

Maratón de programación de la SBC 2022

El set de problemas de este cuaderno es utilizado simultáneamente en las siguientes competencias:

Maratona SBC de Programação 2022
Tercera Fecha Gran Premio de México 2022
Gran Premio de Centroamérica 2022
Torneo Argentino de Programación 2022
The 2022 ICPC Bolivia Preliminary Contest

08 de Octubre de 2022

Cuaderno de Problemas

Información General

Este cuaderno contiene 14 problemas; Las páginas están numeradas de 1 a 19, sin contar esta página. Verifique que su cuaderno está completo.

A) Sobre los nombres de los programas

- 1) Para soluciones en C/C++ y Python, el nombre del archivo de código fuente no es significativo, puede ser cualquier nombre.
- 2) Si su solución es en Java, el archivo debe ser llamado: `codigo_de_problema.java` donde `codigo_de_problema` es la letra mayúscula que identifica al problema. Recuerde que en Java el nombre de la clase principal debe ser igual que el nombre del archivo.
- 3) Si su solución es en Kotlin, el archivo debe ser llamado: `codigo_de_problema.kt` donde `codigo_de_problema` es la letra mayúscula que identifica al problema. Recuerde que en Kotlin el nombre de la clase principal debe ser llamado igual que el nombre del archivo

B) Sobre la entrada

- 1) La entrada de su programa debe ser leída de *entrada standard*.
- 2) La entrada está compuesta de un único caso de prueba, descrito en un número de línea que depende del problema.
- 3) Cuando una línea de entrada contiene varios valores, estos están separados por un único espacio en blanco; la entrada no contiene ningún otro espacio en blanco.
- 4) Cada línea, incluyendo la última, contiene exactamente un caracter de final-de-línea.
- 5) El final de la entrada coincide con el final del archivo.

C) Sobre la salida

- 1) La salida de su programa debe ser escrita en *salida standard*.
- 2) Cuando una línea de salida contiene varios valores, estos deben ser separados por un único espacio en blanco; la salida no debe contener ningún otro espacio en blanco.
- 3) Cada línea, incluyendo la última, debe contener exactamente un caracter de final-de-línea.

Promocional:



Sociedade Brasileira de Computação

v1.0

Problema A

Acertando los Monótonos No Triviales Maximales

En este problema trataremos con secuencias de caracteres, comúnmente llamadas *strings*. Una secuencia es *no-trivial* si contiene al menos dos elementos.

Dada una secuencia s , decimos que el tramo s_i, \dots, s_j es *monotono* si todos sus caracteres son iguales, y decimos que es *maximal* si el tramo no puede ser extendido hacia la izquierda o la derecha perdiendo monotonicidad.

Dada una cadena compuesta únicamente por los caracteres “a” y “b”, determine cuántas veces aparece el carácter “a” en tramos monotonos maximales no-triviales.

Entrada

La entrada consiste de dos líneas. La primer línea contiene un solo entero N , donde $1 \leq N \leq 10^5$. La segunda línea contiene una cadena con exactamente N caracteres, que consta únicamente de los caracteres “a” y “b”.

Salida

Imprima una sola línea conteniendo un entero, representando el número de veces que aparece el carácter “a” en tramos monotonos maximales no-triviales.

<p>Ejemplo de entrada 1</p> <p>7 abababa</p>	<p>Ejemplo de salida 1</p> <p>0</p>
<p>Ejemplo de entrada 2</p> <p>7 bababab</p>	<p>Ejemplo de salida 2</p> <p>0</p>
<p>Ejemplo de entrada 3</p> <p>10 aababaaabb</p>	<p>Ejemplo de salida 3</p> <p>5</p>
<p>Ejemplo de entrada 4</p> <p>10 bbaababaaa</p>	<p>Ejemplo de salida 4</p> <p>5</p>

Problema B

Brincando piedras

Alice y Bob jugarán un juego con 3 montones de piedras. Los jugadores se turnan y, en cada turno, un jugador debe elegir un montón que todavía tiene piedras y quitarle un número positivo de piedras. Gana quien toma la última piedra del último montón que aún tenía piedras. Alice hace el primer movimiento.

El i -ésimo montón tendrá un número aleatorio y uniformemente distribuido de piedras en el rango $[L_i, R_i]$. ¿Cuál es la probabilidad de que Alice gane dado que ambos juegan de manera óptima?

