Concurso de Programación

Clasificatorio Regional Madrid 2022

http://www.ada-byron.es

Cuadernillo de problemas

Realizado en la **Escuela Técnica Superior de Ingeniería de Sistemas Informáticos (UPM)**
12 de marzo de 2022

In almost every computation a great variety of arrangements for the succession of the processes is possible, and various considerations must influence the selections amongst them for the purposes of a calculating engine. One essential object is to choose that arrangement which shall tend to reduce to a minimum the time necessary for completing the calculation.
Ada Byron

12 de marzo de 2022

Índice

Autores de los problemas:

- Sergio Cavero Díaz (Universidad Rey Juan Carlos)
- Alberto Díaz Álvarez (Universidad Politécnica de Madrid)
- Antonio González Pardo (Universidad Rey Juan Carlos)
- Raúl Lara Cabrera (Universidad Politécnica de Madrid)
- Isaac Lozano Osorio (Universidad Rey Juan Carlos)
- Jesús Mayor Márquez (Universidad Politécnica de Madrid)
- Raúl Martín Santamaría (Universidad Rey Juan Carlos)
- Dixon David Morán González (ThoughtWorks)
- Fernando Ortega Requena (Universidad Politécnica de Madrid)
- Sergio Pérez Peló (Universidad Rey Juan Carlos)
- Jesús Sánchez Oro (Universidad Rey Juan Carlos)
- Leonardo Santella (Amazon)
- Taras Sotnikov (Telefónica)

Tiempo: 0.3 segundos

A
Ayudando a los camareros

Cuando vamos a un restaurante, a nuestros compañeros de trabajo les gusta facilitarle la vida a los camareros. Por eso, a la hora de pagar usamos el menor número de cheques restaurante posible. Los cheques restaurante son un medio de pago aceptado en restaurantes que consiste en cheques de diferente valor monetario.

Por ejemplo, en este escenario:

- 3 personas
- Precio de la comida por persona de 11 €
- Valores de los cheques: 6, 2 y 1 €

Cada persona podría pagar por su propia comida con 4 cheques: $6 + 2 + 2 + 1 = 11$, de esta manera le daríamos un total de 12 cheques al camarero. Pero como el coste total es de 33 €, preferimos pagar con solo 7 cheques (6, 6, 6, 6, 6, 2, 1), ¡así el camarero pierde menos tiempo contándolos!

¡Ha llegado la hora de pagar! Dado el número de empleados que han ido a comer al restaurante y el precio de la comida por persona, ¿cuál es el número mínimo de cheques con el que podemos pagar? Ten en cuenta que necesitamos pagar la cantidad exacta de dinero que cuesta la comida. Afortunadamente, el precio de la comida es siempre un entero y siempre tendremos disponibles cheques de valor de 1 €, por lo que siempre será posible pagar la comida, además, de todos los cheques tenemos una cantidad infinita.

Entrada

La primera linea contiene un entero T que indica el número de casos de prueba.

Cada caso de prueba está formado por dos líneas. La primera contiene 3 enteros:

- P, el número de personas
- C, el coste de la comida por persona
- N, el número de distintos valores de cheques

La segunda línea contiene N enteros que indican los distintos valores de los cheques $V_1, V_2, \dots V_N$, en orden ascendente y separados por un espacio en blanco.

Salida

Para cada caso de prueba T, la salida es el número mínimo de cheques a pagar por la comida.

Recuerda que la cantidad a pagar deber ser exactamente el coste de las comidas.

Entrada de ejemplo

```
3
3 11 3
1 2 6
6 10 3
1 2 6
2 13 3
1 5 6
```

Salida de ejemplo

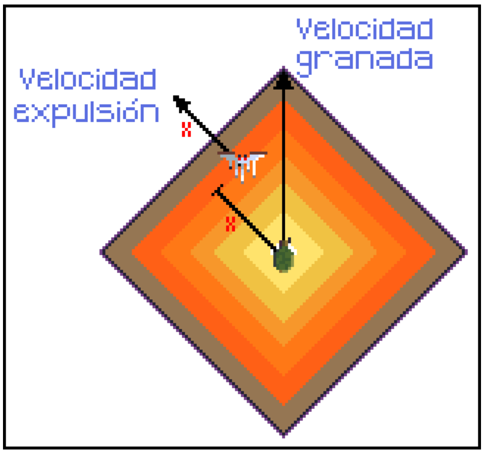
```
7
10
5
```

Límites

- $1 <= T, P, C, V <= 1000$
- $1 <= P * C <= 1000$

Tiempo: 1 segundos

B
¡¡¡Explosiones cuadradas!!!



Antonio es un creador de videojuegos Indie que ama crear mecánicas novedosas. En concreto, Antonio, quiere desarrollar un juego bélico en dos dimensiones con unas reglas algo peculiares. Su siguiente mecánica estrella de su videojuego consistirá en unas granadas explosivas especiales. Estas granadas explotan a control remoto, con el fin de poder alcanzar objetivos que se encuentran en el aire. Es decir, el usuario se encarga de elegir en qué momento quiere que explote dicha granada.

