Concurso de Programación
AdaByron 2024
Final nacional

Cuadernillo de problemas

Patrocinado por

Realizado en la **Facultad de Informática (UCM)**
5-6 de julio de 2024

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

6 de julio de 2024

Listado de problemas

Autores y revisores de los problemas:

- Margarita Capretto (IMDEA Software)
- Catalin Covaci (Universidad Politécnica de Madrid)
- Francisco Criado (CUNEF Universidad)
- Marco Antonio Gómez Martín (Universidad Complutense de Madrid)
- Pedro Pablo Gómez Martín (Universidad Complutense de Madrid)
- Alberto Maurel Serrano (Universidad Complutense de Madrid)
- Manuel Montenegro Montes (Universidad Complutense de Madrid)
- Isabel Pita Andreu (Universidad Complutense de Madrid)
- Rubén Rubio (Universidad Complutense de Madrid)
- Diego Seco (Universidade da Coruña)
- Alberto Verdejo (Universidad Complutense de Madrid)

A
Décimo aniversario

Parece que fue ayer cuando la "*Unión de Concursantes Mayores*" organizó su primer concurso de programación para gente joven. Ha pasado ya tanto tiempo, de hecho, que algunos de esos jóvenes que participaron en las primeras ediciones son ahora también mayores y ayudan en la organización.

Para el décimo aniversario habían pensado hacer un evento extraordinario digno de ser recordado para la posteridad. El número de días dedicados al concurso y el número de equipos admitidos estaría únicamente limitado por las plazas hoteleras de la ciudad.

Para decidir las fechas y número de equipos, el primer paso fue llamar a todos los alojamientos de la ciudad para saber, para cada día dentro de las fechas de celebración posibles, el número de equipos que podrían alojarse en ellos. Tras eso, el objetivo era decidir cuántos equipos admitir y qué días consecutivos celebrarlo teniendo en cuenta que serían al menos dos días y dos equipos y que el número de plazas hoteleras durante todos los días de celebración debía ser suficiente para que todos pudieran dormir.

Y ahí fue donde se vino abajo todo el plan. Con tantos días posibles y tantas plazas hoteleras disponibles, ni siquiera fueron capaces de calcular cuántas posibilidades de elección tenían. Al final decidieron organizar el concurso como el año anterior y esperar al vigésimo aniversario para hacer el evento memorable.

Entrada

La entrada está compuesta de distintos casos de prueba, cada uno ocupando dos líneas. La primera línea contiene el número de días (consecutivos) de los que se tienen los datos de disponibilidad hotelera (entre 1 y 500.000). La línea siguiente contiene, para cada uno de esos días, el número de equipos que podrían alojarse en los hoteles de la ciudad ese día (entre 0 y 10^6).

Después del último caso de prueba aparece una línea con un `0` que no debe procesarse.

Salida

Por cada caso de prueba se escribirá un único número indicando el número de configuraciones distintas posibles del concurso teniendo en cuenta que debe durar un número consecutivo de días (al menos dos), que se necesitan al menos dos equipos para celebrar el concurso y que debe haber plazas suficientes para alojar a todos los equipos mientras el concurso se esté celebrando. Se entiende que dos configuraciones son distintas si el día de comienzo o el día de fin es distinto o si el número de equipos admitidos varía.

Como el número total puede ser muy grande, se debe dar la respuesta módulo 1.000.000.007.

Entrada de ejemplo

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

Salida de ejemplo

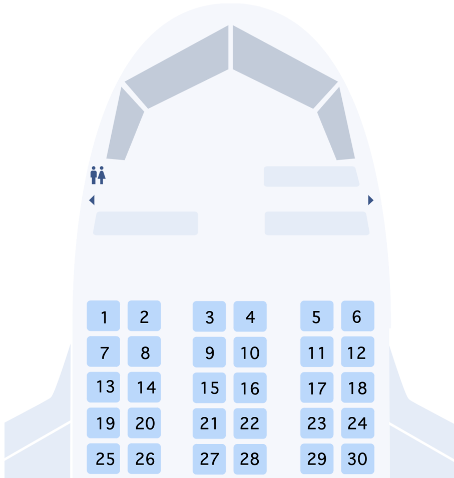
```
3
7
```

Notas

En el primer caso de prueba hay dos posibilidades de concurso de dos días con dos o más equipos admitidos y una configuración con tres días de duración.

En el segundo caso de prueba hay seis formas de organizar concursos con dos equipos: tres posibles concursos de dos días, dos concursos de tres días y un concurso de cuatro días. Además, se podría organizar un concurso con tres equipos y dos días de duración. En total, siete elecciones posibles.

B
¡Pero qué caradura!



Cuando viajo en avión me ocurre con cierta frecuencia que un pasajero se acerca a mí antes de que comience el vuelo y me pide muy amablemente si puedo cambiarle el asiento ya que es muy amigo del pasajero que se sienta a mi lado y le gustaría realizar el viaje con él. A continuación, el pasajero que se sienta a mi lado, con una amplia sonrisa se esfuerza en explicarme que no pudieron reservar los asientos juntos debido a una serie de problemas que nunca alcanzo a comprender. En esta situación es difícil negarse al intercambio, y luciendo mi mejor sonrisa, accedo a cambiar de asiento y recojo mis pertenencias.

Cuando pregunto al pasajero que se acercó a mí dónde se encuentra el suyo para realizar el cambio, quedo asombrada cuando afirma que se encuentra en la última fila del avión. Es el momento en que la sonrisa se trunca en una mueca extraña. ¡Vaya caradura! Quiere que abandone mi posición en la parte delantera del avión y vaya atrás, donde el ruido de los motores no me dejará conciliar el sueño.

