



Cuchicheo

Práxedes quiere regalarle una postal a Yael en el día de San Valentín, pero le da mucha vergüenza dársela directamente. Por ello, ha decidido que se la dará a un amigo, el amigo se la dará a otro amigo suyo, y así sucesivamente hasta que llegue a Yael.

Por desgracia para Práxedes, en su instituto la gente es muy cotilla y no perderá la oportunidad de comentar con sus amigos la situación. Discretamente, eso sí. Así pues, cuando una persona (exceptuando Yael y Práxedes) reciba la postal, el *nivel de cuchicheo* aumentará en una unidad en esa persona y en todos sus amigos. (Inicialmente, el nivel de cuchicheo de todas las personas es 0).

Práxedes no quiere que nadie hable demasiado sobre el tema, por ello, conociendo las relaciones de amistad de todas las personas del instituto, quiere diseñar la “ruta” que seguirá la postal hasta Yael que minimice **el máximo** entre todos los niveles de cuchicheo de todas las personas. Ayuda a Práxedes a encontrar este mínimo nivel y una ruta para conseguirlo.

Entrada y salida

La primera línea de la entrada contiene el número de casos T .

Por cada caso habrá una línea de entrada con dos enteros n, m : el número de personas en el instituto y el número de pares de amigos. Las personas están numeradas de 0 a $n - 1$ y la persona 0 es Práxedes y la $n - 1$ es Yael.

A continuación, siguen m líneas, cada una con un par de enteros u, v , denotando una relación de amistad (simétrica) entre las personas u y v .

Por cada caso, si hay una forma de hacer llegar la postal a Yael, debes imprimir una línea con dos enteros c, l , donde c es el mínimo nivel de cuchicheo máximo al que es posible llegar y l es el número de personas intermediarias (sin incluir Práxedes ni Yael) por las que pasa la postal en la ruta que has encontrado. A continuación, debes imprimir una segunda línea con l enteros: las l personas intermediarias de tu ruta. En caso de que haya varias rutas que minimicen el máximo nivel de cuchicheo, puedes imprimir cualquiera de ellas. En caso de que no haya ninguna forma de hacer llegar la postal a Yael, debes imprimir una única línea con dos enteros 0 0.



Ejemplo

Entrada:

```
3
2 0
3 2
0 1
1 2
6 7
0 1
1 2
1 3
3 0
3 4
4 2
4 5
```

Salida:

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

Restricciones

$1 \leq T \leq 5\,000$.

$2 \leq n \leq 2 \cdot 10^5$, $0 \leq m \leq 2 \cdot 10^5$.

La suma de n para todos los casos y la suma de m para todos los casos son como mucho $2 \cdot 10^5$.

$0 \leq u, v \leq n - 1$, $u \neq v$, no se repetirán pares no ordenados (u, v) en la entrada.

Se garantiza que no habrá relación de amistad entre las personas 0 y $n - 1$.

Subtareas

1. (12 puntos) $n \leq 3$.
2. (20 puntos) $n \leq 8$.
3. (19 puntos) $m = n - 1$, para el i -ésimo par (u, v) dado en la entrada ($1 \leq i \leq m$) se cumple que $u = i$ y $v < i$.
4. (49 puntos) Sin restricciones adicionales.



Goulash

El *goulash de Szeged* es un plato de carne estofada originario de Szeged, Hungría, el lugar donde se celebrará la IOI 2023. Pero a Félix le gusta más el goulash que hacen en República Checa, porque tiene *knedlíky*.

Félix quiere comer exactamente N knedlíky. Hay A restaurantes que ofrecen goulash con 4 knedlíky, B con 5 knedlíky, y C con 6 knedlíky. Si Félix puede ir a comer a algunos de esos restaurantes y no quiere repetir en ningún restaurante, ¿es posible que consiga comer exactamente N knedlíky?

