B. Dark Ride

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

Erika ha obtenido recientemente un trabajo en el parque de atracciones Phantasialand cerca de Bonn. Se la ha asignado al puesto que controla las luces de una atracción a oscuras.

La atracción pasa por N habitaciones, numeradas de 0 a $N-1$. Las habitaciones se atravesarán en orden, comenzando en la habitación 0 y acabando en la $N-1$. La luz es controlada por N interruptores (también numerados de 0 a $N-1$), uno para cada habitación. El interruptor s (donde $0 \leq s < N$) controla la habitación p_s.

El jefe de Erika le ha pedido que solo ilumine la primera y última habitación y que apague el resto. Parece fácil, ¿cierto? Solo necesita encontrar dos interruptores 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 prestó atención cuando su jefe le explicó los controles, y **no recuerda p – es decir, qué interruptor controla qué habitación**.

Erika necesita averiguar esta información antes que su jefe se de cuenta. Al comienzo de la atracción, Erika apagará todas las luces. Luego ella podrá encender un subconjunto de interruptores. A medida que la atracción avance, cuando pase de una habitación encendida a una apagada o viceversa, Erika escuchará a los pasajeros gritar. La velocidad de la atracción puede variar, por lo que Erika no puede determinar qué habitaciones están encendidas pero podrá escuchar el número de gritos. Es decir, aprenderá el número de veces que pasa de una habitación apagada a una iluminada o viceversa.

¿Puedes ayudar a Erika averiguar que dos interruptores controlan la primera y última habitación antes de que su jefe se de cuenta? Puedes hacer como máximo 30 intentos.

Interacción

Este es un problema interactivo.

- Tu programa debería comenzar leyendo un entero N: el número de habitaciones de la atracción.

- Entonces, tu programa debería interactuar con el grader. Para simular un viaje en la atracción, deberías imprimir el caracter "`?`" al inicio de la línea, a continuación la string de longitud N consistiendo de `0`'s (habitación apagada) y `1`'s (habitación encendida), indicando la configuración de los N interruptores. A continuación, tu programa debería leer ℓ ($0 \leq \ell < N$), el número de veces que Erika escuchará a los pasajeros gritar.
- Cuando quieras imprimir tu solución, imprime una línea con un signo de exclamación "`!`", seguido de dos enteros A y B ($0 \leq A, B < N$). Para que tu solución sea aceptada, estos deben de ser los índices que se corresponden con los interruptores para la primera y última habitación, en cualquier orden. Tras esto, tu programa debería terminar.

El grader no es adaptativo, la configuración de p será determinada antes de que la interacción comience.

Asegurate de hacer flush tras describir cada intento, de lo contrario tu programa recibirá Time Limit Exceeded. En Python, esto pasa automáticamente cada vez que usas `input()` para leer nuevas líneas. En C++, `cout << endl;` hace flush e imprime una nueva línea; en el caso de usar printf, utiliza `fflush(stdout)`.

Restricciones y Puntuación

- $3 \leq N \leq 30\,000$.
- Podrás hacer como máximo 30 intentos (la solución final no cuenta). Si excedes este límite, recibirás el veredicto "Wrong Answer".

Tu solución será evaluada en un grupo de casos de prueba, cada uno con una puntuación asociada. Cada grupo contiene un conjunto de casos de prueba. Para obtener la puntuación correspondiente al grupo, debes resolver todos los casos pertenecientes al grupo.

Grupo	Puntuación	Límites
1	9	$N = 3$
2	15	$N \leq 30$
3	17	$p_0 = 0$, es decir, el interruptor 0 controla la habitación 0
4	16	N es par, el interruptor de una habitación se encuentra en la primera mitad ($0 \leq a < \frac{N}{2}$) y el otro en la segunda mitad ($\frac{N}{2} \leq b < N$)
5	14	$N \leq 1000$
6	29	No hay restricciones adicionales

Herramienta de testing

Para facilitar la verificación de tu solución, hemos aportado una herramienta de ejemplo. Mira "attachments" al fondo de la página de Kattis. La herramienta es de uso opcional. El grader oficial difiere del descrito en la herramienta de testing.

