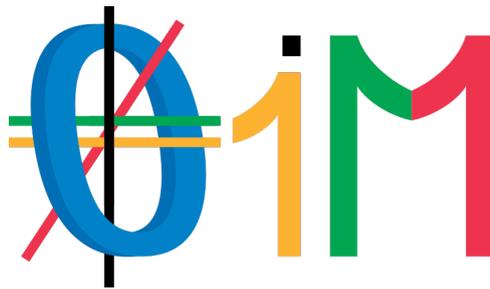


# Olimpiada Informática Española

## Regional de Madrid



### Cuadernillo de problemas



11 de febrero de 2023

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

11 de febrero de 2023

## Listado de problemas

<b>A. Detective de moda</b>	<b>3</b>
<b>B. Ni fu ni fa</b>	<b>5</b>
<b>C. Prohibidos los 11</b>	<b>7</b>
<b>D. La nevada del siglo</b>	<b>9</b>
<b>E. Atrapa la fruta</b>	<b>11</b>
<b>F. Por los anillos de Saturno</b>	<b>13</b>
<b>G. A toda potencia</b>	<b>15</b>

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)
- Huize Mao (Olimpiada Informática de Madrid)
- Manuel Montenegro Montes (Universidad Complutense de Madrid)
- Isabel Pita Andreu (Universidad Complutense de Madrid)
- Javier del Río López (Universidad Complutense de Madrid)
- Rubén Rubio (Universidad Complutense de Madrid)
- Alberto Verdejo (Universidad Complutense de Madrid)



# A

## Detective de moda

Rocío tiene una tienda un tanto peculiar enfrente de su casa. La tienda solo vende camisetas, y cada camiseta tiene un estilo propio que se identifica por un número, aunque se puede dar el caso de que varias camisetas a la venta tengan el mismo estilo.

La tienda tiene todas las camisetas dispuestas en una barra horizontal, y tiene una política de compras un tanto restrictiva: solamente se permiten comprar las camisetas que se encuentren en los extremos de la barra, para que no se desordenen mucho.

Para vestir de exclusiva, Rocío quiere comprar una camiseta que sea única, es decir que no haya otra con su mismo estilo en la tienda en el momento de adquirirla. Está dispuesta a comprar camisetas que no sean únicas con tal de llevarse una que sí lo sea. Pero como tampoco le sobra el dinero, no quiere que sean demasiadas.



### Entrada

La entrada comienza con el número de casos de prueba que vendrán a continuación, cada uno describiendo, a lo largo de varias líneas, la situación de la tienda y los eventos que tienen lugar a lo largo del tiempo.

La primera línea de cada caso contiene un número  $N$ , el número de camisetas que la tienda tiene a la venta actualmente.

En la siguiente línea aparecen  $N$  enteros  $x_i$ , los identificadores de los estilos de las camisetas. Dos camisetas con el mismo estilo tendrán el mismo identificador. En la entrada, las camisetas siguen el mismo orden en el que se encuentran en la tienda, de izquierda a derecha.

A continuación aparece un número  $E$ , indicando el número de eventos que tienen lugar. Las siguientes  $E$  líneas del caso de prueba describen cada una un evento en orden cronológico. Cada evento puede ser de 3 tipos:

- Las letras I y D seguidas de un número  $y_j$  indican que se añade una camiseta a la venta con identificador de estilo  $y_j$  en el extremo Izquierdo o Derecho, respectivamente.
- Las letras i y d indican que se ha vendido la camiseta que se encontraba en el extremo izquierdo o derecho, respectivamente. Se garantiza que cuando se realiza una venta la tienda tiene camisetas.
- La letra P indica que Rocío se pregunta en ese momento cuántas camisetas consecutivas de uno de los extremos debería comprar como mínimo para garantizar que se lleva a casa una camiseta con un estilo único en la tienda en ese momento. Rocío nunca compra las camisetas, solo se hace la pregunta.

### Restricciones

- $1 \leq N \leq 10^5$
- $1 \leq E \leq 10^5$
- $1 \leq x_i, y_j \leq 10^9$

### Subtareas

1. (5 puntos) Todos los números  $x_i$  e  $y_j$  son distintos.
2. (10 puntos)  $N \leq 100$ ,  $E \leq 100$ , y el número de camisetas no excede 100 en ningún momento.
3. (15 puntos) Nunca se venden camisetas y para todo  $y_j$  existe  $x_i$  tal que  $x_i = y_j$ .
4. (25 puntos) Nunca se venden camisetas.
5. (45 puntos) Sin restricciones adicionales.