Más novedoso aún es que cada granada tiene una onda expansiva cuadrada. La forma de dicho cuadrado viene dada por su velocidad y dirección en el momento de explotar. Dicho cuadrado de explosión orientará uno de sus vértices hacia la dirección de movimiento de la granada.

El área de efecto de la explosión dependerá de la velocidad de la granada. A más velocidad, mayor área de explosión. Concretamente, en nuestro juego, las velocidades se expresan como un vector bidimensional que viene dado en unidades de distancia de nuestro juego por segundo. En este vector, su magnitud indica la velocidad. O lo que es lo mismo, dónde se encontraría el objeto dentro de un segundo si no se alterase la velocidad a lo largo del tiempo.

Si un enemigo es alcanzado con una granada, este es despedido con una velocidad diferente a la que cabría esperar. En concreto, la onda expansiva empujará al enemigo en dirección a su arista más cercana. La velocidad será equivalente a la distancia de granada y enemigo proyectada sobre la dirección de repulsión. No hace falta decir que, si el enemigo se encuentra pegado a la granada, no será afectado apenas por el empuje de la explosión. Sin embargo, si el enemigo se encuentra en dicha arista, el efecto de la bomba será el máximo.

Entrada

Se necesita calcular si la granada afecta al enemigo o no.

La primera linea contiene un entero T que indica el número de casos de prueba.

Cada caso de prueba está formado por una línea que contiene 6 números separados por un espacio en blanco en el siguiente orden:

- X_g, coordenada x de la posición del de la granada en el instante que esta explota.
- Y_g, coordenada y de la posición del de la granada en ese instante.
- $\vec{X_v}$, coordenada x del vector velocidad de la granada en ese instante.
- $\vec{Y_v}$, coordenada y del vector velocidad de la granada.
- X_e, coordenada x del enemigo en ese instante.
- Y_e, coordenada y del enemigo.

Salida

Se esperan dos tipos salidas posibles:

- Si la explosión no afecta al enemigo, se indicará con la palabra "FALLO".

- En caso de alcanzarlo, se indicará mostrando el vector velocidad resultante en unidades por segundo del empuje provocado por la onda expansiva de la granada. Este se indicará entre paréntesis, separando por una coma la coordenada x de la y. Con el fin de evitar errores en la salida, todas las coordenadas se imprimirán con un único decimal.

Entrada de ejemplo

```
5
0 0 0 10 2.5 -2.5
100 100 0 20 50 50
0 0 0 50 20 20
50 50 40 40 50 20
25 50 25 50 26 28
```

Salida de ejemplo

```
(2.5,-2.5)
FALLO
(20.0,20.0)
(0.0,-30.0)
(6.7,-20.1)
```

Límites

- El espacio de juego tiene siempre coordenadas positivas entre 0 y 100 para ambos ejes. Sin embargo, la onda expansiva si que podrá exceder esos límites.
- Casos con múltiples soluciones, donde se encuentre el enemigo sobre la propia dirección de la granada o un vector perpendicular a este, no serán objeto del ejercicio.
- La velocidad de la granada en ningún caso podrá ser de magnitud 0.
- La granada y el enemigo no pueden estar en el mismo sitio.

Tiempo: 5 segundos

C
Robo perfecto en Villa Rejilla

Villa Rejilla es mundialmente conocida por su museo regional de gemas y joyas, que incluye en su colección el espectacular Diamantón, el mayor diamante jamás encontrado. Bueno, también es conocida por la peculiar organización de su superficie: todo se distribuye sobre una rejilla cuadrada.

Sly y su banda están planeando el robo del Diamantón. Para ello se han hecho con los planos de todas las salas del museo. Para ser más eficientes han seguido una estrategia divide y vencerás y, actualmente, solo les queda por estudiar y planificar la última sala antes de llegar al Diamantón: la temida sala de los espejos.

La banda sabe de la existencia de un dispositivo de detección de intrusos que funciona emitiendo un rayo de luz infrarrojo. El dispositivo está motorizado, por lo que es capaz de girar 360º, pero la banda no conoce la posición exacta (sobre la rejilla, obviamente) que tendrá el detector en el momento en el que entren en la sala. Además, como su propio nombre indica, todas las paredes de la sala son espejos, por lo que el dispositivo podría detectar a Sly sin tener que estar orientado directamente hacia el, mediante rebotes en las paredes. Afortunadamente para Sly, los recortes en el presupuesto del museo han provocado que el emisor del rayo infrarrojo tenga una distancia efectiva limitada. Además, el rayo se corta en el momento en el que choca con cualquier objeto que no sean las paredes, incluyendo al propio dispositivo de detección.

Para preparar la ruta óptima y burlar al dispositivo de detección de intrusos, la banda quiere desarrollar un programa para simular y evaluar las distintas posibilidades a las que se va a enfrentar Sly cuando entre en la sala, tanto con respecto a la posición del dispositivo como a la suya propia.

¿Puedes ayudar a Sly a planificar el asalto a la sala de los espejos? Para ello, Sly necesita saber desde cuántos ángulos distintos puede ser detectado por el rayo de luz del dispositivo sabiendo la posición que ocupan tanto él como el dispositivo en la rejilla así como el tamaño de la sala.

Entrada

La primera linea contiene un entero T que indica el número de casos de prueba.

