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. Función Alternante

Sea h una función definida en el conjunto de los números enteros positivos tal que $h(n) = (-1)^n \cdot n \cdot p + k$, $(1 \leq p, k \leq 10^{17})$.
Halla el menor entero positivo x tal que $h(x) \geq m$, $(1 \leq m \leq 10^{17})$. El número x siempre existe.

Input

La primera línea de la entrada contiene un número entero t $(1 \leq t \leq 10^5)$, la cantidad de casos de prueba. Las siguientes t líneas contienen 3 números enteros separados por espacios p, k, m, describiendo cada caso de prueba.

Output

Para cada caso de prueba, imprima una línea con un número, el menor entero positivo x tal que $h(x) \geq m$.

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. Juego de Mesa

Alice y Bob deciden probar un nuevo juego de mesa. El juego dispone de un tablero de dimensiones $n \cdot m$, de c monedas ubicadas en casillas específicas del tablero y de un número entero r ($1 \leq r \leq m$), que es escogido por Alice y Bob al inicio de cada partida.
El juego de mesa es un juego por turnos, y por cortesía de Bob, Alice es la primera jugadora en todas las partidas. En su turno, el jugador debe escoger una moneda ubicada en una posición (i, j) tal que $(1 \leq i \leq n, 1 \leq j \leq r)$ y moverla hacia otra casilla (i, k) de la misma fila del tablero que se encuentre hacia la izquierda de (i, j), no importa si esta nueva casilla ya contiene otra moneda. El jugador que no pueda realizar un movimiento, pierde.
Alice y Bob deciden jugar q partidas, se quiere determinar el ganador de cada partida, sabiendo que ambos jugadores juegan de forma óptima.

Input

La primera línea contiene cuatro números enteros n, m, c, and q ($1 \leq n, m, c, q \leq 10^5$), que representan las dimensiones del tablero, la cantidad de monedas y la cantidad de partidas, respectivamente.
Las siguientes c líneas contienen dos números enteros x, y ($1 \leq x \leq n$, $1 \leq y \leq m$) cada una, la casilla donde está ubicada la $i-\acute{e}sima$ moneda.
Las siguientes q líneas contienen un número entero r ($1 \leq r \leq m$), el número entero escogido por Alice y Bob para cada partida.

Output

Para cada una de las q partidas, imprima el ganador de la partida, "Alice" o "Bob".

Example

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

Problem C. Contando Productos

Se conocen los números enteros n y k. Se desea calcular cuántos números enteros distintos x ($1 \leq x \leq n$) pueden expresarse como el producto de x_i, tal que:

- $x_1 + x_2 + \ldots + x_m = k$
- $x_i \geq 1$ y $m \geq 1$

Por ejemplo, sea $n = 20$ y $k = 8$, entonces:

- Como $1 + 2 + 2 + 3 = 8$, el producto será $1 \cdot 2 \cdot 2 \cdot 3 = 12$
- Como $4 + 2 + 2 = 8$, el producto será $4 \cdot 2 \cdot 2 = 16$
- Como $8 = 8$, el producto será $8 = 8$
- Como $1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 8$, el producto será $1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 1$

Por lo tanto los números 1, 12, 16, 8 y otros pueden ser obtenidos.

Input

La primera línea contiene los números enteros n y k ($1 \leq n, k \leq 1000$).

Output

Imprima en una sola línea la respuesta del problema.

Example

standard input	standard output
20 8	14

Problem D. Decorando Árboles

Se tiene un árbol con n vértices numerados de 1 a n y con raíz en el vértice 1. Inicialmente, cada vértice tiene un color c_i.
Se deben realizar q operaciones de los siguientes tipos:

1. Actualizar el color de todos los vértices del subárbol del vértice v. Para cada vértice del subárbol, reemplazar su color dada la siguiente fórmula: $c_i = (c_i + 1)\ mod\ 64$. Donde $x\ mod\ y$ devuelve el resto que se obtiene cuando se divide x entre y.

