A - Riego Avanzado

El nono tiene una granja de NxM celdas, y a su vez cuenta con un regador de dimensiones KxK. El regador se coloca en la esquina superior izquierda y comienza a regar cada subgrilla de KxK celdas, por lo que ciertas celdas pueden ser regadas más de una vez. Se acerca el momento de sembrar y el nono tiene que decidir a qué celda asignar cada paquetes de semillas de forma de maximizar la ganancia. La ganancia está dada por la multiplicación de la cantidad de veces que pasa el regador por una celda y la cantidad de semillas sembradas en dicha celda.

Entrada

Una línea con cuatro enteros: N, M ($1 <= N, M <= 10^5, 1 <= N * M <= 10^5$) las dimensiones de la granja, K ($1 <= K <= min(N, M)$) las dimensiones del regador y P la cantidad de paquetes de semillas disponibles.

Una línea con P enteros C_i ($1 <= C_i <= 10^5$) la cantidad de semillas en el i-ésimo paquete de semillas.

Salida

Un entero con la máxima ganancia que se puede conseguir.

Ejemplos

Entrada	Salida
3 4 2 3 1 2 3	22

En el caso de prueba 1 la cantidad de veces que cada celda es regada es la siguiente:

1	2	2	1
2	4	4	2
1	2	2	1

Luego, la respuesta es $4 * 3 + 4 * 2 + 2 * 1 = 22$.

B - Cocinando ando

La chef de un restaurante que aspira a una estrella Michelin quiere mostrar una selección de sus platos estrella para los inspectores. Para esto, ha asignado un presupuesto máximo B, y quiere maximizar el prestigio acumulado de los platos que muestra a los inspectores.

Para medir el prestigio de sus platos, la chef mantiene una lista de recetas, junto con sus costos e ingredientes. Para cada receta, un plato derivado se obtiene de un plato base agregando un ingrediente. La receta menciona dos piezas adicionales de información: el costo de aplicar la receta, además del costo del plato base, y el prestigio que la receta agrega al prestigio del plato base. La chef mide el prestigio en sus propias unidades, llamadas "unidades de prestigio".

Por ejemplo, una lista de recetas para hacer empanadas se ve así:

- empanada_queso empanada_base queso 1 2
- empanada_jamon_queso empanada_queso jamon 5 5

Aquí, empanada_base es un plato elemental, un plato sin receta asociada, un plato tan simple que su costo es insignificante (fijado en 0) y su prestigio también es 0. La chef puede obtener el plato derivado empanada_queso agregando el ingrediente queso al plato base empanada_base, por un costo de 1 peso y una ganancia de 2 unidades de prestigio. Una empanada_jamon_queso se obtiene de una empanada_queso agregando jamon, por un costo adicional de 5, y un prestigio de 5 agregado al prestigio del plato base; esto significa que el costo total de empanada_jamon_queso es 6 y su prestigio total es 7.

Una selección de platos estrella podría incluir, por ejemplo, tanto una empanada_queso como una empanada_jamon_queso. Tal selección tendría un prestigio total acumulado de 9, y un costo total acumulado de 7.

Armada con la lista de recetas y un presupuesto B, la chef quiere proporcionar una selección de platos estrella a los inspectores de Michelin para que el prestigio total acumulado de los platos sea maximizado, manteniendo su costo total acumulado como máximo B.

Notas importantes

- Ningún plato puede aparecer dos veces en la selección de platos estrella.
- Cualquier plato que no aparezca como plato derivado en ninguna receta se considera un plato elemental, con costo 0 y prestigio 0.
- Un plato puede aparecer más de una vez como plato resultante en la lista de recetas; si hay más de una forma de obtener un plato, se elige siempre la que produce el menor costo total; si los costos totales son iguales, se debe elegir la que produce el mayor prestigio total.
- Las recetas son tales que ningún plato D puede obtenerse agregando uno o más ingredientes a D mismo.

Entrada

- La primera línea consiste del presupuesto B ($0 <= B <= 10000$), un entero.
- La segunda línea consiste del número N ($0 <= N <= 1000000$) de recetas, un entero.
- Cada una de las siguientes N líneas describe una receta, como los siguientes elementos separados por espacios simples: el nombre del plato derivado (string); el nombre del plato base (string); el ingrediente agregado (string); el precio agregado (entero); el prestigio agregado (entero).

Otros límites:

- puede haber como máximo 10 000 platos diferentes (elementales o derivados);
- los costos y prestigios en las recetas están entre 1 y 10000 (inclusive);
- las cadenas contienen como máximo 20 caracteres ASCII (letras, dígitos y '_' solamente).

