

Problema A

Carerra en Pekín Express

Pekín Express es un reality show de Televisión, donde varios equipos formados por dos personas cada uno son enviados a su suerte a un país extranjero. En cada programa, todas las parejas comienzan en el mismo punto, salen corriendo y se las apañan como pueden para llegar a otro punto del país, probablemente a cientos de kilómetros de distancia. El último en llegar puede ser eliminado del programa, y así se continúa hasta tener un ganador. El programa es famoso por coger a parejas divertidas, como AleG y Cherlo Jolms, Dora y HorseLuis, o Thanos y Shrek.

En este programa, la dirección ha decidido dar un giro sorpresa para darle más emoción al concurso. Cada pareja deberá introducir la mano en un bol donde hay distintos sobres, cada uno con un punto distinto de la ciudad marcado. Deberán ir a ese punto y recoger un objeto esencial para el concurso. Eso sí, algunas parejas han acumulado una penalización y saldrán en la carrera más tarde que las demás. (Eso le pasa a AleG por ir gritando “fraude” a la dirección del programa a altas horas de la noche)

Con las parejas ya sabiendo a donde tienen que ir, y habiendo iniciado su viaje tiempo atrás, ahora la dirección tiene que planificar una ruta para la presentadora, pues ella tiene que ir a cada uno de los puntos de finalización a grabar la llegada de cada pareja y entregarles el objeto esencial.

Sabiendo la descripción de la ciudad, el punto al que tiene que ir cada pareja y la penalización con la que han salido, ¿Podrías decirnos en qué orden va a llegar cada equipo a su punto de destino?

Por simplicidad, asumiremos que cada equipo ha cogido la mejor ruta posible hacia su destino, y que todos viajan a la misma velocidad, recorriendo 1 ud de distancia por ud de tiempo. Se garantiza que se podrá llegar a los puntos de destino de cada equipo desde el punto de origen.

Input

La entrada consistirá en un único caso, y comenzará con una línea con dos números separados por un espacio, N y C , N puntos de la ciudad y C carreteras que unen esos puntos. A continuación, aparecerán C líneas con tres números, los dos primeros indicando los dos puntos que une, y el tercero indicando la distancia D de esa carretera.

Le seguirá una línea con un único número, el punto de inicio de los equipos.

Tras ella aparecerá una línea con un número E de equipos, seguida de E líneas con 2 números cada una, en primer lugar, el punto de destino de ese equipo y, en segundo lugar, la penalización P con la que han salido.

La entrada debe ser leída de forma estándar.

Output

La salida será una única línea, con el orden de llegada de los equipos a sus puntos de destino. En la salida se indicará “->” entre dos equipos si uno llega antes que el otro, y “=” si ambos equipos llegan a la vez. En caso de que dos o más equipos lleguen a la vez, se imprimirá primero el que menor índice tenga.

La salida debe ser escrita de forma estándar

Entrada ejemplo	Salida ejemplo
6 6 0 1 10 2 0 15 3 1 5 2 3 5 3 5 20 3 4 4 0 4 1 9 5 10 4 0 3 50	0 = 2 -> 1 -> 3

Constraints

- $2 \leq N \leq 100000$
- $1 \leq C \leq 300000$
- $1 \leq D \leq 500$
- $1 \leq E \leq 50000$
- $0 \leq P \leq 500$

Problema B

Coleccionando Cartas

Manolo es coleccionista de cartas de cierto popular juego que no mencionaremos para que Manolo no se enfade con el autor de este problema.

A Manolo le faltan ciertas cartas de la colección por completar, y quiere completarla, como es normal, gastando el menor dinero posible. Cada mes, a finales, Manolo compra todas las cartas que le falten y pueda comprar con el presupuesto del que dispone, eso sí, comprando siempre las que más baratas pueda conseguir, porque cree que tarde o temprano las cartas acabarán bajando de precio. De cualquier forma, pasado un tiempo desde la salida de la colección, se cansará y dejará de buscar las cartas que le falten.

Al salir a la venta, cada carta común cuesta lo mismo que su número de colección, excepto unas cuantas cartas, consideradas raras, que cuestan el doble. Eso sí, algunas cartas se juegan más que otras, por lo que al principio de cada mes algunas cartas verán su precio modificado, al alza o a la baja.

Por suerte, Manolo dispone de una red de contactos para conseguir cartas, que le ofrecen cualquier carta que le falte a un precio actual o anterior que haya estado, incluido su precio inicial antes de las fluctuaciones de precio del primer mes ¡Menos mal!