Cada caso de prueba está formado por una línea que contiene 7 enteros separados por un espacio en blanco en el siguiente orden:

- W, anchura de la sala de los espejos
- H, altura de la sala de los espejos
- X_d, coordenada x de la posición del dispositivo de detección de intrusos
- Y_d, coordenada y de la posición del dispositivo de detección de intrusos
- X_S, coordenada x de la posición de Sly
- Y_S, coordenada y de la posición de Sly
- D, distancia efectiva del rayo de detección

Salida

Para cada caso de prueba T, el número de ángulos distintos desde los cuales Sly puede ser detectado por el dispositivo en la sala de los espejos. La salida de cada caso de prueba irá en una línea distinta.

Entrada de ejemplo

```
2
3 2 1 1 2 1 4
300 275 150 150 185 100 500
```

Salida de ejemplo

```
7
9
```

Límites

- Las dimensiones de la sala de los espejos están limitadas: $1 < W < 1250$, $1 < H < 1250$
- Tanto el dispositivo detector como Sly siempre estarán dentro de la sala de los espejos: $0 < X_d < W$, $0 < Y_d < H$, $0 < X_S < W$, $0 < Y_S < H$
- La distancia efectiva del rayo está limitada: $1 < D <= 5000$

Tiempo: 1.5 segundos

D
El juego del pañuelo

Muchos institutos tienen olimpiadas internas entre las diferentes clases de los alumnos para fomentar el trabajo en equipo, competitividad, etc. Además de los deportes clásicos, como fútbol, baloncesto, tenis, ajedrez, etc. En mi instituto uno de los juegos favoritos es el juego del pañuelo. Este es un juego de persecución para niños en el que los equipos intentan atrapar un pañuelo sin ser pillados. Las reglas del juego son las siguientes:

- Los jugadores están divididos en dos equipos.
- Cada jugador de un equipo tiene asignada una señal de llamada única, en nuestro caso un número. Cada miembro del equipo contrario tiene otro número que encaja con uno y solo uno de sus contrincantes.
- Los equipos se alinean en bordes opuestos del área de juego y, en el centro, se sitúa el árbitro con el pañuelo.
- El árbitro dice un número y los dos jugadores que lo tienen asignado corren al centro a agarrar el pañuelo. Deben atraparlo y regresar a su zona sin ser pillados por el miembro del otro equipo (el que no ha cogido el pañuelo debe atrapar al que sí).
- Si el jugador que atrapó el pañuelo consigue llevarlo hasta su zona, su equipo se anota un punto.
- Si es pillado, el otro equipo se lo anota.
- Importante: No se puede pillar a un jugador que todavía no ha tocado el pañuelo.

Durante la competición anual de mi instituto, todas las clases se inscriben con los participantes más atléticos. En mi caso no me han seleccionado, pero quiero ayudar a mis compañeros. Hemos visto que la organizadora del evento ha dejado los números de cada uno de los participantes encima de la mesa, y en un descuido los hemos copiado. Dados los números asignados a los participantes, sabemos qué alumno de la clase contraria se enfrentará con alguno de mis compañeros, así que aún podemos cambiar la asignación de los números que han tocado en mi equipo para obtener ventaja. Mis compañeros tienen muy claro a qué personas del equipo contrario son superiores, es decir, a quiénes ganarían con total seguridad en caso de enfrentarse contra ellos. Teniendo esto en cuenta, me han pedido desarrollar un programa que determinará la mejor asignación para el equipo de mi clase dada la asignación numérica de los participantes del equipo contrario, reportando el resultado final.

Entrada

La entrada estará compuesta por un número C que representa el número de casos de prueba de la entrada. Por cada C vendrá un línea con un número A de alumnos que componen el equipo de mi clase (ambos equipos tienen el mismo número de alumnos). A continuación vendrán A líneas, cada una con un número G que indica el número de alumnos de la otra clase que sabe que ganará con total seguridad. En la misma línea, se mostrarán G identificadores I de los alumnos a los que ganará.

Salida

La salida mostrará el resultado de la competición si se realiza la mejor asignación posible. Se escribirán, separadas por un guión, las victorias de mi clase y las de la clase frente a la que competiremos.

Entrada de ejemplo

```
2
5
1 1
1 2
1 3
1 4
1 5
5
2 1 2
0
1 3
1 4
1 5
```

Salida de ejemplo

```
5-0
4-1
```

Límites

- $1 \leq C \leq 1000$
- $1 \leq A \leq 200$
- $0 \leq G, I \leq A$

Tiempo: 0.5 segundos

E
Wordle: el juego del que todo el mundo habla

A estas alturas seguro que ya has oído hablar de Wordle. Pero, ¿sabes lo que es? ¿Has pasado algún tiempo en Twitter recientemente? ¿Has visto todas esos mensajes de tus amigos en grupos con cuadros amarillos, verdes y grises? Sí, eso es Wordle.

Wordle es un juego de palabras diario inspirado en los clásicos crucigramas de periódico que sólo se puede jugar una vez al día. Cada 24 horas hay una nueva palabra del día, y depende de ti averiguar cuál es, para lo que dispondrás de seis intentos.