2. Contar la cantidad de vértices en el subárbol del vértice v con color c.

Input

La primera línea contiene dos números enteros n y q ($1 \leq n, q \leq 10^5$), el número de vértices en el árbol y el número de operaciones a realizar. La segunda línea contiene n números enteros c_i ($0 \leq c_i \leq 63$), los colores iniciales de cada vértice. Las siguientes $n - 1$ líneas contienen dos número enteros a y b ($1 \leq a, b \leq n$) que representan las aristas del árbol. Las siguientes q líneas contienen la descripción de las operaciones en el formato descrito a continuación:

$1\ v$: Actualizar el color de todos los vértices del subárbol del vértice v ($1 \leq v \leq n$).

$2\ v\ c$: Contar la cantidad de vértices en el subárbol del vértice v ($1 \leq v \leq n$) con color c ($0 \leq c \leq 63$).

Output

Imprima el resultado para cada operación de tipo 2. Todas las soluciones deben imprimirse en líneas separadas siguiendo el mismo orden dado en la entrada.

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. Decorando Árboles

El Granjero Juan (GJ) está enseñando a sus vacas los números binarios y ellas aprendieron rápidamente que los números binarios solo contienen los dígitos **0** y **1**. GJ estaba muy feliz con los resultados obtenidos y decidió enseñarles a las vacas cómo crear matrices binarias cuadradas. Sin embargo, las vacas se aburrieron después de la segunda clase. El Granjero Juan se puso un poco triste y pensó, qué pasa si le enseño a mis vacas a codificar matrices binarias con otros símbolos?
El Granjero Juan sabe que sus vacas no soy muy inteligentes. Por esta razón, él definió dos reglas simples para codificar matrices binarias:

1. El bit más frecuente se codificará con el símbolo **'*'** y el menos frecuente se codificará con el símbolo **'o'**.
2. En caso de empate en la frecuencia, el bit que se encuentra en la esquina superior izquierda de la matriz será codificado con el símbolo **'*'** y el bit complementario será codificado con el símbolo **'o'**.

Aparentemente las vacas comprendieron las reglas. Sin embargo, el Granjero Juan no está seguro y desea evaluar las habilidades de las vacas. Escriba un programa para codificar una matriz binaria cuadrada utilizando las reglas propuestas por el Granjero Juan.

Input

La primera línea de la entrada contiene un entero n ($1 \leq n \leq 100$) que representa la dimensión de la matriz. Las siguientes n líneas contienen n símbolos binarios **'0'** o **'1'** sin espacios.

Output

La salida contiene la matriz obtenida con el mecanismo de codificación propuesto por el Granjero Juan.

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. Polígono que Desaparece

Se tienen n puntos en un plano. Cada punto se eliminará con probabilidad 0.5 de forma independiente. Determina el valor esperado del área del *convex hull* de los puntos restantes.

Input

La primera línea contiene un número entero n ($1 \leq n \leq 2000$).
Las siguientes n líneas contienen los pares de números enteros x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$), las coordenadas del i-*ésimo* punto. No existen 3 puntos alineados.

Output

Imprima el valor esperado del área del *convex hull*. Imprima el valor $P \cdot Q^{-1} \mod (10^9 + 7)$, donde P y Q son primos relativos y $\frac{P}{Q}$ es la respuesta al problema.

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. Guiado por el XOR

Se tiene una lista A de n números enteros y un número entero k. Se desea llegar a la posición final de la lista partiendo de la primera posición y solo moviéndose hacia la derecha. Desde cualquier posición i se puede ir a todas las posiciones j tales que ($i < j \leq n$ y $A_i \oplus A_j < k$), donde $\oplus$ denota la operación xor. Calcule cuántas formas diferentes existen de llegar a la posición *n-ésima*. Dos formas son diferentes si existe al menos una posición que es visitada en una forma y no en la otra. Ya que la respuesta puede ser muy grande, imprima su valor módulo $10^9 + 7$.