Salida

La salida debe consistir de dos líneas, cada una con un solo entero. En la primera línea: el prestigio máximo acumulado dentro del presupuesto. En la segunda línea: el costo mínimo acumulado correspondiente al prestigio máximo acumulado, necesariamente menor o igual al presupuesto.

Ejemplos

Entrada	Salida
15 6 empanada_queso empanada_base queso 1 2 empanada_tomate empanada_base tomate 5 10 empanada_clasica empanada_queso tomate 5 5 empanada_clasica empanada_tomate queso 1 2 empanada_bondiola empanada_clasica bondiola 7 6 empanada_picante empanada_queso aji 3 1	25 15

C - Verano

Se acerca el verano y las altas temperaturas en Santa Fe. Para que los estudiantes puedan estudiar y refrescarse, el decanato de la UTN-FRSF (Universidad Tecnológica Nacional - Facultad Regional Santa Fe) ha decidido construir una nueva piscina en uno de los patios de la facultad.

La piscina debe tener una forma **rectangular**. El área total de la superficie, C, ya está predeterminada por el presupuesto y el espacio disponible.

La empresa constructora que ganó la licitación ha entregado una lista de posibles longitudes (en metros) que pueden usar para los lados de la piscina, debido a sus materiales y métodos de construcción. Han estudiado el caso y **garantizan** que se puede construir una piscina con el área exacta C utilizando dos longitudes de esa lista.

Un estudiante de Ingeniería en Sistemas que está ayudando en el proyecto ha hecho algunas observaciones interesantes: 1. Todas las longitudes de los lados disponibles en la lista de la constructora son **números primos**. 2. El área total C de la piscina es siempre un producto de dos números primos **distintos**. (Esto también significa que la piscina nunca será un cuadrado).

Tu tarea es ser más rápido que el estudiante y encontrar qué dos longitudes de la lista se deben usar para construir la piscina.

Entrada

La primera línea de entrada contiene un entero C ($1 \leq C \leq 10^{18}$), el área de la superficie de la piscina.

La segunda línea contiene un entero M ($1 \leq M \leq 2 \cdot 10^5$), la cantidad de posibles longitudes de lado ofrecidas por la constructora.

La tercera línea contiene una lista de M primos P_i ($1 < P_i < 10^9$), únicos y separados por espacios, que representan las longitudes de lado ofrecidas.

Todas las medidas están expresadas en las mismas unidades.

Salida

Imprime una única línea con las dos longitudes de lado de la lista ofrecida que pueden usarse para construir la piscina.

La primera longitud de lado debe ser más pequeña que la segunda.

Ejemplos

Entrada	Salida
15	3 5

Entrada	Salida
5 7 2 5 11 3	

D - Vino en Caja

Vino en Caja es un equipo de programación competitiva reconocido a nivel mundial (no realmente).
En estas competencias de programación, cada equipo recibe una cantidad N de problemas que debe buscar resolver en una cantidad T de minutos. Para esto, deben crear soluciones y enviarlas a un juez, que determinará si la solución tuvo o no éxito de forma instantánea.
Vino en Caja tiene una particularidad, confían más en el azar que en sus propios conocimientos. Debido a esto, nunca están seguros si los problemas que intentan resolver serán o no aceptados, sino que esto depende de una probabilidad P. Su estrategia es la siguiente, en cada minuto de la competencia el equipo intenta resolver un problema, y en caso de tener éxito avanza al siguiente, pero en caso de fallar, siguen intentándolo hasta resolverlo. Debido a este particular método, sus entrenadores se ponen muy nerviosos al no tener idea de cuantos problemas serán capaces de resolver. Por esto, ellos te pidieron que calcules cual es el valor esperado de problemas que el equipo resolverá en una competencia.
Nota: si la competencia dura T minutos, solo pueden realizarse envíos en los minutos 0 a T-1, ya que en el minuto T la competencia habría finalizado.

Entrada

Una línea con tres números N, P y T ($1 <= N, T <= 2000, 0 <= P <= 1$) representando la cantidad de problemas, la probabilidad de resolver un problema y el tiempo en minutos que dura la competencia respectivamente. Los números N y T son enteros, mientras que P es un número real con hasta dos dígitos después de la coma.

Salida

Una línea con un número real, el valor esperado de problemas resueltos por el equipo al terminar la competencia con 6 dígitos después de la coma y redondeando.