Entrada

La entrada consta de una línea con 6 enteros, respectivamente, $L_1, R_1, L_2, R_2, L_3, R_3$. Por cada i , $1 \leq L_i \leq R_i \leq 10^9$.

Salida

Imprime un número entero que represente la probabilidad de que Alice gane módulo $10^9 + 7$.

Se puede demostrar que la respuesta se puede expresar como una fracción irreducible $\frac{p}{q}$, donde p y q son enteros y $q \not\equiv 0 \pmod{10^9 + 7}$, es decir, nos interesa el entero $p \times q^{-1} \pmod{10^9 + 7}$.

Ejemplo de entrada 1 3 3 4 4 5 5	Ejemplo de salida 1 1
Ejemplo de entrada 2 4 4 8 8 12 12	Ejemplo de salida 2 0
Ejemplo de entrada 3 1 10 1 10 1 10	Ejemplo de salida 3 580000005
Ejemplo de entrada 4 5 15 2 9 35 42	Ejemplo de salida 4 1

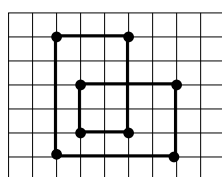
Problema C

Cortes con Laser

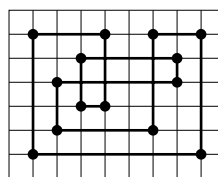
Una máquina laser para cortar hojas de madera tiene una cabeza laser que se puede mover en solo dos direcciones: horizontal y vertical. Has sido contratado para ser parte del equipo de pruebas de esta máquina.

Una de las pruebas consiste en programar la máquina para que realice una secuencia no vacía de cortes que inician y terminan en el mismo punto. Cada corte en la secuencia, excepto el primero, inicia en el punto en el que el corte previo finalizó. Ninguno de los cortes toca los bordes de la hoja que será cortada.

Las figuras (a) y (b) a continuación, muestran dos ejemplos de secuencias de corte con respectivamente 8 y 14 cortes.



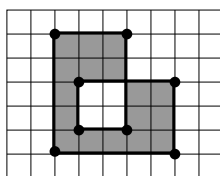
(a)



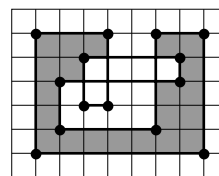
(b)

Tu jefe te ha pedido que determines el area de la pieza más grande producida por la secuencia de cortes, descartando la pieza que contiene a los bordes de la hoja de corte. Es decir, considerando solo las piezas contenidas en el polígono formado por las líneas de corte.

Las figuras (c) y (d) a continuación muestran respectivamente las piezas más grandes producidas por las secuencias de corte de las figuras (a) y (b).

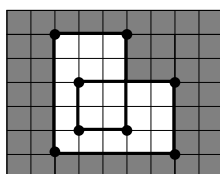


(c)

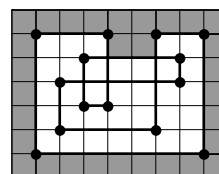


(d)

Para ilustrar, las figuras (e) y (f) a continuación muestran la pieza descartada (que contiene los bordes de la hoja de madera) de las secuencias de corte de las figuras (a) y (b) respectivamente.



(e)



(f)

Entrada

La primera línea contiene un número entero N , el número de cortes en la secuencia ($4 \leq N \leq 10^4$). La segunda línea contiene dos enteros X_0 y Y_0 , la posición inicial del cabezal láser en la secuencia de cortes ($1 \leq X_0 \leq 10^3$ y $1 \leq Y_0 \leq 10^3$). Cada una de las siguientes N líneas contiene dos enteros X_i y Y_i , la posición final del corte i ($1 \leq X_i \leq 10^3$ y $1 \leq Y_i \leq 10^3$, con $1 \leq i \leq N$, y $(X_N, Y_N) = (X_0, Y_0)$). Todas las posiciones dadas en la entrada son distintas, excepto para la primera y la última posición.

Salida

Su programa debe imprimir una sola línea, que contenga un solo número entero, el área de la pieza más grande producida por la secuencia de cortes.