Afortunadamente ya conozco la jugada y reacciono a tiempo. "¿Qué tal si en lugar de ser yo la que se vaya a la parte de atrás, es tu amigo — digo señalando a la persona que tengo a mi lado — el que lo hace? Seguro que tu compañero de fila que ahora mismo está solo detrás preguntándose para qué te has levantado está encantado de venirse a mi lado y ganar tranquilidad".

En el vuelo que nos ocupa, hay muchos pasajeros que quieren cambiar de asiento. Para que no terminemos todos de pie por los pasillos, la tripulación ha puesto orden y vamos a ir haciendo las solicitudes en secuencia: se levantará una persona y se dirigirá a otra para proponerle el cambio; tras hacerlo se procede con la siguiente solicitud y así hasta terminarlas todas. Los cambios de asientos siempre tienen lugar aunque unas veces se intercambiarán los implicados en la solicitud (si el que la recibe no tiene que ir hacia atrás en el avión) y otras veces serán los pasajeros que ocupan sus butacas contiguas los que se intercambiarán.

La disposición de los asientos en el avión es siempre la misma. Hay 6 asientos en cada fila, colocados de dos en dos y separados por dos pasillos. El número de filas varía de un vuelo a otro. Los asientos se numeran consecutivamente por cada fila, empezando por la fila uno. Así los asientos de la primera fila se numeran del 1 al 6 empezando por un extremo. Los de la segunda fila son los asientos del 7 al 12, estando el asiento 7 detrás del 1, y así sucesivamente hasta completar el avión. Los pasajeros se numeran inicialmente por el asiento que ocupan. Luego no vuelven a cambiar de número aunque cambien de asiento.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso comienza con una línea con dos valores. El primero indica el número de filas, F, del avión ($1 \leq F \leq 30$) y el segundo el número de cambios que se producen (es posible que no se produzca ningún cambio). Por cada cambio aparece en una línea el número de asiento del pasajero que pide el cambio y el número de asiento del pasajero al que le pide el cambio. La entrada termina con dos ceros.

Un pasajero nunca pide cambio a sí mismo, ni al pasajero a su lado. Además, una vez que una pareja de conocidos ha conseguido sentarse junta no volverá a pedir cambios ni nadie se los pedirá (todos los pasajeros tienen como mucho un amigo en el avión).

Salida

Para cada caso de prueba se escribirá en F líneas la disposición de los pasajeros una vez realizados todos los cambios. En cada línea se muestra una fila del avión. Después de la descripción de cada avión se escribirá una línea con tres guiones.

Entrada de ejemplo

```
2 2
1 8
10 4
3 3
9 5
1 18
11 7
2 2
9 5
11 5
0 0
```

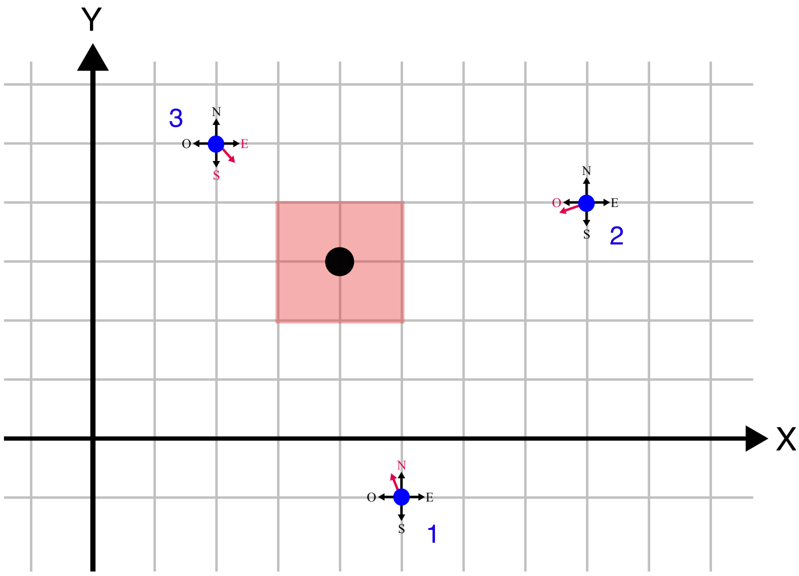
Salida de ejemplo

```
8 2 9 4 5 6
7 1 3 10 11 12
---
18 2 3 4 5 10
11 8 9 6 7 12
13 14 15 16 17 1
---
1 2 3 4 5 12
7 8 9 6 11 10
---
```

C
El enigma de la singularidad

Durante varias décadas, los científicos de Arenópolis llevan estudiando un fenómeno ajeno a las leyes de la física, al que llaman *singularidad*. Saben que la fuente de este fenómeno está en algún lugar remoto del desierto, pero no saben exactamente sus coordenadas. Algunas personas valientes han iniciado una expedición para acercarse a la singularidad, pero ninguna lo ha conseguido. Según cuentan los científicos, alrededor de este fenómeno hay una zona de exclusión a la que es peligroso acercarse, porque la simple presencia de elementos externos en esta zona puede afectar a la singularidad de manera completamente impredecible. Aunque no se conoce el tamaño de la zona de exclusión, sí se sabe que esta zona tiene forma de cuadrado, en cuyo centro está la singularidad, y cuyos lados están alineados con los ejes de coordenadas que los científicos utilizan como referencia para navegar por el desierto. Por convenio, el eje Y apunta hacia el norte, y el eje X apunta hacia el este. El perímetro del cuadrado está incluido en la zona de exclusión.

La singularidad emite un potente campo magnético. Aun así, por desgracia, las brújulas tradicionales no pueden apuntar con exactitud al lugar de la singularidad, porque el campo tiene muchas oscilaciones. No obstante, los científicos de Arenópolis han desarrollado la *brújula discreta* que, aunque no apunta exactamente al sitio de la singularidad, sí que indica el punto cardinal más cercano (`N`, `S`, `E`, `O`) a la dirección en la que se encuentra. Eso sí, si la dirección está exactamente a medio camino de dos puntos cardinales, la brújula discreta no sabe decidirse, y muestra la palabra `IND` (*indefinido*). Por ejemplo, en la siguiente figura, el círculo negro representa la singularidad, el cuadrado rojo su zona de exclusión, y los puntos azules son los lugares en los que se ha utilizado la brújula discreta.



