



The 2019 ICPC Caribbean Local Contests

Real Contest Problem Set

Document composed by 13 pages (not including this cover)

Problem set developers

Marcelo Fonet (NEAR, Cuba)

Carlos Joa (INTEC, República Dominicana)

Reynaldo Gil (Datys, Cuba)

Carlos Toribio (Google, República Dominicana)

Elio Govea (UPR, Cuba)

Rubén Alcolea (UCI, Cuba)

Ernesto Peña (UO, Cuba)

José Carlos Gutierrez (Universität Bremen, Cuba)

Norge Vizcay (Cuba)

Abdel Rodríguez (Cuba)

Rafael Hernández (Cuba)

Aurora Gil (Cuba)

Alexander Bestard (UO, Cuba)

Ariel Cruz (UH, Cuba)

September 21st, 2019

Problem A. String in the tree I

Una nueva familia de árboles ha sido descubierta. Estos árboles tan peculiares tienen exactamente una fruta en cada hoja y en cada bifurcación, similar a la estructura de datos árbol. La característica más rara de estas plantas es que producen hasta 62 frutos de diferentes tipos (representados con letras inglesas minúsculas, mayúsculas y dígitos). Para clasificarlas se necesita una función que, dado un árbol (con todos sus frutos) y un patrón (una secuencia de frutas), cuente la cantidad de caminos simples en el árbol que son iguales al patrón. Un camino simple está definido como una secuencia de nodos donde dos nodos adyacentes en la secuencia son adyacentes en el árbol y la secuencia no contiene nodos repetidos. Decimos que un camino simple es igual a un patrón si, cuando reemplazamos el número del nodo en el camino (la secuencia) por la fruta que contiene, obtenemos una secuencia igual al patrón.

Input

La primera línea contiene dos números enteros N y L ($1 \leq N, L \leq 10^3$) que indican el número de nodos y la longitud del patrón respectivamente. La segunda línea contiene una cadena S ($|S| = N$) donde S_i indica la fruta contenida en el nodo i . La tercera línea contiene el patrón P ($|P| = L$) a encontrar. Luego, las siguientes $N - 1$ líneas contienen la estructura del árbol, donde cada línea contiene dos números enteros a y b ($0 \leq a, b < N$) y que indican que estos dos nodos son adyacentes.

Output

Imprime un número entero que indique el número de caminos simples del árbol iguales al patrón.

Example

standard input	standard output
<pre>4 3 baaa aba 0 1 0 2 0 3</pre>	6
<pre>15 4 ababbabaaabbabb abab 0 1 1 2 2 3 3 4 2 5 5 6 2 7 0 8 8 9 8 10 10 11 10 12 12 13 12 14</pre>	3

Problem B. String in the tree II

Una nueva familia de árboles ha sido descubierta. Estos árboles tan peculiares tienen exactamente una fruta en cada hoja y en cada bifurcación, similar a la estructura de datos árbol. La característica más rara de estas plantas es que producen hasta 62 frutos de diferentes tipos (representados con letras inglesas minúsculas, mayúsculas y dígitos). Para clasificarlas se necesita una función que, dado un árbol (con todos sus frutos) y un patrón (una secuencia de frutas), cuente la cantidad de caminos simples en el árbol que son iguales al patrón. Un camino simple está definido como una secuencia de nodos donde dos nodos adyacentes en la secuencia son adyacentes en el árbol y la secuencia no contiene nodos repetidos. Decimos que un camino simple es igual a un patrón si, cuando reemplazamos el número del nodo en el camino (la secuencia) por la fruta que contiene, obtenemos una secuencia igual al patrón.

Input

La primera línea contiene dos números enteros N y L ($1 \leq N, L \leq 10^6$) que indican el número de nodos y la longitud del patrón respectivamente. La segunda línea contiene una cadena S ($|S| = N$) donde S_i indica la fruta contenida en el nodo i . La tercera línea contiene el patrón P ($|P| = L$) a encontrar. Luego, las siguientes $N - 1$ líneas contienen la estructura del árbol, donde cada línea contiene dos números enteros a y b ($0 \leq a, b < N$) y que indican que estos dos nodos son adyacentes.

Output

Imprime un número entero que indique el número de caminos simples del árbol iguales al patrón.

