

## Maratón de programación de la SBC 2023

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

Gran Premio de México 2023  
Gran Premio de Centroamérica 2023  
The 2023 ICPC Bolivia Preliminary Contest  
Torneo Chileno de Programación 2023

*02 de Septiembre de 2023*

### Cuaderno de Problemas

#### Información General

Este cuaderno contiene 13 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

## Problema A

# Altura Mínima

Conoce a Carlitos, un entusiasta aventurero con un insaciable amor por los parques de diversiones. A pesar de su vibrante pasión, Carlitos enfrenta un reto único – su estatura. Cuando planea entusiasmado su escapada de fin de semana, se da cuenta que sus limitaciones verticales podrían provocar problemas en su experiencia de parques de diversiones. No es solo escoger un parque; es también encontrar uno donde pueda disfrutar la emoción de los juegos.

Imagina el caleidoscopio de colores, la risa jubilosa, y los latidos del corazón durante los juegos. Carlitos siempre ha sido atraído por la energía de los parques de diversiones. Con el fin de semana aproximándose, él analiza detenidamente folletos de parques, estudiando los requerimientos de estatura de cada juego. Su objetivo es maximizar su diversión, y aquí es donde entras tú.

Tu tarea es ayudar a Carlitos a determinar el número de juegos que él puede disfrutar en un parque específico. Considerando su estatura y la estatura mínima requerida en cada juego, guíalo para conseguir la mayor cantidad de diversión en su aventura en el parque.

### Entrada

La primera línea contiene dos enteros,  $N$  y  $H$  ( $1 \leq N \leq 6$  y  $90 \leq H \leq 200$ ), representando el número de juegos en un parque y la estatura de Carlitos en centímetros, respectivamente.

La segunda línea contiene la mínima estatura requerida  $A_1, \dots, A_N$  ( $90 \leq A_i \leq 200$ ) de cada juego en el parque.

### Salida

Imprime una única línea con un entero indicando el número de juegos a los que Carlitos puede subir, esto es, el número de juegos para los cuales la estatura de Carlitos es al menos igual a la estatura mínima requerida.

<b>Ejemplo de entrada 1</b> 1 100 100	<b>Ejemplo de salida 1</b> 1
<b>Ejemplo de entrada 2</b> 6 120 200 90 100 123 120 169	<b>Ejemplo de salida 2</b> 3

## Problema B

# Baraja Barajada

Una barajada justa es una forma de reacomodar  $N$  cartas de una baraja dispuesta horizontalmente de izquierda a derecha. En esta barajada, las cartas son inicialmente divididas en dos partes consecutivas de tamaños posiblemente diferentes en que incluso una de ellas puede tener cero cartas.

Denotemos por  $L$  y  $R$  las partes de izquierda y derecha, respectivamente. Las cartas de  $L$  son, entonces, combinadas con las cartas de  $R$ , de tal forma que el orden relativo entre las cartas de cada partición se mantenga.

Te presentan una disposición final de las cartas y debes descubrir cuál es la cantidad mínima de barajadas justas que deben ser realizadas en la baraja inicial para que se llegue a este estado. Inicialmente, las cartas de la baraja pueden ser vistas como la secuencia  $1\ 2\ \dots\ N$ .

Por ejemplo, comenzando con la secuencia  $1\ 2\ 3\ 4\ 5$  y realizando una barajada justa con  $L = 1\ 2$  y  $R = 3\ 4\ 5$ , podríamos obtener las siguientes permutaciones:

- $1\ 3\ 2\ 4\ 5$
- $1\ 3\ 4\ 2\ 5$
- $3\ 4\ 5\ 1\ 2$
- $1\ 2\ 3\ 4\ 5$
- etc

Cada una de las permutaciones anteriores representa un posible resultado de la barajada justa. Note que  $1\ 3\ 2\ 5\ 4$  no es una barajada posible ya que el orden relativo de las cartas  $4$  y  $5$  de  $R$  no se preserva.

Asuma que el resultado de la primera barajada es  $1\ 3\ 2\ 4\ 5$ . Si hiciéramos una segunda barajada justa en ella, podríamos particionar la baraja en  $L = 1\ 3\ 2\ 4$ ,  $R = 5$ , y combinar ambas para obtener la permutación  $1\ 3\ 2\ 5\ 4$ .

### Entrada

La primera línea contiene un entero  $N$  ( $1 \leq N \leq 10^6$ ), el número de cartas en la baraja. La segunda línea contiene una permutación de números enteros de  $1$  a  $N$  que describe la disposición final de las cartas.

