Torneo Chileno de Programación 2024

05-10-2024

Sedes:
Universidad Cátolica del Norte (Antofagasta)
Universidad Cátolica del Norte (Coquimbo / La Serena)
Universidad Andrés Bello (Viña del Mar)
Universidad de Valparaíso (Valparaíso)
Universidad Técnica Federico Santa María (Santiago)
Universidad de O'Higgins (Rancagua)
Universidad de Talca (Talca)
Universidad de Concepción (Concepción)

Agradecimientos:

Agradecer a: Javier Oliva y Gabriel Carmona, que ayudaron a redactar, testear y armar el contest; Benjamín Rubio, Martín Muñoz y Javier Marinkovic, que ayudaron a preparar problemas; Roberto Asín, Javier Robledo, Gonzalo Fernández, Bernardo Subercaseaux, Martín Andrighetti y Javier Reyes, que entregaron ideas para los problemas; Blaz Korecic, Diego Arias y Ignacio Muñoz a testear los problemas y entregar feedback; Cristían Ruz, Eric Ross, Waldo Gálvez, Ricardo Brrientos, Roberto Muñoz, Rodrigo Olivares, Giannina Costa, Diego Urrutia, Erik Regla que permitieron la organización esta competencia presencialmente en las diversas sedes; Bruno Ribas que nos ayudo a settear al contest en el juez BOCA.

Problema A

Super-carreteras

autor: Benjamín Rubio

La ciudad de Coquimbo se prepara para ser anfitrión de la ICPC por tercera vez, para eso se planea la construcción de unas nuevas súper-carreteras que harán el transporte desde la universidad a diversas partes de la ciudad mucho más rápido.

La estructura original de la ciudad puede ser vista como un árbol de caminos **bidireccionales** con pesos de n nodos $1, \ldots, n$ con raíz en la universidad (representada por el nodo 1). Se planea la construcción de m súper-carreteras, cada súper-carretera puede ser representada por un trío de nodos $a, b, c \in \{1, \ldots, n\}$ y un largo positivo $w \leq 10^9$ permitiendo moverse entre cualquier nodo en el camino entre a y b en el árbol original al nodo c, en w minutos, estas nuevas carreteras son **unidireccionales**, es decir no es posible ir del nodo c a nodos en el camino de a a b usándola.

Dada la representación original de la ciudad y la representación de las m súper-carreteras añadidas, entrega la distancia mínima entre la universidad y el resto de la ciudad, considerando el uso de las súper-carreteras.

Input

El input inicia con una línea con 2 enteros n, m ($2 \leq n \leq 5 \cdot 10^5$, $0 \leq m \leq 5 \cdot 10^5$), la cantidad de nodos en la ciudad y la cantidad de súper-carreteras.

Luego vienen $n - 1$ líneas describiendo los caminos originales de la ciudad. Cada línea consiste de 3 enteros u, v, w ($1 \leq u, v \leq n$, $u \neq v$, $1 \leq w \leq 10^9$), los nodos conectados por este camino y el largo w del camino.

Finalmente hay m líneas describiendo las súper-carreteras. Cada línea consiste de 4 enteros a, b, c, w ($1 \leq a, b, c \leq n$, $1 \leq w \leq 10^9$) describiendo la estructura de la súper-carretera y su largo.

Output

Debes imprimir $n - 1$ enteros, la distancia mínima desde el nodo 1 a cada otro nodo $2, \ldots, n$ en orden.

Ejemplos

Ejemplo de input 1	Ejemplo de output 1
11 3 1 2 20 1 3 1 2 4 5 2 5 3 3 6 1 4 7 5 5 8 3 5 9 3 7 10 5 7 11 5 8 9 11 20 5 4 1 1 3 6 9 1	8 1 13 5 2 18 8 2 23 22

Problema B

La Barra de Sushi

autor: Gonzalo Fernández

Tu equipo ha logrado resolver todos los problemas del Torneo Cosmogónico de Plurivocalización. Aunque no ganaron, están contentos con su desempeño. Como es costumbre, ningún torneo termina sin un merecido festín, así que deciden ir al restaurante de sushi más cercano, sin imaginar que allí los platos se sirven de una manera muy particular.

En este restaurante, los platos se disponen en una barra, alineados uno al lado del otro y numerados consecutivamente desde el 1. Cada plato contiene una cantidad distinta de piezas de sushi. Para ordenar, deben seleccionar un subconjunto contiguo (*) de platos, y el costo final será el número de platos elegidos.

Con un presupuesto limitado y, como buenos amigos, quieren asegurarse de que la cantidad total de sushi de los platos seleccionados sea divisible por el número de miembros del equipo. Deciden entonces buscar la combinación de platos que les permita obtener la mayor cantidad de sushi sin exceder el presupuesto. Tras varios intentos fallidos, se dan cuenta de que es un problema más complejo de lo que parecía, y deciden aprender a programar para resolverlo de una vez por todas.

