

Maratón de programación de la SBC 2021

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

Maratona de Programação da SBC 2021
Tercera Fecha Gran Premio de México 2021
Tercera Fecha Gran Premio de Centroamérica 2021
Torneo Argentino de Programación 2021

30 de Octubre de 2021

Cuaderno de Problemas

Información General

Este cuaderno contiene 14 problemas; Las páginas están numeradas de 1 a 25, 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

Asignación de premios

Pronto se realizará una competencia en Nlogonia para determinar quién es el mejor programador Nlogonio de todos los tiempos.

Participarán N competidores y no habrá empates. Es decir, cada competidor terminará ubicado en una posición diferente desde 1 hasta N inclusive. Una menor posición indica un resultado mejor.

Los organizadores del evento han decidido otorgar a cada competidor un premio, en forma de puntos de rating. Como máximo, cada competidor puede recibir R puntos de rating. Para ser justos con los competidores que tuvieron un mejor desempeño, un competidor nunca recibirá menos puntos de rating que otro competidor con una posición peor.

Algunos competidores sin embargo son más codiciosos, y quieren recibir más puntos de rating para estar contentos. El competidor ubicado en la posición i necesita recibir un premio de al menos p_i puntos de rating para estar contento.

Ina es una organizadora muy curiosa, y se pregunta de cuántas maneras diferentes es posible asignar los premios a los competidores, de modo tal que se satisfagan las condiciones impuestas por la organización y que todos los competidores estén contentos. Además, como este número puede ser muy grande, debes calcularlo módulo $10^9 + 7$.

Dos maneras de asignar premios son diferentes, si al menos un competidor recibe una cantidad diferente de puntos de rating.

Entrada

La primera línea contiene dos enteros N y R ($1 \leq N \leq 5000$, $1 \leq R \leq 10^9$), que indican la cantidad de competidores y la cantidad máxima de puntos de rating que puede recibir cada competidor como premio, respectivamente.

La segunda línea contiene N enteros, p_i ($1 \leq p_i \leq 10^9$), que indican la mínima cantidad de puntos de rating que cada competidor necesita recibir como premio para estar contento.

Salida

Imprima la cantidad de formas diferentes en que se pueden asignar los puntos de rating de premio, módulo $10^9 + 7$.

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

Problema B

Bellas palabras

Dada una cadena A de longitud N y un conjunto S de M cadenas.

Una permutación cíclica B_i de A , en la que i se encuentra entre 1 y N , es la cadena:

$$B_i = A_i A_{i+1} \cdots A_{N-1} A_N A_1 A_2 \cdots A_{i-2} A_{i-1}$$

Y su puntuación se define como la longitud máxima de una subcadena de B_i que también es subcadena de alguna cadena en S .

Una subcadena está definida como una secuencia contigua de letras. Por ejemplo, `ab` y `dc` son subcadenas de `abfdc`, pero, `ad` y `fc` no son subcadenas de `abfdc`.

Su tarea es calcular el puntaje mínimo sobre todas las permutaciones cíclicas de la cadena A .

Entrada

La primera línea contiene dos enteros positivos N y M , ($1 \leq N \leq 10^5$, $1 \leq M \leq 10^4$), indicando, respectivamente, la longitud de la cadena A y el tamaño del conjunto S .

La segunda línea contiene la cadena A .

Cada una de las siguientes M líneas contiene una cadena s_i , representando la i -ésima cadena del conjunto S .

Todas las cadenas contienen solo letras minúsculas del alfabeto inglés. Se garantiza que la suma de las longitudes de las cadenas en S nunca será mayor a 10^5 caracteres.

Salida

Imprima una línea con un número entero, indicando la mínima puntuación sobre todas las permutaciones cíclicas de A .

<p>Ejemplo de entrada 1</p> <pre>7 3 acmicpc acm icpc maratona</pre>	<p>Ejemplo de salida 1</p> <pre>3</pre>
<p>Ejemplo de entrada 2</p> <pre>11 4 competition oncom petition ztxvu fmwper</pre>	<p>Ejemplo de salida 2</p> <pre>5</pre>
<p>Ejemplo de entrada 3</p> <pre>12 4 latinamerica zyvu okp wsg kqpdb</pre>	<p>Ejemplo de salida 3</p> <pre>0</pre>

Problema C

Creando múltiplos

Malba es un niño muy inteligente al que le gusta realizar cálculos. Ha ganado varias competencias, incluida la prestigiosa competencia Tahan, donde obtuvo el primer lugar representando a su país, Logonia.

Ha creado un acertijo, en el cual considera un número N que al escribirse en una cierta base B , está formado por L dígitos. En el acertijo se debe reducir a lo sumo uno de los dígitos, de tal modo que se forme un nuevo número M que sea múltiplo de $B + 1$. Pero, hay una trampa: si hay varias opciones, deberá ser alterado el dígito que haga el valor de M lo más bajo posible.

Por ejemplo, supongamos que $B = 10$ y $N = 23456$. Hay dos maneras en las que se puede obtener M : ya sea que se reduzca el dígito 4 a 0, o que se reduzca el dígito 6 a 2. Entonces, 4 debe ser cambiado a 0, y entonces $M = 23056$. A veces no hay solución alguna, como es el caso si $B = 10$ y $N = 102$. En este caso, si se cambiara el dígito 1 por el 9 obtendríamos un múltiplo de 11, ¡pero no tenemos permitido incrementar el valor de un dígito!