Example

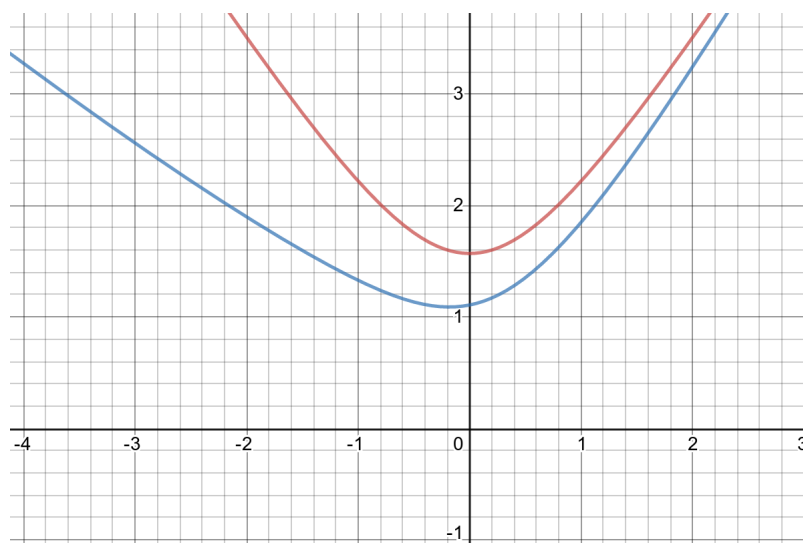
standard input	standard output
<pre>4 3 baaa aba 0 1 0 2 0 3</pre>	6
<pre>15 4 ababbabaaabbabb abab 0 1 1 2 2 3 3 4 2 5 5 6 2 7 0 8 8 9 8 10 10 11 10 12 12 13 12 14</pre>	3

Problem C. Chinese curves

Te encuentras en un mundo con curvas de la forma $f(x) = \arctan(e^x + a)\sqrt{b \cdot x^2 + c}$, donde a , b y c son enteros. En este mundo hay dos tipos de acciones:

1. Dado tres enteros a , b y c , añade una nueva curva definido por los parametros a , b y c .
2. Dada una coordenada p , determina cuál es el mínimo valor de $f(p)$ entre todas las curvas.

La siguiente imagen corresponde al caso de ejemplo, donde la curva azul fue la ultima curva agregada. Aquí tienes una explicación del caso de ejemplo:



1. De la primera línea de entrada, la máscara m se inicializa a 0.
2. La acción 1 crea una curva con los parámetros $a = 1000$, $b = 1$, $c = 1$ (debido a que la máscara $m = 0$).
3. La acción 2 pregunta por el mínimo valor de $f(1)$ entre todas las curvas. Como solo existe una sola, la respuesta es simplemente $\arctan(e^1 + 1000)\sqrt{1^2 + 1}$, aproximadamente 2.22.
4. Luego de la acción 2, la máscara m se convierte a $\lfloor 2.22 \rfloor \oplus 0 = 2$.
5. La acción 3 agrega una curva con los parámetros a , b y c igual a $3 \oplus 2 = 1$.
6. Como $p = 3 \oplus 2 = 1$, la acción 4 pregunta por el mínimo valor de $f(1)$ entre las dos curvas que existen en ese instante. En la imagen, podemos observar que la última curva agregada arroja el valor mínimo de aproximadamente 1.85.
7. Luego de contestar la acción 4, la máscara m adquiere el valor de $\lfloor 1.85 \rfloor \oplus 2 = 3$.
8. La última acción pregunta por el mínimo valor de $f(41 \oplus 3) = f(42)$ entre las dos curvas existentes.

Input

En la primera línea, tenemos dos enteros q y m , donde q es la cantidad de acciones ($1 \leq q \leq 10^5$) y m es una máscara ($0 \leq m \leq 10^5$). Cada una de las siguientes q líneas contiene una operación bajo una de las siguientes notaciones:

1. $a b c$: la acción es del primer tipo, donde $0 \leq b \leq 10^6$, $0 \leq a, c \leq 10^{18}$.

2. p : la acción es del segundo tipo, donde $0 \leq p \leq 10^6$.

En la entrada, los valores de a , b , c y p no serán mostrados directamente. En su lugar, debes recuperarlos calculando $a = a' \oplus m$, $b = b' \oplus m$, $c = c' \oplus m$ y $p = p' \oplus m$, donde a' , b' , c' y p' son los valores de la entrada, m es el valor actual de la máscara m y \oplus es el operador xor (disponible como el símbolo \wedge en muchos lenguajes de programación). Después de cada acción de tipo 2, la máscara cambia a $\lfloor s \rfloor \oplus m$, donde s es la respuesta a esta acción, \oplus es el operador xor y $\lfloor s \rfloor$ es la función floor que devuelve la parte entera del número s . Está garantizado que la primera acción es del primer tipo.

