

Olimpiada Informática Española

Regional de Madrid



Cuadernillo de problemas



Facultad
de
Informática



UNIVERSIDAD COMPLUTENSE
MADRID

19 de febrero de 2022

*Every great developer you know got there by solving problems
they were unqualified to solve until they actually did it*
Patrick McKenzie

Listado de problemas

A Mismo resto	3
B El mayor con dos primos	5
C ¿Dónde construimos los aeropuertos?	7
D Amebas	9
E Juegos online	11
F Suma10	13
G Los parasoles	15
H Representación hiperbinaria	17

Autores de los problemas:

- Marco Antonio Gómez Martín (Universidad Complutense de Madrid)
- Pedro Pablo Gómez Martín (Universidad Complutense de Madrid)
- Blanca Huergo Muñoz (Olimpiada Informática Femenina)
- Isabel Pita Andreu (Universidad Complutense de Madrid)
- Alberto Verdejo (Universidad Complutense de Madrid)

● A

Mismo resto

Dados números naturales A , B , D y R , ¿cuántos números hay en el intervalo $[A..B]$ que al ser divididos por D den como resto R ?

Por ejemplo, si $A = 1$, $B = 10$, $D = 5$ y $R = 3$, la respuesta es 2, los números 3 y 8.

Entrada

La entrada comienza con una línea con el número de casos de prueba que vendrán a continuación.

Cada caso de prueba aparece en una línea, y está formado por cuatro números, los valores A , B , D y R .

Restricciones

$$0 \leq A \leq B \leq 10^{18}$$

$$1 \leq D \leq B$$

$$0 \leq R < D$$

Subtareas

1. (5 puntos) $A = 0$, $B < 10$, $R = 0$
2. (10 puntos) $A = 0$, $B \leq 10^3$
3. (15 puntos) $B \leq 10^9$, $B - A \leq 10^3$
4. (50 puntos) $B \leq 10^9$
5. (20 puntos) sin restricciones adicionales

Salida

Por cada caso de prueba se escribirá una única línea indicando el número de valores en el intervalo $[A..B]$ que al dividirse por D dan como resto R .

Entrada de ejemplo

```
4
0 4 2 0
1 10 5 3
0 13 5 3
10 19 2 0
```

Salida de ejemplo

```
3
2
3
5
```


● B

El mayor con dos primos

Todos los números naturales tienen una única descomposición en factores primos. Por ejemplo, sabemos que:

$$1.547 = 7 \times 13 \times 17$$

Algunos números tienen la característica de estar compuestos únicamente por dos factores primos diferentes. Por ejemplo, el 6, el 12, el 18 o el 36 tienen a 2 y a 3 como únicos factores primos.

Dados dos números primos, p y q , ¿cuál es el número más alto que tiene únicamente esos dos factores, y que es menor o igual que otro número, n ?

Entrada

El programa deberá leer, de la entrada estándar, múltiples casos de prueba, cada uno en una línea. Un caso de prueba estará compuesto de tres números, n , p y q , donde p y q son dos primos distintos.

La entrada acabará con un 0, que no debe procesarse.

Restricciones

$$1 \leq n, p, q \leq 1.000.000.000$$

$p \neq q$ y ambos primos

Subtareas

1. (5 puntos) $n < p \cdot q \leq 1.000.000.000$
2. (35 puntos) $n < p^2 \cdot q^2 \leq 1.000.000.000$
3. (60 puntos) sin restricciones adicionales

Salida

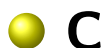
Para cada caso de prueba, el programa escribirá, por la salida estándar, el mayor número menor o igual que n que tenga en su descomposición en factores primos exactamente a p y a q . Si no hay ninguno se escribirá un 0.

Entrada de ejemplo

```
20 2 3
10 3 2
23 11 7
0
```

Salida de ejemplo

```
18
6
0
```

¿Dónde construimos los aeropuertos?

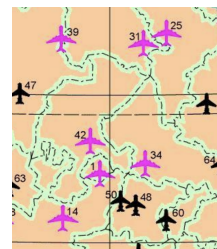
El gobierno de cierta nación en desarrollo quiere mejorar el acceso a una de sus regiones más olvidadas, en un intento de atraer inversiones. La región consta de varias localidades que deben tener acceso a un aeropuerto.

Una opción sería construir un aeropuerto en cada uno de estos lugares, pero se están buscando soluciones más económicas. En particular, se considerará que una ciudad tendrá acceso a un aeropuerto si ella misma tiene un aeropuerto, o si es posible viajar desde allí por autopista a otro lugar que tenga uno.