El funcionamiento de Wordle es sencillo: ofrece a los jugadores seis oportunidades de adivinar una palabra de cinco letras seleccionada al azar. Si tienes la letra correcta en el lugar correcto, aparece en verde. Una letra correcta en el lugar equivocado aparece en amarillo. Una letra que no está en la palabra en ningún lugar aparece en gris. ¿te parece sencillo? Habría que ver tus estadísticas.

De estadísticas va el programa que se le ha encargado hacer a los participantes del AdaByron de este año, vamos allá...

Entrada

La entrada comienza con una línea que contiene un número, n, el número de partidas que un jugador ha realizado. Las siguientes líneas se corresponden con las n partidas. La estructura de las partidas es similar, aunque el número de líneas de cada partida dependerá de cómo de rápido el jugador adivine la palabra oculta. La estructura de una partida es la siguiente:

- La primera línea contiene la palabra oculta. Esta palabra tiene cinco letras.
- Las siguientes líneas contienen los intentos del jugador. Como mínimo, deberá realizar un intento, y podrá fallar hasta en seis ocasiones.

La partida finaliza tan pronto como el jugador adivine la palabra oculta. En ese momento, comenzará una nueva partida, hasta alcanzar el número n de partidas.

Salida

La salida consiste siempre en 6 líneas, cuyo contenido se detalla a continuación:

- La primera línea muestra el porcentaje de partidas que el jugador consiguió encontrar la palabra.
- Las siguiente 5 líneas indicarán el porcentaje de las partidas resueltas en el primer, segundo, tercer, cuarto, quinto, sexto intento.

El formato de cada una de las líneas se especifica en la salida de ejemplo.

Importante: todos los números reales obtenidas a partir de cualquier operación serán redondeados al entero menor más próximo (33.34 es redondeado a 33, 66.67 es redondeado a 66).

Entrada de ejemplo

```
4
LEJOS
ISAAC
JUEGO
MEJOR
LEJOS
OVEJA
SERIO
JOVEN
OVEJA
BAHIA
SECAR
AYUDA
TANTA
PALMA
VACIA
ZAFIA
HUMOR
CERDO
MAYOR
HUMOR
```

Salida de ejemplo

```
75%
1->0%
2->0%
3->50%
4->25%
5->0%
6->0%
```

Límites

- $1 \leq n \leq 100000$

Tiempo: 0.5 segundos

F
La conquista del espacio

Elon Musk está ansioso por lograr que la raza humana se convierta en una especie interplanetaria y, tras años de desarrollo, su empresa SpaceX ha anunciado que dispone del nuevo cohete Falcon Ultimate capaz de viajar de forma real entre los planetas de la Vía Láctea. Para que la raza humana se considere “interplanetaria” es necesario asentar una colonia en algún planeta diferente a la Tierra y, por ello, el primer objetivo del nuevo reto de Musk consiste en tomar muestras de los planetas candidatos de ser habitables mediante una misión no tripulada.

Gracias al telescopio espacial James Web se ha logrado elaborar un mapa estelar que contiene información detallada de las distancias entre los planetas. Este dato es crítico, pues debido a la atracción interplanetaria, se ha descubierto que no es posible viajar entre dos planetas cualesquiera, si no que existen unas rutas preestablecidas que el Falcon Ultimate es capaz recorrer. Además de esta información, el telescopio espacial James Web ha reportado la composición de la atmósfera de cada planeta, información crucial para conocer de antemano si el planeta es potencialmente habitable y si dispone de Helio, pues este gas puede ser transformado por el cohete del Falcon Ultimate en combustible para su propulsor a razón de 100 Kg de combustible por cada tonelada de Helio existente en el lugar del aterrizaje.

Sabiendo que el cohete Falcon Ultimate tiene una capacidad máxima de 2 toneladas de combustible y empleando la información del mapa estelar, se desea determinar la ruta planetaria que permita visitar el mayor número de **planetas diferentes** con el menor número de viajes.

Entrada

La primera linea contiene un entero N con el número de planetas que componen el mapa estelar.

La siguientes N líneas contienen información relativa a la atmósfera de los planetas. En concreto, cada linea está compuesta por un número decimal que indica las toneladas de helio presentes en la zona de aterrizaje del planeta. El primer planeta de la lista siempre será la Tierra, donde no se repostará combustible y, por tanto, tendrá un valor 0.

Las sucesivas líneas contienen los costes de viajar entre los planetas. Cada linea esta compuesta de:

- Un entero entre 0 y $N-1$ indicando el identificador del planeta de origen.
- Un entero entre 0 y $N-1$ indicando el identificador del planeta de destino.
- Un número decimal indicando los kg de combustible necesarios para llegar del planeta origen al planeta destino.

Es importante saber que el coste de viajar del plantea A al planeta B es el mismo que el de viajar del planeta B al planeta A, por lo que la información del coste de viajar entre A y B solo aparece una vez en los datos de entrada.

Salida

Se espera determinar la secuencia de planetas que se deben visitar, separados por espacios, para lograr visitar el mayor número de planetas diferentes con el menor número de viajes interplanetarios.