Output

Para cada acción de tipo 2, imprima una línea con la respuesta requerida, redondeando a dos dígitos después de la coma.

Example

standard input	standard output
5 0	2.22
1 1000 1 1	1.85
2 1	65.99
1 3 3 3	
2 3	
2 41	

Problem D. Set

Encuentra un conjunto tal que:

- Todos sus elementos son enteros positivos.
- Tiene exactamente N elementos.
- Ninguno de sus elementos es mayor que L .
- La suma de todos sus elementos es S .
- La cantidad de elementos del conjunto tales que su antecesor no pertenece a este es menor que 100.

Nota: En el segundo caso de prueba el conjunto elegido fue: 3, 4, 5, 7, 9, 10

Input

Tres enteros N ($1 \leq N \leq 10^9$), L ($1 \leq L \leq 10^9$), S ($1 \leq S \leq 10^{18}$).

Output

Una línea con dos enteros A y B , donde A es la cantidad de elementos del conjunto tal que su antecesor no pertenece a este y B la cantidad de elementos del conjunto tal que su sucesor no pertenece a este.

Una línea con A enteros ordenados de forma ascendente con todos los elementos del conjunto tal que su antecesor no pertenece a este.

Una línea con B enteros ordenados de forma ascendente con todos los elementos del conjunto tal que su sucesor no pertenece a este.

Si no hay solution, imprima "-1" sin comillas.

Example

standard input	standard output
1 1 1	1 1 1 1
6 15 38	3 3 3 7 9 5 7 10

Problem E. Rotate circles

Se tiene un tablero de círculos con n filas y m columnas. Cada círculo está dividido en 4 regiones iguales. Cada una de las 4 regiones está coloreada con colores diferentes. Es posible seleccionar un círculo cualquiera y rotarlo 90 grados en el orden de las manecillas del reloj. Un estado del tablero es hermoso si las dos regiones adyacentes de cada par de círculos vecinos tienen el mismo color. Dado el estado inicial del tablero, cuántos estados hermosos diferentes se pueden obtener ejecutando la acción descrita tantas veces se quiera. Dos estados son considerados diferentes si hay al menos un círculo que tenga rotaciones diferentes en ambos estados.

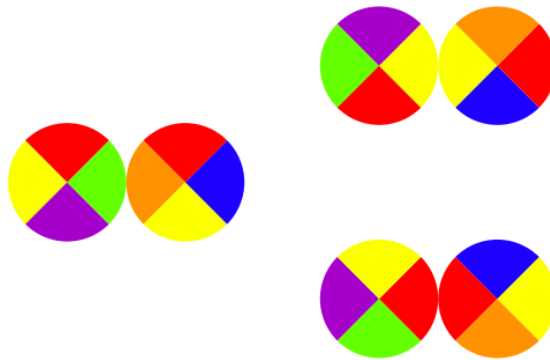


Imagen del segundo caso de prueba. En la izquierda de la imagen está el estado tal y como se describe en la entrada. En la derecha están los dos estados hermosos alcanzables. Nota que los colores por sus nombres en inglés son [r]ed, [b]lue, [p]urple, [y]ellow and [o]range.

Input

La primera línea contiene dos enteros n, m ($1 \leq n, m \leq 10$), número de filas y de columnas del tablero respectivamente. Las siguientes n líneas contienen la descripción de cada fila del tablero. Cada línea contiene $4 \cdot m$ caracteres, cada grupo de 4 caracteres es la descripción de cada círculo. Cada carácter c_i es una letra minúscula del idioma inglés y representa el color de una de las 4 regiones de cada círculo. Las regiones son dadas en el orden de las manecillas del reloj empezando por la región de más arriba. Véase la imagen para más detalles.

Output

Imprima un número entero, la cantidad de estados hermosos diferentes.

Example

standard input	standard output
<pre>2 2 r b k g r b k g r b k g k g r b</pre>	4
<pre>1 2 r g p y r b y o</pre>	2

Problem F. Sigma

Fito y María estaban jugando sobre un mapa del reino de Sigma. El mapa consistía en N ciudades y caminos no dirigidos entre las ciudades. Ambos niños al comienzo del juego eligieron una ciudad para cada uno, F y M , para Fito y María respectivamente. Los niños tuvieron cuidado de escoger dos ciudades distintas que no tuvieran un camino directo entre ellas. Los dos se pusieron de acuerdo en pensar un entero K e intentaron resolver cada uno un reto diferente. Fito desea encontrar a lo sumo K ciudades (distintas de F y M) tal que si las elimina del mapa junto a todos los caminos incidentes a ella no existe ninguna ruta entre la ciudad F y la ciudad M . María por otro lado busca al menos $K + 1$ rutas tal que:

1. Todas las rutas empiecen en la ciudad F .
2. Todas las rutas terminen en la ciudad M .
3. Ninguna ruta contenga ciudades repetidas.
4. Para todo par de rutas las únicas ciudades en común sean F y M (no se repiten ciudades internas de las rutas).