Desde el lugar marcado con 1, el punto cardinal más cercano a la dirección de la singularidad es el norte. Por tanto, la brújula discreta marcaría `N` (Norte). Desde el lugar 2, la brújula discreta marcaría `O` (Oeste). Por último, desde el lugar 3 la brújula marcaría `IND`, ya que la dirección equidista de los puntos cardinales Este y Sur.

Para ayudar a los científicos, debes escribir un programa que determine las coordenadas de la singularidad. El programa debe leer varios casos de prueba. En cada uno de ellos, la singularidad se encuentra en un punto distinto y el tamaño de su zona de exclusión también es distinto. Al inicio de cada caso, el programa debe leer dos números enteros x e y que indican las coordenadas de Arenópolis, la cual siempre está fuera de la zona de exclusión. Después, el programa podrá enviar equipos de expedición a distintos puntos del desierto para que utilicen la brújula discreta desde allí. Para cada punto, el programa debe escribir `?` seguido de las coordenadas desde las que se utilizará la brújula discreta. A continuación podrá leer de la entrada estándar el resultado que se muestra en la brújula discreta cuando se utiliza en ese punto (`N`, `S`, `E`, `O`, `IND`). Cuando el programa conozca el lugar exacto de la singularidad, debe escribir una línea con el texto `Respuesta: X Y`, donde `X` e `Y` son las coordenadas de la singularidad, y pasará al siguiente caso de prueba. El programa finaliza cuando no hay más casos de prueba que procesar.

Ten cuidado, porque no puedes enviar equipos de expedición a la zona de exclusión que rodea la singularidad. Si lo hicieses, la brújula podría desestabilizar a la singularidad hasta hacerla desaparecer del desierto, de modo que ya no tendría sentido investigar sobre ella, aunque conocieses su situación.

A continuación aparece un ejemplo de ejecución. En cursiva aparece lo leído por la entrada estándar y en negrita lo escrito por la salida estándar.

2 1
? 5 -1
N
? 8 4
O
? 2 5
IND
? 7 6
IND
Respuesta: 4 3

Las coordenadas de la singularidad y de Arenópolis son números enteros comprendidos entre -10^8 y 10^8. El desierto es aún más vasto, y es posible enviar equipos de expedición a zonas cuyas coordenadas estén en el intervalo [-10^9, 10^9]. En cada ejecución, el número de casos de prueba es, como mucho, 700.

Notas

Al tratarse de un problema *interactivo* es importante que cada vez que tu solución escriba algo que el juez deba leer se asegure de volcar la salida (usando terminología inglesa, haga "*flush*"). La forma de hacerlo varía entre lenguajes. En los admitidos en la competición puede hacerse con:

- C++: `cout << endl;` o `cout << flush;`
- C: `fflush(stdout);`
- Java: `System.out.flush();`
- Python: `print(..., flush=True)` o `sys.stdout.flush()`

El envío de equipos de expedición a la zona de exclusión o a coordenadas que están más allá de los límites del desierto provocará que el juez devuelva un veredicto `WRONG-ANSWER`.

D
El jardín de las palabras

En clase de lengua, Hannah ha aprendido a clasificar palabras por distintos criterios: según su morfología (sustantivos, adjetivos, artículos, verbos…), según su acentuación (agudas, llanas, esdrújulas, sobreesdrújulas), según su estructura (simples, derivadas, compuestas, parasintéticas), según su significado (léxicas, gramaticales, monosémicas, polisémicas), según su número de sílabas (monosílabas, bisílabas, trisílabas, polisílabas), etc.

Hannah cree que hay otras formas curiosas de clasificar palabras. Ha estado mirando el diccionario de la lengua española y tiene curiosidad por un tipo especial de palabras, a las que ha bautizado convenientemente como *equiliterales*. Una palabra *equiliteral* es aquella en la que todas sus letras aparecen el mismo número de veces. Ha conseguido en formato digital la lista de palabras de una versión antigua del diccionario y tras quitar las palabras conjugadas y eliminar las tildes ha extraído los siguientes datos:

- Existen 16.446 palabras equiliterales donde cada letra distinta aparece una vez. Las más largas son de doce letras: *vomipurgante*, *culteranismo*, *calumbriento*, *gesticulador*, *chupaderitos*, *pregustacion*, *conquistable*, *translucidez*, *centrifugado*. Curiosamente, también estaría entre las más largas la palabra compuesta *reality show*, si consideráramos el espacio como una "letra" más.
- Existen 144 palabras equiliterales donde cada letra distinta aparece dos veces. Las más largas también son de doce letras: *glaciologica*, *condicionada*, *quisquilloso*. Algunas otras de diez letras son: *furrufalla*, *quinaquina*, *pingopingo*, *zambombazo*.
- Existen 4 palabras equiliterales donde cada letra distinta aparece tres veces. Estas cuatro palabras son: *blablabla*, *chachacha*, *ayayay*, *lilili*.
- No existen palabras equiliterales donde cada letra distinta aparezca cuatro veces o más.

A partir de estas ocurrencias a Hannah los resultados le parecen bastante aburridos, así que ha decidido incluir también sus propias palabras inventadas. Para cada una, debes decir si es o no una palabra equiliteral.

Entrada

La entrada comienza con un número indicando cuántos casos de prueba deberán procesarse (no más de 1.000).

Cada caso de prueba ocupa una única línea y contiene una palabra de N caracteres ($1 \leq N \leq 50$). Los caracteres se componen de letras minúsculas del alfabeto inglés, de la 'a' a la 'z', sin espacios, tildes ni eñes.

