one binary string A of length n on the first line and one binary string B of length m on the second line, such that the pair (A, B) is the smallest lexicographically where the *LCS* algorithm returns the given matrix.

Example

standard input	standard output
2 3 0 1 1 0 1 2	00 100
3 4 0 1 1 1 0 1 2 2 0 1 2 3	000 1000
5 5 0 0 1 1 1 1 1 1 1 2 1 2 2 2 2 1 2 2 2 3 1 2 2 2 3	01111 11001

Problem M. Matrix Parity

We call a matrix of integers "even" if the sum of the numbers in each row is even and the sum of the numbers in each column is even.
In the following examples, matrices A, B and C are "even". Matrix D is not "even" because the second column and the last row have odd sums. Matrix E is not "even" because its two columns have odd sums.

$$A = \begin{bmatrix} 2 \end{bmatrix}, B = \begin{bmatrix} 1 & 3 & 0 \\ 3 & 5 & 6 \end{bmatrix}, C = \begin{bmatrix} 3 & 1 \\ 1 & 9 \end{bmatrix}, D = \begin{bmatrix} 1 & 1 & 8 \\ 3 & 0 & 7 \\ 0 & 4 & 3 \end{bmatrix}, E = \begin{bmatrix} 7 & 3 \end{bmatrix}$$

Given a matrix A, convert it into an "even matrix" by performing **as few as possible** of the following type of operation:

- Select a cell of the matrix and increment its value by 1. You may select the same cell in different operations.

Print any final "even" matrix that results of executing the minimum number of operations.

Input

The first line contains two integers r and c ($1 \le r, c \le 50$). Following r lines define matrix A. Each line contains c integers with values between 0 and 100. The j-*th* number of the i-*th* line is the cell $A_{i,j}$ of the matrix.

Output

Print any "even matrix" that results from performing the minimum number of operations on matrix A. Then print r rows, with c numbers in each row, describing the matrix after applying the operations.

Example

standard input	standard output
2 3 1 9 3 2 8 9	2 9 3 2 9 9
2 5 1 2 3 4 5 3 1 4 1 5	1 2 4 4 5 3 2 4 2 5
1 2 2 6	2 6

Problem N. New Combination

You are given a set P with n points in the plane. We want to know if a point w can be written as a linear combination of points from R, where $R \subseteq P$, that is:

$$w = \sum_{i=1}^{|R|} \mu_i \cdot R_i$$

such that $\mu_i \geq 0$ for $i = 1, 2, ..., |R|$ and $\sum_{i=1}^{|R|} \mu_i = 1$.

For each point w you should find a set R with **at most 5** elements ($|R| \leq 5$) such that coefficients satisfy previous rule, or tell if such representation doesn't exist.

Notes:

- Let $p = (x, y)$ be a point and c a real number. The product $c \cdot p$ is defined as point $s = (c \cdot x, c \cdot y)$.
- Let $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ be two points. The sum $p_1 + p_2$ is defined as point $s = (x_1 + x_2, y_1 + y_2)$.

Input

First line of input contains an integer n ($1 \leq n \leq 10^5$), the number of points in P.
Next n lines contains two integers x_i, y_i ($0 \leq x_i, y_i \leq 10^9$), the coordinates of the i-th point.
Next line contains an integer q ($1 \leq q \leq 10^4$), representing the amount of points w to answer.
The remaining q lines contains two integers $x_w, y_w (0 \leq x_w, y_w \leq 10^9)$, the coordinates of each point w.

Output

For each point w:

- If there is no way to represent w as described, print the word "impossible" (without the quotes).
- If there is a solution, print a line with an integer m, indicating the amount of elements in R.

Then print m lines, each of them with an integer i indicating that the i-th point of P belongs to R, followed by a real number μ_i indicating its coefficient.

Note: Your solution will be considered correct if it follows all the described restrictions and the squared distance between w and the point $w' = \sum_{i=1}^{|R|} \mu_i \cdot R_i$ does not exceed 10^{-5}.

Example

standard input	standard output
6 0 0 2 0 4 1 0 2 2 2 3 3 3 1 1 2 1 3 0	4 1 0.25 2 0.25 4 0.25 5 0.25 3 1 0.44444444444444444444 3 0.33333333333333333332 6 0.22222222222222222223 impossible
2 1 3 5 3 3 4 2 3 3 8 3	impossible 2 1 0.5 2 0.5 impossible

Problem O. Or Exclusive Sum

Given a list A with n positive integers. You are allowed to change at most two elements from the list into any two elements of your choice.
Your goal is to maximize the formula:

$$S = \sum_{i=1}^{n-1} A_i \oplus A_{i+1}$$

where the symbol $\oplus$ stands for the *xor* operation (exclusive or).

Input

First line of input contains an integer n ($2 \leq n \leq 10^5$), the size of the list. Second line contains the elements of A separated by space such that ($0 \leq A_i < 2^{30}$) for $i = 1, 2, \ldots, n$.

Output

Print a single integer, the maximum value of S you can get.
Note: The new values of A must remain withing the range $[0, 2^{30})$.

Example

standard input	standard output
2 0 1	1073741823
3 1 2 3	2147483646