Ayude a determinar a alguno de los dos niños la respuesta a su problema. Nota: Es posible que en el mapa hayan caminos múltiples entre el mismo par de ciudades, o caminos que empiecen y terminen en la misma ciudad.

Input

Una línea con cinco enteros N, T, F, M, K ($1 \leq N, T, K \leq 10^5, 1 \leq F, M \leq N$). La cantidad de ciudades, la cantidad de caminos, la ciudad elegida por Fito, la ciudad elegida por María y el número pensado respectivamente. A continuación T líneas cada una con dos enteros u, v ($1 \leq u, v \leq N$) denotando un camino no dirigido entre las ciudades u y v .

Output

Si encuentra solución para el reto de Fito imprima en la primera línea la palabra "FITO" sin comillas. En la siguiente línea imprima un número C ($0 \leq C \leq K$), la cantidad de ciudades que Fito debe eliminar para resolver su reto. A continuación una línea con C enteros distintos u ($1 \leq u \leq N$), las ciudades que debe eliminar. Cualquier respuesta válida será aceptada. Si encuentra solución para el reto de María imprima en la primera línea la palabra "MARIA" sin comillas. En la siguiente línea imprima un número C ($K + 1 \leq C \leq N$), la cantidad de rutas que debe seleccionar María. En las siguientes C líneas imprima una ruta por línea usando el siguiente formato. Al principio imprima un número X , la cantidad de ciudades de la ruta y a continuación en la misma línea X enteros u ($1 \leq u \leq N$) que representan las ciudades de la ruta. Note que las rutas deben satisfacer las condiciones de María. Cualquier respuesta válida será aceptada. Solo debe encontrar al menos una respuesta para alguno de los dos niños. En caso de que el reto de ninguno de los dos niños tenga solución imprima "NONE" sin comillas.

Example

standard input	standard output
4 4 1 4 2 1 2 1 3 2 4 3 4	FITO 2 2 3
4 4 1 4 1 1 2 1 3 2 4 3 4	MARIA 2 3 1 2 4 3 1 3 4

Problem G. Three numbers

Se tienen tres dígitos A, B, C (no necesariamente diferentes). Halle tres dígitos X, Y, Z que cumplan las siguientes restricciones.

1. X, Y, Z , tienen que ser distintos entre sí.
2. $X \neq A$
3. $Y \neq B$
4. $Z \neq C$
5. $1 \leq X, Y, Z \leq 9$

Input

Una única línea con tres dígitos $1 \leq A, B, C \leq 9$.

Output

Tres dígitos X, Y, Z que cumplan con las restricciones del problema. (Cualquier tripla de enteros que satisfaga las restricciones será aceptada).

Example

standard input	standard output
9 8 7	1 2 3
1 1 1	9 6 3

Problem H. Queries with recurrences

Los problemas que incluyen consultas y recurrencias son muy comunes en el ICPC. En este problema, vamos a mezclar un poco ambos temas. Dada la recurrencia $F_n = a \times F_{n-1} + b \times F_{n-2} + k$ y un arreglo A con N valores, escriba un programa para ejecutar Q consultas con el siguiente formato:

- $1 \ a \ b \ k$: Incrementar todos los valores en el rango $[a, b]$ con el valor F_n , donde $n = a \times b$ ($1 \leq a \leq b \leq N$, $1 \leq k \leq 2N$).
- $2 \ a \ b$: Calcular la suma de los valores $A_a + A_{a+1} + \dots + A_b$. Este valor puede ser muy grande, así que imprima el resultado modulo 1000000007. ($1 \leq a \leq b \leq N$).

Input

La entrada contiene en la primera línea dos enteros: N ($1 \leq N \leq 10^5$) y Q ($1 \leq Q \leq 10^5$). Las siguientes Q líneas contienen la información de las consultas, como se explicó anteriormente. La recurrencia siempre tiene dos casos base: $F[0] = 0$ y $F[1] = k$. Además, al inicio cada valor del arreglo $A[i] = 0$.

Output

Para cada consulta de tipo 2, imprima el resultado correspondiente.