Salida

Para cada caso de prueba se escribirá una sola línea indicando `SI`, si se trata de una palabra equiliteral, y `NO` en caso contrario.

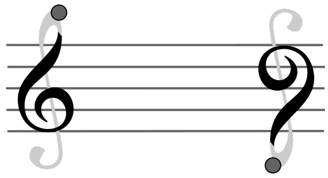
Entrada de ejemplo

```
5
hannah
supercalifragilisticoespialidoso
esporospredanidina
xujxepperr
jajajajajajajajaja
```

Salida de ejemplo

```
SI
NO
SI
NO
SI
```

E
La séptima de Schubert



En 1813, con solo 16 años, Schubert escribió su primera sinfonía, a la que siguieron otras 5, la última de ellas compuesta en 1818.

A partir de ese momento, la historia se complica. Schubert comenzó bocetos de varias sinfonías, muchas de las cuales no llegó a terminar. A final de 1822 había compuesto dos movimientos de una nueva sinfonía y esbozado el tercero en una reducción para piano pero, tras enfermar de sífilis, la abandonó. Al año siguiente envió la partitura a Anselm Hüttenbrenner como obsequio, quien la guardó en un cajón y la olvidó.

En 1825, Schubert, recuperado ya de su enfermedad, comenzó una nueva sinfonía que concluyó al año siguiente. Apodada como "*La grande*" por su longitud, se etiquetó como la séptima, tras las seis de su juventud. Pero cuando mucho después, en 1865, Hüttenbrenner desempolvó la partitura que recibió de Schubert cuatro décadas antes, surgió el conflicto. Si la numeraban como la octava, las etiquetas no serían conformes al orden de composición. Pero si la colocaban como la séptima y movían "*La grande*" al número 8, se generaría una ambigüedad al referirse a la séptima, por ser un término usado para dos sinfonías distintas a lo largo del tiempo. La decisión fue *mover* "*La grande*" al número 9 y dejar un hueco con la séptima[1].

Número	1	2	3	4	5	6	7	8	9
Año composición	1813	1815	1815	1816	1816	1818	~~1826~~	1822	1826
Sobrenombre				Trágica		Pequeña	~~Grande~~	Incompleta	Grande

Entrada

Cada caso de prueba comienza con un número $1 \leq n \leq 200.000$ indicando el número de sinfonías escritas por un compositor. En la línea siguiente aparecen n números con una permutación de los números del 1 al n indicando el orden de descubrimiento de cada sinfonía, donde cada una se referencia por el orden de composición. Por ejemplo, si el primer número es un `4`, significa que la primera sinfonía descubierta (y catalogada) fue la cuarta en ser compuesta.

A continuación se proporciona un número $1 \leq q \leq 20.000$ indicando la cantidad de consultas al catálogo que se realizan. Después aparecen, en una única línea, todas esas consultas.

Cada consulta es un número de sinfonía del catálogo de la que se quiere saber el orden de composición. Se garantiza que los números estarán en orden de menor a mayor y serán valores entre 1 y U, donde U es el número asignado a la última sinfonía compuesta por el autor, sabiendo que cada vez que se descubre una sinfonía en desorden con respecto al orden de composición, se renumeran aquellas mal numeradas dejando huecos.

La entrada termina con un caso de un compositor sin sinfonías, que no debe procesarse.

Salida

Por cada caso de prueba se resolverán las diferentes consultas. Cada una pregunta por una sinfonía dando el número en el catálogo, y hay que indicar *su orden de composición*, K. Si el número fue "abandonado" por desplazar la sinfonía posteriormente, se escribirá `Fue la` *K*. Si es su posición definitiva se escribirá `Es la` *K*.

Después de cada caso de prueba se escribirá una línea con tres guiones (`---`).

[1]No todo el mundo aceptó esa renumeración y, por si fuera poco, algunos de los que sí lo hicieron aprovecharon el hueco de la séptima para una nueva sinfonía, todavía más inacabada, que Schubert dejó inconclusa en 1821. Hoy, no siempre es fácil saber a qué sinfonía se refiere un número en el catálogo sinfónico de Schubert.

Entrada de ejemplo

```
8
1 2 3 4 5 6 8 7
4
6 7 8 9
4
4 3 2 1
4
1 3 6 10
0
```

Salida de ejemplo

```
Es la 6
Fue la 8
Es la 7
Es la 8
---
Fue la 4
Fue la 4
Fue la 4
Es la 4
---
```

Notas

El primer ejemplo representa las sinfonías de Schubert. Se considera que escribió 8 sinfonías. Las 6 primeras se descubrieron y numeraron desde el principio de acuerdo al orden de composición, pero luego se descubrió la octava en ser compuesta ("*La grande*") y finalmente la compuesta en séptimo lugar ("*La incompleta*").

"*La grande*", compuesta en octavo lugar, tuvo asignado dentro del catálogo el número 7, de ahí que a la consulta `7` se conteste `Fue la 8`. Tras descubrirse "*La incompleta*", compuesta en séptimo lugar, se renumeraron. El número 7 dejó de usarse, a "*La incompleta*" (compuesta en séptimo lugar) se le asignó el número 8 y "*La grande*", compuesta en octavo lugar, se recolocó en el número 9. Es por esto por lo que a la consulta `8` se contesta `Es la 7` y a la consulta `9` se contesta `Es la 8`.

F
Gran Torneo de Sabiduría Arcana

¡Bienvenido al Gran Torneo de Sabiduría Arcana, el evento más esperado del año en el reino de los sabios! Este prestigioso torneo reúne a N de los más eruditos hechiceros de todas las tierras. Cada hechicero, al iniciar el torneo, posee un nivel inicial de sabiduría arcana, acumulado a lo largo de años de estudio y práctica de las artes mágicas.