(*) Un subconjunto contiguo es una parte de un conjunto en la que los elementos están uno al lado del otro, sin que haya saltos entre ellos. Por ejemplo, en el conjunto $\{5, 10, 15, 20, 25\}$: $\{10, 15, 20\}$, $\{15, 20\}$ y $\{5\}$ son algunos subconjuntos contiguos; $\{5, 15\}$ y $\{10, 25\}$ no son subconjuntos contiguos.

Input

El input consiste en 2 líneas, la primera línea contiene 2 enteros N ($2 \leq N \leq 10^9$), y M ($1 \leq M \leq 10^5$), indicando la cantidad de integrantes del equipo y el presupuesto máximo a gastar, respectivamente.

La segunda línea contiene un entero S ($1 \leq S \leq 10^5$), indicando la cantidad de platos en la barra, seguido de S enteros S_i ($1 \leq S_i \leq 10^9$), indicando la cantidad de piezas en el plato número i. Todos los enteros están separados por un espacio.

Output

El output debe contener dos enteros separados por un espacio i y j, indicando que se tomarán los platos en el rango $[i, i + j - 1]$. En caso de no existir una solución, se debe imprimir "`-1 -1`". Si hay múltiples soluciones posibles, deben imprimir la que empiece primero.

Ejemplos

Ejemplo de input 1	**Ejemplo de output 1**
`2 3` `5 5 10 15 20 25`	`3 3`

Ejemplo de input 2	**Ejemplo de output 2**
`5 10000` `3 2 6 46`	`-1 -1`

Ejemplo de input 3	Ejemplo de output 3
3 10 6 10 20 30 40 50 60	3 4

Nota

En el primer ejemplo, se tienen 5 platos con las siguientes cantidades de sushi: $\{5, 10, 15, 20, 25\}$. Los siguientes subconjuntos contiguos cumplen con las dos condiciones: no exceden el presupuesto y la cantidad total de sushi es divisible por 2:

- $\{10\}$
- $\{5, 10, 15\}$
- $\{15, 20, 25\}$

Entre estos subconjuntos, el que contiene la mayor cantidad de sushi es $\{15, 20, 25\}$. Por lo tanto, la respuesta es `3 3`, ya que el primer plato de este subconjunto está en la posición 3 y se seleccionan 3 platos en total.

Problema C

Mineras

autor: Javier Reyes

Una minera realizó un estudio sobre la eficiencia de transporte entre puntos de interés y concluyó que enfocarse en la eficiencia de transporte es clave para aumentar sus ganancias.

La red de transporte consta de n puntos de interés, conectados por m conexiones bidireccionales. El objetivo es condensar esta red de tal manera que el transporte entre cualquier par de puntos de interés utilice a lo más una conexión.

Para cumplir el objetivo solicitarán los servicios de Transporte Chileno Plus (TCP) que se dedica a condensar redes de este tipo. Una condensación simple de TCP funciona de la siguiente manera:

1. Se determinan todos los pares de puntos de interés (u, v) tales que el menor número de conexiones a usar para pasar de u a v es 2.
2. Por cada par (u, v) entregado por el punto anterior se agrega una conexión nueva entre u y v.

Un servicio de condensación se representa por un entero k que indica el número de veces que se debe aplicar una condensación simple a una red, las cuales se aplican de forma secuencial.

El costo por un servicio de condensación k es $k + 1$ si en cada una de las condensaciones se agrega al menos una conexión nueva y k en el caso contrario.

Debes escoger el parámetro k de servicio de condensación de tal manera que se cumpla el objetivo de que el transporte para cada par de puntos de interés utilice a lo más una conexión y que el costo del servicio sea óptimo.

Input

La primera línea del input contiene dos enteros n, m ($1 \leq n \leq 5 \cdot 10^5, 1 \leq m \leq 5 \cdot 10^5$), separados por un espacio, representando el número de puntos de interés y el número de conexiones, respectivamente.

Las siguientes m líneas contienen dos enteros a_i, b_i ($1 \leq a_i, b_i \leq n$), separados por un espacio, correspondientes a las conexiones.

Output

El output debe contener un único entero, el parámetro k del servicio de condensación a contratar. Si hay varias soluciones, puedes escoger cualquiera de ellas.

Ejemplos

Ejemplo de input 1	Ejemplo de output 1
3 2 1 2 2 3	1

Ejemplo de input 2	Ejemplo de output 2
6 7 1 2 2 3 3 4 2 6 6 5 5 3 2 5	3

Nota

En el primer caso, en la primera operación se agrega una arista entre 1 y 3. Ninguna otra operación es necesaria así que indicar 1 a TCP implica un costo óptimo. Noten que indicar 2 también habría implicado un costo óptimo.