Ejemplos

Entrada	Salida
1 0.50 1	0.500000

Entrada	Salida
1 0.50 4	0.937500

Entrada	Salida
4 0.20 2	0.400000

E - Revolución Industrial

Estamos en el siglo XVIII en plena *Revolución Industrial* y trabajamos en la mayor empresa carbonífera del país. Sin embargo, tenemos dificultades para poder hacer los envíos de carbón a las ciudades a tiempo. El dueño de la empresa nos acaba de dar una última advertencia: o solucionamos el problema para fin de mes o podemos buscar un nuevo trabajo.

Para poder resolver el problema tenemos un mapa con la siguiente información:

- Las ciudades y sus demandas de carbón.
- Las minas de carbón y su producción mensual.
- Los trenes que conectan las ciudades y minas, y el tiempo de transporte.

¿Cuál es el tiempo mínimo necesario para abastecer a todas las ciudades?

Entrada

Una línea con cuatro enteros: C ($1 <= C <= 100$) con la cantidad de ciudades, P ($1 <= P <= 5$) con la cantidad de plantas de carbón de la empresa, T ($1 <= T <= 200$) la cantidad de trenes que existen y M ($1 <= M <= 4 * 10^4$) la cantidad de minutos que disponemos para hacer las entregas.

C líneas con un entero $Demanda_i$, la demanda de la i-ésima ciudad ($1 <= Demanda_i <= 10^3$).

P líneas con un tnero $Produccion_i$, la producción de la i-ésima planta ($1 <= Produccion_i <= 2 * 10^4$).

T líneas con cuatro enteros describiendo cada uno de los trenes: U y V ($1 <= U, V <= C + P$) los IDs de las ciudades que une el tren (puede usarse en ambas direcciones) y $Minutos$ ($1 <= Minutos <= 10^3$) el tiempo de duración del viaje. Los ID de las ciudades van de los enteros 1 a C y los de las plantas de $C + 1$ a $C + P$. Puede haber varios trenes disponibles para la misma ruta.

Salida

Una línea con un entero, el tiempo mínimo para abastecer a todas las ciudades o en caso de ser imposible el string DESPEDIDO.

Ejemplos

Entrada	Salida
3 1 3 2 1 1 2	2

Entrada	Salida
5 1 4 1 2 4 1 2 3 1	

Entrada	Salida
3 1 3 1 1 1 2 5 1 4 1 2 4 1 2 3 1	DESPEDIDO

F - Palabra Clave

Esta noche es el programa de televisión *Palabra Clave*, en el que dos jugadores se alternan turnos para descubrir una palabra que se encuentra completamente cubierta inicialmente. En cada turno el jugador elige una letra del alfabeto que todavía tiene chances de estar en la palabra, en caso de que haya una letra de ese tipo por descubrir se destapa una de sus ocurrencias. El jugador que descubre la última letra gana el juego.

Te invitaron a participar del programa de esta noche, pero el presentador les dijo sin querer la palabra clave a vos y a tu contrincante ¿Podés ganar si jugás en el primer turno y ambos juegan óptimamente?

Entrada

Una línea con un entero T ($1 <= T <= 1000$), la cantidad de casos de prueba. Cada caso de prueba contiene una sola línea con un string K ($1 <= |K| <= 26$), la palabra clave.

Salida

Una línea por cada caso de prueba con el string ":)" en caso de que puedas ganar o ":(" en caso contrario.

Ejemplos

Entrada	Salida
2	:(
ANARAP	:)
FRUTAFRESCA	

G - TOC

Como todos saben, Monica Geller tiene un Trastorno Obsesivo Compulsivo (TOC) con el orden y la limpieza. Últimamente, ha estado organizando todo en su apartamento (especias, toallas, e incluso las películas de Chandler) y asignando a cada ítem un "valor numérico" basado en algún criterio que solo ella entiende (quizás nivel de limpieza, fecha de caducidad o "sensación de orden").

Esto ha resultado en largas secuencias de números.

El TOC de Monica no la deja tranquila. Para sentirse satisfecha, necesita encontrar "patrones perfectos" en sus secuencias. Para ella, un patrón perfecto es una **secuencia aritmética continua**, es decir, una subsecuencia de números consecutivos donde la diferencia entre cada número y el siguiente es siempre la misma.

Pero hay una condición más (¡obviamente, es Monica!). Para que ella esté *realmente* satisfecha y pueda dejar de organizar, el patrón debe continuar **perfectamente hasta el último elemento de la secuencia**.

De todos los posibles patrones aritméticos que terminan justo al final, Monica quiere encontrar el más largo, ya que un patrón más largo la tranquiliza más.

