

B. Montaña rusa a oscuras

Nombre del problema	Dark Ride
Límite de tiempo	1 segundo
Límite de memoria	1 gigabyte

Erika consiguió recientemente un trabajo en el parque de atracciones Phantasialand, cerca de Bonn. Fue contratada para controlar las luces de los cuartos por los que pasa una montaña rusa a oscuras.

La montaña rusa pasa a través de N cuartos, numerados del 0 al $N - 1$. Los cuartos se recorren en orden, comenzando en el cuarto 0 y terminando en el cuarto $N - 1$. Las luces en los cuartos están controladas por N apagadores (también numerados del 0 al $N - 1$), uno para cada cuarto. El s -ésimo apagador (para $0 \leq s < N$) controla la luz del cuarto p_s .

La jefa de Erika le ha pedido que prenda las luces en el primer y último cuarto y apague las luces en todos los demás. Suena fácil, ¿verdad? Solo necesita encender los dos apagadores A y B tal que $p_A = 0$ y $p_B = N - 1$ (o $p_B = 0$ y $p_A = N - 1$). Desafortunadamente, Erika no puso atención cuando su jefa le explicó los controles, por lo que **ella no recuerda el arreglo p – es decir, qué apagador controla qué cuarto.**

Erika debe resolver esto antes de que su jefa se de cuenta. Al inicio de cada recorrido, Erika apaga todas las luces. Después, ella enciende algún subconjunto de apagadores. Como el recorrido ocurre de cuarto en cuarto, cada vez que el carrito de la montaña rusa pasa de un cuarto iluminado a uno oscuro o viceversa, Erika escuchará a los pasajeros gritar de la emoción. La velocidad del carrito puede variar, por lo que Erika no puede inferir directamente que cuarto está iluminado, pero al menos escuchará cuantas veces gritaron los pasajeros. Esto es, ella conocerá cuantas veces el carrito pasa de un cuarto iluminado a uno oscuro, o de un cuarto oscuro a un cuarto iluminado.

¿Puedes ayudar a Erika a determinar cuales apagadores controlan la luz del primer y último cuarto antes de que su jefa se de cuenta? Puedes realizar a lo más 30 recorridos para lograrlo.

Interacción

Este es un problema interactivo.

- Tu programa debe comenzar leyendo una línea con un entero N : el número de cuartos en la montaña rusa.
- Después, tu programa debe interactuar con el evaluador. Para realizar un recorrido, debes imprimir una línea con un signo de interrogación "?", seguido de una cadena de longitud N formada por 0's (apagado) y 1's (encendido), indicando tu configuración para los N apagadores. Después, tu programa debe leer un entero ℓ ($0 \leq \ell < N$), el número de veces que Erika escucha gritar a los pasajeros.
- Cuando quieras responder, imprime una línea con un signo de exclamación "!", seguido por dos enteros A y B ($0 \leq A, B < N$). Para que tu respuesta sea aceptada, estos deben ser los índices de los apagadores que controlan los cuartos en los extremos del recorrido, en cualquier orden. Después de esto tu programa debe terminar.

El evaluador no se adapta durante la interacción, esto significa que arreglo secreto p es determinado desde antes de que la interacción comience.

Asegúrate de hacer flush del stream de salida después de cada línea que imprimas, de lo contrario tu programa podría obtener el veredicto Time Limit Exceeded. En Python, esto sucede automáticamente si usas `input()` para leer líneas. En C++, `cout << endl;` también hace flush además de imprimir una línea nueva; si usas `printf` deberás usar `fflush(stdout)`.

Límites y Evaluación

- $3 \leq N \leq 30\,000$.
- Puedes realizar a lo más 30 recorridos (imprimir la respuesta final no cuenta como un recorrido). Si excedes este límite, obtendrás el veredicto "Wrong Answer".

Tu solución se evaluará con un conjunto de grupos de casos de prueba, cada grupo otorga un valor determinado de puntos. Cada grupo contiene un conjunto de casos de prueba. Para obtener los puntos de un grupo, tienes que resolver todos los casos de prueba de ese grupo.

Grupo	Puntos	Límites
1	9	$N = 3$
2	15	$N \leq 30$
3	17	$p_0 = 0$, es decir, el apagador 0 controla el cuarto 0
4	16	N es par, con el apagador de uno de los cuartos en los extremos del recorrido en la primer mitad ($0 \leq a < \frac{N}{2}$) y el otro en la segunda mitad de los apagadores ($\frac{N}{2} \leq b < N$)
5	14	$N \leq 1000$
6	29	Sin restricciones adicionales

Herramienta de Pruebas

Para facilitarte poder probar tu solución, te damos una herramienta simple que puedes descargar. Ve a la sección "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 a esta herramienta de pruebas.