### Salida

Imprime un único entero  $K$ , que representa el menor número de barajadas justas necesarias para obtener la permutación dada.

<b>Ejemplo de entrada 1</b> 5 3 4 5 1 2	<b>Ejemplo de salida 1</b> 1
<b>Ejemplo de entrada 2</b> 10 1 6 5 2 10 3 4 8 7 9	<b>Ejemplo de salida 2</b> 3

<b>Ejemplo de entrada 3</b> 5 5 4 2 3 1	<b>Ejemplo de salida 3</b> 2
-----------------------------------------------	---------------------------------

<b>Ejemplo de entrada 4</b> 8 1 6 2 3 7 8 4 5	<b>Ejemplo de salida 4</b> 1
-----------------------------------------------------	---------------------------------

<b>Ejemplo de entrada 5</b> 5 1 2 3 4 5	<b>Ejemplo de salida 5</b> 0
-----------------------------------------------	---------------------------------

## Problema C

# Caminata en la Montaña

Finalmente, los agotadores exámenes quedaron atrás y ha llegado el momento de disfrutar de unas merecidas vacaciones lejos del ajetreo universitario. Con tus maletas empacadas, te embarcas en una aventura para explorar una de las montañas más reconocidas de tu región – un destino con el que has estado soñando visitar durante un período considerable de tiempo.

Esta majestuosa montaña de impresionantes dimensiones, ofrece una gran cantidad de opciones para hacer senderismo. Conformada por un total de  $N$  puntos de referencia, cada uno identificado de manera única por un número que va desde 1 hasta  $N$ , la montaña presenta una cautivadora red de  $N - 1$  senderos interconectados. Estos senderos garantizan que puedas recorrer sin esfuerzo alguno desde cualquier punto de referencia a otro, conectando así toda la montaña en una red continua.

Cada punto de referencia, denotado por el índice  $i$ , tiene asociado un puntaje llamado acertadamente  $v_i$ . Este puntaje refleja la cantidad esperada de “likes” que atraería tu publicación en redes sociales de ese punto de referencia. Lleno de entusiasmo, has decidido añadir un nivel adicional de emoción a tu caminata con el “reto de la moda de fotos”. En este reto, tu objetivo es nada menos que extraordinario: capturar y publicar fotos de manera que cada foto obtenga más “likes” que su predecesora.

Las reglas de este reto indican que tu recorrido debe desarrollarse de la siguiente manera:

1. Comienzas tu caminata desde el punto de referencia 1.
2. Avanzas por los senderos disponibles, pasando por los puntos de referencia, siempre avanzando y sin retroceder.
3. En cada punto de referencia que visitas, tienes la opción de tomar una foto y publicarla en tu plataforma de redes sociales.

A medida que tu caminata se desarrolla, sientes curiosidad por los resultados potenciales. Específicamente, para cada punto de referencia  $i$ , buscas determinar el número máximo de fotos que se pueden publicar si inicias la caminata desde el punto de referencia 1 y concluyes en el punto de referencia  $i$  (aunque no necesariamente hayas tomado una foto del punto de referencia  $i$ ). El reto consiste en seleccionar estratégicamente los puntos de referencia para capturar fotos, asegurándote de que cada foto supere a su predecesora en términos de “likes” esperados.

Tu tarea es calcular este número máximo de fotos para cada punto de referencia. ¿Puedes enfrentar este reto y descubrir el camino más cautivador y atractivo a través de esta montaña?

### Entrada

La primera línea contiene un entero  $N$  ( $1 \leq N \leq 10^5$ ), representando el número de puntos de referencia en la montaña. La segunda línea contiene  $N - 1$  enteros  $p_2, p_3, \dots, p_N$  ( $1 \leq p_i \leq N$ ), donde  $p_i$  representa que hay un sendero entre los puntos de referencia  $i$  y  $p_i$ . La tercera línea contiene  $N$  enteros  $v_1, v_2, \dots, v_N$  ( $1 \leq v_i \leq 10^9$ ), donde  $v_i$  representa la cantidad esperada de “likes” que recibiría una foto en el  $i$ -ésimo punto de referencia.

### Salida

Imprime una línea con  $N - 1$  enteros, donde el  $i$ -ésimo entero representa la máxima cantidad de fotos que puedes publicar si iniciaras la caminata en el punto de referencia 1 y terminarás en el punto de referencia  $(i + 1)$ .

