GPC-UPC Saint Valentine Contest

A. San Glovo

1 second, 256 megabytes

Radewoosh se olvidó de que hoy era Día de San Valentín. Dado que ya habia hecho planes con su familia, no podrá encontrarse con su pareja hoy. Desesperado, buscó todas las alternativas posibles para que su pareja no se moleste y encontró la respuesta a toda la existencia humana: "San Glovo".

Entonces, planeó enviarle un regalo a su pareja usando Glovo. Como Radewoosh debe pagar con tarjeta, el costo del regalo no debe exceder al saldo de su tarjeta de débito.

Como es lo usual, en Glovo se puede ver el precio de envio $envio$, el costo del regalo $costo$ y posible descuento $disc$. El precio total de la transacción será:

$$\text{Precio Total} = envio + \frac{100 - disc}{100} \cdot costo$$

Radewoosh separó las posibilidades en n regalos, de los cuales se conoce su costo de envio, su costo, su descuento y cuánta felicidad le dará a su pareja al serle entregado.

Ayúdalo a determinar la máxima felicidad posible que pueda darle a su pareja.

Input
La primera línea de entrada contiene un entero n y un decimal s $(1 \leq n \leq 10^5, 0 \leq s \leq 10^{14})$ — La cantidad de regalos posibles y el saldo en la tarjeta.

Las siguientes n líneas contienen dos reales e_i, c_i y dos enteros d_i, h_i $(0 \leq e_i, c_i \leq 10^{14}, 0 \leq d_i \leq 99, 0 \leq h \leq 10^{18})$ — El costo de envío, el costo del regalo, el descuento y la felicidad respectivamente del i-ésimo regalo.

Todos los reales serán dados con exactamente 2 decimales de precisión.

Output
Imprime una sola línea — La respuesta al problema. Si no hay ningún regalo posible, imprime 0.

input
2 100.00 99.00 100.00 99 11 99.00 100.00 98 12
output
11

B. Fuego del Corazón

1 second, 256 megabytes

Hoy es San Valentín, así que los miembros del partido "Candidato a Maestro" se han dispersado por la ciudad, que se ve como una matriz de $n \times m$. Como acostumbra su líder, ellos tienen tanta pasión que el fuego de su corazón se vuelve realidad y comienza a incendiar la celda en la que están en el minuto 0, al siguiente minuto el fuego se propaga a las adyacentes (aquellas que compartan un lado) hasta que ya no se puedan extender más.

Tu trabajo es determinar la mínima cantidad de minutos que deben pasar antes de que toda la ciudad esté incendiada.

Input
La primera línea de entrada contiene dos enteros n y m $(1 \leq n \cdot m \leq 10^5)$ — La cantidad de filas y columnas que tiene la matriz de la ciudad.

La segunda línea de entrada contiene un entero x $(1 \leq x \leq n \cdot m)$ — La cantidad de miembros del partido que están en la ciudad.

Las siguientes x líneas contienen dos enteros r_i, c_i cada una $(1 \leq r_i \leq n, 1 \leq c_i \leq m)$ — La posición del i-ésimo miembro del partido.

Se garantiza que todos los miembros están en diferentes celdas.

Output
Imprima una sola línea — La respuesta al problema.

input
3 2 2 1 1 3 2
output
1

C. Botella amorosa

1 second, 256 megabytes

Todos conocemos el juego de la botella borracha. Las personas se ubican en un círculo, colocan una botella y la giran para saber quién le debe dar un beso a quién.

Sin embargo, *Botella amorosa* es completamente diferente: Se decide primero un número k y se van eliminando a las personas que quedan en el círculo. Primero se elimina a la persona k, luego a la que está a k personas y así hasta que solo quede una persona. Esa persona que salga es la que recibirá un beso de la persona que fue elegida antes de iniciar la ronda (y por ende, no participa del proceso). **Como este es un juego rápido, se considera $k = 2$.**

Esta vez, es San Valentín y tus amigos se han reunido; en el primer turno se decidió que tú serías el que besaría a la persona que saliera elegida. Dado que conoces cuántos amigos están en el círculo y el número que tiene tu amor platónico, calcula si vas a poder besarlo o no.

Input
La primera línea contiene un entero t $(1 \leq t \leq 10^5)$ — La cantidad de casos de prueba.