Obsérvese que puede ser necesario reducir el primer dígito a 0. Por ejemplo cuando $B = 10$ y $N = 322$.

¿Puedes decir cuál es el dígito que se debe reducir, y cuál es su nuevo valor?

Entrada

La primera línea contiene dos enteros B y L ($2 \leq B \leq 10^4$, $1 \leq L \leq 2 \times 10^5$), que representan respectivamente, la base y la cantidad de dígitos del número N .

La segunda línea contiene L enteros D_1, D_2, \dots, D_L ($0 \leq D_i < B$ para $i = 1, 2, \dots, L$), que representan los dígitos del número N . El primer dígito, D_1 , es el más significativo y el último, D_L , es el menos significativo.

Salida

Debes imprimir una línea que contenga dos enteros separados por un espacio. El primer entero es el índice del dígito que debe cambiarse (recuerda que el índice del primer dígito: D_1 , es 1; y el índice del último dígito: D_L es L). El segundo entero es el nuevo valor del dígito. Si no hay una solución al problema, imprime -1, -1. Si N ya es un múltiplo de $B + 1$, imprime 0, 0.

Ejemplo de entrada 1 10 5 2 3 4 5 6	Ejemplo de salida 1 3 0
Ejemplo de entrada 2 10 3 1 0 2	Ejemplo de salida 2 -1 -1
Ejemplo de entrada 3 2 5 1 0 1 1 1	Ejemplo de salida 3 4 0
Ejemplo de entrada 4 17 5 3 0 0 0 0	Ejemplo de salida 4 1 0

Ejemplo de entrada 5 16 4 15 0 13 10	Ejemplo de salida 5 1 14
Ejemplo de entrada 6 16 5 1 15 0 13 10	Ejemplo de salida 6 0 0

Problema D

Dividiendo el reino

El reino de Nlogonia ha sido históricamente un lugar muy rico y tranquilo. Sin embargo, las circunstancias actuales podrían llevar esta era de paz y prosperidad a su fin: El rey es padre de dos mellizos, por lo que ambos son herederos al trono.

Los mellizos no se llevan bien y son celosos y excesivamente competitivos uno con otro. Debido a esto, hacer que ambos gobiernen el reino cooperativamente no es una opción viable. El reino tendrá que ser dividido en dos principados independientes, de modo que cada una de ellos sea dado a cada príncipe. Además, la división debe ser totalmente justa para evitar conflicto entre los envidiosos hermanos.

El reino consta de N ciudades y M caminos que conectan pares de ciudades. Los nlogonios son particularmente orgullosos de sus caminos. Cada camino tiene asociado un valor positivo que representa su belleza.

El reino será dividido de este modo: Primero, las ciudades serán particionadas en dos conjuntos tales que cada ciudad está en uno y sólo un conjunto. Entonces, cada principado constará de las ciudades en un conjunto y los caminos conectando ciudades de este mismo conjunto entre sí. Los caminos que conectan ciudades de distintos principados serán destruidos, ya que los príncipes no están interesados en comerciar uno con otro y mantener los caminos sólo incrementaría la probabilidad de guerra.

La belleza de un principado se define como la mayor belleza entre los caminos del principado, o 0 (cero) si el principado no tiene ningún camino. Por razones obvias, el rey quiere que la belleza de ambos principados sea la misma.

Ayuda al rey a determinar todos los posibles valores de la belleza de los principados resultantes, dado que la división es hecha de tal manera que los principados son igual de bellos.

Entrada

La primera línea contiene dos enteros N, M ($1 \leq N, M \leq 5 \times 10^5$), que representan la cantidad de ciudades y la cantidad de caminos, respectivamente.

Cada una de las siguientes M líneas contiene tres enteros x_i, y_i, b_i ($1 \leq x_i < y_i \leq N, 1 \leq b_i \leq 10^9$), que representan que hay un camino que conecta las ciudades x_i e y_i y tiene belleza b_i . No hay dos caminos conectando el mismo par de ciudades.

Salida

Si no es posible dividir el reino de modo que ambos principados tengan la misma belleza, imprime una línea con la cadena "IMPOSSIBLE". En caso contrario, imprime todos los posibles valores para la belleza de los principados resultantes de divisiones en principados con igual belleza. Los valores deben ser impresos en orden ascendente, cada uno en su propia línea.

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

Ejemplo de entrada 2 4 4 1 2 5 2 3 6 1 3 7 3 4 7	Ejemplo de salida 2 IMPOSSIBLE
Ejemplo de entrada 3 2 1 1 2 10	Ejemplo de salida 3 0

Problema E

Escalera mecánica

Acabas de inventar un nuevo tipo de escalera mecánica: la escalera mecánica doble. En una escalera mecánica normal, las personas son transportadas desde un extremo hasta el otro, siempre en esa misma dirección. En cambio, la escalera mecánica doble es capaz de transportar personas desde cualquiera de sus extremos hasta el otro.

La escalera mecánica doble se demora 10 segundos en transportar una persona desde cualquiera de los extremos hasta el otro. Es decir, si una persona ingresa a la escalera desde uno de los extremos en el instante T , entonces saldrá por el otro extremo en el instante $T + 10$ – esta persona ya no estará utilizando la escalera en el instante $T + 10$.

En cualquier instante de tiempo en el que nadie esté utilizando la escalera mecánica doble, esta se detiene inmediatamente. La escalera se encuentra inicialmente detenida.