El Gran Torneo de Sabiduría Arcana sigue un formato de eliminación simple con las siguientes reglas especiales:

- Habrá un total de $N-1$ duelos. En cada duelo, dos hechiceros cualesquiera (que no hayan sido eliminados previamente) se enfrentan en un concurso de conocimientos y habilidades mágicas.
- Tras una serie de pruebas, que pretenden dar una sensación de imparcialidad y justicia al duelo, en realidad siempre es declarado ganador del duelo el hechicero con el mayor nivel de sabiduría. En caso de empate, el ganador puede ser elegido de manera arbitraria por el Gran Consejo de Magos.
- Para reflejar la absorción del conocimiento y experiencia del adversario, el ganador del duelo no solo sigue en la competición, sino que también adquiere el nivel de sabiduría arcana del perdedor, sumándolo a su propio nivel.
- El perdedor del duelo es eliminado del torneo y no participa en duelos futuros.
- Después de $N-1$ duelos, solo quedará un hechicero, quien será coronado como el Gran Archimago del Torneo de Sabiduría Arcana.

Como miembro del honorable Gran Consejo de Magos, tienes la misión de hacer que el torneo sea lo más emocionante posible. Sabes que al público le encantan las historias de magos inesperados que logran la victoria contra todo pronóstico. Por ello, quieres determinar cuál es el hechicero con el menor nivel de sabiduría arcana inicial que puede, con una estrategia óptima de duelos, terminar ganando el torneo.

Entrada

La entrada consta de varios casos de prueba. En la primera línea se indica el número de casos de prueba que vendrán a continuación, ocupando cada caso dos líneas.

En la primera línea de cada caso se indica el número de hechiceros N ($1 \leq N \leq 300.000$) que participan en el torneo. En la segunda línea aparecen N enteros s_i ($1 \leq s_i \leq 10^{12}$), todos distintos, indicando las sabidurías iniciales de cada hechicero, antes de comenzar el torneo.

Salida

Para cada caso de prueba se escribirá una línea con un solo número, indicando la *posición* en la entrada del hechicero con una sabiduría inicial menor que puede ganar el torneo, si puedes elegir tú los contrincantes de cada duelo, quién es el ganador en caso de empate y el orden de los duelos.

Entrada de ejemplo

```
3
1
3
2
2 1
4
1 4 6 2
```

Salida de ejemplo

```
1
1
2
```

G
El profesor Malvado

El profesor Gabriel Malvado es famoso en nuestro instituto, pero no por sus habilidades didácticas, sino por su reputación como maestro despiadado. Es temido por su costumbre de hacer exámenes sorpresa cada poco tiempo. Su clase es un campo de batalla donde solo los más listos sobreviven, y el sufrimiento forma parte del proceso de aprendizaje.

El otro día publicó una hoja con las notas de clase. La calificación final se calcula a partir de la media ponderada de las notas de los tres exámenes que hemos hecho a lo largo del curso. Cada examen se puntúa de 0 a 10 (sin decimales), y su peso en la nota final está determinado por un porcentaje. El peso del primer examen era del 40 % y el de los otros dos era de un 30 % cada uno. Los resultados fueron los siguientes:

Nombre	**Examen 1** (Peso: 40 %)	**Examen 2** (Peso: 30 %)	**Examen 3** (Peso: 30 %)	**TOTAL** (Media ponderada)
Lucas	8	10	9	8.9
Iván	4	8	5	5.5
Martín	6	5	4	5.1
Adrián	6	3	2	3.9

Con estos resultados, la calificación media de toda la clase era 5.85. Estábamos contentos, pero la alegría duró poco. El profesor Malvado había decidido cambiar a posteriori los pesos de los exámenes a la hora de calcular la nota final, de modo que pasaban a ser del 30, 20 y 50 % respectivamente. Con ello consiguió que la nota media global de la clase bajase hasta 5.60. De hecho, no tardamos en darnos cuenta de que esa elección de porcentajes tenía como objetivo *minimizar* la media global de la clase.

Por suerte, el profesor no podía manipular los pesos a su antojo, porque el departamento impone unas horquillas de porcentajes mínimos y máximos que puede tener un examen para su ponderación. Estas horquillas dependen del tipo de examen. Por ejemplo, el peso del primer examen debía estar en el intervalo 20−50 %, el del segundo en el intervalo 20−40 % y el tercero en el intervalo 30−50 %. Aun así, el profesor se las arregló para amargar a sus estudiantes lo máximo posible, pero respetando en todo caso las horquillas de evaluación. ¿Cómo lo haría?

Entrada

La entrada se compone de varios casos de prueba. Cada caso de prueba comienza con una línea con el número N de estudiantes y el número M de exámenes realizados ($1 \leq N \leq 10^4$, $1 \leq M \leq 100$). Después vienen N líneas, una por cada alumno. Cada línea contiene M números, que son las calificaciones obtenidas por ese estudiante en cada examen. Cada calificación es un número entero entre 0 y 10. Por último, siguen M líneas, cada una con dos números X_i, Y_i ($1 \leq X_i \leq Y_i \leq 100$) que determinan la horquilla de porcentajes en la que puede estar la ponderación de cada uno de los ejercicios. Se garantiza que las horquillas permiten que los pesos de los exámenes sumen 100 %.

Salida

Para cada caso de prueba debe imprimirse una línea con M números enteros, que son los pesos que el profesor Malvado ha de asignar a los exámenes para minimizar la calificación media total de clase. Estos pesos han de estar dentro de las horquillas determinadas por la entrada y han de sumar 100.

En caso de que haya varias asignaciones de pesos que lleven a la misma calificación mínima, se escogerá aquella que sea mayor según un orden lexicográfico. Es decir, aquella que otorgue un mayor peso al primer examen, y en caso de empate, la que dé más peso al segundo examen, y así sucesivamente.

