

Preparación

ADA BYRON

MADRID

MASTER



Competitive Programming

UPV

Capítulos repetidos

Tiempo máximo: 1,000-3,000 s Memoria máxima: 20480 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=647>

Andrés es un forofo de las series de televisión, se las ve todas y todos los capítulos que se emiten. Lo que no le gusta es cuando una cadena elige una serie y empieza a emitir sus capítulos de forma desordenada (no siguiendo la cronología de las temporadas) y repitiendo capítulos sin ningún criterio. ¡Hay veces que vuelve a ver el mismo capítulo dos días después!

Para cada una de sus series favoritas, ha estado apuntando los números de los capítulos que se han emitido cada día (uno por día), y ahora se pregunta cuál es, para cada serie, el periodo más largo de días consecutivos en el que se emitieron capítulos todos diferentes entre sí. ¿Le ayudas a averiguarlo?



Entrada

La entrada comienza con el número de series que Andrés ha estado controlando. Para cada serie, aparece en una primera línea el número de capítulos que Andrés ha apuntado (nunca más de 100.000), seguida de otra línea con los números (entre 1 y 10^9) de los capítulos emitidos, separados por espacios.

Salida

Para cada serie, el programa deberá escribir el número máximo de días consecutivos en los que no se emitió ningún capítulo repetido (es decir, que todos los capítulos emitidos en ese periodo son diferentes entre sí).

Entrada de ejemplo

```
3
6
1 2 3 1 2 3
4
2 2 2 2
7
2 1 2 3 4 1 5
```

Salida de ejemplo

```
3
1
5
```

Autor: Alberto Verdejo.

Revisores: Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.

El conteo de la rosa

Tiempo máximo: 2,000-3,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=323>

Estando ya al final de mi vida de pecador, esperando que mi enfermo y pesado cuerpo libere mi alma para poder enfrentarme al Príncipe de los Apóstoles, vienen a mi memoria lejanos recuerdos que aún conservo pese a que, al mismo tiempo, no me sea posible, cual hechizo del Anticristo, recordar lo que desayuné tras mis rezos de Maitines.

Siendo apenas un novicio, desconocedor de la Santa y férrea disciplina del monasterio, me enfrenté con el abad, cuyo nombre prefiero ahora cubrir con un piadoso manto de silencio, por lo que mi joven entendimiento consideró una afrenta del bibliotecario mayor. Éste nos encargó a Adso, otro novicio que a la par llegaría a ser sabio franciscano, y a mí que numeráramos las 200 páginas de una nueva edición de la *Poética* de *Aristóteles* que varios monjes habían estado copiando durante meses atrás. El bibliotecario, que Dios tenga en Su Gloria, le asignó a Adso la numeración de las 100 primeras páginas del manuscrito, de la 1 a la 100, mientras que a mí me fueron asignadas las 100 siguientes, de la 101 a la 200. Yo, que había oído preocupantes rumores sobre una maldición que mataba a todo aquél que se acercaba a ese libro, caí en la cuenta de que me vería obligado a transcribir muchos más *dígitos*, que no páginas, que mi compañero de celda, razón que me llevó a mi enfrentamiento con el abad.

Éste, que consideró una mera lujuria del conocimiento que me hubiera planteado ni siquiera semejante hecho, me forzó a rezar en Laudes, Tercia y Vísperas durante todo un año el Salmo 30 para pedir protección contra las injusticias. Aun así, algo debió ver en mis ojos brillantes de muchacho, que concedió darme la bula de su castigo si le decía hasta qué página debía numerar Adso, y a partir de cuál debía numerar yo para que el reparto fuera justo, de forma que si Adso numerara una página más ya tendría que escribir más dígitos que yo.

En el pecado llevé mi penitencia, porque me tembló el entendimiento y ni siquiera hoy el Señor me ha concedido la gracia de descubrir la respuesta.



Entrada

La entrada consta de una serie de casos de prueba. Cada caso son dos números, el primero indica la página inicial a numerar y el segundo la página final. Se garantiza que $1 \leq p_1 < p_2 \leq 1.000.000$. La entrada termina con dos valores a cero.