En el segundo caso, después de la primera operación se agregan aristas de tal forma que los únicos pares de sitios no conectados directamente son $(1, 4)$ y $(4, 6)$. Una segunda operación es necesaria. Por lo tanto indicar 2 o 3 implica un costo óptimo.

Problema D

Sudoku Malo

autor: Gabriel Carmona

Sudoku es un juego matemático que consiste en rellenar una cuadrícula de 9×9 casillas (81 casillas) dividida en subcuadrículas de 3×3 con las cifras del 1 al 9 partiendo de algunos números ya dispuestos en algunas de las celdas. La regla es que en cada fila, columna y subcuadrícula solo puede aparecer una vez cada número del 1 al 9. Un Sudoku es considerado válido si es que cumple la regla anterior mencionada, en caso contrario no es válido.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

(a) Estado de Sudoku válido, en ninguna fila, columna o subcuadrícula se repite un número del 1 al 9.

5	3			7				
6			1	9	5			
9	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

(b) Estado de Sudoku inválido, en la fila 3 y en la subcuadrícula de arriba la izquierda se repite el número 9.

Martín es un apasionado del Sudoku; juega día y noche, sin descanso, toda la semana. En Pelotillehue, la asociación de jugadores de Sudoku tiene un premio reservado para el mejor jugador.

Al enterarse del premio, Martín decide que debe ganarlo para honrar a su familia. Con determinación, envía todos los Sudokus que ha resuelto, tanto los completos como los incompletos, a la asociación para demostrar su habilidad y así asegurarselo.

Pero cuando está a punto de enviar sus Sudokus, se da cuenta de que varios de ellos son inválidos. A punto de desesperarse, Martín encuentra una solución rápida: para que sus Sudokus parezcan válidos reales, puede intercambiar únicamente dos números de posición en una sola subcuadrícula de 3×3.

El problema es que Martín tiene poco tiempo para corregir sus Sudokus y hacer que todos sean válidos, por lo que te pide ayuda.

Input

El input está compuesto por 9 líneas, cada una de ellas contiene 9 números separados por un espacio. Cada número tendrá un valor entre 0 y 9. Si el valor es 0, entonces significa que en esa casilla no hay número.

Output

Si es posible obtener un estado válido de Sudoku intercambiando a lo más una vez dos números de una sola subcuadrícula, imprime el estado válido obtenido siguiendo el mismo formato del input. Si hay más de una solución, puedes imprimir cualquiera.

Si no es posible obtener un estado válido realizando a lo más un intercambio, imprime -1.

Ejemplos

Ejemplo de input 1	Ejemplo de output 1
5 3 0 0 7 0 0 0 0 6 0 0 1 9 5 0 0 0 0 9 8 0 0 0 0 6 0 8 0 0 0 6 0 0 0 3 4 0 0 8 0 3 0 0 1 7 0 0 0 2 0 0 0 6 0 6 0 0 0 0 2 8 0 0 0 0 4 1 9 0 0 5 0 0 0 0 8 0 0 7 9	5 3 0 0 7 0 0 0 0 6 0 0 1 9 5 0 0 0 0 9 8 0 0 0 0 6 0 8 0 0 0 6 0 0 0 3 4 0 0 8 0 3 0 0 1 7 0 0 0 2 0 0 0 6 0 6 0 0 0 0 2 8 0 0 0 0 4 1 9 0 0 5 0 0 0 0 8 0 0 7 9

Problema E

Ayuda el contest se está quemando

autor: Gabriel Carmona

El equipo de escritores del Torneo Chileno de Programación debe entregar su propuesta de problemas al director, Cris Roz. Sin embargo, debido a retrasos, están preocupados por si su set de problemas está adecuadamente balanceado.

Un set de problemas se considera balanceado si incluye al menos un problema para cada nivel de dificultad esperado. Para facilitar las cosas debido al tiempo limitado, también se aceptará un set que cubra cada rango de dificultad indicado, es decir, al menos un problema para cada rango de dificultad $[x, y]$ donde $y - x = 1$.

Los escritores están muy ocupados preparando los casos de prueba y las soluciones, por lo que te piden que los ayudes a determinar si el set de problemas propuesto está balanceado o no.

Input

La primera línea del input contiene dos enteros p y r ($1 \leq p, r \leq 10000$), separados por un espacio, que representan la cantidad de problemas y la cantidad de rangos de dificultad, respectivamente.

La segunda línea contiene p enteros separados por un espacio, donde cada entero indica la dificultad de un problema propuesto por los escritores. Las dificultades pueden tomar valores entre 1 y 10000.

Las siguientes r líneas contienen dos enteros x e y ($1 \leq x < y \leq 10001$), separados por un espacio, que representan el comienzo (x) y el final (y) de un rango de dificultad, donde $y - x = 1$.