Entrada de ejemplo

```
4 3
8 10 9
4 8 5
6 5 4
6 3 2
20 50
20 40
30 50
1 4
7 9 8 4
20 40
20 40
30 70
10 25
```

Salida de ejemplo

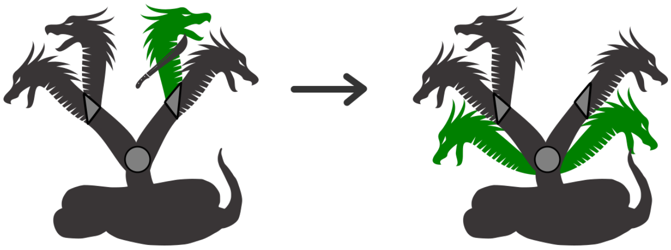
```
30 20 50
25 20 30 25
```

H
Trhidra

Como penitencia por el asesinato, en un ataque de locura, de su mujer, hijos y dos sobrinos, Hércules tuvo que cumplir los que se conocen como "los doce trabajos de Hércules". El segundo de ellos fue matar a la hidra de Lerna, un monstruo acuático con forma de serpiente y nueve cabezas de dragón. Cada vez que Hércules conseguía cortar una cabeza, se regeneraban dos más. Consiguió vencer al monstruo con la ayuda de su sobrino Yolao, quien con una antorcha quemaba el cuello de la hidra tras el corte de cada cabeza para evitar el surgimiento de las dos nuevas.

Un subgénero de las hidras, quizá más espantoso, son las *trhidras*. Del cuerpo de la serpiente surgen cuellos que, a veces, no acaban en cabezas, sino que antes se ramifican, incluso varias veces. Cuando una cabeza es cortada, surgen las réplicas, igual que en las hidras, pero no en el mismo punto donde estaba la recién cortada, sino en la ramificación inmediatamente anterior del cuello.

Por ejemplo, si del cuerpo de una determinada *trhidra* salen dos cuellos, cada uno con una ramificación de la que surgen dos cabezas, al cortar una cabeza surgen dos nuevas en la ramificación anterior que, en este caso, resulta ser el cuerpo.



Cuando se eliminan todas las cabezas de un punto de ramificación, este se convierte en una cabeza que debe también ser cortada, lo que ocasiona el surgimiento de las réplicas un nivel por debajo.

Pese a la fiereza de la *trhidra*, terminar con todas sus cabezas es más sencillo que en el caso de las hidras, porque las cabezas que surgen directamente del cuerpo, y no de un cuello ramificado, no se regeneran, al estar en el primer nivel de ramificación y no haber uno anterior. Cuando todas las cabezas que surgen del cuerpo son cortadas, la *trhidra* muere.

Entrada

Cada caso de prueba comienza con un primer número, entre 1 y 10, que indica el poder de replicación de una *trhidra*, es decir el número de cabezas que surgen, un nivel de ramificación por debajo, cada vez que se corta una cabeza.

En la segunda línea de cada caso de prueba aparecen entre 2 y 10.000 números con la descripción de la *trhidra*. Primero se indica la ramificación del cuerpo (número de cuellos principales) y a continuación aparece la descripción, siguiendo este mismo esquema, de cada uno de ellos. Un cuello sin ramificación se indica con un 0, que terminará en una cabeza. Ningún cuello tiene una ramificación mayor que 10.

A veces la *trhidra* ha pasado por batallas anteriores y tiene algunas cabezas ya cortadas, por lo que algún cuello podría tener una ramificación de 1.

La entrada termina con un 0.

Salida

Por cada caso de prueba se escribirá el número total de cabezas que tiene que cortar Hércules para conseguir matar a la *trhidra*, teniendo en cuenta que las va cortando de una en una. Dado que el número puede ser alto, se dará el valor módulo 1.000.000.007.

Entrada de ejemplo

```
2
2 2 0 0 2 0 0
2
3 3 0 0 0 3 0 0 0 3 0 0 0
3
1 1 1 0
0
```

Salida de ejemplo

```
14
30
18
```

● I
Edificios bonitos

Desde chiquitita la pasión de Vera ha sido siempre la fotografía. Aún recuerda aquellos viajes de su infancia con su familia cuando iban a visitar países cercanos con sus padres. Mientras su hermana mayor se quejaba de tener que "patear" calles y más calles, ella disfrutaba planificando encuadres y configurando los ajustes de blancos de su móvil heredado para conseguir la instantánea perfecta de cada edificio histórico con el que se cruzaban.

A nadie le extrañó que pasados los años decidiera dedicarse a la fotografía de forma profesional. Ahora ¡cobra por sus fotos! Ha tenido la suerte de caer en una empresa que paga sus viajes a lugares exóticos para hacer fotos con las que elaborar guías turísticas, catálogos de viajes y páginas web. Eso sí, los tiempos de paseos relajados por las ciudades se acabaron; ahora no tiene tiempo que perder y cuando visita una ciudad planifica todas sus rutas para conseguir hacer el mayor número de fotos en el menor tiempo posible.

Entrada

La entrada está compuesta por distintos casos de prueba, cada uno representando una ciudad y las rutas que quiere planificar.

En la primera línea aparece el número de intersecciones n ($2 \leq n \leq 10.000$), y de calles de la ciudad m ($2 \leq m \leq 100.000$). A continuación aparecen n líneas describiendo las calles. Cada una contiene dos números distintos con los identificadores de las intersecciones (números entre 1 y n), el tiempo que se tarda en recorrer las calles (número entre 1 y 10.000) y la cantidad de edificios bonitos que se pueden fotografiar en esa calle (hasta 10.000). Todas las calles son de doble sentido y nunca hay más de una calle entre dos intersecciones.