Salida

Para cada caso de prueba se indicará la última página que debería escribir Adso, el primer monje, de tal forma que ambos escriban la misma cantidad de dígitos. Si esto resulta imposible, Adso deberá numerar la mayor cantidad posible de páginas siempre que escriba menos *dígitos* que el segundo monje.

Entrada de ejemplo

```
1 200
99 100
99 101
97 103
0 0
```

Salida de ejemplo

118
99
99
100

Autor: Pedro Pablo Gómez Martín.

Revisores: Alberto Verdejo y Marco Antonio Gómez Martín.

Megaautopista de Oriente

Tiempo máximo: 1,000-3,000 s Memoria máxima: 24576 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=693>

La megaautopista de Oriente está terminada y a punto de estrenarse. A lo largo de sus miles de kilómetros hay áreas de descanso ya habilitadas. Las autoridades han abierto un concurso para la explotación de dichas áreas, y en particular para la construcción de restaurantes que sirvan de esparcimiento gastronómico a los usuarios de la autopista. Para evitar el monopolio han puesto una condición: una misma empresa no podrá tener dos restaurantes a menos de K kilómetros de separación.



En la compañía de restaurantes *Le Nouvelle Kebab* están muy interesados en esta oportunidad de expansión y van a apostar fuerte por conseguir un buen negocio. Han hecho un estudio y valorando cada área de descanso según su posición, las poblaciones cercanas, la cantidad de tráfico previsto, las vistas, etc., han estimado cuál sería el beneficio medio diario en cada área de descanso.

Ahora quieren saber dónde deberían construir sus restaurantes para maximizar el beneficio obtenido, sabiendo que tienen que respetar la restricción sobre separación entre restaurantes.

Entrada

La entrada está formada por diversos casos de prueba. Cada caso se describe en tres líneas: en la primera aparece el número N de áreas de descanso de la megaautopista ($1 \leq N \leq 300.000$) y los kilómetros K de separación mínima entre restaurantes de la misma empresa ($1 \leq K \leq 1.000.000$); en la segunda línea aparecen los N puntos kilométricos donde están localizadas esas áreas de descanso, en orden y medidos todos desde el comienzo de la megaautopista (esta nunca tendrá más de 10.000.000 de kilómetros); y en la tercera aparecen los N beneficios estimados, uno por área, en el mismo orden (los beneficios por área nunca son mayores que 1.000).

Salida

Por cada caso de prueba se escribirá una línea con el máximo beneficio que puede obtener la compañía de restaurantes según su estimación y respetando la restricción de separación entre restaurantes.

Entrada de ejemplo

```
3 15
10 20 30
20 40 10
3 10
10 20 30
20 40 10
```

Salida de ejemplo

```
40
70
```

Autor: Alberto Verdejo.

Revisores: Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.

Ada, Alan y compañía

Tiempo máximo: 2,000 s Memoria máxima: 8192 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=255>

Cuando iba a las fiestas de la alta sociedad, a Ada le gustaba impresionar a los maridos de sus amigas hablándoles de las palabras palíndromas. “Los palíndromos —les decía— son palabras o frases que se leen igual de izquierda a derecha que de derecha a izquierda”. Les ponía ejemplos como “seres”, “somos” de 5 letras, o, más largas, “acurruca” de 8 y “reconocer” de 9. Y siempre acababa su disertación presumiendo de que su propio nombre, Ada, era también palíndromo.



Alan solía acompañarla en esas situaciones e, intentando quedar por encima para impresionar aún más, siempre afirmaba que su nombre era todavía mejor. Desde niño había sido aficionado a los códigos y las frases ocultas, y estaba orgulloso de que su nombre no fuera palíndromo pero ocultara uno en su interior, “ala”.

Pero los dos se cuidaban mucho de presumir cuando su amigo Charles estaba cerca. La primera vez que intentaron impresionarle, al fin y al cabo Charles sólo esconde palíndromos triviales de longitud 1, salieron los dos escaldados. “Queridos —les replicó— mi nombre será vulgar pero mi apellido, Babbage, es mejor que vuestros nombres. No oculta uno, sino tres palíndromos no triviales, y además uno es de mayor longitud que los vuestros”. Y no le faltaba razón. Si se buscan, se pueden encontrar “bb”, “bab” y “abba”, el más largo de todos.