Cuando la escalera se encuentra detenida y una persona ingresa desde uno de los extremos, la escalera se enciende automáticamente y se mueve en la dirección en la que esta persona desea desplazarse.

Si una persona llega a la escalera y esta ya se encuentra en movimiento en la dirección en la que la persona desea desplazarse, la persona ingresa inmediatamente a la escalera. De lo contrario, si la escalera se encuentra moviéndose en la dirección opuesta, la persona esperará hasta que la escalera se detenga, y sólo entonces ingresará. La escalera mecánica es tan grande que tiene lugar para que mucha gente ingrese en ella al mismo tiempo.

La escalera mecánica doble tiene un efecto muy extraño, probablemente relacionado con algún efecto de física cuántica (o quizás sea simplemente por azar): ninguna persona llegará nunca a la escalera en el momento exacto en el que esta se detiene.

Ahora que sabes cómo opera la escalera mecánica doble, tu tarea consiste en simular su funcionamiento. Dada la información acerca de N personas, incluyendo sus tiempos de llegada a la escalera y en qué dirección desean desplazarse, debes calcular en qué momento la escalera se detendrá por última vez.

Entrada

La primera línea contiene un entero N ($1 \leq N \leq 10^4$), que indica la cantidad de personas que utilizarán la escalera mecánica.

Cada una de las siguientes N líneas contiene dos enteros t_i y d_i ($1 \leq t_i \leq 10^5$, $0 \leq d_i \leq 1$), que representan respectivamente el instante de tiempo en el que la i -ésima persona llegará a la escalera, y en qué dirección desea desplazarse. Si d_i es 0, la persona desea desplazarse desde el extremo izquierdo hasta el derecho, y si d_i es 1, la persona desea desplazarse desde el extremo derecho hasta el izquierdo. Todos los valores t_i son distintos, y serán dados en orden ascendente.

Salida

Debes escribir una única línea, que contenga el instante de tiempo en el que la escalera se detendrá por última vez.

Ejemplo de entrada 1	Ejemplo de salida 1
3	23
5 0	
8 0	
13 0	

Ejemplo de entrada 2 3 5 0 7 1 9 0	Ejemplo de salida 2 29
---	----------------------------------

Ejemplo de entrada 3 3 5 0 10 1 16 0	Ejemplo de salida 3 35
---	----------------------------------

Problema F

Fugarse de la prisión

Michael y su hermano Lincoln han sido injustamente encarcelados en la misma prisión, pero Michael tiene un plan para rescatar a su hermano. La prisión se puede ver como un conjunto de polígonos convexos en el plano, en el cual las aristas de los polígonos son paredes. Las paredes de polígonos distintos no se intersectan, pero los polígonos pueden ser anidados, es decir, estar uno dentro de otro. Michael y Lincoln pueden ser vistos como dos puntos en el plano. Para Michael, el camino de rescate consistirá en alcanzar a su hermano y luego ambos necesitan escapar de la prisión.

Caminar para ellos no es un problema, pero trepar paredes es peligroso y difícil, por lo que Michael intentará minimizar la cantidad total de paredes trepadas por él. Michael primero necesita trepar algunas paredes para alcanzar a su hermano si no están en la misma área, y luego trepar más paredes para salir de la prisión. Salir de la prisión quiere decir no estar dentro de ningunas paredes, o también puede verse como alcanzar un punto muy lejano, digamos $(10^{20}, 10^{20})$. Brad está a cargo del posicionamiento de los prisioneros y es consciente del plan de Michael, por lo que ubicará a los prisioneros en dos puntos distintos en el plano, no contenidos por ningún segmento, de modo que la menor cantidad de paredes que deben ser trepadas por Michael sea lo mayor posible. ¿Cuál es la mínima cantidad de paredes que serán trepadas por Michael si Brad coloca a los hermanos de manera óptima?

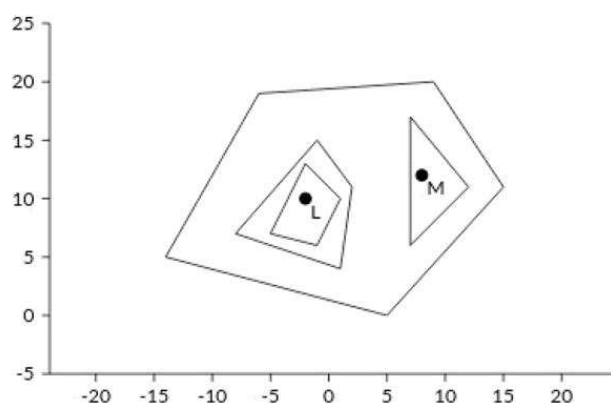


Figura 1: Ilustración de un posicionamiento válido para el ejemplo 1. El punto M representa a Michael y el punto L a Lincoln

Entrada

La primera línea de la entrada contiene un entero N ($1 \leq N \leq 2 \times 10^5$), la cantidad de polígonos convexos. Esta línea es seguida por las descripciones de cada polígono. La i -ésima descripción empieza con un entero k_i ($3 \leq k_i \leq 6 \times 10^5$) seguido de k_i líneas, cada una de las cuales contiene un punto (x_j, y_j) ($-10^9 \leq x_j, y_j \leq 10^9$).

Los puntos forman un polígono convexo y son dados en orden contrario a las agujas del reloj. No hay tres puntos consecutivos del polígono que sean colineales. No hay dos aristas de distintos polígonos que se intersequen. La cantidad total de aristas no excede 6×10^5 , es decir $\sum_{i=1}^N k_i \leq 6 \times 10^5$.

