

¿On ta Wally?

Nombre del problema	whereswaldo
Límite de tiempo	11 segundos
Límite de memoria	1 gigabyte

Existe una permutación escondida P_0, P_1, \dots, P_{N-1} de longitud N . Se garantiza que se generará de manera aleatoria con distribución uniforme. La permutación contiene los números $1, 2, 3, \dots, N$ exactamente una vez, en algún orden desconocido.

Puedes escoger dos posiciones l y r , y hacer preguntas de la forma: "¿Cuál es la suma de $P_l + P_{l+1} + \dots + P_r$?"

Tu tarea es encontrar la posición del 1 en P haciendo la menor cantidad posible de preguntas. Los puntos que obtengas dependerán del número de preguntas hechas.

Interacción

Tu programa debe leer primero dos enteros en una línea, T y N . T es el número de rondas que correrá tu programa para probarlo y N es la longitud de P .

Después vienen las T rondas:

Al inicio de la ronda puedes empezar a hacer preguntas. Imprime una línea con "? a b" para preguntar cuál es la suma de los números que están en el rango entre a y b inclusive ($0 \leq a \leq b \leq N - 1$).

Después de cada pregunta, tu programa debe leer un entero, que es la suma de los números en el intervalo.

Una vez que encuentres la posición del 1, imprime una línea con el formato "! i", donde i es el índice tal que $P_i = 1$. Después de que imprimas este número, empezará la siguiente ronda.

Asegúrate de hacer flush del stream de salida después de hacer una pregunta, si no tu programa puede obtener el veredicto Time Limit Exceeded. En Python, `print()` hace flush automáticamente. En C++, `cout << endl;` hace flush además de imprimir una línea nueva; si usas `printf` deberás usar `fflush(stdout)`.

Límites y Evaluación

Tu programa será evaluado con **un solo caso de prueba, con $N = T = 1000$** . Se garantiza que la permutación en cada ronda será **generada aleatoriamente**.

Si en alguna ronda tu salida es incorrecta, obtendrás el veredicto de *Wrong Answer*.

De lo contrario, tu puntaje será calculado de la siguiente manera:

$$\text{puntaje} = \min\left(220 - \frac{M}{2500}, 100\right) \text{ puntos,}$$

donde M es el número de preguntas que tu programa hizo en total contando todas las T rondas.

El puntaje se redondeará al siguiente entero. Si el puntaje es negativo, se tomará como cero.

Entonces, si haces más de 550 000 preguntas, recibirás 0 puntos, y si haces 300 000 o menos, recibirás 100 puntos. En medio de eso, tu puntuación crece linealmente.

Herramienta de pruebas

Para facilitarte poder probar tu solución, te damos una herramienta simple que puedes descargar. Ve a "attachments" al final de la página del problema en kattis. El uso de la herramienta es opcional y puedes cambiarla. Ten en cuenta que el evaluador oficial en kattis es diferente de esta herramienta de pruebas.

Uso de ejemplo (con $T=1000$, $N=10$):

Para programas en python, si tenemos `solution.py` (se corre normalmente `pypy3 solution.py`) corre:

```
python3 testing_tool.py pypy3 solution.py <<<"1000 10"
```

Para programas en C++, primero compílalo (por ejemplo `g++ -std=gnu++17 solution.cpp -o solution.out`) y luego corre:

```
python3 testing_tool.py ./solution.out <<<"1000 10"
```

Ejemplo

En el caso de prueba de ejemplo, $T = 2$ y $N = 10$. Para la primera de estas dos rondas, la permutación escondida es "6 10 8 7 9 1 2 4 5 3". La primera pregunta ? 0 9 pregunta por la

suma de todos los números que es efectivamente 55, luego la segunda pregunta ? 0 4 pregunta por $6 + 10 + 8 + 7 + 9 = 40$.

salida del evaluador	tu salida
2 10	
	? 0 9
55	
	? 0 4
40	
	? 5 5
1	
	! 5
	? 0 0
1	
	! 0