Entrada de ejemplo

```
6
0.0
1000.0
3000.0
500.0
1000.0
2000.0
0 1 200.0
0 2 300.0
0 3 400.0
1 4 800.0
1 5 500.0
2 3 350.0
2 4 250.0
3 4 1000.0
4 5 400.0
```

Salida de ejemplo

```
0 1 5 4 2 3
```

Límites

- El cohete iniciará siempre su viaje con sus tanques de combustible a su máxima capacidad 2 toneladas (2000 kg).
- Cuando el cohete aterriza en un planeta, obligatoriamente transformar todo el Helio disponible en el planeta en combustible adicional para el cohete:
  - El cohete nunca puede tener más de 2 toneladas de combustible.
  - El exceso de Helio del planeta se desperdicia (el planeta se queda sin Helio).
  - El Helio no se regenera en el planeta.
  - No puede haber una cantidad negativa de Helio en un planeta.
  - El aterrizaje no implica ningún gasto adicional de combustible. Todo el combustible necesario para viajar entre dos planetas viene indicado en el coste de la ruta.
- Es posible re-visitar un planeta si es la ruta que más conviene para lograr visitar el mayor número de planetas diferentes.
- El mapa estelar nunca tendrá más de 8 planetas.
- Todos los planetas estarán conectados por al menos una ruta. El coste de estas rutas será un número estrictamente positivo.
- En caso de que existan dos rutas con que visiten el mismo número de planetas con igual número de viajes, se devolverá la ruta con menor orden lexicográfico. Por ejemplo, si se obtienen las rutas `‘0 2 1’` y `‘0 1 2’` se devolverá la ruta `‘0 1 2’`.

Tiempo: 1.5 segundos

G
El hacker al que le sobraba el dinero

Un hacker con mucha ética ha sido capaz de vulnerar el sistema de su sucursal y puede acceder a un montón de transacciones de clientes; N para ser exactos. Se ha dado cuenta de que un atacante puede cambiar el destinatario de dichas transacciones, aunque el sistema **sólo permite cambiar un segmento contiguo de transacciones**.

Como las transacciones son de recibir (positivas) o enviar (negativas) dinero, le interesa saber qué conjunto contiguo de transacciones maximiza el dinero que un atacante podría llevarse explotando esta vulnerabilidad (evidentemente no tiene sentido si todas las transacciones son negativas, en cuyo caso la ganancia sería 0). Pueden ser las N transacciones, pueden ser menos, puede ser ninguna (a lo mejor todas son transacciones en envío y no recepción) pero lo importante es que tienen que ser contiguas.

Entrada

La primera linea contiene un entero N con el número de transacciones que se han realizado ese día.

La segunda línea contiene N valores enteros con el valor de la transacción, positivo (si es una transacción recibida) o negativo (si es una transacción enviada).

Salida

Una línea con un único valor, la suma máxima de las subsecuencia de transacciones.

Entrada de ejemplo

```
12
273 -164 -51 166 -432 268 -12 70 -232 -299 -271 -33
```

Salida de ejemplo

```
326
```

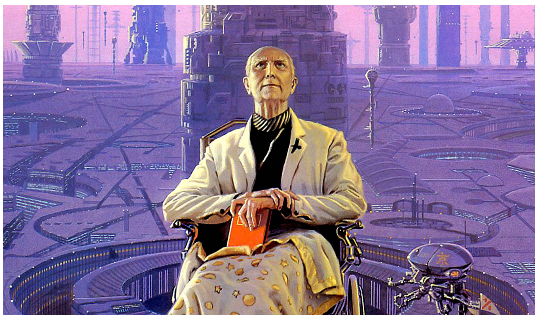
Explicación: La mayor ganancia que se puede obtener de entre los subsegmentos de transacciones es de `326`, correspondiente al subsegmento `268 -12 70`

Límites

- El número de transacciones estará comprendido entre 0 y 10^6.
- El valor de las transacciones estará comprendido entre -10^6 y 10^6.

Tiempo: 0.5 segundos

H
La guerra intergaláctica



Hace un tiempo que existe en la galaxia una ciencia capaz de predecir con bastante precisión los acontecimientos venideros: la psicohistoria. Según esta ciencia, la guerra entre la Fundación y el Imperio está a punto de estallar. Esta vez, ni las habilidades de los gobernantes ni las argucias de los psicohistoriadores podrán evitarlo. Ante este escenario, Hari Seldon necesita planificar el inevitable conflicto para intentar reconducir el destino de la galaxia, evitando que El Mulo se haga con el poder de todo el Universo. Para ello, quiere trazar un mapa de toda la galaxia, desde Askone hasta la Tierra, con el objetivo de decidir cuáles son los planetas estratégicos que descompondrían la unidad galáctica para siempre. Actualmente, las rutas comerciales conectan todos los planetas, pero hay algunos puntos estratégicos que no deberían perderse. Si caen en manos del imperio, la red comercial dejará de ser robusta, quedando desconectada en algún punto. Es decir, existirá al menos un par de planetas que no podrán comerciar entre sí. Para hacerse una idea de qué planetas deben ser defendidos con mayor ímpetu, Hari Seldon ha asociado a cada planeta de la red el coste que supondría su pérdida para la Fundación. Tu objetivo será desarrollar un programa que, dada la red de planetas y su valor asociado, imprima el coste de perder **todos** los planetas estratégicos y un listado con sus nombres.