Output

El output debe contener la frase "`Es balanceado!`" si el set de problemas es balanceado, en caso contrario debe contener "`No es balanceado!`".

Ejemplos

Ejemplo de input 1	**Ejemplo de output 1**
10 6 9 2 1 8 3 7 4 2 10 7 1 2 2 3 8 9 5 6 10 11 9 10	No es balanceado!

Ejemplo de input 2	**Ejemplo de output 2**
10 6 9 2 1 8 3 6 4 2 10 7 1 2 2 3 8 9 5 6 10 11 9 10	Es balanceado!

Problema F

Qué elegancia la de Transilvania

autor: Javier Robledo

En Transilvania se ha logrado que la comunidad de humanos y monstruos convivan en armonía, siendo normal en esta sociedad que existan parejas humano/monstruo.

El Hotel Transilvania ha organizado una cena de etiqueta. Los invitados deben sentarse en una gran mesa redonda siguiendo el protocolo francés, el cual establece que las parejas no se pueden sentar una al lado de la otra y, para apoyar el intercambio cultural, se deben sentar monstruos y humanos de forma alternada. Un humano o monstruo solo puede ser pareja con un humano o monstruo. Un humano o monstruo no puede ser pareja de sí mismo.

Input

La primera línea del input contiene un número positivo p ($1 \leq p \leq 100$) que corresponde a la cantidad de parejas. Luego le siguen p líneas que representan las parejas, las cuales son representadas por strings separados por un espacio.

Finalmente le siguen $2 \cdot p$ líneas que representan a cada invitado, los que se representan por un string indicando su nombre seguido de un espacio y luego un caracter, el cual puede ser `h` si es humano o `m` si es monstruo.

Solo se han invitado parejas a la cena y cada invitado aparece en una sola pareja. Los nombres son de largo máximo 30 y contienen solo letras del abecedario inglés en mayúscula o minúscula.

Output

El output corresponde a una línea que representa el orden en el cual se sentarán los invitados considerando las restricciones del protocolo francés. En esta línea aparece el nombre de un invitado, luego un espacio, seguido del siguiente invitado. Deben aparecer los $2 \cdot p$ invitados.

Si es que no se puede lograr sentar a las parejas de tal forma que se cumpla la condición descrita en el enunciado, imprime -1.

Ejemplos

Ejemplo de input 1	**Ejemplo de output 1**
4 A B C D E F G H A h B m C h D h E m F h G m H m	A E C G D B F H

Problema G

Problema de Emergencia

autor: Martín Andrighetti y Gabriel Carmona

Los organizadores del Torneo de Computación de Pelotillehue están desesperados por buscar un problema, pero no se les ocurre nada hasta que un día Martín ve a un niño dibujando polígonos regulares en un papel. Martín, intrigado, se acerca y observa que no solo está dibujando polígonos regulares, sino que también está trazando segmentos entre todos los pares de vértices del polígono. Ya sin aguantar la curiosidad, Martín le pregunta por qué está haciendo ese tipo de dibujos. El niño le responde que su profesora de kinder les dijo que tenían que averiguar cuántos pares de segmentos intersectan ortogonalmente.

Con mucha emoción Martín le dice al niño que salvó el TCP y corre a su casa para resolver ese problema.

Input

El input se compone de únicamente un entero n ($3 \leq n \leq 10^6$), correspondiente a la cantidad de lados que tiene el polígono regular.

Output

El output debe contener un único entero, correspondiente al número de pares de segmentos que intersectan ortogonalmente.

Ejemplos

Ejemplo de input 1	**Ejemplo de output 1**
10	15

Ejemplo de input 2	**Ejemplo de output 2**
250	15375

Problema H

Otro problema de camino más corto

autor: Javier Oliva

Sofía está jugando su videojuego de rol favorito, llamado Troll & Cave Patrol (TCP). En este juego se te entrega un mapa que consiste de una secuencia de triángulos $t_0, t_1, ..., t_{n-1}$ que satisface las siguientes restricciones:

- El área de t_i es positiva para todo $0 \leq i < n$.
- El área de la intersección entre t_i y t_j $(i \neq j)$ es 0.
- El triángulo t_i comparte un lado (o dos vértices) con el triángulo t_{i+1} para todo $0 \leq i < n-1$.
- Si existe intersección entre t_i y t_j con $i+1 < j$ entonces ésta consiste de un y solo un vértice que también es vértice de todos los triángulos t_k con $i < k < j$.

La secuencia de triángulos se entrega como una secuencia de $3n$ vértices a coordenadas enteras $v_0, v_1, ..., v_{3n-1}$. El triángulo t_i tiene como vértices los vértices $v_{3i}, v_{3i+1}, v_{3i+2}$. Además, se asegura que v_0 y v_{3n-1} son únicos en la secuencia de puntos.