En caso de que Manolo encuentre que dos cartas cuestan lo mismo, preferirá comprar la que tenga el número de colección más bajo. De esta forma, si tiene que enseñar su álbum de cartas a alguien, parecerá que tiene más cartas.

Dada la lista inicial de cartas que busca Manolo, una lista de su presupuesto por mes, y las cartas que verán su precio modificado cada mes, ¿Podrías decirnos cuanto dinero se va a gastar Manolo en completar la colección? Ten en cuenta que cada mes siempre comprará las cartas que le falten y más baratas le resulten. Si al final del mes le sobra dinero y no puede comprar ninguna carta más, se gastará ese dinero en otras cosas.

Nota: No habrá nunca dos cartas con el mismo número de colección.

Input

La entrada consistirá en un único caso de prueba, comenzando por M , el número de meses que pasarán antes de que Manolo se canse de completar la colección, seguido por M líneas con su presupuesto K_M para cada mes.

A continuación, aparecerá un número C , seguido de una línea con C números de colección separados por espacio, las cartas comunes que le faltan a Manolo. Tras ello aparecerá un número R , seguido de una línea con los números de colección de las R cartas raras que le falten por conseguir a Manolo.

Finalmente, aparecerá la descripción con las cartas más y menos jugadas, cada mes. Por cada mes, habrá una primera línea con un número J de cartas jugadas, seguida por J líneas, donde aparecerán dos números, primero el número de colección C de la carta, y luego su nuevo precio P .

La entrada debe ser leída de forma estándar.

Output

La salida serán únicamente dos líneas, el dinero gastado por Manolo para intentar completar la colección en la primera línea, y una segunda línea que indique “COMPLETADA” si Manolo consigue completar la colección, o que, en caso contrario, contenga los números de colección de las cartas que le falten por obtener, en orden ascendente, separadas por espacios.

La salida debe ser escrita de forma estándar

Entrada ejemplo	Salida ejemplo
3 100 250 200 4 30 20 17 38 3 100 60 22 1 100 150 0 2 100 500 20 5	419 COMPLETADA

Constraints

- $1 \leq M \leq 300$
- $0 \leq K_M \leq 50000$
- $1 \leq C, R \leq 3000$
- $0 \leq J \leq C + R$
- $1 \leq P \leq 10000$

Problema C

Comprendiendo el código

Los laboratorios de todo el mundo están intentando secuenciar el código genético del virus que se está propagando a una velocidad alarmante. Normalmente el código (*ADN*) se compone de dos cadenas principales largas unidas por pares formados por cuatro bases nitrogenadas: adenina (*A*), uracilo (*U*), guanina (*G*) y citosina (*C*). Pero en este caso se trata del (*CVA* (*código del virus alfabético*)), en él tanto las dos cadenas como las uniones entre estas están codificadas usando todas las letras mayúsculas del alfabeto inglés, un total de 26 bases.

La información que quieren leer los científicos está en los pares que unen las dos cadenas principales, necesitan leerlos todos para procesar el (*CVA*) con éxito. Pero el proceso que tienen que seguir no es sencillo. Solamente pueden empezar la lectura por uno de los extremos de las cadenas principales en el lado izquierdo e ir transitando a un elemento adyacente, leyendo todos los elementos encima de los que pasan. El proceso termina cuando hayan leído todos los pares que unen las cadenas principales, además de los trozos de las cadenas que han tenido que usar para transitar, y llegado a otro extremo abierto en el lado derecho. Solamente pueden empezar el proceso una vez así que deben tener cuidado para recorrer toda la cadena en un solo trazo. Ahora sí, esta técnica tiene dos problemas. Por un lado el tiempo de la lectura depende del tipo de elementos entre las que se transita. Se puede calcular como la *distancia* entre las letras que representan cada elemento de la doble hélice. Por ejemplo, la distancia entre la *A* y la *Z* es 25, entre la *C* y la *A* es 2 y entre la *R* y la *R* es 0. Por otro lado, la lectura es destructiva, es decir, al leer un elemento no se puede volver a él.

Input

La entrada comenzará con una línea indicando L , la longitud de la doble hélice.

A continuación vendrán 4 líneas de longitud L . La primera y la última estarán compuestas solamente por letras de la *A* a la *Z* y representan las cadenas principales; mientras que las dos líneas centrales incluirán espacios en blanco y representan los pares que unen dichas cadenas.