La región ha recibido poca inversión en infraestructuras y no hay autopistas, pero si construirlas permite reducir el coste total al necesitar menos aeropuertos, el gobierno está dispuesto a crearlas.

Se han hecho diferentes estudios sobre las posibles autopistas que podrían construirse, hasta el extremo de que en ocasiones hay varias opciones para unir las dos mismas ciudades, potencialmente con distintos costes.

Dados los costes de construcción de un aeropuerto en cada una de las localidades y una lista de posibles autopistas entre pares de localidades y sus correspondientes costes, el gobierno necesita ahora tu ayuda para decidir cuál es la forma más barata de garantizar que todas las localidades tengan acceso a un aeropuerto. El objetivo es que el acceso al aeropuerto sea lo más fácil posible, así que si hay varias formas de conseguir el coste mínimo, elige la que tenga más aeropuertos.



Entrada

La entrada está formada por una serie de casos de prueba. Cada caso comienza con una línea con dos enteros: N es el número de localidades y M es el número de autopistas posibles que se pueden construir.

A continuación aparece una línea con N números, $A_1 \dots A_N$, donde A_i es el coste de construir un aeropuerto en la localidad i .

Las siguientes M líneas describen las posibles autopistas: cada línea contiene tres enteros X , Y y C , donde X e Y son dos lugares y C es el coste de construir una autopista entre X e Y .

La entrada termina con dos ceros, 0 0.

Restricciones

$$1 \leq N \leq 200.000$$

$$0 \leq M \leq 200.000$$

$$1 \leq A_i \leq 1.000.000, \sum A_i \leq 10^9$$

$$1 \leq X, Y \leq N$$

$$1 \leq C \leq 100.000$$

$$\text{Si } t \text{ es el número de casos de prueba, } \sum_{i=1}^t N_i + M_i \leq 400.000$$

Subtareas

1. (5 puntos) $N = 2$, $M \leq 1$ y todos los A_i son iguales
2. (5 puntos) $N = 3$, $M = 2$ y todos los A_i son iguales
3. (25 puntos) $N \leq 100$, $M \leq 1.000$ y todos los A_i son iguales
4. (40 puntos) todos los A_i son iguales
5. (25 puntos) sin restricciones adicionales

Salida

El programa escribirá una línea por cada caso de prueba, con dos números: el coste mínimo de construir las autopistas y aeropuertos necesarios para que todas las localidades tengan acceso al menos a un aeropuerto, y el número de aeropuertos a construir. Como ya se ha dicho, si hay varias respuestas con un coste mínimo, se elegirá la que maximice el número de aeropuertos.

Entrada de ejemplo

```
4 4
100 100 100 100
1 2 10
2 3 30
4 1 40
4 3 20
5 3
1000 2000 1000 1000 3000
4 5 10
1 2 20
3 2 30
0 0
```

Salida de ejemplo

```
160 1
2060 2
```

D

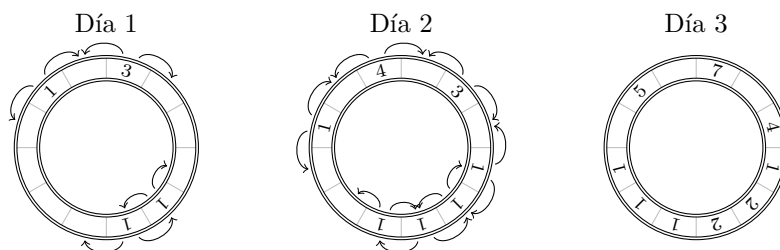
Amebas

La mitosis celular se da cuando el núcleo de una célula eucariota se divide en dos manteniendo intacta la información genética que contienen sus cromosomas. A la mitosis le sigue la división celular de forma que donde antes había una célula ahora hay dos genéticamente idénticas.

La mitosis es esencial para, por ejemplo, el crecimiento o reparación de tejidos de los seres vivos y es la base de la reproducción de organismos simples de una o pocas células. Ese es el caso, por ejemplo, de las amebas, organismos unicelulares que se multiplican asexualmente: cuando llega el momento el material genético se duplica, la célula se alarga, el citoplasma se divide y en lugar de una célula tenemos dos.

En un laboratorio tenemos un cultivo de amebas en el que distintas “parcelas” se distribuyen en una corona circular. Inicialmente en cada parcela tenemos un número fijo de amebas que alcanzan su madurez y se duplican diariamente. Cada vez que una ameba se duplica una de las copias va a la parcela situada a la izquierda y la otra a la parcela de la derecha.

La figura muestra un ejemplo de la evolución de un cultivo durante tres días. El primer día hay seis amebas distribuidas en cuatro parcelas (se omiten los números en aquellas parcelas vacías). Esas amebas se duplican y se mueven de forma que al principio del segundo día hay doce y el tercer día veinticuatro.