<b>Ejemplo de entrada 1</b> 5 1 1 3 3 5 7 7 6 8	<b>Ejemplo de salida 1</b> 2 2 2 3
----------------------------------------------------------	---------------------------------------

<b>Ejemplo de entrada 2</b> 5 3 1 3 1 5 4 7 6 5	<b>Ejemplo de salida 2</b> 2 2 2 1
----------------------------------------------------------	---------------------------------------

## Problema D

# Desvío

En la ciudad de Nlogonia, el alcalde finalmente va a cumplir su promesa de campaña y hará repavimentar algunos tramos de los caminos. Sin embargo, cuando un tramo estuviera siendo repavimentado, los carros no podrán usarlo y por lo tanto un desvío debe ser utilizado.

Cada tramo conecta dos esquinas en la ciudad, tiene una longitud positiva y puede ser recorrido en ambas direcciones.

Un desvío es un camino alternativo que puede servir como un sustituto temporal para el tramo en obras. Más específicamente, si el tramo que conecta las esquinas  $U$  y  $V$  estuviera intransitable, el desvío debe ser una secuencia de tramos que comienza en  $U$ , termina en  $V$ , y no usa el tramo que conecta  $U$  directamente con  $V$ . El objetivo es encontrar el desvío más corto para cada tramo, de forma que se minimice el impacto mientras las obras estuvieran siendo hechas.

Como Integrante del Centro de Pavimentación y Carros, debes ayudar al alcalde a calcular cuál es la longitud del desvío más corto, para cada tramo.

### Entrada

La primera línea contiene dos enteros,  $N$  y  $M$  ( $1 \leq N \leq 300$ ), que representan, respectivamente, el número de esquinas y el número de tramos. Cada una de las  $M$  líneas siguientes contiene tres enteros:  $U$ ,  $V$ , y  $L$  ( $1 \leq U \leq N$ ,  $1 \leq V \leq N$ ,  $U \neq V$ ,  $1 \leq L \leq 10^6$ ), que representan un tramo de dos sentidos de longitud  $L$  que une las esquinas  $U$  y  $V$ . Ningún tramo es representado más de una vez.

### Salida

Imprime  $M$  líneas, donde cada línea contiene un entero. El  $i$ -ésimo entero debe ser la longitud del desvío más corto para el  $i$ -ésimo tramo, o -1 si no fuera posible hacer un desvío. El orden de los tramos en la salida debe ser el mismo orden proporcionado en la entrada.

Ejemplo de entrada 1	Ejemplo de salida 1
4 5	9
1 2 4	5
1 3 8	9
2 3 4	11
4 1 2	10
3 4 3	

Ejemplo de entrada 2	Ejemplo de salida 2
2 1	-1
1 2 1	

## Problema E

# Extrayendo Polen

Ha llegado la primavera, marcando el comienzo de otra intensa temporada de trabajo en la Sociedad de las abejas de Chapecó (SBC). En el jardín de la SBC, han florecido  $N$  hermosas flores, cada una con una cierta cantidad de granos de polen que serán recolectados por las laboriosas trabajadoras de la sociedad.

Para mantener un entorno de trabajo seguro, la SBC tiene reglas muy estrictas para la recolección de polen, que son las siguientes:

- Cuando una abeja visita una flor, debe recolectar una cantidad de polen igual a la suma de los dígitos del total de polen actual en esa flor. Por ejemplo, si una flor tiene 123 granos de polen, la abeja que la visita debe recolectar  $1 + 2 + 3 = 6$  granos, dejando la flor con  $123 - 6 = 117$  granos. Si la flor tiene 201 granos, la abeja recolectará  $2 + 0 + 1 = 3$  granos, dejando la flor con  $201 - 3 = 198$  granos de polen.
- Todas las abejas deben formar una fila al comienzo del día; aquella que esté en la primera posición de la fila debe recolectar polen de alguna de las flores que tienen la mayor cantidad de polen. Si la abeja visita una flor con 0 granos, recolecta cero granos de polen. Después de recolectar el polen de una flor, la abeja termina su turno de trabajo y regresa a la colmena.

A la abeja Gertrude estas reglas le parecen muy extrañas, así que buscó tu ayuda para saber cuánto polen recolectará en su turno. Gertrude tiene una vista increíble y descubrió que actualmente está en la  $K$ -ésima posición de la fila.

### Entrada

La primera línea de entrada tiene dos enteros  $N$  ( $1 \leq N \leq 10^6$ ) y  $K$  ( $1 \leq K \leq 10^9$ ) que representan, respectivamente, el número de flores y la posición de Gertrude en la fila. La segunda línea contiene  $N$  enteros, donde el  $i$ -ésimo entero  $F_i$  ( $1 \leq F_i \leq 10^6$ ) para  $1 \leq i \leq N$  representa la cantidad inicial de polen de la  $i$ -ésima flor.