Formalmente, debes decir si existen enteros $0 \leq x \leq A, 0 \leq y \leq B, 0 \leq z \leq C$ tales que $4x + 5y + 6z = N$.

Entrada y salida

La primera línea de la entrada contiene el número de casos T .

Por cada caso habrá una línea de entrada con cuatro enteros N, A, B, C .

Por cada caso, debes imprimir una línea con SI si Félix puede comer N knedlíky y NO en caso contrario.

Ejemplo

Entrada:

```
4
0 1 2 3
26 1 2 3
4 0 0 0
11 2 1 0
```

Salida:

```
SI
SI
NO
NO
```

Explicación: En el primer caso, Félix puede comer 0 knedlíky no yendo a ningún restaurante. En el segundo, puede ir a un restaurante del primer tipo, a 2 del segundo y a 2 del tercero para comer $1 \cdot 4 + 2 \cdot 5 + 2 \cdot 6 = 26$ knedlíky.

Restricciones

$1 \leq T \leq 100\,000$.

$0 \leq N, A, B, C \leq 10^8$.



Subtareas

1. (20 puntos) $A, B, C \leq 100$.
2. (19 puntos) $N, A, B, C > 100$.
3. (18 puntos) $B = 0$.
4. (17 puntos) $A \cdot B \cdot C = 0$.
5. (26 puntos) Sin restricciones adicionales.



Lista Valiosa

Dada una lista de n enteros positivos $L = (a_1, \dots, a_n)$, Ana define su valor $V(L)$ como:

$$V(L) = n + \sum_{i=1}^{n-1} |a_i - a_{i+1}| = n + |a_1 - a_2| + |a_2 - a_3| + \dots + |a_{n-1} - a_n|$$

Dados v y k , ayuda a Ana a encontrar la k -ésima lista en orden lexicográfico que tiene valor igual a v .

Recordamos que, dadas dos listas $L_1 = (a_1, \dots, a_n)$ y $L_2 = (b_1, \dots, b_m)$, L_1 viene antes que L_2 en orden lexicográfico si:

- Existe un $j \leq \min(n, m)$ tal que $a_i = b_i$ para todo $1 \leq i < j$ y $a_j < b_j$, o:
- $a_i = b_i$ para todo $1 \leq i \leq n$ y $n < m$.

Entrada y salida

La primera línea de la entrada contiene el número de casos T .

Por cada caso habrá una línea de entrada con dos enteros v , k .

Por cada caso debes imprimir una línea con los enteros de la lista pedida separados por espacios.

Ejemplo

Entrada:

```
5
5 3
1 1
2 4
3 10
4 10000000000000000
```

Salida:

```
1 1 2 2
1
4 4
4 4 4
142857142857144 142857142857144 142857142857143
```

Explicación del primer caso: Las listas de valor 5 son, en orden lexicográfico: $(1, 1, 1, 1, 1)$, $(1, 1, 1, 2)$, $(1, 1, 2, 2)$, $(1, 1, 3)$, $(1, 2, 1)$, $(1, 2, 2, 2)$, $\dots$



Restricciones

$$1 \leq T \leq 200\,000.$$

$$1 \leq v \leq 10^6.$$

$$1 \leq k \leq 10^{15}.$$

La suma de v para todos los casos será como mucho 10^6 .

Subtareas

1. (3 puntos) $v = 1$.
2. (10 puntos) $v \leq 5$.
3. (4 puntos) $k = 1$.
4. (8 puntos) $k \leq 5$.
5. (18 puntos) $v, k \leq 100$, la suma de $v \cdot k$ para todos los casos será como mucho 10^6 .
6. (19 puntos) $v \leq 1\,000$.
7. (20 puntos) La suma de k para todos los casos será como mucho 10^6 .
8. (18 puntos) Sin restricciones adicionales.



Números cíclicos

Empúries fue una ciudad griega, posteriormente romana, situada cerca del mar en el golfo de Roses en la costa Brava. Actualmente son unas ruinas que se pueden visitar. Pepa y María son dos arqueólogas que trabajan en ellas.