Dado el número de amebas en cada parcela de un cultivo, ¿cuántas amebas habrá en cada parcela un día concreto?

Entrada

La entrada estará compuesta de t casos de prueba, cada uno ocupando dos líneas.

La primera línea de cada caso de prueba tendrá dos números, n y d indicando, respectivamente, el número de parcelas y el número del día por el que se pregunta. En la segunda línea aparecen n enteros con el número de amebas que hay en cada parcela el día 1 (como mucho 10^9).

Recuerda que como las parcelas se distribuyen como una corona circular la parcela correspondiente al primer número tiene como vecina izquierda la parcela correspondiente al último.

Al último caso de prueba le sigue una línea con dos ceros que no debe ser procesada.

Restricciones

$$3 \leq n \leq 1000, 1 \leq d \leq 10^9$$

Subtareas

1. (3 puntos) $n \leq 5$, $d \leq 2$, $t \leq 2000$ y parcelas con hasta 1000 amebas al principio
2. (2 puntos) $n \leq 5$, $d \leq 2$, $t \leq 2000$
3. (20 puntos) $d \leq 100$, $t \leq 10$
4. (50 puntos) $n \leq 100$, $t \leq 2$
5. (25 puntos) sin restricciones adicionales (con $t \leq 2$)

Salida

Por cada caso de prueba se escribirán, en una línea independiente, n números que indican las amebas totales el último día en cada parcela. Como pueden ser muchas, en lugar de las cantidades totales se dará el resto de dividir esas cantidades entre 10^9+7 .

Entrada de ejemplo

```
4 2
0 2 0 0
4 2
2 0 0 0
12 3
1 0 3 0 0 0 1 1 0 0 0 0
0 0
```

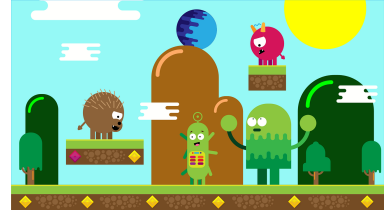
Salida de ejemplo

```
2 0 2 0
0 2 0 2
5 0 7 0 4 1 2 2 1 1 1 0
```

● E

Juegos online

Mi abuelo es un gran aficionado a los juegos online. Por la noche después de la cena decide con qué juego se entretendrá un rato, y se hecha una o varias partidas, sin cambiar de juego, hasta que se acuesta. Le gusta tanto que es capaz de recordar a qué jugó cada noche. A veces le pregunto si recuerda cuándo fue la tercera vez que eligió jugar al Minecraft y enseguida me contesta “Jugué por tercera vez el día 15”. Para mi abuelo, el día 1 fue el que jugó a su primer juego online, algo que no ha dejado hacer desde entonces cada noche. ¡Los videojuegos le han cambiado la vida!



¿Podrías ayudar al abuelo a recordar cuándo ha jugado cada partida cuando ya no tenga tan buena memoria?

Entrada

La entrada está formada por una serie de casos de prueba. Cada caso comienza por una línea con la cantidad N de noches que el abuelo ha jugado después de cenar. En la siguiente línea aparecen los nombres de los juegos a los que ha jugado cada noche. Cada nombre es una cadena de no más de 20 caracteres sin espacios.

A continuación aparece un número M que indica el número de preguntas que se van a hacer. En las siguientes M líneas aparecen las preguntas, cada una compuesta de un número k y el nombre de un juego `Juego`, que representan la pregunta “¿Cuándo jugaste por k -ésima vez a `Juego`?”

La entrada termina con un 0.

Restricciones

$$1 \leq N \leq 100.000$$

$$1 \leq M \leq 100.000$$

$$1 \leq k \leq N$$

Subtareas

1. (5 puntos) el abuelo juega solamente a LoL
2. (5 puntos) $M = 1$, $k = 1$
3. (10 puntos) $N \leq 100$, $M \leq 100$
4. (15 puntos) el abuelo juega solamente a LoL y Minecraft
5. (20 puntos) el abuelo juega como mucho a 20 juegos diferentes
6. (45 puntos) sin restricciones adicionales

Salida

Para cada caso de prueba, se escribirá una línea por cada pregunta con el día en el que el abuelo jugó por k -ésima vez al juego. Recuerda que el primer día en que el abuelo jugó (a cualquier juego) se considera el día 1. Si no ha jugado nunca a ese juego, o no tantas veces, se escribirá `NO JUEGA`.

Detrás de cada caso se escribirán tres guiones, ---.