### Salida

Imprime un único entero  $Q$ , que representa la cantidad de polen que será recolectada por Gertrude.

Ejemplo de entrada 1	Ejemplo de salida 1
5 3 22 15 7 2 1	6

*Explicación del ejemplo 1:*

La primer abeja recolectará polen de la primer flor, dejándola con  $22 - (2 + 2) = 18$  granos restantes. La segunda abeja también recolectará de la primera flor, dejándola con  $18 - (1 + 8) = 9$  granos de polen restante. Finalmente, Gertrude, la tercer abeja de la fila, recolectará polen de la segunda flor, recolectando en total  $1 + 5 = 6$  granos de polen, que es la respuesta para este caso de prueba.

Ejemplo de entrada 2	Ejemplo de salida 2
3 10 21 21 21	0

<b>Ejemplo de entrada 3</b> 3 9 21 21 21	<b>Ejemplo de salida 3</b> 9
------------------------------------------------	---------------------------------

## Problema F

# Fiestas Agotadoras

William está planeando sus próximas vacaciones. Un problema recurrente en todas sus vacaciones es la necesidad de lidiar con el cansancio. Algunos días no los disfruta mucho, pues después de varias actividades el cansancio comienza a ser mayor de lo que puede soportar.

En esta ocasión, William tuvo una idea. Va a estimar el impacto que tiene en su disposición cada una de las actividades de turismo que puede realizar. Él notó que algunas de las actividades de las vacaciones, como deportes y caminatas, son agotantes, consumiendo su disposición, mientras que otras actividades como las obras de teatro y musicales, son revitalizantes, recuperando su disposición.

Más precisamente, William comienza con  $D$  unidades de disposición y separa las actividades en dos grupos:  $C$  actividades agotantes, y  $R$  actividades revitalizantes. Cada actividad agotante requiere una cierta cantidad de disposición, y por lo tanto, consume esa cantidad cuando es realizada. Cada actividad revitalizante le proporciona una cierta cantidad de disposición cuando es realizada. Además, ordena las actividades de cada grupo según un orden de preferencia, pues hay actividades que está más ansioso por realizar. Es importante mencionar que las actividades de los dos grupos pueden ser intercaladas, pero William nunca realizará una actividad de un grupo sin haber realizado todas las actividades anteriores, pues esto iría en contra de sus preferencias.

A lo largo de sus vacaciones, al decidir qué actividad realizar en seguida, el elegirá la primera actividad agotante que aún no haya realizado, si es que tiene suficiente disposición para hacerlo. En caso contrario, llevará a cabo la siguiente actividad revitalizante que aún no haya realizado, si es que aún hay alguna, recuperando así cierta cantidad de disposición. Naturalmente, si en algún momento ya no quedan actividades agotantes por realizar, simplemente podrá llevar a cabo todas las actividades revitalizantes restantes.

Teniendo en cuenta este proceso, William ha pedido tu ayuda para determinar cuántas actividades (incluyendo tanto agotantes como revitalizantes) puede llevar a cabo.

### Entrada

La primer línea de entrada contiene tres enteros:  $D$ ,  $C$ , y  $R$ . Representando respectivamente, la cantidad de disposición inicial, el número de actividades agotantes, y el número de actividades revitalizantes ( $1 \leq D \leq 10^5$ ,  $1 \leq C \leq 10^4$ , y  $1 \leq R \leq 10^4$ ). Cada una de las siguientes  $C$  líneas contiene un entero  $C_i$  ( $1 \leq C_i \leq 10^5$  para  $1 \leq i \leq C$ ), representando el consumo de disposición de una actividad agotante, en orden de preferencia. Finalmente, cada una de las siguientes  $R$  líneas contiene un entero  $R_i$  ( $1 \leq R_i \leq 10^5$  para  $1 \leq i \leq R$ ), representando el retorno de disposición de una actividad revitalizante, en orden de preferencia.

### Salida

Imprima una única línea que contenga un único entero, el número total de actividades (incluyendo tanto agotantes como revitalizantes) que William puede llevar a cabo.

Ejemplo de entrada 1	Ejemplo de salida 1
40 3 3	5
30	
20	
10	
5	
5	
5	

<b>Ejemplo de entrada 2</b>	<b>Ejemplo de salida 2</b>
40 2 2 60 80 5 10	2

<b>Ejemplo de entrada 3</b>	<b>Ejemplo de salida 3</b>
100 3 1 60 60 50 10	2

## Problema G

# Gran Tratado de Byteland

La Gran Guerra de Byteland ha terminado. Los reinos restantes están discutiendo ahora el Tratado de División, que dividirá toda la tierra en el mundo entre ellos. Se referirá no solo al mundo conocido, sino también a cualquier territorio aún por descubrir o habitar, incluyendo tierra o mar. Podemos asumir que el mundo es un plano infinito.