## Salida

Por cada caso de prueba el programa deberá responder a todas las preguntas que se haga Rocío, en orden cronológico. Cada respuesta ocupará una línea independiente.

Para responder a una pregunta, se indicará el número mínimo de camisetas consecutivas que es necesario comprar, empezando por uno de los extremos, para llevarse a casa una camiseta que sea única en la tienda. Al número le seguirá, separado por un espacio, el nombre del extremo en cuestión: IZQUIERDA o DERECHA. Si pudiesen ser ambos, el nombre del extremo se sustituirá por la palabra CUALQUIERA.

Si en el momento de hacer una pregunta no hay ninguna camiseta con un estilo único en la tienda (o simplemente no hay camisetas), se escribirán las palabras NADA INTERESANTE.

Al final de cada caso de prueba se escribirán tres guiones ---.

## Entrada de ejemplo

```
2
5
8 5 9 23 8
9
P
I 22
P
I 23
D 5
D 22
P
I 9
P
5
2 1 3 1 2
5
P
i
P
D 2
P
```

## Salida de ejemplo

```
2 CUALQUIERA
1 IZQUIERDA
5 CUALQUIERA
NADA INTERESANTE
---
3 CUALQUIERA
1 DERECHA
2 IZQUIERDA
---
```

# B

## Ni fu ni fa

Cuenta la historia que el matemático G. H. Hardy, tras bajarse de un taxi que le había llevado a ver a S. Ramanujan, le indicó a este que el número del taxi en el que había llegado era bastante aburrido. Vamos, que era un número insulso y carente de todo interés por el que difícilmente alguien podría sentir algo de aprecio. En definitiva, que era un número *ni fu ni fa*.

$$1729$$
$$1^3 + 12^3 = 9^3 + 10^3$$

Ramanujan sintió curiosidad por saber qué número era ese tan vulgar y al enterarse de que Hardy estaba hablando del 1729 mostró su completo desacuerdo. Ese número resulta ser el más pequeño que puede escribirse como la suma de dos cubos positivos de dos formas diferentes.

Y es que, como bien nos dice la paradoja de los números interesantes, es difícil encontrar números aburridos, porque simplemente no los hay. La “demostración” de que todos los números son interesantes es fácil utilizando reducción al absurdo: si hubiera números aburridos y los pusiéramos todos juntos, uno de ellos sería el más pequeño, lo que automáticamente lo convertiría en interesante pues sería el número más pequeño que es aburrido<sup>1</sup>.

A pesar de lo difícil que parece, nuestro objetivo será el mismo que el de Hardy: encontrar algún número *ni fu ni fa* en una secuencia de números ordenados. Para nosotros los números interesantes serán únicamente los de los extremos: el valor más grande y el más pequeño. Y los *ni fu ni fa* son aquellos que no son ni el mínimo ni el máximo de la colección.

Por ejemplo, en  $[1, 2, 3, 4]$  tanto el 2 como el 3 son números *ni fu ni fa*. Sin embargo, en la secuencia  $[10, 10, 10, 13, 13]$  no hay ningún número *ni fu ni fa*.

La búsqueda de algún número así se hará de forma *interactiva*: al arrancar la solución, se leerá de la entrada estándar el número de secuencias que habrá que analizar. Para cada colección se leerá primero el número  $n$  de elementos de la secuencia. En ese momento la aplicación escribirá todas las consultas que necesite. Cada consulta consiste en una línea comenzando con el símbolo ? y, después de un espacio, la posición de la secuencia que quiere consultarse (entre 0 y  $n - 1$ ). Tras la consulta se podrá leer, de la entrada estándar, el valor de la secuencia en esa posición.

Cuando la aplicación conozca la respuesta escribirá los caracteres => y, tras un espacio:

- NINGUNO si la secuencia no tiene ningún elemento “ni fu ni fa”.
- Uno de los valores de la secuencia que sea “ni fu ni fa”. Si esta tiene varios, cualquiera vale.

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

```
2
1
=> NINGUNO
10
? 0
10
? 1
13
? 2
15
=> 13
```

### Restricciones

- $1 \leq n \leq 10^9$

---

<sup>1</sup>El problema de esa demostración es la ambigüedad en la propiedad “ser interesante”; al no estar bien definida no tiene sentido separar los enteros en interesantes y no interesantes.

- Los valores de la secuencia son números naturales menores o iguales que  $10^9$ .