Tras la descripción de la ciudad aparecerá un número q con el número de rutas que quiere planificar Vera (no más de 100). Por cada ruta habrá una línea con dos números distintos indicando la intersección origen y la intersección destino de la ruta. Se garantiza que existe al menos un camino que une ambos puntos.

Salida

Por cada ruta a planificar se escribirá la ratio *edificios_bonitos* / *tiempo_total* más grande que puede conseguirse en una ruta que una ambos puntos, con seis decimales de precisión.

Ten en cuenta que las distintas rutas son independientes por lo que la misma calle podría aparecer en más de una. Eso sí, los pasos que se dan en cada ruta *deben siempre acercar a Vera al destino*: puede no ir por el camino más corto si eso mejora la relación de edificios bonitos fotografiados por unidad de tiempo, pero si en un momento dado estuvo a un tiempo mínimo X de llegar al objetivo, no podrá ir a una intersección que no haga que ese tiempo mínimo decrezca.

Tras cada caso de prueba se escribirá una línea con tres guiones, `---`.

Entrada de ejemplo

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

Salida de ejemplo

```
2.333333
1.333333
---
```

● J
Misión de rescate: pantalla defectuosa

¡Atención, cadetes de la Flota Estelar! Durante una misión de rutina en el espacio profundo, la nave *SS Codebreaker*, una de las naves más avanzadas de la Flota, equipada con tecnología de última generación para explorar los confines del universo y descifrar los misterios que guarda, ha sufrido un fallo en su sistema de visualización.

La pantalla principal de la nave, que utiliza varios displays de siete segmentos para mostrar información crucial sobre la navegación y el estado de los sistemas, como coordenadas espaciales y códigos de acceso, ha comenzado a fallar debido a una tormenta de radiación cósmica que ha causado estragos en los sistemas electrónicos de la nave. Algunos de los píxeles en estos displays han comenzado a funcionar de forma errónea, haciendo difícil la interpretación de los datos.

Debido a la importancia de estos, es vital determinar cuántos valores posibles pueden estar representados en la pantalla defectuosa. La misión y la seguridad de la tripulación dependen de una interpretación correcta de estos dígitos.

En condiciones normales, los dígitos se muestran de la siguiente manera:

```
...... ...... ...... ...... ...... ...... ...... ...... ...... ......
..##.. ...... ..##.. ..##.. ...... ..##.. ..##.. ..##.. ..##.. ..##..
.#..#. ....#. ....#. ....#. .#..#. .#.... .#.... ....#. .#..#. .#..#.
.#..#. ....#. ....#. ....#. .#..#. .#.... .#.... ....#. .#..#. .#..#.
...... ...... ..##.. ..##.. ..##.. ..##.. ..##.. ...... ..##.. ..##..
.#..#. ....#. .#.... ....#. ....#. ....#. .#..#. ....#. .#..#. ....#.
.#..#. ....#. .#.... ....#. ....#. ....#. .#..#. ....#. .#..#. ....#.
..##.. ...... ..##.. ..##.. ...... ..##.. ..##.. ...... ..##.. ..##..
...... ...... ...... ...... ...... ...... ...... ...... ...... ......
```

Debido al comportamiento defectuoso, algunos de los píxeles que forman los segmentos titilan, sin que podamos saber si deberían estar encendidos o apagados. Por ejemplo, la pantalla podría mostrar un código, en este caso de tres dígitos, de la siguiente manera (donde aparecen marcados con una X con fondo rojo los píxeles titilantes):

```
..................
..##....X#....XX..
....#..#..#.....#.
....#..#..X.....#.
..##..............
.X..X..#..#.....#.
.X..X..#..#.....#.
..##....##....XX..
..................
```

El primer dígito que muestra la pantalla podría ser tanto un 2 como un 3; el segundo dígito, aunque tiene también píxeles defectuosos, es inequívocamente un 0; y el tercer dígito podría ser tanto un 1 como un 7. Por lo tanto, el código mostrado por la pantalla podría ser 201, 207, 301 ó 307.

Viendo un código mostrado por la pantalla defectuosa, ¿sabrías decir cuántos códigos diferentes podría estar representando?

Entrada

La entrada consta de una serie de casos de prueba (nunca más de 1.000). Cada caso comienza con una línea con el número N de dígitos mostrados por la pantalla ($1 \leq N \leq 9$). A continuación aparecen 9 líneas de caracteres, cada una con $6 \times N$ caracteres, los correspondientes a representar los N dígitos. El carácter '`.`' representa un píxel apagado, el carácter '`#`' representa un píxel encendido y el carácter '`X`' representa un píxel titilante, que no sabemos si debería estar encendido o apagado. Solamente los píxeles pertenecientes a alguno de los siete segmentos puede estar encendido o ser titilante. Llamamos segmento a cada uno de los siete grupos de dos caracteres adyacentes en vertical u horizontal que en la representación del 8 se indican con caracteres '#'.

La entrada termina con un 0 que no debe procesarse.

Salida

Para cada caso de prueba se escribirá el número total de códigos que puede estar representando la pantalla defectuosa.

Entrada de ejemplo

```
3
..................
..##....X#....XX..
....#..#..#.....#.
....#..#..X.....#.
..##..............
.X..X..#..#.....#.
.X..X..#..#.....#.
..##....##....XX..
..................
1
......
..XX..
.X..X.
.X..X.
..##..
.X..X.
.X..X.
..XX..
......
0
```

Salida de ejemplo

```
4
7
```

K
Le championnat n'a plus de sens

Scrabble es un juego de mesa en el que, dadas ciertas letras, los jugadores intentan construir palabras en un tablero. Cada letra tiene una puntuación predefinida, y cuantas más letras con mayor puntuación se usen, mayor será la puntuación total obtenida.