Entrada

La primera linea contiene dos enteros N y M, que representan el número de planetas que existen en la galaxia y cuántas conexiones comerciales hay entre ellos, respectivamente. Las siguientes N líneas contienen una cadena S de, como máximo, 100 caracteres y un entero C, indicando el nombre de un planeta y el coste de perderlo durante la guerra. Las siguientes M líneas contendrán dos cadenas de caracteres S_a y S_b, separadas por un espacio, que indican los nombres de los planetas que tienen una relación comercial.

Salida

Se debe imprimir una línea con el coste de perder todos los planetas críticos (estratégicos) y un listado con sus nombres, en el mismo orden en el que se recibieron en la entrada (esto es, en el listado de planetas, antes de recibir la información sobre las relaciones comerciales entre ellos).

Entrada de ejemplo

```
6 5
LORIS 50
SMYRNO 22
DARIBOW 58
ANACREONTE 99
KALGAN 38
TERMINUS 21
LORIS SMYRNO
SMYRNO DARIBOW
DARIBOW ANACREONTE
ANACREONTE KALGAN
KALGAN TERMINUS
```

Salida de ejemplo

```
217
SMYRNO
DARIBOW
ANACREONTE
KALGAN
```

Límites

- $6 \leq N \leq 1000$
- $5 \leq M \leq 30000$
- $10 \leq C \leq 1000$
- $0 \leq a, b < N$

Tiempo: 0.5 segundos

I
Bajo presión

Durante esta semana un pequeño grupo de turistas españoles ha sido secuestrado por una célula radical en Israel. A día de hoy, se desconoce el paradero de los rehenes. Pero la cooperación del Centro Nacional de Inteligencia con el Mossad resulta clave para poder rescatar a los turistas españoles sin que sufran daños.

Por suerte para los turistas, el grupo radical que les ha secuestrado está siendo investigado por la gente del Mossad. Diversos agentes se encuentran infiltrados entre la población israelí y varios de ellos tienen relaciones con algunos miembros de la banda radical.

En este tiempo, estos agentes (expertos en el comportamiento humano y la psicología) han notado que el nivel de tensión dentro de los miembros del grupo radical está creciendo. Además, han logrado elaborar un sistema por el cual pueden determinar el nivel máximo de presión que puede soportar cada radical. Saben que cuando el estrés de un secuestrador es igual o superior a dicho nivel, hay muchas probabilidades de que el radical sea capaz de revelar información confidencial. Por último, conocen que el líder de la célula va a asignarles nuevas tareas que deben realizar, y han evaluado la cantidad de estrés que dichas tareas van a añadir al propio estrés que ya sufren los radicales.

Vuestra labor en este ejercicio será la de ayudar tanto al Centro Nacional de Inteligencia como al Mossad, desarrollando un programa que sea capaz de determinar qué radicales son los que no van a soportar la presión de las tareas asignadas. Esta labor es clave, ya que los agentes del Mossad saben que cuando los radicales están muy estresados son más propensos a revelar secretos de la organización... como el lugar donde tienen cautivos a los turistas españoles.

Entrada

La entrada de este problema contendrá únicamente un caso de prueba. La primera línea contiene dos enteros R y T, que representan el número de radicales que participa en la célula y el número de tareas que van a realizar. Las siguientes R líneas contienen dos números: R_i, que representa el identificador del radical, y E_{R_i} que representa el nivel de estrés límite de dicho radical. Después, aparecen T líneas que contienen dos valores: T_i que es el identificador de la tarea, y E_{T_i} que indica las unidades de estrés que va a añadir dicha tarea al nivel de estrés del radical al que se le asigne. Por último, el caso terminará con las asignaciones de tareas a los radicales. Esto se indicará en T líneas diferentes que contienen dos números: T_i que es el identificador de la tarea, y R_i que representa el identificador del radical.

Salida

La salida de este programa serán los nombres de aquellos radicales que no podrán manejar la presión de las tareas asignadas. Estos nombres serán del tipo `Radical X`, donde `X` es el identificador del radical.

En caso de que haya varios radicales que deban ser mostrados en la salida, la lista de nombres se imprimirá en el mismo orden en el cual los secuestradores no van soportando la presión. También cabe resaltar que si un radical no soporta la presión al asignarle una tarea, y se le asigna más tareas, su nombre solo aparecerá la primera vez que supera el límite.