### Subtareas

1. (2 puntos)  $n \leq 2$
2. (5 puntos) La secuencia tiene todos los elementos iguales.
3. (7 puntos) Todos los elementos de la secuencia son distintos.
4. (10 puntos)  $n \leq 20$
5. (76 puntos) Sin restricciones adicionales.

### 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()`.

# C

## Prohibidos los 11

Que hay distintos sistemas de numeración es algo conocido por todos. Quien más quien menos sabe que los humanos utilizamos la base 10 y que los ordenadores usan la base 2 (aunque en realidad si la máquina hablara te diría que también utiliza base 10 pues, al fin y al cabo, en base 2 el “10” representa, precisamente, el número 2. . .). Bases mucho menos conocidas pero con propiedades simpáticas son, por ejemplo, la base 1, la base raíz de 10, la base 3 balanceada o la base factorial.

Hay otro curioso sistema de numeración que, igual que el sistema binario, restringe los dígitos utilizados al 0 y al 1 pero que impone una restricción adicional: no se pueden tener dos unos seguidos en los números. Con esa restricción, tras el 10 viene el 100 y tras el 101 viene el 1000. Los primeros números del sistema serían:

Base 10	Binarios sin “11”
0	0
1	1
2	10
3	100
4	101
5	1000
6	1001
7	1010
...	...

Dado un número  $n$  en decimal, ¿cuál es su representación en este sistema de numeración que prohíbe los 11?

### Entrada

La entrada comienza con el número de casos de prueba que vendrán a continuación. Cada caso ocupa una línea y tendrá un número  $n$  en base 10.

### Restricciones

$$0 \leq n \leq 10^{15}$$

### Subtareas

1. (3 puntos)  $n \leq 7$
2. (2 puntos)  $n \leq 12$
3. (5 puntos)  $n \leq 33$
4. (40 puntos)  $n \leq 10000$
5. (50 puntos) Sin restricciones adicionales.

### Salida

Por cada caso de prueba se escribirá, en una línea independiente, el número  $n$  utilizando el sistema de representación binario que prohíbe tener dos unos seguidos.

### Entrada de ejemplo

3
0
3
7

## Salida de ejemplo

```
0  
100  
1010
```

# D

## La nevada del siglo

¡Y pasamos a dar las noticias del tiempo! Continúa el frío intenso en el norte del país. La borrasca Tremendinne sigue azotando la región Una Candena Montañosa, como saben ustedes con muchos pueblitos dispersos por las montañas. Tras la gran nevada de esta noche, todas sus carreteras han quedado bloqueadas. Cada una de estas carreteras conecta dos pueblos, y se ha estimado el tiempo en horas que se va a tardar en limpiar y volver a abrir cada una de las carreteras. Estos tiempos son independientes entre sí, ya que se trabaja en todas las carreteras a la vez. Ya se sabe que en todos los pueblos están muy concienciados de la importancia de no quedarse aislados, por lo que tienen equipos de voluntarios que salen a limpiar cada una de las carreteras que llegan al pueblo cada vez que hay una nevada como esta.



Los vecinos se preguntan cuándo podrán volver a viajar a cualquier otro pueblo de la región (todos saben que el mejor queso se encuentra en Villa Leche de Masarriba, o el mejor pan en Masamadre de Centeno). O lo que es lo mismo, cuál es el mínimo número de horas que tienen que pasar para que todos los pueblos vuelvan a estar conectados. ¿Lo conseguirán antes de la próxima nevada?

¡Seguiremos informando!

### Entrada

La entrada comienza con el número  $T$  de casos de prueba que aparecerán a continuación.

La primera línea de cada caso contiene dos números enteros: el número de pueblos  $N$  y el número de carreteras  $M$ . A continuación aparecen  $M$  líneas, cada una describiendo una carretera: primero aparecen los números de los pueblos  $A$  y  $B$  que la carretera conecta (en ambos sentidos) y a continuación el número de horas  $H$  estimadas que requerirá la limpieza de esa carretera.

No hay carreteras que unan un pueblo consigo mismo, ni dos carreteras que unan el mismo par de pueblos.

### Restricciones

- $1 \leq N \leq 200.000$
- $0 \leq M \leq 200.000$
- $1 \leq A, B \leq N$
- $1 \leq H \leq 10.000$
- $\sum_{i=1}^T N_i + M_i \leq 400.000$

### Subtareas

1. (5 puntos)  $N \leq 3$
2. (10 puntos)  $M = N - 1$ , y las carreteras (una vez que todas están limpias) permiten ir de cualquier pueblo a cualquier otro.
3. (35 puntos)  $N \leq 100$ ,  $M \leq 1.000$
4. (50 puntos) Sin restricciones adicionales.