Sofía debe viajar desde v_0 hasta v_{3n-1} y debe hacerlo siguiendo el camino más corto posible. Además, Sofía solo puede moverse por los bordes o interior de los triángulos.

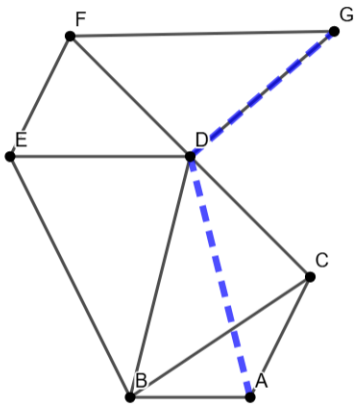


Figura 2: El camino más corto desde el primer vértice A al último vértice G es el camino azul.

Dada la secuencia de triángulos, determina un camino óptimo para Sofía.

Input

La primera línea del input contiene un único entero n $(2 \leq n \leq 5 \cdot 10^5)$, indicando el número de triángulos.

Luego vienen $3 \cdot n$ líneas donde la $i-$ésima de éstas contiene dos enteros x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$), separados por un espacio, representando las coordenadas del vértice v_{i-1}.

Se asegura que la secuencia de triángulos respeta las restricciones entregadas arriba.

Output

El output debe contener el camino representado como una secuencia de puntos a coordenadas enteras $p_1, p_2, \ldots, p_k$ tal que el camino corresponde a la unión de los segmentos $[p_i, p_{i+1}]$ para $1 \leq i < k$ y $p_1 = v_0$. Se debe entregar un punto por línea con coordenadas separadas por un espacio. Además, no deben haber tres puntos consecutivos colineales.

Se puede demostrar que existe un camino óptimo que corresponde a la unión de una cantidad finita de segmentos definidos por dos puntos a coordenadas enteras.

Ejemplos

Ejemplo de input 1	**Ejemplo de output 1**
5 2 0 0 0 3 2 0 0 3 2 1 4 0 0 1 4 -2 4 1 4 -2 4 -1 6 -1 6 1 4 3 6	2 0 1 4 3 6

Problema I

Un juego extraño

autor: Javier Oliva

Javier y Gabriel encontraron un juego en el antiguo departamento de Marinkovic y decidieron probarlo. El juego se juega sobre un árbol enraizado de n nodos cuya raíz es el nodo 1. Cada nodo tiene un cierto número no negativo de fichas. En un turno (con turnos alternados), un jugador escoge un nodo i que tenga una cantidad positiva de fichas y elimina una de esas fichas, luego escoge un subconjunto del conjunto de sus ancestros (*) y, para cada ancestro en su conjunto, puede eliminar o agregar una ficha. Además, no puede haber una cantidad negativa de fichas en un nodo.

El jugador que no pueda jugar pierde. Si Javier empieza el juego, ¿quién ganará si ambos juegan de forma óptima?

(*) El conjunto de ancestros de un nodo i corresponde al conjunto de todos los nodos que se encuentran en el camino único entre i y 1, excluyendo al nodo i.

Input

La primera línea del input contiene un entero n ($1 \leq n \leq 10^6$), correspondiente al número de nodos del árbol.

La segunda línea contiene n enteros $a_1, a_2, \ldots, a_n$ ($0 \leq a_i \leq 10^5$) donde el entero a_i indica el número de fichas con las que parte el nodo i.

Luego vienen $n-1$ líneas donde la i-ésima contiene dos enteros a_i, b_i ($1 \leq a_i, b_i \leq n$) indicando que los nodos a_i y b_i están conectados por una arista.

Se asegura que el grafo corresponde a un árbol.

Output

Debes imprimir "`Javier`" si Javier ganará o "`Gabriel`" si Gabriel ganará.

Ejemplos

Ejemplo de input 1	Ejemplo de output 1
4 1 2 0 0 1 2 1 3 2 4	Javier

Nota

En el ejemplo gana Javier sacando la ficha del nodo 1, luego Gabriel solo puede jugar sacando una ficha del nodo 2 y no puede hacer nada más, finalmente Javier saca la última ficha y gana.

Problema J
El Dado Que Rueda™

autor: Martín Muñoz

Hace dos meses comenzaste tu doctorado en la Universidad de Polandia y todo lo que has hecho es leer papers y dormir. ¡Oh no!

Para superar este tedio infinito decidiste visitar el Club de Juegos de Mesa de Polandia con la idea de hacer algo distinto un día a la semana. No fue grata tu sorpresa que en este Club solo juegan un juego que se llama El Dado Que Rueda™, y es el juego más simple que has visto en la vida. Pero como estás tan desesperado por tener un panorama sonríes y les permites que te expliquen su pavada de juego.