John, amigo de los tres, siempre esquivaba las discusiones sobre palíndromos. Odiaba los juegos en los que siempre perdía.

Entrada

El programa deberá procesar múltiples casos de prueba, cada uno en una línea. Cada caso de prueba contendrá una sucesión de un mínimo de 1 y un máximo de 2.000 letras minúsculas del alfabeto inglés, sin símbolos especiales ni espacios.

Salida

El programa debe indicar, en líneas independientes, la longitud del palíndromo más largo que contiene la sucesión de letras de cada caso de prueba.

Entrada de ejemplo

```
ada
alan
babbage
john
```

Salida de ejemplo

```
3
3
4
1
```

Autores: Pedro Pablo Gómez Martín y Marco Antonio Gómez Martín.

Revisor: Alberto Verdejo.

Parkímetros

Tiempo máximo: 3,000-4,000 s Memoria máxima: 8192 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=251>

Debido a su afán recaudatorio, el alcalde del pueblecito de la sierra Pese Taspamí ha decido poner parkímetros en la zona centro. Toda la oposición está en contra, y consideran que la medida es una demostración más de la poca inteligencia del alcalde: la zona centro es peatonal desde hace años.

Además, por ahorrarse dinero, los parkímetros los ha adquirido de saldo y son muy malos; no sólo no devuelven cambio, algo habitual en este tipo de aparatos, sino que además sólo permiten pagar con un número máximo de monedas. Si un ciudadano, al ir a pagar, introduce más monedas de las que el parkímetro es capaz de gestionar, éste comienza a devolverlas y no las admite.

Al descubrir esta limitación, el alcalde se ha quedado muy preocupado porque significa que el número de pagos posibles está limitado, lo que restringe los precios que puede cobrar.



Entrada

La entrada comenzará con un número indicando la cantidad de casos de prueba que vendrán a continuación.

Cada caso de prueba estará compuesto de dos líneas. La primera tendrá dos números mayores que 0 indicando la cantidad de monedas diferentes que están en uso en el pueblo, y el número máximo de monedas simultáneas que los parkímetros son capaces de procesar. Ninguno de los dos valores será mayor que 10.

A continuación, se indicará, en la segunda línea del caso de prueba, el valor de las diferentes monedas existentes, medidas en céntimos. El valor de la moneda más alta no superará nunca los 200 céntimos.

Salida

Por cada caso de prueba, el programa indicará el número de posibles precios distintos que se pueden pagar en los parkímetros.

Entrada de ejemplo

```
4
8 1
1 2 5 10 20 50 100 200
4 2
1 2 5 10
8 2
200 100 50 20 10 5 2 1
3 3
1 2 5
```

Salida de ejemplo

```
8
12
39
13
```

Autores: Pedro Pablo Gómez Martín y Marco Antonio Gómez Martín.

Revisores: Alberto Verdejo, Isabel Pita y Clara Segura.

Racha afortunada

Tiempo máximo: 1,000-2,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=316>

Vladimir es un jugador de *póquer* profesional que lleva muchos años perfeccionando su técnica de juego. Sabe que la práctica y la constancia son la clave para llegar a dominar cualquier cosa y, desde el principio, lleva registros exhaustivos de todas sus partidas.

Como todos los jugadores, tiene días mejores y días peores. Forma parte del juego, a veces se gana y a veces se pierde. Sin embargo, mirando sus registros se ha dado cuenta de que a veces tiene rachas muy buenas en las que gana casi siempre y le gustaría identificarlas.

En particular, teniendo en cuenta las ganancias o pérdidas de cada día, le gustaría conocer qué día debería haber empezado a jugar y qué día debería haberlo dejado para maximizar sus ganancias, suponiendo que jugase todos los días intermedios. ¿Puedes ayudarlo?



Entrada

El primer número de la entrada indica el número de casos de prueba que aparecen a continuación.

Cada caso de prueba está compuesto por dos líneas. La primera contiene el número de días anotados en el registro (al menos 1 y como mucho 100.000), y la segunda línea las ganancias (números positivos) o pérdidas (números negativos) de cada día. Cuando las ganancias o las pérdidas de un día llegan a 10.000, lo deja antes de que la suerte le abandone, o de que tenga que volver a casa desnudo tapándose con un barril.