Por último, si se diera el caso de que se está tratando con una célula en la cual ningún radical supera el límite de presión, la salida del programa será: `Tenemos un problema`

Entrada de ejemplo

```
3 4
0 3
1 1
2 5
0 9
1 7
2 10
3 8
0 2
3 2
2 1
1 2
```

Salida de ejemplo

```
Radical 2
Radical 1
```

Entrada de ejemplo

```
5 2
0 15
1 15
2 12
3 38
4 23
0 2
1 2
1 4
0 2
```

Salida de ejemplo

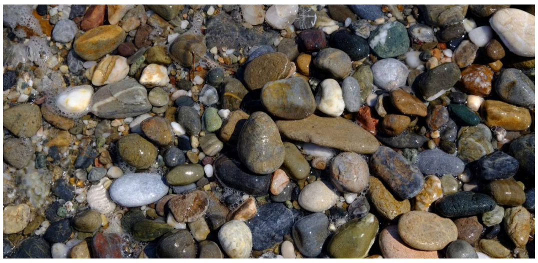
```
Tenemos un problema
```

Límites

- $2 \leq R \leq 30000$
- $2 \leq T \leq 20000$
- $0 \leq R_i \leq R$
- $0 \leq T_i \leq T$
- $2 \leq E_{R_i} \leq 20000$
- $2 \leq E_{T_i} \leq 10^6$

Tiempo: 0.5 segundos

J
El juego de las piedras



Hay días que apetece jugar a algo sencillo, casi metódico, solo por pasar un rato. Para esas ocasiones especiales, siempre tienes el juego de las piedras. Las reglas son simples: se colocan en una superficie plana N montones de M piedras cada uno. Los jugadores, sentados alrededor de la mesa, deben decidir en cada uno de sus turnos si cogen una piedra de cualquiera de los montones, o pasan turno. El jugador que coja la última piedra del último montón, pierde.

Si todos los jugadores intentan ganar, y como máximo cada jugador puede decidir saltar su turno tres veces, ¿podrías predecir quien pierde para una disposición de jugadores dada?

Entrada

La primera linea contiene tres enteros N, M y J, que representan el número de montones de piedras que existen encima de la mesa, el número de piedras que tiene cada montón, y el número de jugadores respectivamente.

Para cada una de las siguientes J líneas, aparece el nombre del jugador correspondiente. Los turnos ocurren en el orden dado en la entrada, de forma circular.

Salida

Se debe imprimir una única línea con el nombre del jugador que crees que pierde la partida.

Entrada de ejemplo

```
2 5 3
Juan
Ana
Ricardo
```

Salida de ejemplo

```
Juan
```

Límites

- $1 \leq N \leq 2^{30}$
- $1 \leq M \leq 2^{30}$
- $1 \leq J \leq 10000$
- $1 \leq len(Nombre) \leq 10$

Tiempo: 0.5 segundos

K
¡A Fortnitear!

Elmorenus,
el nuevo streamer estrella de Twitch, quiere mejorar en Fortnite porque desde que entró en la arena, no queda por encima del puesto 50 en cada partida. Una de sus principales desventajas es que es muy indeciso a la hora de elegir qué armas llevar en cada partida, y mientras piensa qué arma llevar y qué arma dejar suele recibir algún que otro disparo en la cabeza.

Para evitar que esto siga sucediendo, ha decidido pedirnos ayuda para que implementemos un código que permita elegir la combinación óptima de armas en función de las que se ha ido encontrando a lo largo de la partida. Elmorenus quiere que implementemos este software para la siguiente temporada de Fortnite, donde en lugar de tener seis espacios fijos para las armas, dispondremos de una capacidad fija y cada arma ocupará un tamaño diferente. Para saber qué arma es mejor, Elmorenus nos ha hecho un ranking de potencia de cada una de ellas.

El objetivo de Elmorenus es llevar aquellas armas que garanticen una máxima potencia. Hay que tener en cuenta que, en caso de que un arma no quepa en el inventario, en Fortnite van a permitir meter una última arma que ocupe más de lo debido, pero se reducirá su potencia de manera proporcional a lo que se pase de tamaño.

Como buen maniático, dependiendo de la partida, Elmorenus quiere utilizar unas armas u otras, así que nos indicará qué tipo de armas no quiere utilizar bajo ningún concepto en cada partida para evitar cualquier arma cuyo nombre coincida, total o parcialmente, con el del arma que estamos tratando de elegir.

Entrada

La primera linea contiene un entero N que indica el número de armas que podemos elegir. Las siguientes N líneas contienen una cadena C, y dos enteros S y P que indican el nombre del arma, su tamaño en el inventario, y su potencia, respectivamente.

La siguiente línea contiene un entero M que indica el número de situaciones que quiere probar Elmorenus. Las últimas M líneas contienen un posible escenario para Elmorenus, representada por un entero T y una cadena R que indicarán el tamaño máximo del inventario y el tipo de arma prohibida en ese escenario, respectivamente.

Salida

Por cada posible situación, se debe imprimir el potencial total de las armas elegidas con dos decimales de precisión.

Entrada de ejemplo

```
10
PistolaPartepiernas 88 88
EscopetaDestrozaculos 30 86
EscopetaRompecraneos 83 94
EscopetaRompecraneos 10 69
EscopetaDestrozaculos 38 4
LanzagranadasRompecraneos 88 78
ArcoPartepiernas 19 11
PistolaAmputabrazos 96 56
EscopetaPartepiernas 46 30
LanzagranadasRevientacorneas 73 55
8
301 Lanzagranadas
267 Arco
319 Rifle
307 Pistola
298 Rifle
278 Escopeta
289 Lanzagranadas
251 Pistola
```

Salida de ejemplo

```
392.67
386.64
430.07
397.00
414.11
237.92
385.67
357.14
```

Límites

- $10 \leq N \leq 1000$
- $10 \geq |R| \leq |C| \leq 128$
- $1 \leq S, P \leq 100$
- $100 \leq T \leq 10000000$

Tiempo: 0.5 segundos

● L
Los primos traviesos

Los primos Cocho son muy famosos en su pueblo. Hacen travesuras hasta más no poder, sin embargo, para Elvis y Elsan, estas travesuras potencialmente han llegado a su fin. Sus padres han visto sus notas de sus últimas asignaturas presentadas en este cuatrimestre y ¡Madre mía como se han puesto! Ha sido tanto así que su vecino, Sean Cabezas, ha escuchado todo como si le hubiesen gritado a él.