Nigel Richards, nacido en Nueva Zelanda en 1967, es comúnmente considerado el mejor jugador de *Scrabble* de todos los tiempos. Desde el comienzo de su carrera competitiva, ha ganado aproximadamente el 75% de sus partidas en torneos, y ha cosechado infinidad de títulos. Además, al contrario que en otros juegos en los que los algoritmos y la inteligencia artificial ya pueden ganar al ser humano, en *Scrabble* es muy común que en la mayoría de análisis de partidas la gente confíe más en las jugadas de Nigel que en las de una máquina.

Sin duda, uno de sus logros más curiosos es que en 2015 ganó el campeonato del mundo de *Scrabble* en francés... sin saber francés. Según sus propias declaraciones, pasó unas nueve semanas estudiando y memorizando las palabras del diccionario francés usado oficialmente para aceptar palabras en el juego, sin saber realmente articular una frase en este idioma. En 2018 volvió a ganar el mismo título, y desde 2016 ha ganado la versión francesa de *Scrabble* duplicado en numerosas ocasiones. *Scrabble* duplicado es una variante del juego en la que en lugar de enfrentarse dos jugadores entre sí, todos empiezan con la misma posición del tablero (mismas letras ya colocadas y letras disponibles para ser usadas) y el objetivo es encontrar la mejor jugada, es decir, aquella que mayor puntuación ofrezca. En palabras de un jugador experimentado: *"Con un jugador que no habla una palabra de francés y obtiene la mayor puntuación en todas las partidas, el campeonato ya no tiene sentido"*.

Después de un largo descanso, Nigel tiene ganas de participar en unas partidas de *Scrabble* duplicado en francés. Quiere obtener la mayor puntuación posible en total, pero también quiere dejar alguna partida a los jugadores francófonos, así que no participará en más de un cierto número de partidas seguidas. En las partidas en las que no participa no obtiene puntos, y en las que sí, asumimos que obtendrá la puntuación máxima (no sería de extrañar sabiendo que lo ha hecho varias veces en el pasado). Averigua cuál es la máxima puntuación que puede obtener siguiendo una estrategia óptima de selección y abandono de partidas.

Entrada

La entrada consta de varios casos de prueba. En la primera línea se indica el número de casos de prueba. Cada caso de prueba consta de dos líneas.

En la primera línea de cada caso se indica el número de partidas en total N ($1 \leq N \leq 300.000$) y el número de partidas K ($1 \leq K \leq N$) en las que como mucho Nigel va a participar de forma consecutiva.

En la segunda línea de cada caso se indican N enteros p_i ($1 \leq p_i \leq 10^9$), siendo cada uno la puntuación que obtendrá Nigel si participa en esa partida.[2]

Salida

Para cada caso de prueba se escribirá una línea con un solo número, indicando la puntuación máxima que Nigel puede obtener si elige de forma óptima qué partidas juega y cuáles se salta, de tal forma que no participe en más de K partidas consecutivas.

[2]En el juego original, la puntuación de *Scrabble* duplicado se da como un número negativo indicando a cuántos puntos se ha quedado el jugador de la puntuación perfecta, pero para simplificar se da un número positivo indicando la máxima puntuación.

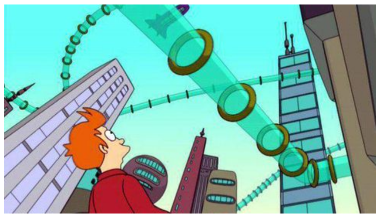
Entrada de ejemplo

```
3
2 2
1 2
5 3
2 3 4 5 6
5 3
10 8 9 9 9
```

Salida de ejemplo

```
3
17
37
```

L
Veinte mil leguas de viaje en hipertubo



El transporte soñado por Matt Groening en Futurama ya es una realidad. La ciudad de Madrid acaba de inaugurar su nuevo sistema de tubos de propulsión hidráulica, el *hipertubo*, que permitirá transportar personas y paquetes velozmente por toda la urbe. Este portento de la ingeniería está compuesto por una red de tubos interconectados por los que circula (siempre en la misma dirección) aire a presión que empuja a sus ocupantes hacia su destino. En cada intersección, los transeúntes o paquetes pueden apearse o continuar su viaje por alguno de los tubos siguientes. Sin embargo, como la construcción ha requerido muchos metros de tubo, no todos se han podido encontrar del mismo diámetro. Algunos tramos son estrechos y solo caben por ellos pequeños paquetes, mientras que en otros un hipopótamo circularía holgadamente.

Quieres enviar un bulto de un punto a otro de la ciudad y te gustaría usar el hipertubo, pero antes debes asegurarte de que no se va a quedar atascado por el camino. A tu disposición tienes un plano del hipertubo con el diámetro de todos sus tramos y un ordenador, ¿podrías calcular qué objetos pueden realizar el trayecto?

Entrada

La entrada consistirá en una serie de casos de prueba. Cada caso comienza con una línea con el número N de intersecciones de la red ($2 \leq N \leq 25.000$) y el número M de tramos de hipertubo ($0 \leq M \leq 200.000$). Siguen M líneas, cada una con tres números: los extremos del tubo (números distintos entre 1 y N) en la dirección del aire propulsado y el diámetro del tubo (entre 1 y 10^6). Nunca hay más de un tramo que conecte en la misma dirección un mismo par de intersecciones.

Por último, aparece una línea con dos números distintos (entre 1 y N) para el origen y el destino del trayecto deseado.

Salida

Para cada caso de prueba se escribirá una línea con el diámetro máximo de los objetos que pueden desplazarse por el hipertubo del punto de origen al punto de destino. En el caso de que no sea posible enviar un objeto del origen al destino se escribirá un 0.

Entrada de ejemplo

```
5 6
1 2 1
1 4 5
3 2 4
4 5 5
5 2 10
5 3 8
4 2
3 1
2 3 10
1 2
```

Salida de ejemplo

```
5
0
```