Entrada de ejemplo

```
7
Pacman Parchis Parchis SuperMario Pacman Parchis Minecraft
5
2 Pacman
1 Minecraft
3 Parchis
1 Oca
2 SuperMario
0
```

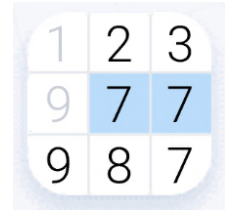
Salida de ejemplo

```
5
7
6
NO JUEGA
NO JUEGA
---
```

● F

Suma10

Izan está cansado de pelearse con el nuevo juego de moda: *Suma10*. El juego comienza con una lista con n números, todos ellos del 1 al 10. En cada movimiento, se pueden eliminar dos números de la lista si ambos son iguales, o si suman 10. El objetivo es consumir todos para terminar con la lista vacía.



También se ha comprado la nueva versión del juego, *SumaS*, donde hay que conseguir sumar la cantidad S , en lugar de 10. Es decir, dos números de la lista pueden ser emparejados y eliminados si son iguales o si suman S .

Dada una lista inicial A , ¿sabrías decirle a Izan si hay alguna solución? Así dejaría de agobiarse en caso de que dada esa lista inicial no hubiera victoria posible.

Entrada

La entrada comienza con una línea con un único entero, el número t de partidas a jugar, cada una descrita en dos líneas.

La primera línea de cada partida contiene dos números: el número S que hay que conseguir sumando, y el número n , la longitud de A . La segunda línea contiene n enteros, $A_1 \dots A_n$.

Restricciones

$$10 \leq S \leq 100$$

$$1 \leq n \leq 10^6$$

$$1 \leq A_i \leq S$$

Subtareas

1. (3 puntos) $S = 10$ y n será impar
2. (10 puntos) $S = 10$ y todas las posiciones de A contendrán el mismo número
3. (10 puntos) $S = 10$ y $A_i = 5$ o $A_i = 10$ para todo i
4. (15 puntos) $S = 10$ y $n \leq 10$
5. (20 puntos) $S = 10$ y $n \leq 1000$
6. (42 puntos) sin restricciones adicionales

Salida

Para cada partida, se escribirá una línea con una frase en función del resultado: si hay alguna posibilidad de dejar la lista A vacía, se imprime SE PUEDE, IZAN! y, en caso contrario, se imprime NO MERECE LA PENA :(.

Entrada de ejemplo

```
2
10 6
2 7 4 3 4 8
20 1
7
```

Salida de ejemplo

```
SE PUEDE, IZAN!
NO MERECE LA PENA :(
```


● G

Los parasoles

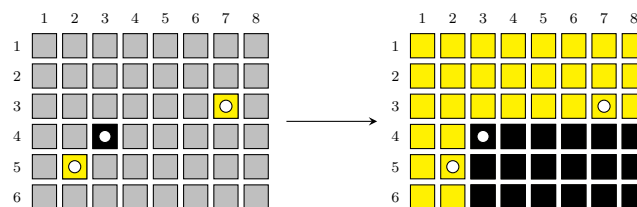
Nuestro tranquilo pueblo lleva siglos dedicado a la explotación agrícola con plantaciones de girasoles. De hecho, cuenta la leyenda que fuimos unos de los primeros en cultivarlos tras su llegada a Europa.

Desgraciadamente, en los últimos tiempos nuestras tierras están siendo afectadas por unos pequeños gusanos, los *parasoles*, que mordisquean los tallos de nuestras plantas e impiden que estas giren para mirar al Sol.

El ataque de los *parasoles* es siempre el mismo. Los gusanos entran en un cultivo por la esquina inferior derecha (mirando el cultivo desde arriba) y se van extendiendo formando un rectángulo. El resultado es que cuando se hace una foto del terreno se ve que una sección rectangular en la parte inferior derecha está llena de plantas que no miran al Sol.

Para poder evaluar los daños, el ayuntamiento ha comprado un dron que podemos enviar a un punto concreto de la plantación para saber si en esa zona las plantas han girado o no. Lo que necesitamos ahora es decidir a qué puntos mandamos al dron para saber cómo de afectada está un área determinada.

Como ejemplo, si tenemos un cultivo de tamaño 8x6 y mandamos al dron a las coordenadas (3,4), (7,3) y (2,5) y solo en la primera detecta parasoles, entonces sabremos con certeza que la zona afectada va desde la posición (3,4) hasta la esquina inferior derecha. La figura siguiente ilustra este ejemplo mostrando en amarillo las zonas sanas y en negro las zonas afectadas por los parasoles:



Nuestra tarea consiste en crear una aplicación que ayude al técnico del ayuntamiento a decidir dónde enviar el dron para poder averiguar rápidamente la posición de la esquina superior izquierda del rectángulo afectado de varios cultivos. La aplicación recibirá el tamaño de la parcela e irá solicitando al operario que envíe al dron a distintas coordenadas. En base a lo que vea el dron en cada posición, la aplicación decidirá dónde le envía a continuación.