Salida

Imprime un entero, la menor cantidad de paredes que deben ser trepadas por Michael para rescatar a su hermano, en caso que Brad asignara a los hermanos posiciones de modo que tal cantidad de paredes sea máxima.

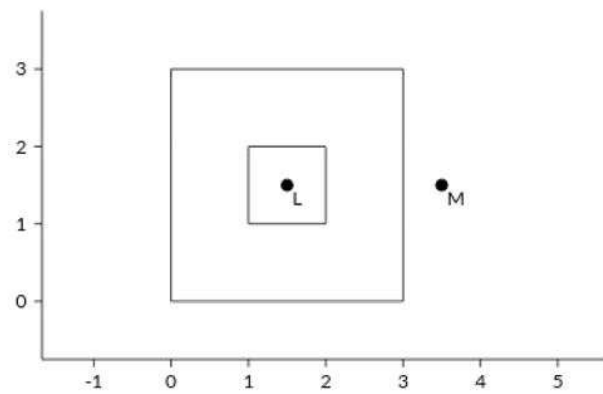


Figura 2: Ilustración de un posicionamiento válido para el ejemplo 2

Ejemplo de entrada 1	Ejemplo de salida 1
<pre> 4 4 1 10 -2 13 -5 7 -1 6 5 15 11 9 20 -6 19 -14 5 5 0 4 -1 15 -8 7 1 4 2 11 3 7 17 7 6 12 11 </pre>	<pre> 6 </pre>

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

Problema G

Garantizando el entrenamiento

Juan decidió comenzar a hacer ejercicio y se está preparando para su rutina de entrenamiento.

Sabe que algunos días podría no querer hacer todos los ejercicios de su rutina. Así que ha establecido algunas reglas para evitar saltarse todos los ejercicios de su rutina y no entrenar en absoluto, pero permitiéndose omitir opcionalmente algunos de los ejercicios.

Las reglas son:

- Solo habrá dos tipos de ejercicios: A , y B .
- Después de finalizar un ejercicio de tipo B , Juan pasará al siguiente ejercicio en caso de que haya alguno. De otro modo, la rutina habrá terminado.
- Después de finalizar un ejercicio de tipo A , hay dos posibilidades: Juan puede pasar al siguiente ejercicio, o, puede saltarse el siguiente ejercicio y pasar al siguiente después de ese.
- El último ejercicio de la rutina siempre debe ser del tipo B .

Entonces, puede haber diferentes maneras en las que Juan puede completar su rutina. Por ejemplo, si los tipos de los ejercicios en una rutina son: $BAAB$, hay 3 maneras en las que la rutina puede ser completada: Hacer todos los ejercicios, saltarse el tercer ejercicio, o saltarse el último ejercicio.

Juan quiere preparar su rutina de entrenamiento de tal manera que haya exactamente N diferentes formas en las que su rutina pueda ser completada. ¿Puedes ayudarlo?

Entrada

Un entero positivo N ($2 \leq N \leq 10^{15}$), que indica la cantidad de formas diferentes en las que la rutina de entrenamiento debe ser completada.

Salida

Imprima una línea que contenga una cadena, formada únicamente por los caracteres ‘A’, y ‘B’, que indiquen los tipos de ejercicios en la rutina de entrenamiento. En caso de que haya más de una respuesta válida, imprima la menor lexicográficamente. En caso de que no haya una rutina de entrenamiento válida, imprima una línea que contenga la cadena “IMPOSSIBLE” (sin las comillas).

Ejemplo de entrada 1 2	Ejemplo de salida 1 AB
Ejemplo de entrada 2 4	Ejemplo de salida 2 ABAB
Ejemplo de entrada 3 7	Ejemplo de salida 3 IMPOSSIBLE

Problema H

Habrá ordenamiento

Un amigo tuyo ha inventado un juego y quiere saber si eres capaz de resolverlo.

Tu amigo ha construido una secuencia de N bloques. Cada bloque tiene un número escrito, y es de un color. Todos los números son distintos, entre 1 y N , pero diferentes bloques pueden tener el mismo color.

En el juego se pueden llevar a cabo tantos turnos como quieras. En cada turno, debes elegir dos bloques diferentes que tengan el mismo color e intercambiarlos.

Debes decir si es posible hacer que toda la secuencia esté en orden ascendente de acuerdo a los números escritos en los bloques.

Entrada

La primera línea contiene dos enteros N y K ($1 \leq N \leq 10^5$, $1 \leq K \leq N$), que representan respectivamente, el número de bloques en la secuencia y el número de colores diferentes.

Cada una de las siguientes N líneas contiene dos números enteros n_i , y c_i ($1 \leq n_i \leq N$, $1 \leq c_i \leq K$), que representan respectivamente, el número que tiene escrito, y el color del i -ésimo bloque.

Salida

Debes imprimir una línea que contenga solo una letra. Si la secuencia puede ser acomodada en orden ascendente, imprime la letra mayúscula ‘Y’; En caso contrario, imprime la letra mayúscula ‘N’.