Nota 1 : Las líneas centrales tendrán espacios por el final hasta alcanzar la longitud indicada y seguirán el mismo patrón que la entrada de ejemplo.

Nota 2 : Se asegura que la longitud de la doble hélice siempre se podrá representar como $3 + 2n$ con $0 \leq n$.

La entrada debe ser leída de forma estándar.

Output

Se debe imprimir el tiempo mínimo necesario para procesar el (*CVA*) de la entrada.

La salida debe ser escrita de forma estándar

Entrada ejemplo	Salida ejemplo
5 ABCDD C A D D ZACAA	19

Entrada ejemplo	Salida ejemplo
7 CACABDB C C A D B B BBCBDDD	21

Constraints

- $3 \leq L \leq 200003$

Problema D

Comunidades en Cuarentena

Ante la situación actual, provocada por el COVID-19, el gobierno ha cerrado varias carreteras para reducir el número de puestos de control necesarios. Para ello han usado un algoritmo para generar un árbol de recubrimiento mínimo y han dejado solamente las calles imprescindibles. El problema es que aún así, parece que las medidas no son suficientes. El nuevo plan es dividir el territorio en varias comunidades aisladas cortando algunas de las carreteras que las unían después de la medida anterior.

Antes de poner el plan en marcha hay que barajar las opciones viables y tener en cuenta una cosa, las comunidades resultantes de esta división tienen que ser del mismo tamaño para facilitar las labores administrativas. Sería posible obtener una lista de todos los tamaños que permitan una división así del territorio? Así será más fácil elegir la opción adecuada.

Input

La entrada comenzará con una línea indicando N , el número de ciudades en el territorio.

A continuación vendrán $N - 1$ líneas con parejas de números indicando dos ciudades entre las que existe una calle que no ha sido cortada. Se asegura que de una ciudad se puede llegar a cualquier otra a través de la red de carreteras.

La entrada debe ser leída de forma estándar.

Output

Se deben imprimir todos los tamaños de comunidad en los que se puede dividir el territorio, separados por espacios en orden ascendente.

La salida debe ser escrita de forma estándar

Entrada ejemplo	Salida ejemplo
10 1 2 3 4 5 6 4 2 4 5 4 7 8 7 7 9 10 9	1 2 10

Constraints

- $1 \leq N \leq 100000$

Problema E

Cajas de juegos desordenados

Ya está. Es el fin. La madre de David se ha cansado de tantas tonterías. Pero empecemos por el principio.

David es gamer. De los buenos. Todos sus juegos los tiene en físico, con su caja y su disco. Y siempre los va a tener así. Como cuando se pone a jugar lo hace deprisa, siempre hace lo mismo. Coge el juego al que quiere jugar, lo mete en la consola, saca el juego anterior y lo deja en la caja que acaba de abrir. Así, no tiene que buscar la caja del juego que estaba en la consola. El problema viene cuando, más o menos una vez al mes, su madre le pide el *Danza Máxima 2020*, que claro, con el sistema de David, no está en su caja. Siempre pasa igual, y a David le toca reordenar los juegos.

Hoy es diferente. La madre de David ha ido a coger el *Danza Máxima 2020*, y como siempre, no lo ha encontrado. Así que ha llamado a David, y le ha dicho que o lo encuentra rápido, o le quita la consola. David, desesperado, ha sido listo y ha pensado en una estrategia. Como en la caja del *Danza Máxima 2020* está el *Coches de Carreras 4K*, abre la caja de este último juego, y dentro encuentra el *Cruce de Animales*. Para su suerte, dentro de la caja de *Cruce de Animales* estaba el juego que buscaba, así que se ha salvado. Con esta estrategia, recorreremos los juegos que ha jugado anteriormente David, encontrando así siempre, y seguro, el que buscamos.

David se pregunta ahora qué habría pasado de no encontrar el *Danza Máxima 2020*, o si hubiera agotado la paciencia de su madre.

Sabiendo que la madre de David puede pedir cualquier juego de la colección, y que se habrá cansado tras haber buscado en más de P cajas, dada la descripción de qué juego hay en qué caja, ¿Podemos asegurar que, siguiendo la estrategia de David, se salvará seguro? ¿O habrá algún juego que, al pedirlo, no sea capaz de encontrar antes de que se canse su madre?