El funcionamiento exacto es el siguiente. Al arrancar, la aplicación se quedará esperando a que el operario introduzca por teclado el tamaño de la parcela a analizar (dos números, TamX y TamY). A continuación irá pidiendo al operario que mande al dron a los sitios que estime convenientes para ver la situación en esa zona. Para eso, escribirá en líneas independientes comandos de la forma: “? posX posY”. El técnico del ayuntamiento irá mandando al dron a cada una de esas posiciones y a su vuelta escribirá en el teclado un 0 si la zona está intacta y un 1 si la zona ha sido afectada por los parasoles. Cuando la aplicación conozca el resultado, escribirá una línea con “=> posX posY” con la esquina solicitada; en ese momento el operario escribirá un nuevo tamaño de parcela comenzando de nuevo el ciclo o bien escribirá un tamaño vacío (“0 0”) lo que hará que nuestra aplicación termine.

A continuación aparece un ejemplo de ejecución. En cursiva aparecen los datos introducidos por el operario y en negrita los escritos por la aplicación.

```

2 3
? 1 1
1
=> 1 1
8 6
? 3 4
1
? 7 3
0
? 2 5

```

0
=> 3 4
0 0

Restricciones

$$1 \leq \text{TamX}, \text{TamY} \leq 10^5$$

Subtareas

1. (2 puntos) $\text{TamY} \leq 10$, $\text{TamX} \leq 10$ y la respuesta es $(\text{TamX}, \text{TamY})$
2. (10 puntos) $\text{TamY} = 1$, $\text{TamX} \leq 10$
3. (4 puntos) $\text{TamY} = 2$, $\text{TamX} \leq 10$
4. (40 puntos) $\text{TamY} = 1$, $\text{TamX} \leq 10^5$
5. (4 puntos) $\text{TamY} = 2$, $\text{TamX} \leq 10^5$
6. (40 puntos) sin restricciones adicionales

Notas

Este es un problema *interactivo*. Es importante que cada vez que tu solución escriba algo que el juez deba leer (en la ambientación, cada vez que tu aplicación escriba algo para el “operario del ayuntamiento”) 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;`.
- Java: `System.out.flush();`.
- Python: `stdout.flush()`.

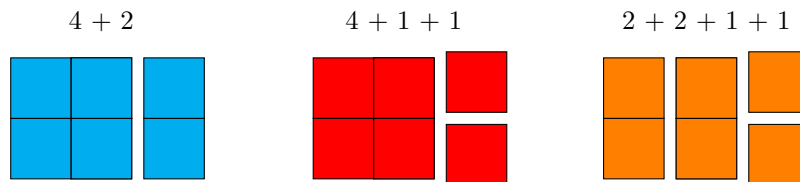


Representación hiperbinaria

Es ampliamente conocido que un número puede expresarse de una única forma como suma de potencias de dos. De hecho las potencias utilizadas en esa suma son las que dan forma a la representación en base 2 del número.

Por ejemplo, $6 = 4 + 2$ lo que hace que la representación binaria del 6 sea el 110_2 .

La *representación hiperbinaria* de un número es aquella en la que el número se expresa como suma de potencias de dos en las que cada potencia, 2^k , aparece como mucho dos veces. Al permitir más de una aparición de una misma potencia la representación no es única. Por ejemplo el 6 puede representarse de tres formas distintas:



Dado un número cualquiera, ¿cuántas representaciones hiperbinarias tiene?

Entrada

La entrada comienza con una línea que contiene el número de casos de prueba que aparecerán a continuación (como mucho 2.000). Tras él aparece una línea por cada caso con un número n entre 1 y 10^{18} .

Restricciones

$$1 \leq n \leq 10^{18}$$

Subtareas

1. (2 puntos) $n \leq 6$
2. (9 puntos) $n \leq 14$
3. (4 puntos) $n \leq 1023, n = 2^k - 1$
4. (45 puntos) $n \leq 1023$
5. (20 puntos) $n \leq 10^9$
6. (20 puntos) sin restricciones adicionales

Salida

Por cada caso de prueba se escribirá una línea con la cantidad de representaciones hiperbinarias del número.

Entrada de ejemplo

```
4
1
2
3
6
```

Salida de ejemplo

1
2
1
3