<p>Ejemplo de entrada 1</p> <pre>4 2 3 1 4 2 1 1 2 2</pre>	<p>Ejemplo de salida 1</p> <pre>Y</pre>
<p>Ejemplo de entrada 2</p> <pre>4 2 2 1 4 2 1 1 3 2</pre>	<p>Ejemplo de salida 2</p> <pre>N</pre>
<p>Ejemplo de entrada 3</p> <pre>3 1 1 1 2 1 3 1</pre>	<p>Ejemplo de salida 3</p> <pre>Y</pre>

Problema I

Invirtiendo todo

El gobierno de Nlogonia está molesto por la falta de eficiencia de su sistema ferroviario. Todos los pares de ciudades están conectados por una única vía de tren, pero debido a problemas de presupuesto, algunas de éstas están inactivas.

Una configuración ideal del tejido ferroviario es una tal que para todo par de ciudades existe exactamente un camino que conecta esas ciudades usando sólo vías activas.

El gobernador de Nlogonia te ha contratado para transformar su conjunto de vías en una configuración ideal. Desafortunadamente, no hablas nlogonio ni puedes activar o desactivar vías: Sólo los líderes de cada ciudad, todos los cuales solamente hablan nlogonio, pueden activarlas o desactivarlas.

El idioma nlogonio tiene algunas frases realmente específicas, así que esperas contar con ello. Tienes un amigo que habla nlogonio, y para ayudarte (y también desafiarte un poco), te ha enseñado una frase que puedes intentar usar: “`lupDujHomwIj luteb gharghmey`”. Te dijo que si dices eso al líder de una ciudad, entonces el líder activará todas las vías que están conectadas a su ciudad y que previamente estaban inactivas, y desactivará todas las vías que están conectadas a su ciudad y que previamente estaban activas. En otras palabras, los estados de “activación” de todas las vías que conectan a la ciudad de ese líder serán dados vuelta.

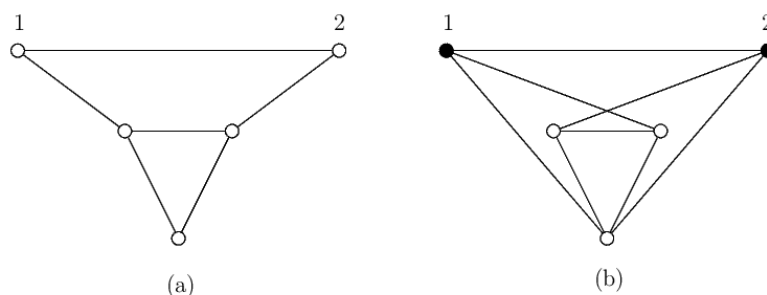


Figura 3: Resultado de “`lupDujHomwIj luteb gharghmey`” las ciudades 1 y 2

Conociendo esta frase, puedes llamar a algunos líderes y hacerles “`lupDujHomwIj luteb gharghmey`” las vías de sus ciudades.

Tu amigo duda que puedas hacer ideal el sistema ferroviario solamente usando esta frase. Realmente quieres probar que está equivocado. Tú le has dicho que no sólo encontrarás una solución, sino que también le dirás de cuántas formas diferentes lo puedes lograr. Más precisamente, decimos que un conjunto de líderes es *bueno* si contactando a ellos y sólo a ellos una vez se logra una configuración ideal. Quieres decirle a tu amigo cuántos conjuntos buenos de líderes hay.

Dos conjuntos de líderes son distintos si hay al menos un líder que está en uno de los conjuntos pero no en el otro.

Como este número puede ser muy grande, debes imprimir el resto de dividirlo por $10^9 + 7$.

Entrada

La primera línea contiene dos enteros, N y M ($1 \leq N \leq 100$, $0 \leq M \leq N \times (N - 1)/2$), que representan la cantidad de ciudades y la cantidad de vías que inicialmente están activas.

Cada una de las siguientes M líneas contiene dos enteros, u y v ($1 \leq u < v \leq N$), que representan la existencia de una vía inicialmente activa entre las ciudades u y v . Ningún par de ciudades será listado dos veces.

Salida

Imprime una línea conteniendo la cantidad de conjuntos *buenos* de líderes dada la configuración inicial del tejido ferroviario, módulo $10^9 + 7$.

Ejemplo de entrada 1 5 6 1 2 1 3 2 4 3 4 3 5 4 5	Ejemplo de salida 1 8
Ejemplo de entrada 2 3 2 1 2 2 3	Ejemplo de salida 2 6
Ejemplo de entrada 3 4 4 1 2 2 3 3 4 1 3	Ejemplo de salida 3 4
Ejemplo de entrada 4 3 1 1 2	Ejemplo de salida 4 0
Ejemplo de entrada 5 2 0	Ejemplo de salida 5 2

Ejemplo de entrada 6	Ejemplo de salida 6
10 15 1 6 1 2 1 5 6 7 6 10 2 3 2 9 7 3 7 8 3 4 8 9 8 5 4 5 4 10 9 10	0

Problema J

Jugando fútbol

El fútbol no siempre ha sido el deporte más popular en América. Los historiadores han encontrado registros de un deporte antiguo que era jugado en muchas civilizaciones a lo largo del continente. Debido a la falta de tradición oral acerca del mismo, su nombre original es desconocido, pero en tiempos modernos ha sido llamado muy creativamente “bútfol”.

No sabemos mucho acerca del bútfol, ni siquiera las reglas básicas. Sin embargo, los arqueólogos han encontrado un montón de notas hechas por entrenadores de bútfol al tratar de armar sus equipos, lo cual nos da algo de información acerca de cómo se formaban los mismos. Estas notas están repletas de números y cálculos. Los entrenadores de bútfol intentaban meticulosamente optimizar sus equipos, asignando jugadores a las mejores posiciones posibles. Para facilitar esta tarea, desarrollaron una métrica para determinar el rendimiento de cada asignación.