Cada reino en el continente de Byteland tiene una única capital, y el Tratado de División se basará en sus ubicaciones: declara que cada pedazo de tierra pertenece al reino cuya capital es la más cercana en un vuelo de pájaro (o en línea recta). En otras palabras: donde sea que te encuentres en el mundo, si  $C$  es la capital más cercana a ti, estarás en el territorio del reino de  $C$ . Si hay un empate entre las distancias de dos o más capitales, ese lugar estará en la frontera entre sus reinos.

Bajo este tratado, algunos reinos pueden quedar encerrados entre otros, mientras que otros reinos pueden terminar con territorio ilimitado. Por lo tanto, algunos monarcas están cuestionando el tratado. Para informar esta discusión, exigen tu ayuda. Dadas las coordenadas de ubicación de cada capital en el continente de Byteland, debes averiguar qué reinos tendrían un territorio infinitos según el Tratado de División.

### Entrada

La primer línea de entrada contiene un solo entero  $N$  ( $2 \leq N \leq 10^5$ ), el número de reinos. Cada reino es identificado por un número entero único entre 1 y  $N$ . Cada una de las siguientes  $N$  líneas contiene dos enteros  $X$  y  $Y$  ( $0 \leq X, Y \leq 10^4$ ), que representan la coordenada en 2D de la ubicación de la capital de un reino. Las ubicaciones de las capitales son dadas en orden creciente según el identificador de los reinos, no hay dos capitales que tengan la misma ubicación, y puedes suponer que cada capital tiene un tamaño insignificante.

### Salida

Imprime una sola línea con una lista de enteros separados por espacio en orden creciente: Los identificadores de los reinos que tendrían un territorio infinito según el Tratado de División. Se garantiza que siempre habrá al menos uno de esos reinos.

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

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

## Problema H

# Honrado Trabajador

Rafael vive en una sociedad igualitaria ideal: todos los trabajos ofrecen un salario diario igual y solo necesitas trabajar en un empleo que sea de tu interés. Infelizmente, ni siquiera esta utopía estaba preparada para el hecho de que Rafael no tenga ninguna habilidad.

Para compensar esta situación, Rafael irá a trabajar solo en obras, donde empleadores potenciales no tendrán tiempo de reparar en cuán poco cualificado Rafael es para el trabajo. Así mismo, dada su poca cualificación para esos trabajos, él necesita comprar cartas de recomendación falsas para ser contratado.

Rafael tiene  $N$  trabajos a su disposición en los próximos días. El  $i$ -ésimo trabajo comienza en el día  $\ell_i$ , termina en el final del día  $r_i$  y paga exactamente  $S$  monedas de oro por día trabajado. Rafael es realmente malo para el trabajo, y no logra trabajar en dos empleos al mismo tiempo; además, él puede solo comenzar en el empleo  $i$  en el día  $\ell_i$  pero, una vez contratado, él puede renunciar al final de cada día, manteniendo el dinero de los días trabajados (inclusive el último), siendo así capaz de comenzar otro trabajo a partir del día siguiente (pero no en el mismo día de su renuncia). Rafael también sabe que necesita  $c_i$  monedas de oro para comprar la carta de recomendación falsa para el trabajo  $i$ . Consciente de su falta de habilidades y de la necesidad de las cartas falsas, Rafael reservó un dinero y siempre es capaz de comprar cuantas cartas fueran necesarias, incluso antes de comenzar en cualquiera de los  $N$  trabajos.

Dadas las descripciones de los trabajos disponibles, ¿cuál es el lucro máximo que Rafael puede conseguir, tomando en cuenta los costos para la compra de las cartas falsas?

### Entrada

La primera línea contiene dos enteros  $N$  ( $1 \leq N \leq 10^6$ ) y  $S$  ( $1 \leq S \leq 10^9$ ), que representan el número de trabajos disponibles y cuánto es la paga de cada uno por día trabajado.

Cada una de las  $N$  líneas siguientes contiene tres enteros:  $\ell_i$ ,  $r_i$  y  $c_i$  ( $1 \leq \ell_i \leq r_i \leq 10^9$ ,  $1 \leq c_i \leq 10^9$ ), que representan la fecha de inicio, la fecha de fin, y el costo de comprar una carta de recomendación falsa para el trabajo  $i$ , respectivamente.