<p>Ejemplo de entrada 1</p> <pre> 8 2 1 7 1 7 4 3 4 3 2 5 2 5 6 2 6 2 1 </pre>	<p>Ejemplo de salida 1</p> <pre> 17 </pre>
<p>Ejemplo de entrada 2</p> <pre> 14 1 1 8 1 8 6 6 6 6 2 2 2 2 4 7 4 7 5 3 5 3 3 4 3 4 6 1 6 1 1 </pre>	<p>Ejemplo de salida 2</p> <pre> 21 </pre>

Problema D

Desplazamiento de Partículas

Un cuadrado tiene sus vértices en las coordenadas $(0, 0)$, $(0, 2^N)$, $(2^N, 2^N)$, $(2^N, 0)$. Cada vértice tiene un atractor. Una partícula se coloca inicialmente en la posición $(2^{N-1}, 2^{N-1})$. Cada atractor se puede activar individualmente, cualquier número de veces. Cuando se activa un atractor en la posición (i, j) , si una partícula está en la posición (p, q) , se moverá al punto medio entre (i, j) y (p, q) .

Dado N y un punto (x, y) , calcula el número mínimo de veces que tienes que activar los atractores para que la partícula acabe en la posición (x, y) .

Entrada

La entrada consiste de una sola línea que contiene tres enteros N , x y y , tales que $1 \leq N \leq 20$ y $0 < x, y < 2^N$.

Salida

Imprima una sola línea, que contenga la menor cantidad de veces que tiene que activar los atractores.

Ejemplo de entrada 1 1 1 1	Ejemplo de salida 1 0
Ejemplo de entrada 2 4 12 4	Ejemplo de salida 2 1
Ejemplo de entrada 3 4 3 1	Ejemplo de salida 3 3

Problema E

Eliminando Globos

Una enorme cantidad de globos flotan en el Salón de Convenciones al finalizar la Ceremonia de Premiación del Concurso Subregional del ICPC. El gerente del Salón de Convenciones está enojado porque mañana será el anfitrión de otro evento y los globos deben ser removidos. Afortunadamente, este año Carlinhos vino preparado con su arco y flechas para reventar los globos.

Además, por suerte, debido al flujo de aire acondicionado, los globos están todos en el mismo plano vertical (es decir, un plano paralelo a una pared), aunque en distintas alturas y posiciones.

Carlinhos disparará desde el lado izquierdo del Salón de Convenciones, a una altura elegida, apuntando a la dirección del lado derecho del Salón de Convenciones. Cada flecha se mueve de izquierda a derecha, a la altura a la que se disparó, en el mismo plano vertical de los globos. Cuando una flecha toca un globo, el globo revienta y la flecha continúa su movimiento hacia la derecha a su altura disminuida en 1. En otras palabras, si la flecha estaba en la altura H , después de reventar un globo continuará en la altura $H - 1$.

Carlinhos quiere reventar todos los globos disparando la menor cantidad de flechas posible. ¿Puedes ayudarlo?

Entrada

La primera línea de entrada contiene un entero N , el número de globos ($1 \leq N \leq 5 \times 10^5$). Como todos los globos están en el mismo plano vertical, definamos que la *altura* de un globo se da en relación con el eje y y la *posición* de un globo se da en relación con el eje x . Los globos están numerados del 1 al N . Los números en los globos indican sus posiciones, desde el extremo izquierdo (globo número 1) hasta el extremo derecho (globo número N), independientemente de sus alturas. La posición del globo número i es diferente de la posición del globo número $i + 1$, para todo i . La segunda línea contiene N enteros H_i , donde H_i indica la altura del globo número i ($1 \leq H_i \leq 10^6$ para $1 \leq i \leq N$).

Salida

Su programa debe imprimir una sola línea, que contenga un solo número entero, el número mínimo de flechas que Carlinhos necesita disparar para reventar todos los globos.

Ejemplo de entrada 1 5 3 2 1 5 4	Ejemplo de salida 1 2
Ejemplo de entrada 2 4 1 2 3 4	Ejemplo de salida 2 4
Ejemplo de entrada 3 6 5 3 2 4 6 1	Ejemplo de salida 3 3

Problema F

Fanático del Ahorcado Multidimensional

El Juego del Ahorcado Multidimensional tiene reglas muy peculiares. En cierto modo, es como si estuvieras jugando varios juegos del tradicional juego del Ahorcado al mismo tiempo, con la diferencia de que las palabras no tienen por qué existir en el diccionario. Si nunca has jugado el juego del Ahorcado, no te preocupes: toda la información que necesitas está a continuación.