Durante el último año han encontrado trozos de piedra con números en representación decimal grabados; parece ser que algún comerciante los trajo de algún lugar desconocido. Por lo que han observado, estos números tienen una propiedad muy especial. Si movemos la última cifra al principio nos queda un número que es múltiplo del original.

Por ejemplo, el número 142857 tiene esta propiedad ya que si movemos la última cifra al principio nos queda 714285, que es 142857×5 .

El problema que tienen ahora es que han encontrado algunas piedras que están incompletas, lo que significa que puede ser que falten cifras al principio y al final del número.

Conociendo la propiedad característica de los números que se representan en estas piedras, Pepa y María quieren saber cuál es el mínimo número de cifras que tienen que añadir delante o detrás de los números encontrados para conseguir formar un número con la propiedad deseada. Si no es posible conseguirlo, también quieren saberlo, ya que esto significaría que su investigación va por mal camino.

Siguiendo el ejemplo anterior, si encontramos una piedra con el número 4285, la respuesta sería 2 ya que si añadimos un 1 delante y un 7 detrás conseguimos el número 142857.

(Consideramos que un número válido no tiene ceros a la izquierda)

Entrada y Salida

Cada archivo empieza con un entero T , el número de casos.

A continuación siguen T líneas cada una con un entero k representando el número inscrito en una piedra.

Para cada caso, escribid una línea con el número de cifras que hay que añadir como mínimo para conseguir un número con la propiedad deseada o "No" si no es posible conseguirlo.

Ejemplos

Ejemplo 1:

Entrada:

```
4
142856
142857
230768
230769
```

Salida:

```
No
0
No
0
```



Ejemplo 2:

Entrada:

4
5517
9831508
4745762
7128

Salida:

24
No
51
No

Restricciones

$$1 \leq T \leq 10^5$$

$$1 \leq k \leq 10^{1000000}$$

Sea S la suma del número de dígitos de k para todos los casos. Entonces $S \leq 1\,000\,000$.

Subtareas

1. (5 puntos) La respuesta será 0 o “No” y $k \leq 10^{15}$.
2. (10 puntos) $k \leq 10^{15}$.
3. (4 puntos) La respuesta será 0 o “No” y $S \leq 1000$.
4. (32 puntos) $S \leq 1000$.
5. (14 puntos) La respuesta será 0 o “No”.
6. (35 puntos) Sin restricciones adicionales.



Prisión

Los prisioneros de la cárcel de Szeged quieren aprovechar que su ciudad se encuentra distraída por la IOI para escapar de la cárcel. Se encuentran en la sala de llaves de la prisión, pero no saben qué llave abre la única puerta de salida.

Hay n llaves en fila, dejando 1 metro entre llave y llave, y disponen de un dispositivo que al preguntar por una de las llaves les responde con un número d , la distancia a la llave que abre la prisión. El caso es que el dispositivo se ha contagiado por el espíritu de la IOI, ahora este responderá con $d \& x$, donde $0 \leq x \leq \frac{n}{2}$ es un entero que desconocen.

La operación $\&$ (AND bit a bit) de dos enteros no negativos se calcula de la siguiente forma: se escriben los dos números en binario y el resultado es el número que escrito en binario tiene 1 en las posiciones en las que ambos operandos tienen un 1, y 0 en las posiciones en las que alguno de los operandos tiene un 0.

Por ejemplo, $26 \& 14 = 10$, porque 26 en binario es 11010, 14 en binario es 1110, y el número resultante de poner 1 en las posiciones en las que ambos operandos tienen un 1 es 1010, que si lo escribimos en base decimal es 10.

$$\begin{array}{r} 11010 \\ \& 01110 \\ \hline 01010 \end{array}$$

Manuel, el cabecilla de la operación, también se ha contagiado por el espíritu de la IOI y se ha planteado un nuevo problema: encontrar el número x . Por eso te pide a ti que crees un programa que dado n , analice con el dispositivo una serie de llaves y responda con el valor de x (el lugar donde se encuentra la llave ya le es irrelevante). Ya que el dispositivo tarda mucho tiempo en realizar los cálculos, Manuel quiere reducir el número de preguntas que realiza por temor a que los guardias le descubran.