## Salida

Para cada caso de prueba se escribirá una línea con el número de horas que tienen que pasar para que los pueblos vuelvan a estar conectados. Es decir, para que para cualquier pareja de pueblos exista al menos una ruta entre los dos pueblos que pase por una o más carreteras que ya han sido limpiadas y abiertas a la circulación.

Si no es posible conectar todos los pueblos, se mostrará una línea con la palabra IMPOSIBLE.

## Entrada de ejemplo

```
3
3 3
1 2 10
2 3 20
3 1 30
4 2
1 3 10
4 2 10
4 3
1 2 20
1 3 30
1 4 10
```

## Salida de ejemplo

```
20
IMPOSIBLE
30
```

# E

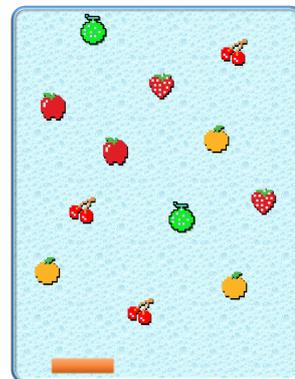
## Atrapa la fruta

“Atrapa la fruta” es el nuevo juego de moda para dispositivos móviles que tiene a la gente como loca compitiendo unos contra otros para ver quién es capaz de obtener la mejor puntuación.

El juego realmente no es muy complicado. En cada uno de los niveles aparecen frutas distribuidas por la pantalla que van cayendo a velocidad constante mientras el jugador intenta recogerlas. La forma de recogerlas es colocar la barra manejada por el jugador de forma que esta esté en la vertical de la fruta en el momento en el que llega al suelo.

Tienes sospechas de que en muchos niveles es imposible atrapar toda la fruta para obtener la puntuación máxima, y que los niveles están diseñados así de forma intencionada para que la gente siga jugando al pensar que aún pueden mejorar su puntuación.

Para no perder tiempo en estos niveles, te has propuesto crear un programa que calcule cuál es el número máximo de piezas de fruta que puedes conseguir, y así saber cuándo no merece la pena intentar mejorar tu puntuación. El primer paso ya lo tienes, pues has analizado cómo funciona y has llegado a las siguientes conclusiones:



- Aunque en pantalla las frutas ocupan más, desde el punto de vista del cálculo de la colisión entre la fruta y la barra del jugador, la fruta ocupa un único píxel.
- Tanto las frutas como la barra se mueven a intervalos regulares, todas a la vez.
- Se considera que el suelo (y la barra) están a altura 0.
- La velocidad de cada pieza de fruta es constante, por lo que desciende siempre a la misma velocidad. Eso sí, las velocidades de caída pueden ser distintas entre frutas.
- La velocidad de la barra del jugador depende de cuánto gire este el móvil, pero siempre será un número entero y nunca excederá (en valor absoluto) una cierta cantidad que depende del nivel.
- Los niveles están diseñados de tal forma que se garantiza que tras dejar pasar el número de intervalos necesarios, las frutas terminarán tocando el suelo (es decir, estarán a altura 0). Es en ese momento cuando, si el jugador quiere coger esa pieza de fruta, deberá tener la barra situada justo debajo.
- La barra tiene un ancho  $L$  medido en píxeles. Eso significa que si el píxel de más a la izquierda de la barra está en la posición  $V$  del eje  $x$  de coordenadas, la barra podría recoger todas las piezas de fruta que se encuentren en el intervalo  $[V, V + L)$ , el cual incluye la posición  $V$  pero no la  $V + L$ .

### Entrada

La entrada comienza con el número de casos de prueba que vendrán a continuación, cada uno describiendo un nivel del juego a lo largo de varias líneas.

La primera línea de cada caso contiene cuatro números enteros:  $N$ , el número de piezas de fruta presentes en el nivel;  $L$ , la longitud en píxeles de la barra del jugador;  $K$ , el número máximo de píxeles que la barra puede desplazarse horizontalmente por cada intervalo de tiempo; y  $S$ , la coordenada  $x$  del píxel izquierdo de la barra al empezar el nivel.