Para usar la herramienta, crea un archivo de entrada, como "sample1.in", el cual debe empezar con un número N seguido de una línea con p_0, p_1, \dots, p_{N-1} , describiendo la permutación secreta de apagadores. Por ejemplo:

```
5
2 1 0 3 4
```

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

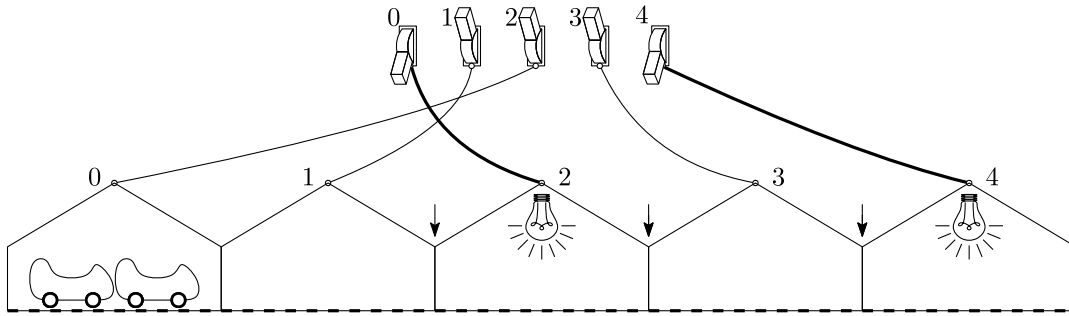
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

Para programas en C++, primero compílalo (por ejemplo `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) y luego corre:

```
python3 testing_tool.py ./solution.out < sample1.in
```

Ejemplo

En el primer ejemplo, la permutación secreta es $[p_0, p_1, p_2, p_3, p_4] = [2, 1, 0, 3, 4]$. Este ejemplo está dentro de los límites de los grupos 2, 5, y 6. Primero, el programa lee un entero $N = 5$. Después, el programa solicita un recorrido con $K = 2$ apagadores encendidos: el apagador 4 y el apagador 0. Estos controlan los cuartos $p_4 = 4$ y $p_0 = 2$; observa la imagen de abajo. Erika escucha 3 gritos (marcados por flechas en la imagen): el primero cuando el carrito pasa del cuarto oscuro 1 al cuarto iluminado 2; el segundo cuando pasa del cuarto iluminado 2 al cuarto oscuro 3; y el tercero cuando pasa del cuarto oscuro 3 al cuarto iluminado 4. Entonces, el programa solicita otro recorrido donde los cuartos p_0, p_2 , and p_3 son iluminados, haciendo que Erika escuche 3 gritos. Finalmente, el programa responde con $A = 2$ y $B = 4$, que es la respuesta correcta ya que controlan el primer y último cuarto ($p_2 = 0$ and $p_4 = 4$). Observa que $A = 4$ y $B = 2$ también es considerada una respuesta correcta.



En el segundo ejemplo, la permutación secreta es $[p_0, p_1, p_2] = [2, 0, 1]$. Este ejemplo está dentro de los límites de los grupos 1, 2, 5, y 6. El programa solicita un recorrido donde todos los apagadores están encendidos. Dado que todos los cuartos estarán iluminados, Erika no escuchará ningún grito. En el segundo recorrido, los apagadores 1 y 0 están encendidos, haciendo que los cuartos $p_1 = 0$ y $p_0 = 2$ estén iluminados, mientras que el cuarto 1 está oscuro. Erika escuchará dos gritos: cuando el carrito pase del cuarto 0 (iluminado) al cuarto 1 (oscuro), y cuando pase del cuarto 1 (oscuro) al cuarto 2 (iluminado). En el último recorrido, ningún apagador está encendido, por lo que todos los cuartos están oscuros, y de nuevo, Erika no escuchará gritos. Entonces, el programa responde con los apagadores 1 y 0, los cuales, en efecto, controlan el primer y último cuarto. Ambas, $! 0 1''$ y $! 1 0''$, son respuestas correctas.

En el tercer ejemplo, la permutación secreta es $[p_0, p_1, p_2, p_3] = [0, 1, 2, 3]$. Este ejemplo está dentro de los límites de los grupos 2, 3, 4, 5, and 6. Observa que no es necesariamente posible inferir la respuesta después de este único recorrido, pero la solución de ejemplo adivinó la respuesta y tuvo suerte.

Primer Ejemplo

salida del evaluador	tu salida
5	
	? 10001
3	
	? 10110
3	
	! 2 4

Segundo Ejemplo

salida del evaluador	tu salida
3	
	? 111
0	
	? 110
2	
	? 000
0	
	! 1 0

Tercer Ejemplo

salida del evaluador	tu salida
4	
	? 1010
3	
	! 0 3