El juego comienza en un tablero con forma de plano cartesiano infinito, y tu oponente te entrega dos valores a y b. Tu objetivo es tomar El Dado™, colocarlo en la coordenada $(0, 0)$ y hacerlo llegar a la coordenada (a, b) gastando la menor cantidad de puntos. Este dado es un cubo de dimensiones $1 \times 1 \times 1$ donde cada cara tiene un valor que puede ser 1 o 2. Los valores que están adelante, atrás, a la izquierda y a la derecha en el dado son todos 1, y los valores que están arriba y abajo son 2.

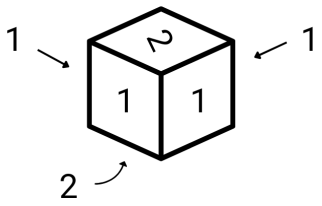
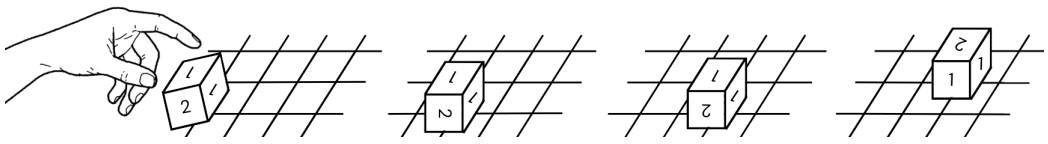


Figura 3: El Dado™

Poner el dado en el tablero toma tantos puntos como el valor en la cara que toca el tablero; tú dedides en qué orientación colocarlo. Mover el dado solo se puede hacer volteándolo en las cuatro direcciones paralelas a los ejes x o y. Es decir, si el dado está en la coordenada (x, y), lo puedes voltear hacia las coordenadas $(x + 1, y)$,$(x - 1, y)$, $(x, y + 1)$ y $(x, y - 1)$. Al voltearlo una nueva cara toca el tablero, y esta movida cuesta tantos puntos como el valor que está en esta cara.

La figura de abajo muestra una forma óptima de partir en $(0, 0)$ y llegar a $(2, 1)$, que cuesta 5 puntos.



Tú has jugado El Dado que Rueda™ un par de horas ya, y ya sabes cómo obtener el puntaje perfecto para cada posición. El Club de Juegos de Mesa de Polandia quiere entrenar dentro de la semana para tratar de ganarte en la próxima junta, así que les vas a entregar un programa al que ellos pueden ingresar un par de números a y b, y entregue la mínima cantidad de puntos que se pueden gastar para hacer llegar El Dado™ a la coordenada (a, b).

Input

El input contiene únicamente dos enteros separados por un espacio, correspondiente a los valores a y b ($0 \leq a, b \leq 10^8$).

Output

Imprimir un único entero, indicando el costo mínimo de hacer llegar el dado a la coordenada (a, b).

Ejemplos

Ejemplo de input 1	**Ejemplo de output 1**
`0 0`	`1`

Ejemplo de input 2	**Ejemplo de output 2**
`0 2`	`3`

Ejemplo de input 3	**Ejemplo de output 3**
`2 1`	`5`

Ejemplo de input 4	**Ejemplo de output 4**
`3647 2`	`3651`

Problema K

Problema constructivo choro

autor: Martín Muñoz

Eres el diseñador principal de una prestigiosa competencia, y tu tarea es definir las rutas que los equipos deben seguir desde el inicio hasta la gran final. En esta competencia, los equipos se mueven a través de una red de n nodos empezando del nodo 1, el equipo ganador es aquel que llega al nodo n primero. Sin embargo, como requerimiento especial, las rutas deben ser cuidadosamente planificadas para garantizar que haya exactamente k caminos diferentes entre el punto de inicio y la meta.

Tu misión es diseñar un mapa de rutas que cumpla con esta condición. Pero hay un límite: no puedes usar más de 100 nodos para representar todas las rondas y enfrentamientos, y las rutas deben estar bien organizadas para que los equipos siempre avancen hacia la final, sin posibilidad de retroceder o quedar atrapados en ciclos.

Input

El input contiene únicamente el entero k ($1 \leq k \leq 10^9$).

Output

El output debe contener una posible red para la competencia con el siguiente formato:

- Imprime una línea con dos enteros n, m ($n \leq 100$), separados por un espacio, que representan la cantidad de nodos y aristas en el grafo.
- Luego imprime m líneas con números u, v ($1 \leq u, v \leq n$), separados por un espacio, para decir que hay una arista de u a v en el grafo. El grafo tiene que ser simple (cada arista no puede aparecer más de una vez) y tener exactamenente k caminos distintos desde el nodo 1 hasta el nodo n.