En la versión multidimensional del juego, hay varias palabras en un tablero, inicialmente desconocidas, todas de la misma longitud. En cada turno del juego, descubres algunos caracteres de ciertas posiciones de las palabras (el cómo estos caracteres fueron descubiertos no es importante para este problema). En un cierto momento del juego, cuando sólo queda un caracter desconocido en cada palabra del tablero, el juego entra en la fase de *todo o nada*. En este punto, se debe elegir una palabra que maximice el número de compatibilidades con las palabras del tablero. Para una palabra P elegida, decimos que es compatible con una palabra T en el tablero si todas las letras conocidas en T aparecen exactamente en las mismas posiciones en P .

Dada la información conocida sobre las palabras en el tablero, debe determinar qué palabra elegir en la fase de *todo o nada* que maximice el número de compatibilidades. Si hay más de una solución, escriba la menor lexicográficamente. Decimos que una palabra P es lexicográficamente más pequeña que una palabra Q si $P_i < Q_i$ donde P_i es el i -ésimo caracter de P , Q_i es el i -ésimo caracter de Q y i es el índice más pequeño tal que $P_i \neq Q_i$.

Entrada

La primera línea de la entrada contiene dos enteros N y C que satisfacen $1 \leq N \leq 10^4$ y $1 \leq C \leq 12$, indicando el número de palabras en el tablero y la longitud de las palabras que contiene. Cada una de las siguientes N líneas contiene una palabra de longitud C compuesta únicamente de caracteres de la “a” a la “z” excepto por una de sus posiciones, que contendrá un caracter “*”, que indica que el caracter en esa posición aún se desconoce.

Salida

Imprima una sola línea, que contenga, en orden, una palabra T , de longitud C , y un entero M , tal que M es el mayor número de compatibilidades que una palabra puede tener con las palabras de entrada y T es lexicográficamente la más pequeña entre las palabras con compatibilidad M .

Ejemplo de entrada 1	Ejemplo de salida 1
<pre>5 4 rat* ru*d rot* r*ta r*ta</pre>	<pre>rata 3</pre>

Ejemplo de entrada 2	Ejemplo de salida 2
<pre>5 4 bon* fon* n*no *eto *ano</pre>	<pre>nano 2</pre>

Problema G

Geometría Triangular

Todo polígono se puede construir uniendo triángulos. En particular, podemos hacer esto iterativamente: comenzamos con un triángulo, agregamos un segundo triángulo que identifica uno de sus lados con uno de los lados del triángulo inicial, añadimos un tercer triángulo identificando uno de sus lados a uno de los lados libres de alguno de los triángulos originales, y así sucesivamente. Solo consideraremos polígonos que se pueden construir de esta manera, donde cada triángulo agregado toca (y se identifica con) exactamente un lado de un triángulo colocado previamente.

Dado un polígono P , sea T el conjunto de triángulos utilizados para formarlo. Los lados de cada triángulo son segmentos de recta. Sea L el conjunto de segmentos que son lados de algún triángulo en T . Tenga en cuenta que cada elemento de L es un lado de uno o dos elementos de T .

Una vez que tenemos un polígono posicionado en el plano, en algunos casos podemos eliminar algunos de los triángulos que lo componen, sin cambiar el conjunto L . Queremos eliminar triángulos de tal manera que se mantenga el conjunto L sin cambios y el área total de los triángulos restantes sea mínima. De manera equivalente, queremos seleccionar un subconjunto S de triángulos de T tal que:

1. Cada elemento de L es el lado de al menos un triángulo en S ; y
2. La suma de las áreas de los elementos de S es lo más pequeña posible.

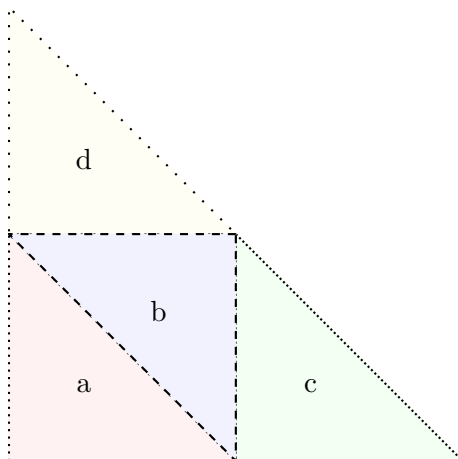
Entrada

La primera línea de la entrada contiene un entero N , $1 \leq N \leq 10^5$ correspondiente a el número de triángulos en la triangulación de P . Cada una de las siguientes líneas N contiene 6 números, x_1, y_1, x_2, y_2, x_3 y y_3 , indicando la existencia de un triángulo con coordenadas (x_1, y_1) , (x_2, y_2) y (x_3, y_3) . Los triángulos se dan en orden arbitrario. Todas las coordenadas serán números enteros con valor absoluto como máximo 10^6 .