Hay M posiciones en un campo de bútfol, que están distribuidas en una línea. Un equipo de bútfol consta de N jugadores, cada uno de los cuales es asignado a alguna posición (todos los jugadores deben ser asignados a exactamente una posición, cada posición puede ser ocupada por uno o más jugadores o puede ser dejada sin ocupar).

Naturalmente, los jugadores no son iguales entre sí: Cada jugador puede tener diferente rendimiento cuando juega en diferentes posiciones en el campo. Concretamente, para cada jugador i y cada posición j , hay un valor positivo $P_{i,j}$ que representa el rendimiento del jugador i cuando juega en la posición j .

Para complicar las cosas aún más, los entrenadores también consideran el aspecto de la interacción entre jugadores. Hay algunos pares de jugadores que son “mejores amigos”. Cuando los mejores amigos están lejos uno de otro en el campo, eso tiene un impacto negativo en el rendimiento del equipo. Hay un valor positivo C que representa la penalidad de rendimiento que se paga por mover a los mejores amigos lejos uno de otro.

Una vez que los jugadores son asignados a sus posiciones, el valor del rendimiento del equipo se calcula de esta forma: Primero, sumamos los rendimientos de los jugadores cuando juegan en su posición asignada. Luego, por cada par de jugadores que son mejores amigos, sustraemos C por la distancia entre los dos jugadores, donde la distancia entre dos jugadores se define como la diferencia (en valor absoluto) entre las posiciones a las que los jugadores fueron asignados.

Queremos saber qué tan buenos eran los entrenadores de bútfol a la hora de armar equipos. Para hacer eso, nos gustaría saber cuál es el máximo valor posible para el rendimiento de un equipo que se puede conseguir asignando a los jugadores en las posiciones óptimas, dados los rendimientos de los jugadores en cada posición y los pares de jugadores que son mejores amigos.

Entrada

La primera línea contiene cuatro enteros N , M , K y C ($1 \leq N, M \leq 50$, $0 \leq K \leq 50$, $0 \leq C \leq 10^6$), que representan la cantidad de jugadores, la cantidad de posiciones, la cantidad de pares de mejores amigos y la penalidad por tener a los mejores amigos lejos uno de otro.

Cada una de las siguientes N líneas contiene M enteros. El j -ésimo entero de la i -ésima línea es $P_{i,j}$, que representa el rendimiento del jugador i si juega en la posición j ($0 \leq P_{i,j} \leq 10^6$).

Cada una de las siguientes K líneas contiene 2 enteros a_i y b_i ($1 \leq a_i < b_i \leq N$), que representan que los jugadores a_i y b_i son mejores amigos. No se repiten pares de jugadores en esta lista.

Salida

Imprime una línea conteniendo un entero, representando el máximo rendimiento posible para el equipo.

Ejemplo de entrada 1	Ejemplo de salida 1
3 3 2 5 5 2 1 3 2 8 1 9 3 1 2 1 3	14

(En este caso, la solución óptima es asignar los jugadores 1 y 3 a la posición 2, y el jugador 2 a la posición 3, entonces la suma de los rendimientos de los jugadores es $2+8+9=19$, y pagamos una penalidad de 5 por los jugadores 1 y 2 estando a distancia 1, y penalidad 0 por los jugadores 1 y 3 estando en la misma posición).

Problema K

Katmandú

Por fin la pandemia está mejorando y podrás hacer esa cosa con la que has estado soñando tanto durante los últimos años: comer en tu restaurante favorito. Resulta que este restaurante está en Katmandú, pero no hay problema, siempre puedes tomar un avión.

El problema es que los aviones casi siempre te dejan cansado. Consideras que tienes un descanso adecuado si puedes dormir por T minutos sin interrupción, esto es, no estás despierto en ningún momento entre t y $t + T$. Además, eres muy bueno durmiendo: te puedes quedar dormido al inicio de cualquier minuto y despertar al finalizar cualquier minuto.

Pero si duermes demasiado ¡Podrías perder las comidas que sirven en el vuelo! Eso sería inaceptable, ninguna oportunidad de tener comida gratis debe ser desperdiciada.

Afortunadamente, la aerolínea te ha enviado por anticipado el cronograma completo del viaje: la duración D del vuelo en minutos, el número M de comidas que se van a servir, y el momento y_i en el que cada una de estas comidas se va a servir. Para poder comer una de ellas, debes estar despierto en el momento en el que se sirve, de otro modo la aeromoza no te la servirá. Como siempre estás hambriento, devorarás la comida instantáneamente al servirse.

Ahora te preguntas, para tener la mejor experiencia en el avión, ¿Puedes tener un descanso adecuado, y además comer todas las comidas del vuelo?

Entrada

La primera línea de entrada contiene tres enteros, T , D , M ($1 \leq T, D \leq 10^5$, $0 \leq M \leq 1000$), que representan, respectivamente, el número de minutos que debes dormir sin interrupción para tener un descanso adecuado, la duración del vuelo, y el número de comidas que se van a servir durante el vuelo.

Cada una de las siguientes M líneas contiene un entero y_i ($0 \leq y_i \leq D$). Estos enteros representan los momentos en los que cada comida se va a servir, y están dados en orden cronológico.

