Volteando Segmentos

Sea n un entero positivo dado. El juez tiene un número $1 \leq x \leq n$ oculto que debes determinar. Para ello, puedes hacer una serie de preguntas.

Hay una permutación $p_1, \ldots, p_n$ donde inicialmente $p_i = i$ para todo $1 \leq i \leq n$. Puedes hacer la siguiente pregunta: das un índice i y el juez te responde si $p_i < x$, $p_i = x$ o $p_i > x$. Pero después de cada pregunta, si j es el índice tal que $p_j = x$, se da la vuelta al segmento $[i, j]$ (si $i \leq j$) o $[j, i]$ (si $i > j$) de la permutación, es decir, denotando por p' la nueva permutación tras la pregunta, se tiene que $p'_i = p_j, p'_{i+1} = p_{j-1}, \ldots, p'_j = p_i$ si $i \leq j$ y análogamente $p'_j = p_i, p'_{j+1} = p_{i-1}, \ldots, p'_i = p_j$ si $i > j$.

Tu objetivo es determinar el número x sin hacer demasiadas preguntas.

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 $1 \leq i \leq n$ es el índice. A continuación, se te responderá con una línea con el carácter `<`, `>`, o `=`, según si $p_i < x$, $p_i > x$, o $p_i = x$ respectivamente. En caso de que realices una pregunta inválida o excedas el límite de preguntas, se te responderá con `-`, y en ese caso tu programa debe terminar inmediatamente.

Para dar la respuesta, debes imprimir una línea con el formato `! x`, donde $1 \leq x \leq n$ 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.

En cualquier caso, tu programa debería terminar sólo después de haber leído un `-`.

Ejemplo

Aquí mostramos una interacción de ejemplo en la que, para clarificar, se muestra el valor de x de cada caso y el estado de la permutación p al hacer cada pregunta.

Juez	Programa	Comentario
2		
5		$x = 3$
	? 1	$[1, 2, 3, 4, 5]$
<		$p_1 = 1 < 3$
	? 2	$[3, 2, 1, 4, 5]$
<		$p_2 = 2 < 3$
	? 3	$[2, 3, 1, 4, 5]$
<		$p_3 = 1 < 3$
	? 4	$[2, 1, 3, 4, 5]$
>		$p_4 = 4 > 3$
	? 5	$[2, 1, 4, 3, 5]$
>		$p_5 = 5 > 3$
	? 3	$[2, 1, 4, 5, 3]$
>		$p_3 = 4 > 3$
	! 3	
+		
3		$x = 1$
	? 3	$[1, 2, 3]$
>		$p_3 = 3 > 1$
	? 3	$[3, 2, 1]$
=		$p_3 = 1 = x$
	? 1	$[3, 2, 1]$
>		$p_1 = 3 > 1$
	? 2	$[1, 2, 3]$
>		$p_2 = 2 > 1$
	! 1	
−		

Restricciones

$1 \leq T \leq 100$.

$2 \leq n \leq 30\,000$.

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

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

Subtareas

1. (4 puntos) $n = 2$.
2. (18 puntos) $n \leq 40$.
3. (14 puntos) O bien $x \leq 40$, o bien $n - x + 1 \leq 40$, o bien $\left|\frac{n}{2} - x\right| \leq 20$.
4. (31 puntos) $n \leq 600$.
5. (33 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 **15** 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:

$$M(q) = \begin{cases} 0 & q > 50 \\ 0{,}5 + \frac{50-q}{70} & 50 \geq q > 15 \\ 1{,}0 & 15 \geq q \end{cases}$$