Salida

Imprima el área mínima posible, respetando las condiciones del problema, con exactamente un decimal.

<p>Ejemplo de entrada 1</p> <pre>4 0 0 0 10 10 0 10 10 0 10 10 0 10 10 0 10 0 20 10 10 20 0 10 0</pre>	<p>Ejemplo de salida 1</p> <pre>150.0</pre>
<p>Ejemplo de entrada 2</p> <pre>3 0 0 0 10 10 0 10 10 0 10 10 0 10 10 20 0 10 0</pre>	<p>Ejemplo de salida 2</p> <pre>150.0</pre>
<p>Ejemplo de entrada 3</p> <pre>1 0 0 1 0 0 1</pre>	<p>Ejemplo de salida 3</p> <pre>0.5</pre>



En la figura de arriba, las triangulaciones $T_1 = \{a, b, c, d\}$ y $T_2 = \{a, b, c\}$ representan, respectivamente, el primer y segundo ejemplo. Note como $S_1 = \{a, c, d\}$ es un subconjunto válido en el primer caso. Aunque el triángulo b no se toma, todos sus lados están presentes en los triángulos seleccionados.

Problema H

Habilitando el movimiento

El presidente de Nlogonia decidió, por decreto, que todas las calles de Nlogonia deberían ser de un solo sentido. Debido a la falta de conocimientos en ciencia elemental, no hubo una planeación adecuada de los cambios. Después de que el nuevo sistema fue implementado, las personas no podían llegar al trabajo, o no podías regresar a sus casas desde su trabajo, por ejemplo. Como resultado, hubo caos y disturbios en muchas ciudades.

El presidente fue destituido y la nueva administración del país contrató a un equipo de científicos para resolver el problema. A su vez, el comité lo contrató a usted, un experto en complejidad de algoritmos, para ayudarlos con el cálculo eficiente de la solución.

Entonces, para cada ciudad, se te dan los *puntos de referencia* de la ciudad y las calles de un solo sentido, cada una de las cuales conecta dos puntos de referencia. Tu tarea es determinar la cantidad mínima de puentes de un solo sentido que se deben construir para tener una conectividad total en la ciudad. Cada puente también debe conectar dos puntos de referencia.

Entrada

La primera línea de entrada contiene dos enteros, N y M ($1 \leq N \leq 10^4, 1 \leq M \leq 10^6$), donde N es el número de puntos de referencia y M es el número de calles. cada una de las siguientes M líneas contiene dos enteros, R y S , $1 \leq R, S \leq N, R \neq S$, que corresponde a una calle que conecta R con S , por lo que cada vehículo en esa calle debe alejarse de R , hacia S .

Salida

Su programa debe imprimir una sola línea que contenga el número mínimo de puentes que son necesarios para hacer felices a los habitantes.

Ejemplo de entrada 1	Ejemplo de salida 1
7 7 1 2 2 3 3 1 6 1 6 4 4 5 7 6	2

Ejemplo de entrada 2	Ejemplo de salida 2
7 7 2 1 3 2 1 3 1 6 4 6 5 4 6 7	2

Ejemplo de entrada 3 2 1 1 2	Ejemplo de salida 3 1
Ejemplo de entrada 4 3 3 1 2 2 3 3 1	Ejemplo de salida 4 0
Ejemplo de entrada 5 2 0	Ejemplo de salida 5 2
Ejemplo de entrada 6 6 4 1 2 1 3 4 6 5 6	Ejemplo de salida 6 3

Problema I

Información Interceptada

Spies Breaching Computers (SBC) es una agencia de espías digitales que está desarrollando un nuevo dispositivo para interceptar información usando ondas electromagnéticas, el cual, permite espiar incluso sin tener contacto físico con el objetivo.

Este dispositivo intenta obtener la información un byte a la vez, esto es, una secuencia de 8 bits donde cada uno de ellos, naturalmente, puede tener un valor de 0 o 1. En ciertas situaciones, debido a interferencia por otros dispositivos, la lectura no puede ser completada satisfactoriamente. En este caso, el dispositivo regresa un valor de 9 para el bit correspondiente, informando que la lectura no pudo ser realizada.

Para automatizar el reconocimiento de la información que el dispositivo obtiene, se ha realizado una solicitud para que se desarrolle un programa, el cual, dada la información leída por el dispositivo, informe si todos los bits fueron leídos de manera exitosa o no. Tu tarea es escribir este programa.