Ejemplos

Ejemplo de input 1	**Ejemplo de output 1**
7	6 9 1 2 1 3 2 4 3 4 1 4 4 5 5 6 4 6 1 6

Problema L
UF

autor: Javier Marinkovic

Martín, tras ahorrar por **40** años y vender dos riñones, por fin tiene suficiente dinero ahorrado para poner el pie de un crédito hipotecario para comprar un estudio. Obtuvo un crédito flexible, que le permite pagar cuanto quiera cada mes. Martín ya tiene planeado cuánto pagará en pesos para los siguientes N meses, apuntando a pagar toda su deuda, pero se olvidó de un detalle importante: ¡En Chile los créditos se adeudan en UF, no en pesos!

Martín debe un monto de M UFs, a una tasa de interés compuesto del $t\,\%$. Es decir, que si al comienzo de un mes Martín adeuda x, al final de ese mismo mes adeudaría $(1 + t/100) \cdot x$. Si Martín adicionalmente decide pagar un monto de p UFs en ese mes, terminaría el mes con una deuda de $(1 + t/100) \cdot x - p$.

Martín se intranquiliza, pues si bien entiende cómo funciona el proceso de pago con los montos en UF, su plan de pago está en pesos. Martín revisa el valor de la UF en pesos mes a mes, pero como no es muy hábil con las matemáticas, no sabe bien cómo calcular cómo se refleja su pago en pesos en el monto adeudado en UF cada mes. ¿Puedes ayudarlo a diseñar un programa que pueda hacer los cálculos por él, y que determine si termina de pagar su estudio al final del plazo de N meses?

Input

La primera línea del input contiene tres valores separados por un espacio: $N(1 \leq N \leq 360)$, $M(1 \leq M \leq 10^4)$ y $t(0{,}0 \leq t \leq 20{,}0)$; que representan el plazo del crédito en meses, el monto adeudado en UF inicialmente y la tasa de interés, respectivamente.

A continuación, vienen N líneas donde la $i-$ésima contiene dos valores, separados por un espacio, $p_i(0 \leq p_i \leq 10^6)$ y $UF_i(10^3 \leq UF_i \leq 10^5)$, que representan el monto pagado por Martín y el valor de la UF en el mes i, respectivamente.

Output

El output debe contener como respuesta la palabra "**Si**" en el caso de que Martín logre pagar su estudio en N meses, y "**No**" de caso contrario. Podrás considerar que Martín logró pagar la deuda si es que el monto adeudado por Martín es $\leq 0{,}01$ UFs.

Ejemplos

Ejemplo de input 1	**Ejemplo de output 1**
3 10.0 0.0 2000.0 1000.0 50000.0 10000.0 15000.0 5000.0	Si
4 20.0 10.0 6000.0 1200.0 17000.0 1700.0 18000.0 2000.0 1000.0 4000.0	No

Nota

Podría ser el caso de que Martín no se de cuenta de cuándo termine de pagar la deuda, y siga pagando. En ese caso puedes pensar que los pagos “rebotan”, y la deuda no se vuelve negativa.

Problema M

Problema de backtracking????

autor: Martín Muñoz y Bernardo Subercaseaux

El famoso problemsetter BerSub escribió este enunciado hace mucho tiempo:

"Te entregan 10 coeficientes $a_0, \ldots, a_9$ y un valor k. Encuentra la permutación $p_0, \ldots, p_9$ de los valores 0 al 9 lexicográficamente menor (*) que cumpla con $p_0 \cdot a_0 + \cdots + p_9 \cdot a_9 \leq k$."

El problema se supone que era un ejercicio clásico de backtracking y todo funciona bien.

Lamentablemente, cuando el profesor Parceló quiso usar este mismo problema para su curso, se dio cuenta que había transcrito mal el enunciado. Había cambiado los límites de 10 a 100, y la permutación ya no es de 0 a 9 sino que de 0 a 99.

Parceló se dio cuenta de su error y quiso avisarle a sus alumnos que el problema no era así y que lo lamentaba mucho. Justo antes de mandar su correo su alumna Alejandra le habló diciéndole "Hola profesor, gracias por el problema, estuvo difícil pero entretenido." y el mensaje tenía un pantallazo de un envío que decía Accepted.

Nuestro profesor Parceló no sabe qué hacer. Para él, el problema era imposible, y ya no puede decirle a su curso que el problema no se podía hacer–había una alumna que lo había resuelto. ¿Puedes resolverlo por él?

(*) Un arreglo (en particular, una permutación) $A = [a_1, \ldots, a_n]$ es *lexicográficamente menor* a otro arreglo $B = [b_1, \ldots, b_n]$ si es que existe un $i \in [1, n]$ tal que $a_i < b_i$ y para cada $j < i$ se cumple que $a_j = b_j$. Por ejemplo, la permutación $[0, 2, 1, 3]$ es lexicográficamente menor a $[0, 2, 3, 1]$.