### Salida

Imprime un único entero que corresponda al lucro máximo de Rafael después del fin de todos los trabajos.

Ejemplo de entrada 1	Ejemplo de salida 1
<pre>3 3 1 5 10 2 10 4 5 15 1</pre>	<pre>37</pre>

*Explicación del ejemplo 1:*

Es óptimo iniciar el segundo trabajo, después pasar al tercero, para un total de  $14 \times 3 - 4 - 1 = 37$  monedas de oro.

<b>Ejemplo de entrada 2</b> 3 5 1 1 3 2 3 4 3 3 1	<b>Ejemplo de salida 2</b> 8
---------------------------------------------------------------	---------------------------------

<b>Ejemplo de entrada 3</b> 1 1000 1 1 654	<b>Ejemplo de salida 3</b> 346
--------------------------------------------------	-----------------------------------

*Explicación del ejemplo 3:*

Aunque tiene que tomar dinero de sus ahorros para comprar cartas de recomendación falsas puede obtener ganancias.

<b>Ejemplo de entrada 4</b> 1 5 1 3 20	<b>Ejemplo de salida 4</b> 0
----------------------------------------------	---------------------------------

*Explicación del ejemplo 4:*

No vale la pena gastar dinero en cartas de recomendación falsas para este trabajo.

## Problema I

# Intervalos Combinados

Te encuentras en un misterioso mundo binario, donde te espera un arreglo de  $N$  dígitos binarios para que lo examines. Cada dígito es un cero o un uno, creando un patrón único en todo el paisaje. Tu misión es descubrir los patrones ocultos de este reino binario desentrañando el significado de los subarreglos con un número impar de unos.

El arreglo de dígitos se representa como  $b_1, b_2, \dots, b_N$ . Tu tarea es emprender un viaje para descubrir los enigmáticos subarreglos, segmentos de dígitos consecutivos, y determinar el número de subarreglos que contengan un número impar de unos.

Mientras recorres este paisaje binario, recuerda que un subarreglo está definido por sus dígitos de inicio y finalización. Por ejemplo, en la secuencia  $[b_1, b_2, b_3]$ , los subarreglos incluyen  $[b_1]$ ,  $[b_2]$ ,  $[b_3]$ ,  $[b_1, b_2]$ ,  $[b_2, b_3]$ , y  $[b_1, b_2, b_3]$ .

Tu misión es diseñar un algoritmo que determine el número total de subarreglos que contienen un número impar de unos en esta secuencia binaria. Por favor, no olvides que es posible que la respuesta no quepa en un entero de 32 bits.

### Entrada

La primer línea contiene un entero  $N$  ( $1 \leq N \leq 10^5$ ) representando la longitud de la secuencia binaria.

La segunda línea contiene  $N$  dígitos binarios  $b_1, b_2, \dots, b_N$  ( $b_i \in \{0, 1\}$ ) representando los elementos de la secuencia.

### Salida

Imprime una sola línea con un entero representando la cantidad de subarreglos en la secuencia que tienen un número impar de unos.

<b>Ejemplo de entrada 1</b> 3 0 1 0	<b>Ejemplo de salida 1</b> 4
<b>Ejemplo de entrada 2</b> 10 1 0 0 1 1 0 1 1 1 0	<b>Ejemplo de salida 2</b> 30

## Problema J

# Juego de Volley

El escenario está listo para las finales de los Juegos Olímpicos de súper volleyball, ¡y las tensiones están muy altas! Ricardão, el entrenador de uno de los equipos competidores, está posicionando meticulosamente a sus jugadores para obtener una ventaja estratégica. Sin embargo, una preocupación lo aqueja: la posibilidad de que algunas áreas de la cancha puedan quedar inadecuadamente cubiertas por sus jugadores. Esto podría llevar a saltos laterales agotadores para alcanzar la pelota.

En este partido de alto riesgo, tu tarea es ayudar al entrenador Ricardão analizando la configuración de la cancha y las posiciones de los jugadores. Calcula la distancia máxima de un salto que se requiere para que cualquier jugador intercepte la pelota asegurándote de que ningún punto en la cancha quede vulnerable. Los equipos profesionales de súper volleyball son muy buenos y pueden determinar fácilmente cuál es el jugador más cercano al lugar donde caerá la pelota; solo este jugador intentará alcanzarla, los demás permanecerán quietos.