Salida

Debes escribir una línea que contenga solo una letra. Si puedes tener un descanso adecuado y comer todas las comidas durante el vuelo, escribe la letra mayúscula ‘Y’; en caso contrario, escribe la letra mayúscula ‘N’.

Ejemplo de entrada 1 3 10 3 2 4 7	Ejemplo de salida 1 Y
Ejemplo de entrada 2 4 10 3 2 4 7	Ejemplo de salida 2 N
Ejemplo de entrada 3 5 5 0	Ejemplo de salida 3 Y

Ejemplo de entrada 4 4 8 2 5 7	Ejemplo de salida 4 Y
--	---------------------------------

Ejemplo de entrada 5 4 8 2 3 4	Ejemplo de salida 5 Y
--	---------------------------------

Problema L

Listando contraseñas

Michael es el jefe de una oficina apenas conocida y dentro de su habitación hay una caja de seguridad con el dinero para pagarle a los empleados. Desafortunadamente, Michael olvidó la contraseña de la caja de seguridad y ahora es responsabilidad de Dwight ayudar a su jefe.

La contraseña es una secuencia de N dígitos, que contiene sólo ceros y unos, y Michael recuerda el valor en algunas posiciones de la secuencia, pero no la contraseña entera. Michael también recuerda M intervalos de la contraseña que son palíndromos – su mente memoriza palíndromos, por algún motivo.

Un intervalo es un palíndromo si, y sólo si, el primer y último dígitos son iguales, el segundo y el penúltimo son iguales, y así sucesivamente.

Ahora Dwight quiere saber qué tan difícil será recuperar la contraseña completa. Debes ayudar a Dwight, calculando el número de contraseñas posibles que son acordes a lo que Michael recuerda.

Como la respuesta puede ser muy grande, se debe imprimir el resto de dividirla por $10^9 + 7$.

Entrada

La primera línea contiene los dos enteros N y M ($1 \leq N \leq 3 \times 10^5$, $1 \leq M \leq 3 \times 10^5$), que representan cuántos dígitos tiene la contraseña y la cantidad de intervalos que Michael recuerda como palíndromos, respectivamente.

La segunda línea contiene N caracteres s_i , representando cuál dígito Michael recuerda para cada posición de la contraseña. Si s_i es '0' ó '1', entonces significa que el i -ésimo dígito de la contraseña es 0 ó 1 respectivamente. Si s_i es '?', entonces Michael no recuerda el i -ésimo dígito.

Cada una de las siguientes M líneas contiene dos enteros l_i y r_i ($1 \leq l_i \leq r_i \leq N$), lo cual significa que el intervalo de la contraseña desde el dígito en la posición l_i al dígito en la posición r_i , inclusive, es un palíndromo.

Salida

Imprime el resto de dividir el número de contraseñas posibles por $10^9 + 7$. Como algunos de los recuerdos de Michael pueden ser contradictorios, en caso de que no haya contraseñas que cumplan con todo lo que recuerda, imprime '0'.

<p>Ejemplo de entrada 1</p> <pre>5 2 1??0? 1 3 2 4</pre>	<p>Ejemplo de salida 1</p> <pre>2</pre>
<p>Ejemplo de entrada 2</p> <pre>3 2 ??? 1 1 1 3</pre>	<p>Ejemplo de salida 2</p> <pre>4</pre>
<p>Ejemplo de entrada 3</p> <pre>5 2 1???0 1 3 3 5</pre>	<p>Ejemplo de salida 3</p> <pre>0</pre>

Problema M

Monarquía vertiginosa

La sucesión monárquica es un asunto muy delicado, ya que se deben tener en cuenta múltiples factores como el parentesco, el sexo, la legitimidad y la religión. Habitualmente la corona es heredada por el hijo de un soberano, o por la línea colateral más cercana de un soberano sin hijos. No es algo sencillo ¿cierto? Y esa es una de las razones, por las cuales en todo el mundo la monarquía está al límite.

Nlogonia aún es gobernada por una monarquía, aunque por suerte posee reglas de sucesión simples. En general hay solamente dos aspectos a tener en cuenta: “Los hijos antes que los hermanos” y “los viejos antes que los jóvenes”.

Los sirvientes del reino mantienen un hermoso y enorme tapiz en donde se encuentra dibujado en forma de árbol todo el linaje de Constante, el primer gobernante de Nlogonia. Cada vez que nace un nuevo miembro de la familia, se dibuja una nueva rama desde el padre hasta el hijo en el tapiz. Este es un evento tan importante que, de acuerdo a la leyenda, cuando un descendiente de Constante tiene un hijo, nunca morirá antes de ver el nombre de su hijo agregado al tapiz.

Cuando alguien muere, se marca su nombre con una cruz en el tapiz. Cada vez que el monarca actual muere, el tapiz es utilizado por los sirvientes para determinar quién debe ser el próximo gobernante. Para determinarlo los sirvientes comienzan desde Constante y recorren el árbol de acuerdo a las reglas antes explicadas, “Los hijos antes que los hermanos” y “los viejos antes que los jóvenes”. Descienden por el árbol comenzando con Constante, luego el primer hijo de Constante, luego el primer hijo de ese hijo y así siguiendo, hasta llegar a la primera persona viva o a un miembro de la familia sin más hijos, en cuyo caso vuelven al padre de esa persona y pasan a su siguiente hijo, continuando con este proceso hasta determinar el nuevo monarca.