Tu tarea es ayudar a Monica a calmar su TOC. Dada una secuencia de números, debes encontrar la posición (índice) donde comienza la subsecuencia aritmética continua más larga que se extiende hasta el final.

Entrada

Habrá múltiples casos de prueba. Cada caso de prueba consiste en dos líneas.

La primera línea contiene un único entero N ($1 \leq N \leq 1000$), que especifica la longitud de la secuencia de Monica.

La segunda línea contiene una secuencia de N enteros X_i ($0 \leq X_i \leq 10^9$), separados por espacios.

Salida

Para cada caso de prueba, imprime una única línea con el índice del primer miembro de la subsecuencia aritmética continua más larga que se extiende hasta el final de la secuencia dada.

El índice del primer elemento en la secuencia es siempre 1.

Ejemplos

Entrada	Salida
5	1

Entrada	Salida
1 2 3 4 5	6
7	2
1 2 3 4 5 8 8	
3	
7 5 2	

H - SpeedForces

SpeedForces es la principal página de programación competitiva en la ciudad de Santa Fe, donde podés participar de competencias semanales para subir de ELO, o lo que es más común, bajar de ELO.

La más reciente novedad de *SpeedForces* es el *SpeedPredictor*, lo último en predicción de cambios de ELO en competencias. Hay N competencias este año y *SpeedPredictor* ya ha generado las predicciones para tu cambio de ELO para cada una de ellas. Una predicción puede ser de de uno de los siguientes tipos:

1. $= E$, luego de la competencia tu ELO es igual a E.
2. $+E$, luego de la competencia a tu ELO se le suma E.

¡Pero todavía hay más! Tu entrenador, Matías, esta interesado en saber cual sería tu máximo ELO si compitieras sólo en un rango $[L, R]$ de todas estas competencias, comenzando con $ELO = 0$. Cabe aclarar que podés saltearte las competencias que quieras con tal de maximizar tu ELO, siempre respetando el orden de las mismas ¿Podés darle una mano a Mati para encontrar la respuesta a cada una de sus consultas?

Entrada

Una línea con un entero N ($1 <= N <= 2 * 10^5$), la cantidad de competencias en el año.

Una línea por cada competencia con un símbolo "=" o "+", y un entero E ($-600 <= E <= +600$): - "= E" significa que tu ELO luego de la competencia es E. - "+ E" significa que tu ELO luego de la competencia se le suma el entero E.

Una línea con un entero C ($1 <= C <= 2 * 10^5$), la cantidad de consultas de tu entrenador.

Una línea por cada consulta de tu entrenador con dos enteros L y R ($1 <= L <= R <= N$).

Salida

C líneas, la i-esima línea debe contener un entero M con la respuesta a la i-ésima consulta de Mati.

Ejemplos

Entrada	Salida
5 + -2 = 3	3 5

Entrada	Salida
= -2	
+ 2	
+ -1	
2	
1 2	
1 5	

J - Cascadados

¿Acaso un momento de felicidad no alcanza para toda una vida? ¿Acaso aferrarse a un instante no nos ayuda a contener el devenir del tiempo? ¿Acaso...?

—¿Va a apostar o no, señor? —interrumpió el croupier, preparado para cerrar las apuestas.

Demonios.

Tengo solo tres segundos para calcular el valor esperado de tiradas necesarias para finalizar el juego, si quiero entrar en esta ronda. Probablemente sean dos más de los que necesito.

Por suerte, conozco bien *cascadados*. Es un juego sencillo... o eso parece.

El juego inicia con un conjunto de dados cuyo número de caras puede variar. Cada ronda consiste en lanzar todos los dados que queden. Si en un lanzamiento ninguno muestra un 1, la ronda termina y se avanza a la siguiente conservando todos los dados. Pero si alguno de los dados obtiene un 1, ese/esos dado/s se eliminan del conjunto y -como si el azar despertara una reacción en cadena- se activa una *cascada*: volviendo a lanzar los dados restantes en esa misma ronda. Si en esa repetición aparecen nuevos 1, esos dados también se eliminan y se inicia una nueva cascada para la ronda en curso.

Lamentablemente, la casa impone un límite: por cada tirada solo puede iniciarse una *cascada*, sin importar la cantidad de dados mostrando un 1.

El juego se repite ronda tras ronda hasta que no queden dados. Quien apueste con el número de rondas más cercano para finalizar el juego se llevará el pozo.

Voy a ir por lo seguro: calcular, dado el conjunto inicial de dados, cuál es el número esperado de rondas adicionales (sin contar la primera) que el juego necesitará hasta que todos los dados hayan sido eliminados.