La cancha está definida por un rectángulo  $\mathcal{R}$  alineado con los ejes, definido por sus cuatro vértices. Los jugadores se modelan como un conjunto de  $N$  puntos dentro de este rectángulo. Tu tarea es determinar la distancia más pequeña  $d$  tal que cualquier punto en la cancha pueda ser alcanzado por al menos un jugador, si un jugador puede saltar una distancia de hasta  $d$  en cualquier dirección. ¡Recuerda: el borde de la cancha también forma parte de la cancha y debe ser cubierto por los jugadores!

### Entrada

Las primeras cuatro líneas contienen los cuatro vértices de  $\mathcal{R}$ ; es decir, la  $i$ -ésima línea contiene dos enteros  $x_i$  e  $y_i$  ( $-10^5 \leq x_i, y_i \leq 10^5$ ), que representan las coordenadas del  $i$ -ésimo vértice de  $\mathcal{R}$ . La quinta línea contiene un único entero  $N$  ( $1 \leq N \leq 10^5$ ), que representa el número de jugadores. Cada una de las siguientes  $N$  líneas contiene las coordenadas de los jugadores, donde la  $i$ -ésima línea contiene dos enteros  $x_i$  e  $y_i$  ( $-10^5 \leq x_i, y_i \leq 10^5$ ), que representan las coordenadas del  $i$ -ésimo jugador. Se garantiza que cada jugador se encuentra dentro de la cancha.

### Salida

Imprime una sola línea con el valor más pequeño de  $d$  que sea suficiente para cubrir toda la cancha. La salida se considerará correcta si está dentro de un error absoluto o relativo de  $10^{-5}$  con respecto a la respuesta correcta.

Ejemplo de entrada 1	Ejemplo de salida 1
-1 -1	1.41421356237
1 -1	
1 1	
-1 1	
1	
0 0	

<b>Ejemplo de entrada 2</b>	<b>Ejemplo de salida 2</b>
1 -1 -1 3 1 3 -1 -1 3 0 0 1 3 -1 3	1.666666666667

## Problema K

# $k$ para Más, $k$ para Menos

La vida de aquellos que estudian ciencias de la computación no siempre es tan fácil como parece. Unos días puedes estar implementando un algoritmo revolucionario, y otros te encuentras leyendo el mismo libro por décima vez. Sin embargo, en todo momento, estamos buscando lo mismo: optimizar y automatizar tareas. En este caso, un profesor necesita tu ayuda para guiar a sus estudiantes hacia su próximo examen. En opinión del profesor, no es fácil decidir cuánto tiempo los estudiantes deberían dedicar al estudio de temas teóricos y cuánto tiempo a implementar algoritmos.

No es la primera vez que el profesor imparte esta materia, así que la cantidad de datos disponibles es tan grande que pudo crear dos polinomios para describir la calificación final de cada estudiante. Si el estudiante dedica  $x$  unidades de su tiempo al estudio de la teoría su calificación aumentará en  $t(x)$ . Si el estudiante dedica  $x$  unidades de su tiempo a implementar algoritmos, su calificación aumentará en  $p(x)$ . Así que el estudiante que dedique la misma cantidad  $x$  de tiempo en cada una de las áreas tendrá una calificación final de  $t(x) + p(x)$ .

Recientemente, uno de los estudiantes ha estado destacándose de manera impredecible. No oculta su técnica a nadie: “¡Estudio mucho más teoría que práctica!”. El profesor cree que esto es una mentira, para confirmar su sospecha, decidió estimar las calificaciones de los estudiantes si estudiaran más teoría que práctica (o más práctica que teoría). ¿Puedes calcular el polinomio  $q(x) = t(x + K) + p(x - K)$ ? Esto le permitiría describir la calificación de cada estudiante si cambiaran su estrategia de estudio.

### Entrada

La entrada consta de tres líneas. La primera línea contiene dos enteros:  $N$ , que representa el grado de los polinomios  $t$  y  $p$  ( $1 \leq N \leq 10^5$ ), y  $K$  ( $-10^5 \leq K \leq 10^5$ ). La segunda línea contiene los  $N + 1$  coeficientes de  $t$ , y la tercera línea contiene los  $N + 1$  coeficientes de  $p$ . Los coeficientes se dan en orden creciente de grado, de modo que el último coeficiente de la línea correspondiente al término de grado  $N$ , todos son no negativos con un valor máximo de  $10^6$ .

### Salida

Tu programa debe imprimir una línea con  $N + 1$  enteros, los coeficientes del polinomio  $q(x)$  en orden creciente de grado modulo 998244353.