Entrada y salida

Este es un problema interactivo. Debes refrescar la salida cada vez que imprimas datos (cout << endl o cout << flush en C++, System.out.flush() en Java, stdout.flush() en Python).

La primera línea de la entrada contiene el número de casos T .

Por cada caso, deberás leer primero una línea con el número n . Después de leer este valor, debes hacer preguntas al interactor:

Para hacer una pregunta, debes imprimir una línea con el formato $? i$, donde $0 \leq i < n$ es el índice de la llave (nótese que comienza en 0). A continuación, se te responderá con una línea con el número $d \& x$ escrito en decimal. En caso de que realices una pregunta inválida o excedas el límite de preguntas, se te responderá con -1, y en ese caso tu programa debe terminar inmediatamente.

Para dar la respuesta, debes imprimir una línea con el formato $! x$, donde $0 \leq x \leq \frac{n}{2}$ es el valor de x que has determinado. A continuación, debes leer una respuesta con un carácter, que será:

- +, si se pasa al caso siguiente. En ese caso tu programa debe leer el nuevo entero n a continuación y continuar con el nuevo caso.



- -, si no se pasa al caso siguiente, o bien porque ya se han acabado los casos o bien porque has dado una respuesta incorrecta. En ese caso, tu programa debe terminar inmediatamente.

Ejemplo

Aquí mostramos una interacción de ejemplo en la que, para clarificar, se muestra para cada caso el valor de x y de la posición k de la llave.

Juez	Programa	Comentario
2		
3		$x = 1, k = 0$
	? 0	
0		$ 0 - 0 \& 1 = 0$
	? 1	
1		$ 1 - 0 \& 1 = 1$
	! 1	
+		
5		$x = 2, k = 1$
	? 0	
0		$ 0 - 1 \& 2 = 0$
	? 1	
0		$ 1 - 1 \& 2 = 0$
	? 2	
0		$ 2 - 1 \& 2 = 0$
	? 3	
2		$ 3 - 1 \& 2 = 2$
	? 4	
2		$ 4 - 1 \& 2 = 2$
	! 2	
—		

Restricciones

$$1 \leq T \leq 100.$$

$$2 \leq n \leq 10^9.$$

$$0 \leq x \leq \frac{n}{2}.$$

Puedes hacer como mucho 50 preguntas. Dar la respuesta no cuenta como pregunta.

El valor de x y el índice de la llave están fijo al comienzo de la interacción, es decir, no cambian adaptivamente según las preguntas que hagas.



Subtareas

1. (10 puntos) $n = 2$.
2. (15 puntos) $n \leq 50$.
3. (25 puntos) $n \leq 1000$.
4. (15 puntos) La llave que abre la prisión se halla en la posición 0.
5. (20 puntos) La llave que abre la prisión se halla en la posición 0 o en la posición $n - 1$.
6. (15 puntos) Sin restricciones adicionales.

Adicionalmente, la puntuación que obtienes en una subtarea depende del número de preguntas que hagas: para obtener una puntuación completa debes realizar como máximo 4 preguntas y para obtener una puntuación positiva debes realizar como máximo 50 preguntas. La puntuación de cada subtarea es multiplicada por un multiplicador $M(q)$, donde q es el máximo número de preguntas que has hecho en los casos de esa subtarea. El valor de $M(q)$ viene dado por:

$$M(q) = \begin{cases} 0 & q > 50 \\ 0,8 + \frac{50-q}{250} & 50 \geq q > 4 \\ 1,0 & 4 \geq q \end{cases}$$