



Múltiplo o Divisor

El juez tiene un número n oculto que debes determinar. Para ello, puedes hacer preguntas de los siguientes dos tipos:

1. Proporcionas un entero m y el juez te responde “Sí” si m es múltiplo de n o si m es divisor de n , y “No” en caso contrario. Esta pregunta tiene coste 1.
2. Proporcionas un entero m y el juez te responde un entero d que es la mínima distancia de m a alguno de los múltiplos o divisores de n , es decir, $d = \min_{x \in D_n \cup M_n} |m - x|$, donde D_n es el conjunto de divisores de n y M_n es el conjunto de múltiplos de n . La i -ésima vez que haces esta pregunta, la pregunta tiene coste i^2 .

Tu objetivo es determinar el número n sin que la suma de costes de tus preguntas sea demasiado grande.

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 . En cada caso, el juez tendrá un número n que deberás adivinar haciendo preguntas.

Para hacer una pregunta, debes imprimir una línea con el formato `? t m`, donde $1 \leq t \leq 2$ es el tipo de pregunta y $1 \leq m \leq 3 \cdot 10^{18}$ el parámetro de la pregunta. A continuación, se te responderá con una línea las palabras `SI` o `NO` en el caso de la primera pregunta, o con un entero d en el caso de la segunda pregunta. En caso de que realices una pregunta inválida o excedas el límite de coste, 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 `! n`, donde n es el valor 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 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

Juez	Programa	Comentario
2		$n = 2$ en el primer caso
	? 1 1	
SI		1 es divisor de 2
	? 2 3	
1		$1 = 3 - 2 = 3 - 4 $
	! 2	
+		$n = 6$ en el segundo caso
	? 2 8	
2		$2 = 8 - 6 $
	? 2 13	
1		$1 = 13 - 12 $
	? 2 4	
1		$1 = 4 - 3 $
	? 1 5	
NO		5 no es divisor ni múltiplo
	! 6	
-		

Restricciones

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

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

Los m de tus preguntas deben satisfacer $1 \leq m \leq 3 \cdot 10^{18}$.

Las preguntas que hagas pueden tener como máximo coste total 200 en cada uno de los casos.

El valor de n está fijo al comienzo de la interacción, es decir, no cambia adaptivamente según las preguntas que hagas.

Subtareas

1. (10 puntos) $n \leq 100$.
2. (20 puntos) $n \leq 2000$.
3. (30 puntos) $n \leq 10^9$.
4. (10 puntos) n es un número *regular*: no es divisible por primos diferentes de 2, 3 o 5.
5. (30 puntos) Sin restricciones adicionales.

Adicionalmente, la puntuación que obtienes en una subtarea depende del coste de las preguntas que hagas: para obtener una puntuación completa el coste debe ser como máximo 25 y para obtener una puntuación positiva como máximo 200. La puntuación de cada subtarea es multiplicada por un multiplicador $M(q)$, donde q es el máximo coste total de preguntas que has hecho en cada caso de esa subtarea. El valor de $M(q)$ viene dado por:

$$M(q) = \begin{cases} 0 & q > 200 \\ 0,5 + \frac{200-q}{750} & 200 \geq q > 50 \\ 0,7 + \frac{150-3q}{250} & 50 \geq q > 25 \\ 1,0 & 25 \geq q \end{cases}$$