Input

El input consiste de una línea con los 100 valores $a_0, \ldots, a_{99}$ ($1 \leq a_i \leq 10^9$), separados por un espacio, y una segunda línea con el número k ($1 \leq k \leq 10^9$).

Output

Imprime la permutación $p_0, \ldots, p_{99}$ de los valores 0 al 99 lexicográficamente menor que cumpla con $p_0 \cdot a_0 + \cdots + p_{99} \cdot a_{99} \leq k$, cada número debe estar separado por un espacio. Si no existe, imprime -1.

Ejemplos

Ejemplo de input 1	**Ejemplo de output 1**
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 200000	0 1 2 3 4 5 6 61 97 99 98 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7

Ejemplo de input 2	Ejemplo de output 2
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 100000	-1

Nota

En los ejemplos se muestran los 100 valores en varias líneas debido a la cantidad de números. El formato correspondiente se describe en las sección del input y output.

Problema N
Desempate TCP

autor: Roberto Asín-Achá

La ceremonia de premiación del Torneo Chileno de Programación (TCP) no podría haber terminado de forma más espectacular. La Universidad Teórica Computina de Pelotillehue (TCP1) y la Universidad del Trabajo Constante de Pueblo Lavanda (TCP2) han empatado con 10 problemas resueltos y un tiempo acumulado de 800 segundos. Este inédito hecho llevó a los organizadores a consultar a los equipos si querían compartir la gloria o buscar una forma de desempate. Al igual que en el evento de salto alto de las Olimpiadas de París 2024, los equipos decidieron no compartir la gloria. Así, los organizadores definieron un singular método de desempate. A la mañana siguiente, los dos equipos deberán dirimir al campeón de la competencia en un singular juego.

El juego funciona de la siguiente manera. Cada equipo tendrá frente a si mismo, a 100 metros de distancia, N láminas de cartón desplegadas en forma horizontal, de izquierda a derecha, dadas vuelta. Cada lámina contiene, en el lado no visible, un número entero. Adicionalmente, se desplegará en una pantalla gigante un único número entero *obj* que será de referencia para ambos equipos. Por turnos, cada participante del equipo correrá los 100 metros, tomará la lámina más a la izquierda y volverá con sus compañeras(os). Con las láminas recolectadas, todo el equipo decidirá si es posible, mediante sumas de los números recolectados, igualar a *obj*. En dicho caso, el equipo deberá correr a la pantalla (a 200 metros de cada equipo) y levantar los brazos.

Como la velocidad de cada concursante es conocida y la pereza la mañana después de una cena de premiación es infinita, los equipos han acordado aceptar el desenlace simulando el juego.

¿Cuál fue el equipo ganador?

Input

El input consiste en 5 líneas. La primera línea contiene dos números enteros positivos N ($1 \leq N \leq 1000$) y *obj* ($1 \leq obj \leq 10000$).

La segunda línea contiene N números enteros positivos a_i ($1 \leq a_i \leq 10000$), separados por espacios en blanco, que corresponden a las láminas de izquierda a derecha colocadas para la Universidad Teórica Computina de Pelotillehue.

La tercera línea contiene contiene N números enteros positivos b_i ($1 \leq b_i \leq 10000$), separados por espacios en blanco, que corresponden a las láminas de izquierda a derecha colocadas para la Universidad del Trabajo Constante de Pueblo Lavanda.

La cuarta línea contiene 3 valores enteros positivos p_1, p_2 y p_3 ($1 \leq p_i \leq 100$), separados por un espacio, representando el tiempo, en segundos, que cada participante del equipo TCP1 tarda en correr los 100 metros, en el orden definido por el equipo para realizar los relevos.

Finalmente, la quinta línea contiene 3 valores enteros positivos q_1, q_2 y q_3 ($1 \leq q_i \leq 100$), separados por un espacio, representando el tiempo, en segundos, que cada participante del equipo TCP2 tarda en correr los 100 metros, en el orden definido por el equipo para realizar los relevos.

Output

El output corresponde a una línea conteniendo "Universidad Teorica Computina de Pelotillehue", o "Universidad del Trabajo Constante de Pueblo Lavanda", dependiendo de qué equipo sea el ganador del TCP. En el improbable caso de que se vuelva a dar un empate, la salida será una línea conteniendo "Vuelvan a correr".

Ejemplos

Ejemplo de input 1	Ejemplo de output 1
5 17 3 4 10 6 1 10 7 2 2 2 60 60 60 60 60 60	Universidad del Trabajo Constante de Pueblo Lavanda

Ejemplo de input 2	Ejemplo de output 2
5 17 3 4 10 6 1 10 7 2 2 2 30 30 50 60 60 60	Universidad Teorica Computina de Pelotillehue

Ejemplo de input 3	Ejemplo de output 3
5 17 3 4 10 6 1 10 7 2 2 2 60 30 30 60 60 60	Vuelva a correr