Para usar la herramienta, crea un fichero de entrada, como "sample1.in", que debería comenzar por N seguido de una línea con $p_0, p_1, ..., p_{N-1}$ especificando la permutación. Por ejemplo:

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

Para programas de Python, como `solution.py` (normalmente ejecutados como `pypy3 solution.py`), ejecuta:

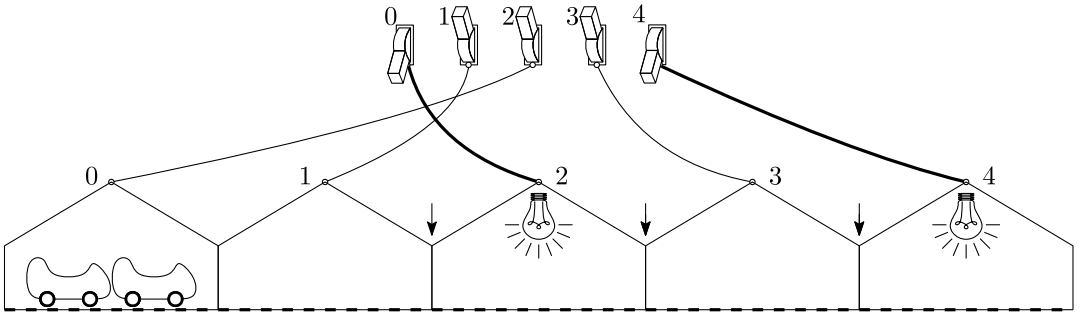
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

Para los programas en C++, primero compila el programa (por ejemplo con `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) y luego ejecuta:

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

Ejemplo

En el primer ejemplo, la permutación oculta es $[p_0, p_1, p_2, p_3, p_4] = [2, 1, 0, 3, 4]$. Esto satisface las restricciones para los grupos 2, 5, y 6. Primero, el programa lee el entero $N = 5$. Entonces, el programa simulará la atracción con dos interruptores encendidos: el interruptor 4 y el interruptor 0. Estos controlan las habitaciones $p_4 = 4$ y $p_0 = 2$; como se puede ver en la imagen inferior. Erika escucha 3 gritos (marcados como flechas en la figura): el primero cuando la atracción pasa de la habitación apagada 1 a la 2 que se encuentra encendida; el segundo grito al pasar de la habitación encendida 2 a la apagada 3; y finalmente cuando se pasa de la apagada 3 a la encendida 4. El programa luego simula otra pasada con las habitaciones p_0, p_2, y p_3 encendidas, haciendo que Erika escuche 3 gritos. Finalmente, el programa termina con $A = 2$ y $B = 4$, lo que es correcto ya que controlan las habitaciones deseadas ($p_2 = 0$ y $p_4 = 4$). Observa que $A = 4$ y $B = 2$ también sería una solución correcta.



En el segundo ejemplo, la permutación oculta es $[p_0, p_1, p_2] = [2, 0, 1]$. Esto satisface las restricciones de los grupos 1, 2, 5, y 6. El programa simula una pasada con todos los interruptores encendidos. Dado que esto significa que todas las habitaciones estarán encendidas, Erika no escuchará ningún grito. En la segunda pasada, los interruptores 1 y 0 están encendidos, encendiendo las habitaciones $p_1 = 0$ y $p_0 = 2$, mientras que la 1 estará apagada. Erika oye dos gritos: cuando la atracción pasa de la habitación 0 (encendida) a la 1 (apagada), y de la 1 (apagada) a la 2 (encendida). En la pasada final, no se enciende ningún interruptor, todas las habitaciones estarán apagadas, de nuevo Erika no escuchará ningún grito. El programa entonces acaba con los interruptores 1 y 0, los cuales se corresponden con los índices de la primera y última habitación. Tanto "`! 0 1`" como "`! 1 0`" se considerarán como respuestas correctas.

En el tercer ejemplo, la permutación oculta es $[p_0, p_1, p_2, p_3] = [0, 1, 2, 3]$. Esto satisface las restricciones de los grupos 2, 3, 4, 5, y 6.

Primer ejemplo

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

Segundo ejemplo

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

Tercer ejemplo

salida grader	tu salida
4	
	? 1010
3	
	! 0 3