Entrada

La entrada consiste de una sola línea, que contiene 8 enteros $N_1, N_2, N_3, N_4, N_5, N_6, N_7$ y N_8 , indicando los valores leídos por el dispositivo (N_i es 0, 1 o 9 para $1 \leq i \leq 8$).

Salida

Imprima una sola línea que contenga la letra mayúscula “S” si todos los bits fueron leídos satisfactoriamente; de lo contrario, imprima una sola línea que contenga la letra mayúscula “F”, indicando una falla.

Ejemplo de entrada 1 0 0 1 1 0 1 0 1	Ejemplo de salida 1 S
Ejemplo de entrada 2 0 0 1 9 0 1 0 1	Ejemplo de salida 2 F

Problema J

Jugando 23

Veintitrés es un juego simple de cartas que juegan los niños. Como su nombre lo indica, es una variación del juego veintiuno (*blackjack*) en inglés), el cual es uno de los juegos más jugados en casinos y sitios de juegos.

En veintitrés gana el jugador que tiene más puntos siempre y cuando no supere los 23. Si un jugador tiene más de 23 puntos decimos que el jugador *reventó*.

El juego utiliza una baraja de 52 cartas, con cuatro palos, cada palo con 13 cartas (as, 2, 3, 4, 5, 6, 7, 8, 9, 10, jota, reina y rey). Los palos de cartas no son relevantes. Las cartas con imagen (jota, reina y rey) valen diez puntos, las cartas con números valen su número en puntos (por ejemplo, la carta 4 vale cuatro puntos) y el as vale un punto.

Las reglas del juego son simples: en cada juego, inicialmente se barajan las cartas del mazo, las cartas se colocan en una pila y cada jugador recibe dos cartas de la pila. Todas las cartas se reparten boca arriba (todos los jugadores ven las cartas de todos los jugadores). El siguiente paso, llamado *ronda*, se repite mientras haya jugadores activos: se toma una carta de la pila, boca arriba. Esta carta, llamada *carta común*, se aplica a todos los jugadores. Si un jugador revienta, deja el juego. El ganador será aquel jugador que en una determinado ronda sume 23 (sumando sus dos cartas iniciales más las cartas comunes), o si el jugador es el único jugador activo al final de la ronda. Tenga en cuenta que puede haber más de un ganador (cuyas cartas suman 23) y que puede no haber ganador en un juego.

Juan y María están jugando veintitrés. Los dos son los únicos jugadores, ninguno de ellos reventó y ninguno de ellos tiene 23 puntos. Además, la puntuación de los jugadores es tal que la siguiente carta común puede hacer que el juego termine.

Dadas las cartas iniciales de Juan y María y las cartas comunes, determine ¿Cuál es el valor de la carta de menor valor que se debe tomar de la pila en la siguiente ronda para que María gane el partido?

Entrada

La primera línea de entrada contiene un solo número entero N ($1 \leq N \leq 8$), el número de rondas jugadas hasta ese momento. Cada carta está descrita por un entero I ($1 \leq I \leq 13$). Tenga en cuenta que las cartas con imagen (jota, reina y rey) están representadas en la entrada por los valores 11, 12 y 13, que no es cuántos puntos valen. La segunda línea contiene dos enteros, que describen las dos cartas iniciales de Juan. La tercer línea contiene dos números enteros, que describen las dos cartas iniciales de María. La cuarta y última línea contiene N enteros, que describen cada una de las cartas comunes, en el orden en que fueron tomadas de la pila.

Salida

Su programa debe imprimir una sola línea, que contenga un solo número entero, el valor de la carta de menor valor que debe tomarse de la pila en la siguiente ronda para que María gane el juego, o -1 si María no puede ganar el juego en la siguiente ronda.

Ejemplo de entrada 1	Ejemplo de salida 1
1	3
10 5	
9 10	
1	

Ejemplo de entrada 2 1 10 5 8 7 2	Ejemplo de salida 2 6
Ejemplo de entrada 3 1 9 10 10 5 1	Ejemplo de salida 3 4
Ejemplo de entrada 4 2 8 4 4 1 4 4	Ejemplo de salida 4 5
Ejemplo de entrada 5 8 2 1 1 1 1 2 2 2 3 3 3 3	Ejemplo de salida 5 -1

Problema K

Kalel, la rana saltarina

Kalel es una rana a la que le gusta saltar sobre piedras.