La familia Cocho se ha reunido en la plaza del pueblo y han decidido poner manos a la obra para evitar que estos primos vuelvan a sacar tales calificaciones. Han propuesto a Leo Frentes, un vecino del pueblo que sabe muchísimo de números primos, que les eche una mano a los primos. Lo que no saben, es que Leo, lejos de ser un profesor convencional, decide hacer que los niños colaboren con la comunidad pintando edificaciones.

Frentes entonces les propone un ejercicio a Elvis y Elsan. Les dará los planos en 2D de una hilera de adosados del pueblo. Por cada adosado se conocerá su altura. Entonces, utilizando botes de pintura, los primos empezarán a pintar el adosado, de abajo hacia arriba.

Pero hay truco, Leo ha decidido que tienen que usar el menor número primo que garantice al menos K unidades pintadas teniendo en cuenta todas las casas posibles. Se considera que una casa i tiene K_i unidades pintadas desde la distancia del piso (0) hasta K_i. Además, deben pintar todas las casas hasta la misma altura, si una casa fuese pintada por su totalidad, se considera que solo cubre la altura de dicha casa. Por simplicidad del problema, asumiremos que las casas son referenciadas como líneas en un espacio bidimensional y no tomaremos en cuenta el ancho de la misma para pintar.

Por si fuera poco ya con esto. La manera en la que Leo Frentes les dará a los primos los planos de los adosados será utilizando una función para generar alturas al 'azar' de cada edificación (ya que quiere saber que tan buenos son los primos sacando sus cuentas). Lo dará de tal forma que.

f(X) = (A*X_{n-1} + B) % M, f(0) = B % M

Siendo f(x) la función que se quiere calcular desde 0 hasta $N - 1$ (inclusive).

Entrada

La primera línea contiene un número T de casos de prueba.

Por cada caso de prueba vendrán dos líneas. La primera con dos números N y K, denotando el número de adosados y el número de unidades mínima que Leo Frentes espera cubrir.

A continuación, vendrán 3 números A, B y M denotando cada variable para implementar la función para generar escenarios.

Salida

Por cada caso de prueba se debe imprimir el menor número primo necesario para lograr el objetivo de Leo. En caso que esto no fuese posible, se imprime "IMPOSIBLE"

Entrada de ejemplo

```
3
3 4
1 1 10
3 400
1 1 10
5 87
5 13 2000
```

Salida de ejemplo

```
2
IMPOSIBLE
19
```

En el último caso, las edificaciones generadas tienen altura:
13, 78, 403, 28 y 153. Usando el número primo 19. Tendríamos 13+19*4 unidades =¿99, sin embargo, usando 17, solo tendríamos 13+17*4 =¿81 unidades. Por lo tanto, nos quedamos con 19.

Límites

- $1 \leq N \leq 100,000$
- $1 \leq K \leq 1,000,000,000$
- $1 \leq M \leq 1,000,000$
- $1 \leq A, B \leq 10,000$

Tiempo: 6 segundos

M
Jakub y los cuadrados

Jakub, el famoso científico de la computación, siempre se ha sorprendido al ver la geometría en su alrededor. Por ejemplo, la ventana de su casa es un cuadrado perfecto, el plato donde se sirve su comida es una circunferencia, su cafetera de prensa francesa es similar a un cilindro, etc.

Sin embargo, tal día como hoy, se levanto con malos recuerdos acerca de la clase de geometría que dictaba la profesora Ada. Jakub hoy no quiere saber nada de geometría!

Sin embargo, esta interesado en un concepto un poco menos geométrico relacionado con cuadrados, ya que en su sueños tuvo una conversación con Byron, un gran matemático (Jakub es fan de Byron)

En particular, Jakub esta interesado en contar los números enteros menores o iguales que N tal que su factorización en números primos no contenga ninguna potencia mayor a 1.

Jakub esta triste porque en su sueño, justo cuando Byron le iba a comentar acerca de como resolver este problema, sonó su alarma despertadora (justo a las 3:14 AM. Si, Jakub duerme muy poco)

Jakub necesita tu ayuda para encontrar la respuesta o no podrá dormir mas nunca!

Entrada

La primera linea contiene un entero T que indica el número de casos de prueba.

Cada caso de prueba está formado por una línea que contiene el numero N.

Salida

Para cada caso de prueba T, la salida es el numero de enteros tales que su descomposición en números primos no contengan ninguna potencia mayor a 1.

Entrada de ejemplo

```
3
1
10
1000
```

Salida de ejemplo

```
1
7
608
```

Límites

- $1 <= T <= 100$
- $N <= 10^{14}$