Salida

Para cada caso de prueba se deberá imprimir una línea con dos números que delimitan el intervalo de días consecutivos que Vladimir debería haber jugado para maximizar sus ganancias (suponiendo que sólo jugase esos días). Ten en cuenta que Vladimir juega tanto el día de inicio como el día de fin del intervalo, y que los días empiezan a numerarse en 1.

En caso de existir varios intervalos en los que Vladimir podría obtener el máximo beneficio posible, se debe elegir el de duración más corta y, a igualdad de duración, el que empiece primero. Al fin y al cabo, a nadie le gusta trabajar más de lo necesario y todos preferimos terminar cuanto antes.

Vladimir es un profesional, por lo que siempre hay al menos un día en el que gana algo.

Entrada de ejemplo

```
3
4
-10 4 8 -2
5
-3 0 7 -20 -7
11
30 -7 1 -25 -20 25 10 3 -1 8 -100
```

Salida de ejemplo

```
2 3
3 3
6 10
```

Autores: Antonio Sánchez y Pedro Pablo Gómez Martín.

Revisor: Marco Antonio Gómez Martín.

Examen de pociones

Tiempo máximo: 1,000-3,000 s Memoria máxima: 10240 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=655>

Mañana es el examen de *pociones*. Lo bueno es que no tienes que estudiar porque ya te lo sabes. Lo malo es que te lo sabes porque... eres el profesor de la asignatura y... aún tienes el examen sin poner. Estás seguro de que tus alumnos estarán ya dormidos tan a gusto mientras tú estás ahí dando vueltas a qué examen poner para que no suspendan todos.



Después de mucho pensar ya sabes qué poción vas a pedir que te hagan y ahora tienes que decidir cuánto tiempo les dejas en total para hacerlo. Ese cálculo no es fácil porque una misma poción puede hacerse de muchas formas distintas y quieres que, elijan la forma que elijan, puedan terminar en plazo.

En general, las pociones se componen de pasos distintos. Cada uno de los pasos consiste en una acción (de duración despreciable) como echar algún ingrediente en el caldero o poner el fuego más alto. Tras esa acción hay que esperar un tiempo determinado mientras se remueve con un cucharón, se recita un conjuro, se baila una danza o cualquier otra cosa. La espera (y lo que se haga en ella) lleva a la poción a un estado distinto al original en el que o bien el brebaje ya está listo o bien toca echar otro ingrediente y volver a esperar.

Como buen profesor de pociones, el proceso completo lo tienes bien documentado: para cada estado posible del caldero y la poción que contiene, sabes qué ingredientes hay que echar, de cuántas formas distintas se puede esperar y, para cada posible espera, cuánto dura y a qué estado de la poción conduce.

Con toda esa información, lo que queda es pan comido. Hay que calcular el tiempo máximo de elaboración de la poción, hacer el enunciado, pasarlo a los pergaminos de examen y ya podrás irte a dormir.

Entrada

La entrada está compuesta de distintos casos de prueba

Cada caso de prueba comienza con una línea con dos números: el número de estados posibles en los que puede estar la poción (hasta 10.000) y el número de transiciones posibles entre estados (al menos una no más de 100.000). Tras eso aparece una línea por cada transición indicando el estado origen y destino (números entre 1 y el número total de estados) y el tiempo que hay que esperar (hasta 10.000)

Se garantiza que la receta nunca “va hacia atrás”, es decir que ninguna secuencia de acciones hace que el brebaje vuelva a estar como estuvo anteriormente.

Ten en cuenta que puede haber distintas alternativas para la primera acción a realizar sobre el caldero vacío y que hay distintos estados que se consideran que tienen la receta terminada. Además los distintos caminos para hacer la receta podrían no tener ningún estado común.

Tras el último caso de prueba viene una línea con dos ceros que no debe procesarse.

Salida

Por cada caso de prueba se escribirán dos líneas. En la primera se indicará la duración mínima del examen (o lo que es lo mismo, la duración máxima en la elaboración de la receta). La segunda línea contiene cuál es el camino de estados por los que pasa el brebaje con esa elaboración. Si hay varios caminos de coste máximo se escribirá aquel cuyo primer nodo tenga un identificador más pequeño en la entrada; si aún así sigue habiendo varios, se escribirá el que tenga el segundo nodo más pequeño y así sucesivamente.