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

Ejemplo de entrada 2	Ejemplo de salida 2
2 0 1 2 3 4 5 6	5 7 9

Ejemplo de entrada 3	Ejemplo de salida 3
2 -1 3 3 3 1 0 0	4 998244350 3

## Problema L

# Lexicográficamente Agradable

En la pintoresca villa de Lexiconia, vivían dos amigos, Lily y Ethan. Un día, una misteriosa carta llegó a la casa de los amigos, cerrada con una encantadora insignia. Dentro del sobre, ellos encontraron una adivinanza muy compleja incluso para las mentes más sabias de la villa.

En esa adivinanza, fueron dados un entero  $K$  y una cadena  $S$  conteniendo solo letras minúsculas, que puede ser alterada de acuerdo a una curiosa regla. En cada momento, los aldeanos tienen la libertad de escoger un índice  $i$  y, mágicamente, los caracteres  $S_i$  y  $S_{i+K}$  son intercambiados. La adivinanza es resuelta cuando la cadena lexicográficamente mínima, usando solo operaciones que respeten la regla impuesta para la adivinanza, es encontrada.

La villa quedó muy curiosa y emocionada con la adivinanza. Lily y Ethan, siempre con hambre voraz de aventuras, decidieron sumergirse en ese desafío. Sin embargo, a medida que los dos observaban la cadena, ellos notaron que, en el camino para el éxito, habían innumerables cambios posibles.

Con la cadena de la adivinanza vívida en sus mentes, los amigos se preguntaron: ¿cómo navegar en ese gigantesco mar de posibilidades y determinar la cadena lexicográficamente mínima? Cada cambio realizado parecía como hojear un libro mágico, que revelaría nuevos secretos y misterios.

El tiempo pasó y hasta ahora Lily y Ethan no terminaron de resolver esta adivinanza. ¿Será que puedes ayudarlos?

### Entrada

La primera línea de entrada contiene la cadena  $S$  ( $1 \leq |S| \leq 10^5$ ). La segunda línea de entrada contiene el entero  $K$  ( $1 \leq K < |S|$ ).

### Salida

Imprime una única línea conteniendo la cadena lexicográficamente mínima posible de ser hecha utilizando esas operaciones.

Ejemplo de entrada 1	Ejemplo de salida 1
zaaab 4	baaaz

Ejemplo de entrada 2	Ejemplo de salida 2
njoab 2	banjo

## Problema M

# Malla Aérea

En el reino de Quadradonia, el monarca quiere revisar todas las tarifas aéreas. Para esto, le ha pedido a su contador una tabla con las propuestas para nuevos precios.

Sin embargo, el monarca estudió en el Instituto de Computación y Programación de Chapecó (ICPC) y tiene conocimiento suficiente para exigir coherencia en la tabla. Una tabla es *coherente* si ninguna ruta con escalas es más barata que un vuelo directo.

Verificada la coherencia de la tabla, al monarca le gustaría disminuir el número de vuelos directos, sin que se aumente el costo de ningún viaje.

Su problema es verificar la coherencia de la tabla, y, si esta es coherente, informar al monarca cuántos vuelos directos pueden ser eliminados sin hacer más costoso ningún viaje.

### Entrada

La primera línea contiene  $N$  ( $1 \leq N \leq 100$ ), que indica el número de ciudades de Quadradonia que tienen vuelos. Seguido de  $N$  líneas,  $L_1, L_2, \dots, L_N$ . La línea  $L_i$  contiene  $N$  enteros  $C_{i1}, C_{i2}, \dots, C_{iN}$ , donde  $C_{ij}$  es el costo del vuelo directo entre las ciudades  $i$  y  $j$ . El costo de ida y vuelta entre dos ciudades es siempre la misma, esto es,  $C_{ij} = C_{ji}$ , para todos los pares  $i, j$  tales que  $1 \leq i \leq N$  y  $1 \leq j \leq N$ . Cuando  $i = j$ ,  $C_{ij} = 0$ . Cuando  $i \neq j$ ,  $1 \leq C_{ij} \leq 10^3$ .

### Salida

Imprima una línea con un entero. Si la tabla es incoherente, el entero debe ser igual a  $-1$ . Si la tabla es coherente, el entero debe ser igual al máximo número de vuelos directos que pueden ser eliminados sin que se aumente el costo de algún viaje para los pasajeros.

<b>Ejemplo de entrada 1</b> 3 0 1 2 1 0 1 2 1 0	<b>Ejemplo de salida 1</b> 1
<b>Ejemplo de entrada 2</b> 3 0 2 2 2 0 2 2 2 0	<b>Ejemplo de salida 2</b> 0
<b>Ejemplo de entrada 3</b> 3 0 2 9 2 0 2 9 2 0	<b>Ejemplo de salida 3</b> -1