Las siguientes t líneas contienen dos enteros n y X $(1 \leq n \leq 10^9, 1 \leq X \leq n)$ — La cantidad de amigos en el círculo y el número de tu amor platónico.

Output
Imprima t líneas — La i-ésima línea debe contener la respuesta del i-ésimo caso de prueba. Da como respuesta **Buen dia** si podrás besar a tu crush o **He tenido peores** si no.

input
2 2 1 2 2
output
Buen dia He tenido peores

D. Reunion de Amigos

2 seconds, 256 megabytes

Varios amigos han hecho una reunión con n personas por San Valentín. Ellos quieren dividirse para hacer un juego entre 2 equipos.

Todos son muy desconfiados, así que un equipo es válido si todos sus miembros son amigos entre sí. Es decir, si el equipo es S, entonces para todo $x, y \in S$ con $x \neq y$ se cumple que x y y son amigos.

La distribución no debe ser justa, solamente basta que ambos equipos sean válidos y que haya al menos una persona en cada uno. Ayuda a deducir si existe o no alguna configuración e imprime alguna si es que hay.

Input
La primera línea de entrada contiene un entero n $(1 \leq n \leq 5000)$ — La cantidad de personas en la reunión.

Las siguientes n líneas contienen n enteros M_{ij} $(M_{ij} \in \{0, 1\})$ — El j-ésimo entero de la i-ésima fila es 1 si las personas i y j son amigos y 0 si no lo son.

Se garantiza que $M_{ii} = 0$ para todos los $i = 1, \ldots, n$ y que además $M_{ij} = M_{ji}$.

Output
Imprime en una línea **YES** o **SI** si hay una configuración válida y **NO** si no existe ninguna.

Si existe una configuración, imprime en las siguientes dos líneas las personas que pertenecen a cada grupo.

input
2 0 1 1 0

output
YES 2 1

E. Rosatel al rescate

1 second, 256 megabytes

Karboncito es un repartidor de Rappi. Hoy, como es Día de San Valentín, debe ir a su trabajo, cumplir con algunos pedidos y regresar a su casa.

La ciudad puede ser modelada como un árbol con n nodos (casas), de los cuales el nodo 1 es la casa de Karboncito. Karboncito solo tiene combustible suficiente para atravesar cada arista del árbol dos veces.

Cada pedido tiene un tiempo de confirmación luego de haber sido recibido. Es decir, si Karboncito entrega un pedido a la casa i en el minuto t, el pedido es confirmado recién en el minuto $t + c_i$.

Finalmente, una vez que Karboncito regrese a su casa, debe confirmar que su jornada laboral ha terminado y para este proceso también hay un tiempo de confirmación.

Una vez confirmados todos los pedidos y el final de su jornada laboral, puede enviar los datos a la sede principal para que le adjunten el pago en su sueldo del mes.

Considerando que Karboncito puede elegir a donde ir de manera óptima, ¿Cuál es el menor minuto posible en el que pueda enviar los datos a la sede principal sin problemas? Nótese que Karboncito puede enviar los datos en el mismo minuto en que todos los datos están confirmados.

Input
La primera línea de entrada contiene un entero n $(1 \leq n \leq 5 \cdot 10^5)$ — La cantidad de casas en la ciudad.

La segunda línea de entrada contiene n enteros c_i $(1 \leq c_i \leq 10^9)$ — El primer entero es el tiempo de confirmación de la jornada laboral de Karboncito, mientras que el i-ésimo entero (para $i > 1$) es el tiempo de confirmación del pedido de la i-ésima casa.

Las siguientes $(n - 1)$ líneas contienen dos enteros a_i, b_i cada una $(1 \leq a_i, b_i \leq n, a_i \neq b_i)$ — La i-ésima línea es la i-ésima arista del árbol.

Output
Imprima una sola línea — La respuesta al problema.

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

output
11

Karboncito debe entregar los pedidos a las casas en el siguiente orden: 3, 2, 4, 5, 6 y luego regresar a su casa. De esta forma, las confirmaciones por casa serán en los minutos 11, 10, 10, 10, 8, 9; por lo tanto, podrá enviar los datos en el minuto 11.

Si Karboncito entregara los pedidos en el orden: 3, 4, 5, 6, 2 y luego regresara a su casa, las confirmaciones por casa serían en los minutos 11, 16, 10, 8, 6, 7; por lo que enviaría los datos recién en el minuto 16.