Entrada de ejemplo

```
5 4
1 2 1
2 3 1
3 4 1
3 5 1
5 6
1 2 3
2 3 3
1 4 5
4 2 2
4 5 1
5 3 1
4 2
1 2 10
3 4 20
0 0
```

Salida de ejemplo

```
3
1 2 3 4
10
1 4 2 3
20
3 4
```

Autor: Marco Antonio Gómez Martín.

Revisor: Pedro Pablo Gómez Martín.

Cinco prohibido

Tiempo máximo: 1,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=652>

El sistema decimal ha muerto. Científicos de todo el mundo han realizado estudios que aseguran que el cerebro humano y las máquinas pueden realizar cálculos más rápido si el número 5 no existe. De esta forma, lo que antes era 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 y 15 ahora se convierte en 0, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16 y 17.

Han surgido movimientos sociales en contra de esta medida. Algunos sugieren que el lobby de empresas de calculadoras está detrás de este cambio; otros dicen que es la Real Academia Española (RAE) quien está detrás, harta de rimas malsonantes.

Independientemente de quien esté detrás, nos han encargado realizar una calculadora que multiplique números en este nuevo sistema.



Entrada

La entrada estará compuesta por varios casos de prueba, cada uno en una línea.

Cada caso de prueba serán dos números a y b de hasta 5 dígitos expresados con el nuevo formato (por lo que no tendrán cincos).

Salida

Por cada caso de prueba se escribirá el resultado de multiplicar ambos números utilizando, también, el formato de numeración nuevo.

Entrada de ejemplo

```
2 2
3 3
9 2
4 4
6 6
```

Salida de ejemplo

```
4
10
18
18
28
```

Autores: Pablo Trinidad y Marco Antonio Gómez Martín.

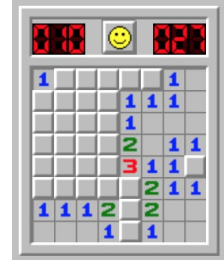
Revisor: Pedro Pablo Gómez Martín.

Jugando al Buscaminas

Tiempo máximo: 1,000-2,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=315>

Hoy vamos a jugar al *Buscaminas*. Seguro que lo conoces. En un tablero dividido en celdas se esconde cierto número de bombas. El objetivo es encontrarlas todas. Para ello hay que destapar todas las celdas que no esconden bombas. Como ayuda, cuando se descubre una celda sin bombas, aparece en ella el número de bombas en las (como mucho) 8 celdas colindantes. Así, si al descubrir una celda aparece un 3 eso significa que de las 8 celdas que hay alrededor 3 de ellas esconden una bomba y 5 no. Cuando se descubre una celda que no tiene bombas alrededor, en vez de un 0 la celda se deja vacía y se descubren de forma automática las 8 celdas colindantes (ahorrando al jugador tener que descubrirlas por sí mismo sin ningún riesgo). Esto provoca que en ocasiones se descubran muchas celdas de golpe. Si se descubre una celda con una bomba, se pierde la partida.



La figura muestra el estado del juego después de haber descubierto consecutivamente las cuatro esquinas del tablero.

Lo que queremos es conocer el estado del juego después de que el jugador haya descubierto una serie de celdas.

Entrada

La entrada constará de una serie de casos de prueba. Cada caso comienza con una línea con los números de filas F y columnas C del tablero ($1 \leq F, C \leq 50$). A continuación aparecen F líneas, cada una con C caracteres: un $*$ significa que en la celda correspondiente se esconde una bomba; las casillas vacías se indican con $-$. Después aparece el número K de celdas a descubrir, seguido de K líneas que indican las coordenadas de cada celda descubierta: una fila entre 1 y F y una columna entre 1 y C . Pueden intentarse descubrir celdas ya descubiertas anteriormente (quizás de forma automática), en cuyo caso el estado del tablero no cambia. Si al descubrir una celda aparece una bomba, esa será la última celda a descubrir que aparezca en la entrada para ese caso.