Hay N piedras en una fila, numeradas del 1 al N de izquierda a derecha. Kalel comienza en la piedra 1 y quiere llegar a la piedra N .

En cada movimiento, Kalel puede elegir entre M tipos de salto. El j -ésimo salto le permite saltar de la piedra x a la piedra $x + d_j$ y cuesta p_j puntos de energía. Puede suceder que p_j sea igual a 0 para algunos j . Puedes suponer que Kalel nunca se queda sin energía.

Dados N y K , calcula de cuántas maneras Kalel puede alcanzar la piedra N gastando como máximo K puntos de energía en total. Dos formas se consideran diferentes si la secuencia de opciones de salto es diferente. Como este número puede llegar a ser muy grande, solo nos interesa su resto módulo 10^9 (mil millones).

Entrada

La primera línea contiene tres enteros, N , M y K ($1 \leq N \leq 10^9$, $1 \leq M \leq 10^5$, $0 \leq K \leq 400$). Las siguientes M líneas contienen dos enteros cada una, los números d_j y p_j ($1 \leq d_j \leq 10$, $0 \leq p_j \leq K$).

Salida

Imprima una sola línea, que contenga de cuántas maneras diferentes Kalel puede llegar a la roca N gastando un máximo de K puntos de energía, módulo 10^9 (mil millones).

<p>Ejemplo de entrada 1</p> <pre>5 3 10 1 3 2 0 3 1</pre>	<p>Ejemplo de salida 1</p> <pre>6</pre>
<p>Ejemplo de entrada 2</p> <pre>100000 3 10 1 9 2 0 7 3</pre>	<p>Ejemplo de salida 2</p> <pre>85449877</pre>

Problema L

Listando caminos tediosos

Un árbol es un grafo conectado (existe un camino entre cualquier par de vértices), no dirigido (las aristas del grafo no tienen dirección) y acíclico (no tiene ciclos).

Un árbol colorido es un árbol en el que cada uno de sus vértices tiene un color específico.

Un camino tedioso es un camino en el árbol tal que tanto el vértice inicial como el final tienen el mismo color, y no hay vértice o aristas que aparezcan más de una vez en el camino. Tenga en cuenta que el color de los vértices intermedios, si los hay, son irrelevantes.

Dado un árbol colorido, con N vértices, tu tarea es calcular, para cada una de las aristas, el número de caminos tediosos que pasan por esa arista.

Entrada

La primera línea de entrada contiene el número de vértices N ($1 \leq N \leq 10^5$). La segunda línea contiene N enteros C_1, \dots, C_N , donde C_i ($1 \leq C_i \leq N$) representa el color del vértice i . Las siguientes $N - 1$ líneas contienen dos enteros cada una, u y v , que representan una arista ($1 \leq u, v \leq N$ y $u \neq v$). Se garantiza que el grafo dado es un árbol.

Salida

Imprime $N - 1$ enteros, que representan el número de caminos tediosos que pasan por cada arista, siguiendo el mismo orden de las aristas que se dan en la entrada.

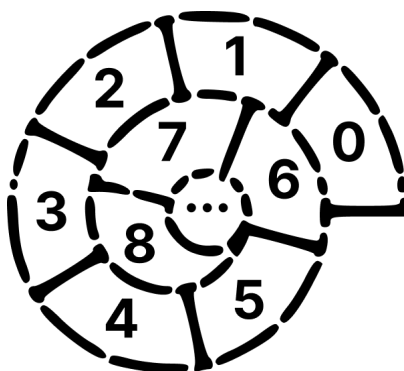
<p>Ejemplo de entrada 1</p> <pre>6 1 1 1 2 2 1 1 2 2 3 4 6 2 4 1 5</pre>	<p>Ejemplo de salida 1</p> <pre>4 3 3 4 1</pre>
<p>Ejemplo de entrada 2</p> <pre>12 1 2 3 1 2 2 1 3 2 3 1 2 1 2 2 3 2 4 4 5 4 6 1 7 7 8 7 9 9 10 6 11 6 12</pre>	<p>Ejemplo de salida 2</p> <pre>10 2 10 4 9 9 2 6 2 3 4</pre>

Problema M

Maratón de Rayuela

8 de octubre de 2022. Esta es la fecha del evento más esperado del año por los estudiantes de informática de todo el país. No, no estamos hablando del ICPC.