Input

La primera línea de la entrada contiene dos números enteros n, k ($1 \leq n \leq 2 \cdot 10^5, 0 \leq k \leq 2^{20}$).
La segunda línea de la entrada contiene n números enteros $a_1, a_2, \ldots, a_n$ ($0 \leq a_i \leq 2^{20}$), donde a_i es el valor de la *i-ésima* posición de la lista.

Output

Imprima un número entero, el número de formas de llegar a la posición *n-ésima* módulo $10^9 + 7$.

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. Grafo Pesado

Se tiene un grafo ponderado no dirigido con V vértices y E aristas. Cada vértice tiene asociado un peso $p_1, p_2, \ldots, p_V$.
Encuentre un subconjunto S de vértices con la puntuación máxima, donde la puntuación se define como la suma de los pesos de todos los vértices de S y los pesos de todas las aristas entre vértices de S, dividida por el número de vértices de 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

La primera línea contiene 2 enteros V $(1 \le V \le 100)$ y E $(1 \le E \le 1000)$.
La segunda línea contiene V enteros $p_1, p_2, \ldots, p_V$ $(0 \le p_i \le 1000)$.
Las próximas E líneas, cada una contiene 3 enteros u_i, v_i, w_i $(1 \le u_i \ne v_i \le V$, $1 \le w_i \le 1000$, $1 \le i \le E)$ denotanto una arista entre los nodos u_i y v_i con peso w_i.
Se garantiza que habrá a lo sumo una arista no dirigida entre cada par de vértices.

Output

Imprima el subconjunto de vértices S con la puntuación máxima de la siguiente manera:
Una línea con un entero $|S|$, la cantidad de vértices de S.
Una línea con $|S|$ enteros separados por espacios: los vértices de S, en cualquier orden.
Si existe más de un subconjunto con puntuación máxima, imprima cualquiera de ellos.

Example

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

Problem I. Triángulo Isorectángulo

Son dadas las coordenadas de n puntos en el plano y un conjunto de m triángulos isorrectángulos (isósceles y rectángulos). Los catetos de estos triángulos son paralelos a los ejes de coordenadas.
Por cada triángulo debe contar cuantos de los puntos de la entrada este contiene.

Input

La primera línea contiene dos números enteros positivos n y m ($1 \leq n, m \leq 10^5$), la cantidad de puntos y la cantidad de triángulos, respectivamente.
La siguientes n líneas contienen dos números enteros x y ($-10^9 \leq x, y \leq 10^9$), las coordenadas de cada uno de los puntos dados.
Las siguientes m líneas contienen seis números enteros x_1 y_1 x_2 y_2 x_3 y_3 ($-10^9 \leq x_1, y_1, x_2, y_2, x_3, y_3 \leq 10^9$), las coordenadas de los vértices de los triángulos dados.

Output

Imprima m números, la cantidad de puntos que contiene cada triángulo.

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. Uniendo Ciudades

¡Se ha fundado una nueva ciudad! La ciudad tiene n casas, pero aún no tiene carreteras. El ayuntamiento de la ciudad ha contratado a una empresa constructora para hacer carreteras entre las casas, tal que se pueda ir desde cualquier casa a otra mediante algún camino de carreteras. El costo de construir una carretera se calcula dependiendo del valor a de cada casa. El costo de la carretera entre la casa i y la casa j, es el valor absoluto de la diferencia entre sus valores: $|a_i - a_j|$. El ayuntamiento necesita minimizar el costo total de construir las carreteras.

Input

La primera línea contiene el número entero n ($1 \leq n \leq 10^5$), la cantidad de casas de la ciudad.
La segunda línea contiene n números enteros separados por espacios $a_1, a_2, ..., a_n$ ($1 \leq a_i \leq 10^9$), el valor de cada casa.

Output

Imprima en una sola línea el costo total mínimo para conectar las casas de la ciudad.

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 la Koala

Se tienen n gatos y n perros, numerados cada uno desde 1 hasta n. Cada animal puede observar a otros gatos y perros. En particular, se sabe que:

1. Cada **gato** observa exactamente a a **perros** (es decir, a números enteros distintos de 1 a n).
2. Cada **perro** observa exactamente a b **gatos** (es decir, b números enteros distintos de 1 a n).