Salida

Para cada caso de prueba, se escribirá el tablero después de haber descubierto todas las celdas indicadas en la entrada (y las que lo hayan hecho de forma automática). Por cada celda descubierta se escribirá un número indicando el número de bombas que hay alrededor, salvo que este sea 0, que se escribirá $-$. Las celdas sin descubrir se indicarán con X .

Si durante el juego se ha descubierto una bomba, se escribirá **GAME OVER** en vez del tablero final.

Entrada de ejemplo

```
8 8
- * - - * - -
- - - - -
* * - - -
- - - * - -
- - - * - - *
- * - - * - -
- - - - * - -
- - - - -
4
1 1
1 8
8 8
8 1
```

Salida de ejemplo

```
1XXXXX1-  
XXXX111-  
XXXX1---  
XXXX2-11  
XXXX311X  
XXXXX211  
1112X2--  
---1X1--
```

Autor: Alberto Verdejo.

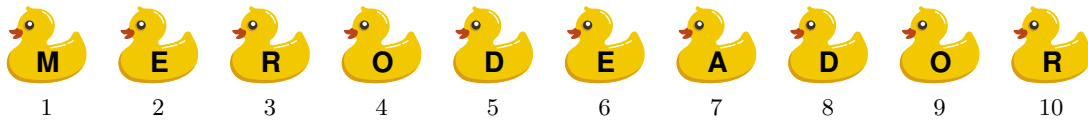
Revisor: Pedro Pablo Gómez Martín.

Tiro al patíndromo

Tiempo máximo: 1,000-3,000 s Memoria máxima: 10240 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=322>

Ya es tradición que cuando en el pueblo de Juan Filloy¹ están en fiestas, él se pase por el puesto del tiro con escopeta. Cuando saben que va a visitarles, en vez de premios en los patos a los que hay que disparar colocan letras de tal forma que Juan pueda divertirse disparando a algunos de los patos de manera que se forme un palíndromo con las letras en los patos supervivientes (es decir, que al leer tanto de izquierda a derecha como de derecha a izquierda, se lea siempre la misma palabra). Por ejemplo, si se enfrentara a la serie de patos de la figura, Juan dispararía a los patos en las posiciones 1, 2 y 6, formando el palíndromo RODADOR.



Además, a Juan le gusta alardear ante las mozas del pueblo que le miran expectantes, por lo que no se conforma solamente con encontrar un palíndromo sino que lo intenta conseguir tirando el menor número de patos. En caso de empate, Juan prefiere tirar los patos de la izquierda, porque a la derecha le molesta más el Sol que le deslumbra. ¿Sabrías tú hacer lo mismo?

Entrada

La entrada estará compuesta por múltiples casos de prueba, cada uno en una línea. Cada caso consiste en una sucesión de un mínimo de 1 y un máximo de 1.000 letras mayúsculas del alfabeto inglés, sin símbolos especiales ni espacios.

Salida

Por cada caso de prueba se escribirá una línea que contenga el palíndromo más largo que puede formarse eliminando (si es necesario) algunas de las letras de la entrada. En caso de empate, se debe imitar a Juan, eliminando letras más a la izquierda.

Entrada de ejemplo

```
MERODEADOR
RECONOCER
ARADAROSOSOMI
OSORASODAR
```

Salida de ejemplo

```
RODADOR
RECONOCER
OSOSO
RADAR
```

Autor: Alberto Verdejo.

Revisores: Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.

¹Escritor argentino nacido en 1894, autoproclamado “*recordman mundial de palindromía*”, gracias al cual hoy conocemos más de 8.000 palíndromos en español.

Simplificación adornada

Tiempo máximo: 2,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=686>

Facundo Toriz A. (al que todos llaman cariñosamente *Fac*) es un profesor de matemáticas que siempre busca formas de sorprender a sus alumnos con los ejercicios que les pone. Cree que eso les animará a resolverlos, aunque no tiene pruebas concluyentes de que su esfuerzo sirva realmente para algo.

Está preparando una hoja de ejercicios para sus clases de simplificación de fracciones, y la verdad es que no está muy inspirado. Lo único que se le ha ocurrido es poner fracciones que tengan, juntando el numerador y el denominador, todos los dígitos del 1 al 9 exactamente una vez. Por ejemplo, se ha dado cuenta que:

$$\frac{2}{9} = \frac{3924}{17658} = \frac{7596}{34182}$$

de modo que en lugar de pedir que simplifiquen la aburrida $4/18$ (que también da como resultado $2/9$), va a ponerles alguna de las otras dos, que son mucho más vistosas.