Input

La entrada consistirá en numerosos casos de prueba, cada uno comenzando por una línea con dos números, J y P , el número J de juegos en la colección y la paciencia P de la madre de David, seguido de J líneas con dos números A y B , el juego A contenido en la caja del juego B . La entrada finalizará con un caso de prueba con 0 juegos y paciencia 0, que no deberá procesarse.

La entrada debe ser leída de forma estándar.

Output

Para cada caso de prueba se deberá imprimir una sola línea, indicando “SI” si, pida el juego que pida la madre de David, lo encontrará y se salvará, y “NO” si hay algún juego que, al ser pedido por su madre, no encontrará antes de que pierda la paciencia y, por tanto, le quitarán la consola.

La salida debe ser escrita de forma estándar

Entrada ejemplo	Salida ejemplo
5 4 0 1 1 2 2 3 3 0 4 4 3 1 0 0 2 1 1 2 0 0	SI NO

Constraints

- $1 \leq J \leq 300000$
- $0 \leq P \leq J$
- $0 \leq A, B < J$

Problema F

Cero Virus

Se ha descubierto un nuevo virus que esta aterrorizando a la población estudiantil, EL CERO VIRUS. Todavía no se conocen bien sus mecanismos de propagación, pero se cree que guarda relación con mantener relaciones de riesgo con los exámenes sin tomar precauciones, como haber estudiado antes.

Para intentar minimizar sus efectos devastadores, eres responsable de crear una aplicación que sepa detectar de forma eficiente qué personas es posible que estén infectadas y cuáles podemos considerar como sanas. Se considera que una persona puede estar infectada si ha tenido contacto con alguna otra persona infectada, o bien ha dado positivo en la prueba. La infección es transitiva, por lo que si A tiene contacto con B, y B con C, y se detecta que A esta infectado tanto B como C se consideran posibles infectados.

Input

En la primera línea, aparecerán dos números N y M, que representarán el número de personas bajo estudio, y el número de líneas que vienen a continuación.

Cada una de las siguientes M líneas tendrá uno de los siguientes formatos:

- POSITIVO 3 Cuando a una persona que se le ha realizado una prueba aleatoriamente ha dado positivo.
- CONTACTO 7 5 Cuando conocemos que las personas 7 y 5 han estado en contacto.
- ? 9 Debes responder "POSIBLE", si es posible que la persona 9 este infectada, o "NO", si no hay motivos para pensar que esta infectada.

La entrada debe ser leída de forma estándar.

Output

Para cada línea que empiece por "?", debes emitir una decisión: "POSIBLE", si es posible que la persona que va a continuación esté infectada, o "NO", si no hay motivos para pensar que está infectada. Para emitir una decisión, solo debes tener en cuenta la información conocida hasta este momento, por lo que es posible que el veredicto de una persona cambie durante la ejecución del programa.

La salida debe ser escrita de forma estándar

Entrada ejemplo	Salida ejemplo
6 8 CONTACTO 1 2 CONTACTO 3 4 CONTACTO 2 3 ? 5 POSITIVO 1 ? 4 CONTACTO 2 5 ? 5	NO POSIBLE POSIBLE

Constraints

- $0 \leq N \leq 10000$
- $1 \leq M \leq 100000$

Problema G

Capturando humanos

Un meteorito ha impactado contra la Tierra. Pero no ha sido por casualidad. Los alienígenas han enviado este meteorito a conciencia.

Los aliens han sido listos y han elaborado un plan para abducir a todos los humanos que puedan. Dentro del meteorito se hallan exactamente tantas naves como ciudades en la Tierra.

El plan es el siguiente: Cada nave tiene asignada una ciudad de la Tierra, a la que debe ir (Desde el meteorito) y abducir a tantas personas como pueda. Tras llegar a esa ciudad, deberá ir a la lanzadera que han instalado para devolver a cada nave al espacio exterior, capturando definitivamente a tantos humanos como cargue esa nave.

Sin embargo, la Gran Jefa Alien se ha dado cuenta de un detalle: Las naves se están saltando las reglas. Además de abducir personas en la ciudad asignada, están abduciendo a toda persona que pueden a su paso, efectivamente aumentando el número de personas que abducen para impresionar a la Jefa. Eso sí, los humanos no son tontos, bueno, no del todo al menos. Podemos asumir que en cada ciudad hay infinitas personas, pero que los aliens solo encontrarán a un número específico de ellas según la ciudad por la que estén pasando. Con lo que sí que no pueden mentir los aliens es con el tiempo. En ningún caso llegarán más tarde de lo previsto, así que no pueden desviarse de la ruta más rápida que les lleve a la ciudad asignada.