Luego de miles de años en el poder, el linaje de Constante es enorme. Mantener el tapiz y, cuando llega la hora, determinar quién debe ser el siguiente monarca son procesos largos y los sirvientes Nlogonios decidieron que ya es hora de modernizarse. Quieren crear un programa que se utilice para mantener el linaje de Constante, y que pueda además indicar quién debe ser el siguiente monarca tras la trágica muerte de un gobernante. Dada la importancia de esta tarea, los sirvientes de la monarquía desean probar el programa verificando que produzca la salida correcta para todos los eventos que ya ocurrieron. El único problema es que ninguno de ellos es bueno programando, y es por eso que han venido en busca de tu ayuda.

Formalmente, cada persona en el linaje de Constante se representa mediante un único identificador, que es un entero positivo. Cada vez que nace una nueva persona, su identificador es el siguiente número más chico disponible. El identificador de Constante es 1, e inicialmente él es la única persona viva. Recibirás muchos eventos para procesar, en orden cronológico. Cada vez que alguien muera, deberás ayudar a los sirvientes a descubrir quién es el monarca entonces. Se garantiza que siempre habrá alguien vivo para gobernar.

Entrada

La primera línea contiene un entero Q ($1 \leq Q \leq 10^5$), que indica la cantidad de eventos que se deben procesar.

Las siguientes Q líneas contienen dos enteros t_i y x_i cada una, que representan el tipo y el argumento del i -ésimo evento.

Si t_i es 1, el evento significa que la persona con identificador x_i tuvo un nuevo hijo. Si t_i es 2, el evento significa que la persona con identificador x_i murió.

Salida

Por cada evento en el que alguien muera, se debe escribir una línea con un entero, que indica el identificador del monarca luego de la muerte.

Ejemplo de entrada 1	Ejemplo de salida 1
8 1 1 1 1 1 2 2 1 2 4 1 2 2 2 2 5	2 2 5 3

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

Problema N

No tienen suerte

Vini es un pintor de carros muy dedicado. Desde que aprendió a pintar carros su sueño ha sido participar en la Competencia Internacional de Pintar Carros (ICPC por sus siglas en inglés).

Cada año la región de Vini tiene una competencia local para clasificar a los equipos de pintores de carros de esta región. Los pintores en los equipos que fueron clasificados en los mejores x lugares irán a competir al ICPC. Es una competencia muy emocionante, con muchos nuevos competidores cada año, hasta que los humos nocivos de la pintura de carros eventualmente los lleva a retirarse permanentemente.

Debido a la variación en el presupuesto nacional de la pintura de carros y las restricciones del ICPC, el número x puede variar entre años, lo que puede causarle disgusto a algunos competidores.

En el último año de Vini como competidor, su equipo estuvo a solo un lugar de clasificar al ICPC. ¡Qué mala suerte! Para hacer que su “mala suerte” se sienta aún peor, ¡en el siguiente año el equipo que obtuvo la misma posición clasificó! A pesar de este sentimiento, después de hablar con otros ex competidores, se dio cuenta de que muchos habían tenido el mismo sentimiento de tener mala suerte de una u otra forma.

Los ex competidores normalmente siguen los resultados de la competencia regional por algunos años después de retirarse. Entonces, un competidor podría no sentirse con mala suerte por los cambios que pudieran ocurrir en x algunos años después de retirarse. Siendo más precisos, cada ex competidor tuvo su última participación en el año a_i , obteniendo la posición p_i , y, después de retirarse, siguió los resultados por los siguientes f_i años.

Un competidor que no clasificó al ICPC en su último año de participación se ha sentido con mala suerte cada año de los que siguió los resultados en los que hubiera clasificado al ICPC si hubiera competido. En otras palabras, para cada año hasta f_i años después de que se retiró, si no clasificó en su última participación, se sintió con mala suerte si el número de equipos que clasificaron al ICPC en ese año fue al menos p_i .

Dados el número de clasificados por año y la información sobre cada ex competidor, queremos saber para cada ex competidor, ¿Cuántos años se sintió con mala suerte?.

Entrada

La primera línea de entrada contiene dos enteros Y y N ($1 \leq Y, N \leq 3 \times 10^5$), que representan respectivamente, el número de años de competencias y el número de ex competidores con los que Vini ha hablado. (Sí, ¡pintar carros es una tradición milenaria, y muy popular!).

La siguiente línea contiene Y enteros x_1, x_2, \dots, x_Y ($0 \leq x_i \leq 10^5$), que indican cuántos equipos clasificados al ICPC hubo en la región en cada año.

Cada una de las siguientes N líneas contiene tres números enteros a_i, p_i , y f_i ($1 \leq a_i \leq Y$, $1 \leq p_i \leq 10^5$, $0 \leq f_i \leq Y - a_i$), que representan respectivamente, el año en el que el i -ésimo ex competidor tuvo su última participación, la posición que obtuvo el i -ésimo competidor en su último año de participación, y por cuántos años el i -ésimo ex competidor siguió los resultados después de retirarse.

Salida

Imprima N líneas, la i -ésima línea debe contener cuántos años el i -ésimo ex competidor se sintió con mala suerte.

Ejemplo de entrada 1 5 3 1 2 3 4 5 1 3 4 2 6 3 3 4 1	Ejemplo de salida 1 3 0 1
Ejemplo de entrada 2 4 1 8 8 8 8 1 7 3	Ejemplo de salida 2 0