Algo muy malo pasará si un gato y un perro se observan entre sí al mismo tiempo, por lo que Koa la Koala ayudará a resolver esta situación. Ella tiene que determinar si es posible organizar qué animal mira cada cual cumpliendo que:

- Se satisfacen (1.) y (2.)
- no existe un par de **gato** y **perro** que se observen entre sí, esto es: no pueden existir números enteros i y j ($1 \leq i, j \leq n$) tal que el gato i observe al perro j y el perro j observe al gato i.

¡Ayuda a Koa!

Input

La primera línea de la entrada contiene el número entero t ($1 \leq t \leq 100$), el número de casos de prueba. A continuación t casos de prueba.
La única línea de cada caso de prueba contiene los números enteros n, a y b ($1 \leq n \leq 100; 1 \leq a, b \leq n$). Se garantiza que la suma de n, para todos los casos de prueba no excede el valor de 100 ($\sum n \leq 100$).

Output

Por cada caso de prueba: Imprima "Yes" o "No" (sin comillas), dependiendo de si existe la distribución deseada.
Si la respuesta es "Yes":

- Entonces, exactamente n líneas deben seguir, indicando los perros que son observados por cada gato.
- La i-ésima ($1 \leq i \leq n$) línea debe consistir de exactamente a números enteros distintos $w_1, w_2, \ldots, w_a$ ($1 \leq w_i \leq n$) indicando que el gato i observa a los perros $w_1, w_2, \ldots, w_a$. Estos números enteros pueden estar en cualquier orden.
- Entonces, exactamente n líneas deben seguir, indicando que gatos son observados por cada perro.
- La i-ésima ($1 \leq i \leq n$) línea debe consistir de exactamente b enteros distintos $m_1, m_2, \ldots, m_b$ ($1 \leq m_i \leq n$) indicando que el perro i observa a los gatos $m_1, m_2, \ldots, m_b$. Estos números enteros pueden estar en cualquier orden.

Si hay muchas distribuciones posibles, imprima cualquiera.

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. Recuperando LCS

El algoritmo *LCS* (Subsecuencia Común más Larga) de dos cadenas binarias A y B devuelve una matriz de la siguiente manera:

```
LCS(A, B):
    n = len(A)
    m = len(B)
    M = Array[n, m] // la matriz está llena de ceros
    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
```

Dada una matriz M, encuentre el menor entre todos los pares posibles de cadenas binarias A y B tal que *LCS(A, B)* devuelva la matriz M. Un par (A, B) es menor que (C, D) si $(A + B)$ es lexicográficamente menor que $(C + D)$ donde el operador $+$ denota la concatenación de cadenas.

Input

La primera línea de la entrada contiene dos números enteros n y m $(1 \leq n, m \leq 2 \cdot 10^3)$, el número de filas y el número de columnas en la matriz.
Las siguientes n líneas contienen m números enteros cada una. El elemento *j-ésimo* en la línea *i-ésima* es $M[i][j]$.
Se garantiza que existen al menos dos cadenas binarias tales que el algoritmo *LCS* devuelve la matriz dada.
Tenga en cuenta que la matriz dada difiere de la del pseudocódigo al no tener la fila 0 y la columna 0 por simplicidad.

Output

En la primera línea, imprima una cadena binaria A de longitud n y en la segunda línea, imprima una cadena binaria B de longitud m tal que el par (A, B) sea el menor donde el algoritmo *LCS* devuelve la matriz dada.

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. Paridad de la Matriz

Decimos que una matriz de números enteros es "par" si las sumas de los números por cada fila y cada columna son números pares.
En los siguientes ejemplos, las matrices A, B y C son "pares". La matriz D no es "par" debido a que su segunda columna y su última fila tienen sumas impares. La matriz E no es "par" ya que sus dos columnas tienen sumas impares.

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

Se tiene una matriz A, y se desea obtener una matriz "par" ejecutando la **mínima** cantidad de veces la siguiente operación:

- Seleccionar una celda de la matriz e incrementar su valor en 1. Se permite seleccionar la misma celda en operaciones distintas.