¡Piensa, cerebro, piensa! Si tan solo alguien me ayudara a calcularlo...

Entrada

La primera línea contiene una expresión de la forma "Ad4 Bd6 Cd8 Dd10 Ed12 Fd20" donde A, B, C, D, E y F ($0 <= A, B, C, D, E, F, A+B+C+D+E+F <= 18$) representan la cantidad de dados de 4, 6, 8, 10, 12, y 20 caras respectivamente en el conjunto inicial de dados.

Salida

La salida consta de una línea, el número esperado de rondas (excluyendo la primera) requeridas para finalizar el juego, dado el conjunto inicial de dados. La salida siempre tendrá cinco dígitos después del punto decimal.

Entrada	Salida
0d4 1d6 0d8 0d10 0d12 0d20	5.00000

Entrada	Salida
1d4 2d6 0d8 3d10 1d12 0d20	16.51845

Entrada	Salida
1d4 0d6 1d8 0d10 0d12 0d20	7.18182

K - Tree Ninja

Fruit Ninja es un juego para celulares donde el jugador debe utilizar sus dedos para cortar las frutas que aparezcan en la pantalla antes de que caigan al piso o de lo contrario el jugador pierde una vida.

Como la gente se estaba aburriendo de jugar *Fruit Ninja* los desarrolladores crearon una nueva versión llamda *Tree Ninja* en la que el jugador debe realizar un único corte a un arbol planar. El jugador recibe un punto por cada parte en la que el árbol planar es dividido después del corte. Un corte consiste en una línea recta que separa todas las aristas por las que pasa.

Dado un árbol planar ¿podrías calcular la máxima cantidad de puntos que podés conseguir?

Entrada

Una nueva línea con un entero N ($1 <= N <= 10^3$), la cantidad de nodos en el árbol planar.

N líneas con dos enteros X e Y ($-10^8 <= X, Y <= 10^8$), las coordenadas del i-ésimo nodo.

$N - 1$ líneas con dos enteros U y V ($1 <= U, V <= N$), indicando una arista entre el nodo U y V.

No hay tres puntos colineales y nunca se cruzan dos aristas.

Salida

Un entero positivo P, la máxima cantidad de puntos que puede conseguir un jugador.

Ejemplos

Entrada	Salida
4 1 2 1 0 1 1 2 0 1 2 2 3 2 4	4

L - Frutas Frescas

Juan y Mariano están realizando un nuevo emprendimiento: una verdulería llamada “Frutas Frescas”

Para abastecer su negocio, tienen que buscar frutas a lo largo de toda la ciudad, la cual podemos representar como un plano. Para ello, quieren contratar repartidores que las busquen, ya que no tienen muchas ganas de caminar.

Cada repartidor como cada fruta poseen una coordenada específica X,Y en la cual están ubicados, y a su vez, la verdulería tiene su propia coordenada.

En esta ciudad todas las calles son rectas, por lo tanto, el costo de ir de una posición A,B hasta una posición X,Y es de |A-X| + |B-Y| (es decir, distancia Manhattan, para los más experimentados)

Cada fruta tiene que ser llevada a la verdulería, y puede ser buscada por cualquiera de los repartidores. Cada repartidor solo puede llevar una fruta a la vez en viajes separados. El costo de llevar una fruta a la verdulería, es la distancia del repartidor a la fruta, más la distancia de la fruta a la verdulería. Notar que una vez que el repartidor lleva una fruta a la verdulería, su punto de inicio para la siguiente busqueda será desde ahí.

El pago que recibe cada repartidor equivale a 1 peso por cada unidad de distancia recorrida.

Como Juan y Mariano son optimizadores natos, quieren minimizar el costo de llevar todas las frutas a la verdulería, ¿podrías ayudarlos?

Entrada

El input consiste de varias líneas:

- Una línea con elementos N y M, (1 <= N,M <= 1000), representando N frutas y M repartidores.
- N líneas con coordenadas X,Y (-1000 <= X,Y <= 1000), representando la coordenada de cada una de las frutas.
- M líneas con coordenadas X,Y (-1000 <= X,Y <= 1000), representando la coordenada inicial de cada uno de los repartidores.
- La última línea consiste de una coordenada X,Y (-1000 <= X,Y <= 1000), la coordenada de la verdulería.

Salida

El output consiste de un solo entero P, la mínima cantidad de pesos que tienen que pagar para llevar todas las frutas a la verdulería.

Ejemplos

Entrada	Salida
2 2 1 0 0 -1 -1 1 2 -1 0 0	5