Las siguientes  $N$  líneas contienen cada una los valores enteros  $X_i$ ,  $Y_i$  y  $V_i$ , indicando que la pieza de fruta  $i$ -ésima se encuentra en el píxel  $(X_i, Y_i)$  al empezar el nivel, y caerá con una velocidad constante de  $V_i$  píxeles por intervalo de tiempo ( $Y_i$  será múltiplo de  $V_i$  para que toque el suelo).

### Restricciones

- $1 \leq N \leq 50$

- $1 \leq L \leq 10^4$
- $0 \leq K \leq 10^9$
- $-10^9 \leq S \leq S+L \leq 10^9$
- $-10^9 \leq X_i < 10^9$
- $0 \leq Y_i \leq 10^9$ ,  $Y_i$  es múltiplo de  $V_i$
- $1 \leq V_i \leq 10^9$

### Subtareas

1. (2 puntos)  $K = 0$
2. (5 puntos) Todas las coordenadas  $X_i$  son iguales.
3. (8 puntos) Todas las coordenadas  $Y_i$  son iguales y todas las velocidades  $V_i = 1$ .
4. (20 puntos)  $L = 1$
5. (20 puntos)  $L \leq 50$
6. (45 puntos) Sin restricciones adicionales.

### Salida

Por cada caso de prueba el programa deberá escribir en una línea independiente un único número indicando el número máximo de piezas de fruta que es posible recoger en el nivel.

### Entrada de ejemplo

```
2
4 10 10 -5
-6 4 2
13 1 1
20 10 2
-20 10 2
3 10 5 0
-11 10 5
21 10 5
-1 0 10
```

### Salida de ejemplo

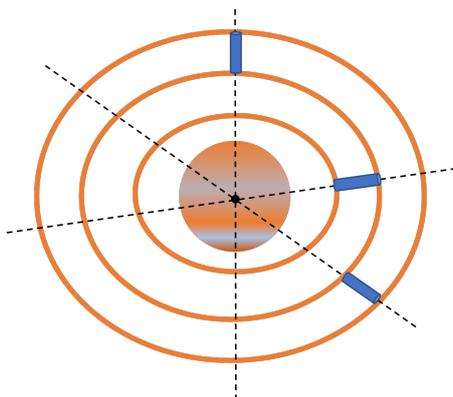
```
3
0
```

# F

## Por los anillos de Saturno

La NASA acaba de publicar el *tweet* más viral de la historia: “Hemos encontrado vida alienígena en Saturno”. Así, sin más información. En pocas horas han explotado todas las redes sociales, con millones de usuarios pidiendo saber más. Sin embargo, la NASA está ocupada estudiando el sistema de transporte de estos alienígenas. Resulta que en vez de vivir en el planeta, ¡viven en sus anillos!

Pero claro, entre estos anillos hay grandes distancias, por lo que los alienígenas han construido tubos de propulsión entre ellos, es decir, túneles que lanzan a los alienígenas de un anillo a otro ¡instantáneamente! Los túneles funcionan en ambas direcciones y tienen la peculiaridad de que conectan posiciones con el mismo ángulo en anillos contiguos, como se ve en el diagrama:



Además, dentro de un mismo anillo, los alienígenas pueden moverse caminando. La velocidad a la que se mueven en un mismo anillo es constante, pero puede ser distinta en distintos anillos. Dentro de un anillo, los alienígenas pueden moverse en ambas direcciones.

Dadas dos posiciones, ¿cuál es el mínimo tiempo que puede tardar un alienígena en ir de una a otra?

### Entrada

La entrada comienza con el número de casos de prueba que aparecen a continuación. Cada caso consta de:

- Una línea con dos enteros, el número de anillos  $N$  y el número de túneles  $K$ .
- $K$  líneas, cada una con dos enteros  $a$  y  $b$ , siendo  $a$  el índice del anillo con índice inferior de los dos que conecta el túnel y  $b$  el ángulo en el que se encuentra el mismo.
- Una línea con  $N$  enteros  $t_1, t_2, \dots, t_N$ , siendo  $t_i$  el tiempo que tarda un alienígena en recorrer un grado de distancia angular en el  $i$ -ésimo anillo.
- Una línea con 4 enteros:  $d$ , el índice del anillo en el que empieza el alienígena y  $g$  el grado en el que empieza;  $e$ , el índice del anillo en el que acaba el alienígena y  $h$  el grado en el que acaba.

### Restricciones

- $1 \leq N \leq 3.000$
- $0 \leq K \leq 10^5$
- $1 \leq t_i \leq 1.000$  para todo  $i$
- Los índices de túneles serán números entre 1 y  $N$  y los grados serán números entre 0 y 359.