Imprima cualquier matriz final "par" que resulte de ejecutar la operación anterior la menor cantidad de veces posible.

Input

La primera línea contiene dos números enteros r y c ($1 \leq r, c \leq 50$). Las siguientes r líneas definen la matriz A. Cada línea contiene c números enteros con valores entre 0 a 100. El j-*ésimo* número de la i-*ésima* línea corresponde a la celda $A_{i,j}$ de la matriz.

Output

Imprima cualquier matriz "par" resultante en el mismo formato indicado en la sección de entrada: r filas con c números en cada fila.

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. Nueva Combinación

Dado un conjunto P de n puntos en el plano, se desea saber si un punto w se puede expresar como combinación lineal de los puntos de $R \subseteq P$, es decir, si:

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

tal que $\mu_i \geq 0$ para $i = 1, 2, ..., |R|$ y además $\sum_{i=1}^{|R|} \mu_i = 1$.

Para cada punto w debe encontrar un conjunto R con **a lo sumo 5** elementos ($|R| \leq 5$) tal que los coeficientes satisfagan la condición anterior, o decir que es imposible.

Nota:

- Sea p un punto de la forma (x, y) y c un escalar. El producto $c \cdot p$ se define como el punto $s = (c \cdot x, c \cdot y)$.
- Sean $p_1 = (x_1, y_1)$ y $p_2 = (x_2, y_2)$ dos puntos. La suma $p_1 + p_2$ se define como el punto $s = (x_1 + x_2, y_1 + y_2)$.

Input

La primera línea de entrada contiene un número entero n ($1 \leq n \leq 10^5$), el número de puntos de P.
Las próximas n líneas contienen dos enteros x_i, y_i ($0 \leq x_i, y_i \leq 10^9$), las coordenadas del i-*ésimo* punto.
La siguiente línea contiene un número entero q ($1 \leq q \leq 10^4$), la cantidad de puntos w a analizar.
Las restantes q líneas contienen dos números enteros x_w, y_w ($0 \leq x_w, y_w \leq 10^9$), las coordenadas de cada punto w.

Output

Para cada punto w:

- En caso de que no exista solución imprimir la palabra "impossible" (sin las comillas).
- En caso de existir solución imprimir una línea con un número m indicando la cantidad de elementos del conjunto R encontrado.

Luego imprima m líneas, cada una con un entero i indicando que el i-*ésimo* punto de P pertenece a R, seguido por un número real μ_i indicando su coeficiente.

Nota: Su solución será considerada correcta si cumple con las restricciones del problema y además el cuadrado de la distancia entre w y el punto $w' = \sum_{i=1}^{|R|} \mu_i \cdot R_i$ obtenido no excede 10^{-5}.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td>6
0 0
2 0
4 1
0 2
2 2
3 3
3
1 1
2 1
3 0</td><td>4
1 0.25
2 0.25
4 0.25
5 0.25
3
1 0.44444444444444444444
3 0.33333333333333333332
6 0.22222222222222222223
impossible</td></tr>
<tr><td>2
1 3
5 3
3
4 2
3 3
8 3</td><td>impossible
2
1 0.5
2 0.5
impossible</td></tr>
</table>

Problem O. Suma XOR

Dada una lista A con n números enteros positivos, usted puede cambiar hasta 2 elementos por los que desee.
Su objetivo es maximizar la expresión:

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

donde el símbolo $\oplus$ representa la operación binaria *xor* (or exclusivo).

Input

La primera línea de entrada contiene el número entero n ($2 \leq n \leq 10^5$) que representa el tamaño de la lista. La segunda línea contiene los elementos de A separados por espacio tal que ($0 \leq A_i < 2^{30}$) para $i = 1, 2, \ldots, n$.

Output

Imprima un único entero, el mayor valor de S que se puede obtener.
Nota: Los nuevos valores de A deben permanecer en el rango $[0 \leq A_i < 2^{30})$

Example

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