Example

standard input	standard output
10 5 1 1 2 1 2 1 6 1 1 3 2 2 1 4 2 2 6	4 40 26
10 5 1 3 5 15 2 1 10 1 4 6 5 2 1 4 2 6 10	854873249 469363035 566114207

Problem I. Build me a fence

Dado una lista con n enteros positivos (a_1, a_2, \dots, a_n) determine n puntos (p_1, p_2, \dots, p_n) que formen un polígono cerrado tal que:

1. La distancia entre los puntos p_i y p_{i+1} sea igual a a_i si $1 \leq i < n$.
2. La distancia entre los puntos p_n y p_1 sea igual a a_n .
3. El polígono formado tenga área positiva (no sea degenerado).
4. Los lados no consecutivos no se intersecten.

Input

Una línea con un entero n ($3 \leq n \leq 100$), la cantidad de lados del polígono. Luego una línea con n enteros positivos ($1 \leq a_i \leq 100$).

Output

En caso de que no exista ningún posible polígono que cumplan las restricciones del problema imprima "No" sin las comillas. En otro caso imprima "Si" sin las comillas y a continuación imprima n líneas con las coordenadas de los vértices x, y ($-10^6 \leq x, y \leq 10^6$) en orden consecutivo. Cualquier polígono que cumpla con las restricciones del problema será considerado válido. La distancia real entre los puntos reportados y la distancia esperada debe ser menor que 10^{-6} .

Example

standard input	standard output
4 1 1 1 1	Si 0 0 0 1 1 1 1 0
4 1 1 3 1	No

Problem J. Elevator

En un edificio de n pisos han llegado algunos paquetes para cada departamento y es necesario transportarlos a sus respectivos destinatarios. Hay m paquetes en el piso 0 y cada uno tiene anotado el piso al que debe ser enviado. Por problemas de seguridad, el elevador puede transportar como máximo k paquetes al mismo tiempo. Una entrega consiste en el proceso siguiente: seleccionar el piso máximo de la entrega, mover el elevador a este piso y finalmente entregar los paquetes en el piso actual y los inferiores (a medida que el elevador baja al piso 0). En cada entrega de los paquetes, el elevador debe terminar en el piso 0. Su tarea es encontrar la estrategia óptima para transportar todos los paquetes con la menor cantidad de entregas.

Input

Una línea con tres enteros n, m, k , la cantidad de pisos, la cantidad de paquetes que se entregarán y la cantidad máxima de paquetes que se pueden transportar en el elevador en cada momento respectivamente. Luego una línea con m enteros. El entero a_i es el piso al que debe ser enviado el i -ésimo paquete. Límites: $1 \leq n, m, k \leq 100000, 1 \leq a_i \leq n$.

Output

Un entero que indica el mínimo número de entregas para transportar todos los paquetes como se explicó anteriormente.

Example

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

Problem K. Polygon

Como todos deben saber, el ICPC es una de las competencias más prestigiosas en el área de la programación competitiva. Este año el equipo Limscape es uno de los más fuertes en la Región del Caribe, y están entrenando con el objetivo de clasificar a la Final Mundial del ICPC del año 2020. Ellos recientemente participaron en el Concurso Local Caribeño y solo les quedó un problema sin resolver, por lo que quedaron empatados en cantidad de problemas resueltos con los equipos UZ**, Frickyfox y VLCU.h. Ahora ellos quieren resolver dicho problema, y le han pedido ayuda a usted, debido a sus conocimientos en el campo de la Programación Competitiva. El problema fue el siguiente: Dado un polígono simple (no se intersecta consigo mismo) consistiendo de n vértices. Una triangulación de este polígono es una subdivisión en $n - 2$ triángulos, definida por un conjunto de diagonales que no se intersectan, en la cual los vértices de los triángulos son también vértices del polígono. Cada vértice i tiene un valor especial v_i asociado. Definimos el costo de un triángulo como el operador XOR de los valores especiales de los vértices que conforman el triángulo, y el costo de una triangulación como la suma de los costos de los triángulos que la conforman. Estamos interesados en calcular el mínimo valor posible que puede tener una triangulación para dicho polígono.

Input

La entrada consiste de una línea conteniendo el entero n , el número de vértices que componen el polígono ($3 \leq n \leq 300$). La próximas n líneas contienen dos enteros separados por espacio x_i y y_i , describiendo las coordenadas de los vértices en sentido contrario a las manecillas del reloj ($-1000 \leq x_i, y_i \leq 1000$). Finalmente sigue una línea con n enteros v_i separados por espacio, que representan los valores especiales de los vértices, en el mismo orden en que son dados en la entrada ($0 \leq v_i < 2^{10}$).

Output

La salida estará compuesta por una sola línea con el mínimo costo de una triangulación para el polígono dado.

Example

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