El único problema es que sospecha que no todas las fracciones irreducibles pueden ser escritas de esa forma, aunque hay otras que se pueden escribir de más de una.

Entrada

La entrada consistirá en un número arbitrario de casos de prueba. Cada uno estará compuesto de dos números, N y D , separados por un espacio, que constituyen el numerador y el denominador de *la solución* del problema que nuestro amigo *Fac* quiere poner a sus alumnos. Se cumple que es una fracción irreducible (los números son primos entre sí), y que $0 < N < D < 1.000.000$.

La entrada termina con dos ceros.

Salida

Por cada caso de prueba el programa escribirá el número de formas de reescribir la fracción utilizando todos los dígitos del 1 al 9 sin repetir ninguno. Ten en cuenta que deben contarse *únicamente* las reescrituras; si la fracción irreducible original usa los 9 dígitos *no* debe contarse.

Entrada de ejemplo

```
2 9
3 10
1 5
0 0
```

Salida de ejemplo

```
2
0
12
```

Autor: Pedro Pablo Gómez Martín.

Revisor: Marco Antonio Gómez Martín.

Multitarea

Tiempo máximo: 1,000-2,000 s Memoria máxima: 8192 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=649>

Los calendarios o agendas electrónicas controlan nuestra vida diaria. Para personas como yo, que no se nos da bien la multitarea, es importante tener como mucho una tarea planificada para cualquier minuto de nuestra vida.

En mi calendario hay dos tipos de tareas: tareas únicas y tareas periódicas. Las tareas únicas tienen un instante de comienzo, c y otro de finalización, f , lo que significa que la tarea me tendrá ocupado durante el intervalo de tiempo $[c, f)$. Las tareas periódicas tienen además de los instantes de comienzo y finalización de su primera aparición, un periodo de repetición. Se supone que estas tareas se repiten para siempre sin fin. Para simplificar las cosas, todos los tiempos se expresan en minutos desde un tiempo inicial 0. Por ejemplo, una tarea periódica con tiempo de inicio 5, tiempo de finalización 8 y periodo de repetición 100 ocurriría en los intervalos de tiempo $[5..8)$, $[105..108)$, $[205..208)$, etc.



Tu trabajo consiste en averiguar si mi calendario está libre de conflictos durante un periodo de mi vida. Se considera que dos tareas están en conflicto si y solo si sus intervalos de tiempo se superponen, por ejemplo $[2..5)$ y $[4..6)$ se superponen, pero no $[2..8)$ con $[9..10)$, o $[2..8)$ con $[8..10)$.

Entrada

La entrada está formada por una serie de casos de prueba, ocupando cada uno de ellos varias líneas.

En la primera línea aparecen tres números: N , que representa el número de tareas únicas; M , que representa el número de tareas periódicas; y T , que representa el número de minutos en los que quiero averiguar si hay o no conflictos. A continuación, aparecen N líneas, cada una con los instantes de comienzo y finalización de una tarea única. A éstas les siguen M líneas más, cada una con tres números: el instante de comienzo y finalización de la primera aparición de una tarea repetitiva, y el periodo de repetición.

Siempre hay al menos 1 tarea y nunca más de 10.000. Todos los tiempos son números entre 0 y 10^9 . Se garantiza que todas las tareas tardan al menos un minuto y que los intervalos de una misma tarea periódica no se solapan entre sí.

Salida

Para cada caso de prueba el programa escribirá SI si hay conflictos entre algunas tareas en el periodo de tiempo $[0..T)$, y NO en caso contrario. Se puede dar por hecho que las tareas terminadas antes del primer conflicto (o durante el intervalo de tiempo si no hay conflictos) serán como mucho 100.000.

Entrada de ejemplo

```
2 0 10
2 5
4 6
0 2 100
1 4 8
5 7 8
2 1 10
8 20
1 5
6 7 10
```


Salida de ejemplo

SI
NO
NO

Autor: Alberto Verdejo.

Revisores: Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.