¡Estamos hablando, por supuesto, de Rayuela! Para aquellos que no estén familiarizados, Rayuela es una competencia anual que tradicionalmente se lleva a cabo como un evento paralelo del ICPC. Transmitida en vivo para espectadores de todos los continentes y para practicantes de los lenguajes de programación más esotéricos, esta variante exótica del popular juego infantil tiene lugar en una cancha infinita en forma de espiral, subdividida en áreas numeradas secuencialmente comenzando en cero, como se muestra a continuación.



Este año, Rayuela ha atraído un número récord de N participantes, numerados secuencialmente de 1 a N . Se sabe que el i -ésimo participante comienza en el área numerada A_i .

Rayuela consta de Q rondas. Durante la q -ésima ronda, Carlão, amado organizador de Rayuela desde hace más tiempo del que nadie puede recordar, comunicará dos números enteros a los participantes: c_q y d_q . Esta es una orden para todos los participantes con un número de identificación i tal que i y c_q comparten un factor entero común mayor que 1 para retroceder d_q posiciones en la cancha Rayuela, una por una, sin retroceder nunca más allá que la posición 0. (Cualquier participante que eventualmente regrese a la posición 0 debe permanecer allí indefinidamente, ignorando cualquier otra orden de retroceso, para no abandonar la cancha).

Bajo el supuesto de que los participantes han ejecutado las instrucciones perfectamente (nunca querrían decepcionar a Carlão), tu tarea es determinar, para cada participante, el número de la ronda en la que él o ella llegará a la posición 0 (o indicar de otra manera que esto nunca suceda).

Entrada

La primera línea contiene los enteros N y Q ($1 \leq N, Q \leq 10^5$). La segunda línea contiene N enteros, A_1, A_2, \dots, A_N ($1 \leq A_i \leq 10^9$). Cada una de las siguientes Q líneas contiene dos números enteros, c_q y d_q ($1 \leq c_q \leq 10^5$, $1 \leq d_q \leq 10^9$).

Salida

Imprima N líneas. La i -ésima línea debe contener un solo número entero, que indica el número de la ronda cuando el i -ésimo participante llegará a la posición 0 (o el valor -1 , si eso nunca sucede).

Ejemplo de entrada 1	Ejemplo de salida 1
7 6	-1
10 20 30 40 50 60 70	1
2 25	2
3 36	3
100 42	4
5 10	2
7 70	5
1 1000	

Ejemplo de entrada 2	Ejemplo de salida 2
6 4	-1
100 100 100 100 100 100	-1
2 50	-1
3 50	-1
5 99	4
5 1	2

Problema N

Números al otro lado

Acabas de ganar una baraja de cartas con N cartas. Cada una de estas cartas tiene dos números escritos: uno al frente y otro al reverso.

Tu amigo te ha retado a un juego. Barajó las cartas y las puso en una mesa. Las cartas se colocan en una línea, lado a lado, con el frente hacia arriba.

De izquierda a derecha, sabes que el número escrito en el frente de la i -ésima tarjeta es A_i , y que el número escrito en el reverso de la i -ésima tarjeta es B_i .

El juego se divide en dos partes.

En la primera parte debes elegir K cartas de la baraja. Para elegir una carta, debe elegir la primera carta a la izquierda, o la primera carta a la derecha de la mesa, y recogerla.

Después de eso, debes elegir L de las cartas que recogiste y voltearlas.

Su puntaje será igual a la suma de los números escritos en el frente de todas las K cartas que escogiste, más la suma de los números escrito al reverso de las L cartas que volteó.

¿La meta? ¡Lograr la puntuación más alta posible, por supuesto!

Entrada

La primera línea contiene un entero N ($1 \leq N \leq 10^5$). La segunda línea contiene N enteros A_1, A_2, \dots, A_N , ($1 \leq A_i \leq 10^9$). La tercera línea contiene N enteros B_1, B_2, \dots, B_N , ($1 \leq B_i \leq 10^9$). La cuarta línea contiene dos enteros K y L ($1 \leq L \leq K \leq N$).

Salida

Imprima una sola línea que contenga un entero, representando la puntuación más alta posible.

<p>Ejemplo de entrada 1</p> <p>5 9 7 2 2 9 5 2 2 3 1 2 1</p>	<p>Ejemplo de salida 1</p> <p>23</p>
<p>Ejemplo de entrada 2</p> <p>5 9 7 2 2 9 5 9 2 3 1 2 1</p>	<p>Ejemplo de salida 2</p> <p>25</p>