- No habrá dos túneles conectando el mismo par de anillos en el mismo grado y la salida y la meta no serán idénticas entre sí.

### Subtareas

1. (7 puntos)  $N = 1, K = 0$
2. (9 puntos)  $K = 360 \cdot (N - 1)$
3. (14 puntos) Habrá exactamente un túnel entre cada pareja de anillos contiguos.
4. (30 puntos) Todos los  $t_i$  son iguales.
5. (40 puntos) Sin restricciones adicionales.

### Salida

Se escribirá una línea por cada caso de prueba con el tiempo mínimo que tardará el alienígena en llegar desde el punto donde comienza al punto donde termina. Si no es posible que el alienígena llegue a su destino, se escribirá IMPOSIBLE. Se garantiza que, en caso de haber una solución, esta no superará el valor  $10^9$ .

### Entrada de ejemplo

```
2
3 4
1 20
1 350
2 170
2 185
10 10 10
1 0 3 170
4 2
1 90
3 0
10 20 30 40
3 90 1 0
```

### Salida de ejemplo

```
1700
IMPOSIBLE
```

# G

## A toda potencia

El servicio secreto británico ha proporcionado a su agente más extravagante, Agustín Powers, un nuevo bólido experimental para su nueva misión. El prototipo es tan potente como difícil de controlar y el espía está entrenándose para ello en un circuito del cuartel general en Londres. Según las especificaciones, el irresistible botón turbo produce un desplazamiento instantáneo,  $D$ , que depende de la concentración de partículas acelerantes en sus cinco motores ( $a$ ,  $b$ ,  $c$ ,  $d$  y  $e$ ) y de la cantidad de oxígeno aportada ( $n$ ) que debe ser de al menos  $m$  para que la reacción tenga lugar. La expresión concreta que relaciona todas las cantidades es:



$$D = (a^n + b^n + c^n + d^n + e^n) \text{ con } n \geq m$$

Estando como está en un circuito circular de longitud  $k$ , el valor de la expresión anterior módulo  $k$  determina el punto exacto del circuito en el que el coche se parará tras el impulso conseguido con la cantidad de oxígeno  $n$ . Si quiere parar exactamente en un punto  $p$ , ¿cuál es la cantidad más pequeña de oxígeno que se lo permitirá, teniendo en cuenta que al menos debe ser  $m$  para que la combustión tenga lugar?

### Entrada

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

Cada caso comienza con una línea con la descripción del circuito (longitud  $k$ ) y del motor (la concentración de partículas acelerantes en los cinco motores,  $a$ ,  $b$ ,  $c$ ,  $d$  y  $e$ ).

A continuación aparece una línea con el número de consultas ( $1 \leq nc \leq 10.000$ ) que se quieren hacer. Por último aparecerán tantas líneas como consultas que tendrán dos números: la mínima cantidad de combustible que habrá que aplicar,  $m_i$ , y la posición  $p_i$  del circuito en la que hay que terminar (valores entre 0 y  $k - 1$ ).

### Restricciones

- $2 \leq k \leq 1000$
- $0 \leq a, b, c, d, e \leq 10^{1000}$
- $0 \leq m \leq 10^{1000}$
- Todos los números son enteros.

### Subtareas

1. (5 puntos) Ninguna respuesta es IMPOSIBLE y la distancia recorrida total no supera  $10^9$ .
2. (3 puntos)  $0 \leq a, b, c, d, e, m \leq 10$  y ninguna respuesta es IMPOSIBLE.
3. (10 puntos)  $k = 2$
4. (15 puntos)  $nc = 1$  y ninguna respuesta es IMPOSIBLE .
5. (20 puntos)  $m = 0$  y todas las bases, excepto una, son nulas.
6. (10 puntos) Todas las bases, excepto una, son nulas.
7. (37 puntos) Sin restricciones adicionales.

## Salida

Por cada caso de prueba se escribirán tantas líneas como consultas hay. Para cada una se escribirá la cantidad de combustible a aplicar que, siendo al menos  $m_i$ , deje al coche exactamente en la posición  $p_i$ . Si no es posible terminar en esa posición, se escribirá IMPOSIBLE.

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

## Entrada de ejemplo

```
2
10 0 2 0 0 0
4
0 4
2 4
3 4
3 5
24 1 2 3 4 5
2
1000 9
1000 19
```

## Salida de ejemplo

```
2
2
6
IMPOSIBLE
---
1001
1000
---
```