Como los números previstos se están cumpliendo antes de lo previsto, la Jefa Alien ha decidido largarse de la Tierra antes de lo anunciado, dejando tirados a sus compañeros. Sabiendo la cantidad de personas que una nave abducirá en cada ciudad a su paso, y el objetivo impuesto por la Jefa Alien en cuanto a personas a abducir, ¿Exactamente tras cuanto tiempo saldrá la Jefa Alien de vuelta a su planeta?

Nota: Podemos asumir que todas las naves saldrán del meteorito en el minuto 0, y que lo harán a la vez. En caso de que haya más de una ruta para llegar a la ciudad asignada o a la lanzadera, la nave irá por aquella en la que pueda abducir más humanos.

Input

La entrada consistirá en un único caso de prueba, comenzando en un número N de ciudades en la Tierra, seguido de C caminos que los aliens pueden tomar.

A continuación, acompañarán C líneas con 3 números, A , B , y T . A y B indicarán que existe un camino que va de A a B , y solo en esa dirección, que se tarda un tiempo T en recorrer.

Seguidamente aparecerán $N - 2$ líneas, cada una indicando cuantas personas K serán abducidas en cada ciudad. La primera línea indicará las personas de la ciudad 1, la segunda la de ciudad 2... Hasta llegar a $N - 2$.

Finalmente continuará una línea con un único número P , el objetivo de personas impuesto por

la Jefa Alien.

La ciudad 0 indicará siempre el meteorito, y la ciudad $N - 1$ indicará la localización de la lanzadera. En ninguno de estos dos puntos se puede abducir personas.

La entrada debe ser leída de forma estándar.

Output

Se deberá imprimir una sola línea, el minuto exacto en el que la Jefa Alien despegará de vuelta a su planeta, o "IMPOSIBLE", si cuando vuelva la última nave no se habrá cumplido el objetivo de personas a abducir.

La salida debe ser escrita de forma estándar

Entrada ejemplo	Salida ejemplo
5 17 4 2 5 1 2 7 2 0 4 0 2 1 2 3 10 3 4 8 1 4 3 3 0 8 0 3 10 0 4 5 4 1 6 3 1 10 4 0 4 0 1 6 2 4 5 2 1 6 3 2 6 13 24 11 40	18

Constraints

- $1 \leq N \leq 100000$
- $1 \leq C \leq 1000000$

- $1 \leq T \leq 1000$
- $1 \leq P \leq 1000000$
- $1 \leq K \leq 1000000$

Problema H

Compras extremas

En este periodo de confinación ir a los supermercados se ha convertido en todo un reto. Se deben hacer colas, entrar 1 por uno, llevar guantes, gel antiséptico, etc. Esto ha provocado además que las compras que se realizan tienen un precio mucho superior y se compran para una cantidad de días superior para respetar las restricciones que ha impuesto este estado de alarma. El director de cada empresa del sector alimentario quiere premiar al que realice la compra más divertida, por esta misma razón nos ha pedido un programa para que detecte a la compra más divertida.

Input

La primera línea contendrá un número T , denotando el número de clientes que han realizado la compra. Por cada cliente vendrá un número N este número será la cantidad de alimentos comprados y una secuencia de caracteres en el alfabeto inglés P siendo este el nombre de la persona que realiza la compra. Seguidamente vendrán N líneas, donde en cada línea tendremos una secuencia de caracteres en el alfabeto inglés S denotando el producto y el valor V de lo divertido que es el producto el cual el responsable del supermercado ha establecido. Se asegura que todos los nombres de clientes y productos no contendrán espacios, serán una palabra.

La entrada debe ser leída de forma estándar.

Output

En la salida tenemos que imprimir la compra más divertida de la siguiente manera. El nombre de la persona que realizó la compra más divertida según el responsable de la empresa del sector alimentario junto con su valor. Se asegura que no existirán empates.

La salida debe ser escrita de forma estándar

Entrada ejemplo	Salida ejemplo
3 2 ISAAC PAN 1 AGUA 2 2 GUIZA CERVEZA 20 CERVEZA 20 1 ALICIA LIMONADA 10	GUIZA 40

Constraints

- $1 \leq T \leq 100$
- $1 \leq TamanodeP \leq 10$
- $1 \leq N \leq 1000$
- $1 \leq TamanodeS \leq 10$
- $-100 \leq V \leq 100$