

## B. Dark Ride

Nome Problema	Dark Ride
Limite di tempo	1 secondo
Limite di memoria	1 gigaottetto

Di recente Erika ha trovato un lavoro estivo al parco divertimenti Phantasialand vicino a Bonn. È stata assunta per controllare le luci nelle stanze della *dark ride*, un particolare tipo di giostra.

Il percorso della *dark ride* attraversa  $N$  stanze, numerate da  $0$  a  $N - 1$ . Le stanze vengono attraversate in ordine, iniziando dalla stanza  $0$  e terminando nella stanza  $N - 1$ . Le luci nelle stanze sono controllate da  $N$  interruttori (anch'essi numerati da  $0$  a  $N - 1$ ), uno per ogni stanza. L'interruttore  $s$  (dove  $0 \leq s < N$ ) controlla la luce nella stanza  $p_s$ .

Il capo di Erika le ha chiesto di accendere le luci nella prima e nell'ultima stanza e di spegnere tutte le altre. Sembra facile, vero? Deve solo accendere i due interruttori  $A$  e  $B$  in modo che  $p_A = 0$  e  $p_B = N - 1$  (o  $p_B = 0$  e  $p_A = N - 1$ ). Sfortunatamente, Erika non ha prestato molta attenzione quando il suo capo le ha descritto i comandi, e **non ricorda l'array  $p$ , ovvero quale interruttore controlla quale stanza.**

Erika deve capirlo prima che il suo capo se ne accorga. All'inizio di ogni corsa, Erika spegne tutte le luci. Può quindi accendere un sottoinsieme di interruttori. Mentre la corsa si sposta da una stanza all'altra, ogni volta che la corsa passa da una stanza illuminata a una stanza non illuminata o viceversa, Erika sentirà i passeggeri urlare dall'eccitazione. La velocità della corsa può variare, quindi Erika non può dedurre direttamente quali stanze siano illuminate, ma almeno sentirà il numero di urla. In altre parole, saprà quante volte la corsa passa da una stanza illuminata a una non illuminata, o viceversa.

Puoi aiutare Erika a capire quali sono i due interruttori che controllano le luci della prima e dell'ultima stanza prima che il suo capo se ne accorga? Puoi usare al massimo 30 corse.

## Interazione

Questo è un problema interattivo.

- Il programma deve iniziare leggendo una riga con un intero  $N$ : il numero di stanze nella corsa al buio.

- Quindi, il programma deve interagire con il grader. Per iniziare una corsa, il programma deve stampare una riga che inizia con un punto interrogativo "?", seguito da una stringa di lunghezza  $N$  composta da 0 (spento) e 1 (acceso), che indica come hai impostato gli  $N$  interruttori. Quindi, il programma dovrebbe leggere un singolo intero  $\ell$  ( $0 \leq \ell < N$ ), il numero di volte in cui Erika sente i passeggeri urlare.
- Per rispondere il programma deve stampare una riga con un punto esclamativo "!", seguito da due interi  $A$  e  $B$  ( $0 \leq A, B < N$ ). Perché la tua risposta sia accettata, questi devono essere gli indici degli interruttori che controllano le due stanze finali, in qualsiasi ordine. Dopodiché il programma deve terminare.

Il grader non è adattivo, il che significa che l'array nascosto  $p$  viene determinato prima che inizi l'interazione.

Assicurati di flushare lo standard output dopo ogni corsa, altrimenti il tuo programma potrebbe essere giudicato errato restituendo l'errore "limite di tempo superato". In Python, questo avviene automaticamente purché si usi `input()` per leggere le righe. In C++, `cout << endl;` esegue il flush oltre a stampare una nuova riga; se si usa `printf`, usare `fflush(stdout)`.

## Assunzioni e punteggio

- $3 \leq N \leq 30\,000$ .
- Puoi richiedere al massimo 30 corse (stampare la risposta finale non conta come corsa). Se superi questo limite, riceverai il verdetto "Risposta errata".

La tua soluzione verrà testata su una serie di subtask, ognuno dei quali vale un certo numero di punti. Ogni subtask contiene un insieme di casi di test. Per ottenere il punteggio relativo ad un subtask, è necessario risolvere tutti i casi di test che lo compongono.

Subtask	Punteggio	Assunzioni
1	9	$N = 3$
2	15	$N \leq 30$
3	17	$p_0 = 0$ , ovvero l'interruttore 0 controlla la stanza 0
4	16	$N$ è pari, l'interruttore per una delle stanze finali si trova nella prima metà ( $0 \leq a < \frac{N}{2}$ ) e l'altro nella seconda metà ( $\frac{N}{2} \leq b < N$ )
5	14	$N \leq 1000$
6	29	Nessuna limitazione aggiuntiva

## Testing Tool

Per facilitare il test della tua soluzione, abbiamo messo a disposizione un semplice tool che puoi scaricare. Vedi "Allegati" in fondo alla pagina dei problemi di Kattis. L'utilizzo del tool è facoltativo. Si noti che il grader ufficiale di Kattis è diverso dal tool di test fornito.

Per utilizzare il tool, crea un file di input, ad esempio "sample1.in", che dovrebbe iniziare con un numero  $N$  seguito da una riga con  $p_0, p_1, \dots, p_{N-1}$  che specifica la permutazione nascosta. Ad esempio:

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

Per i programmi Python, ad esempio `solution.py` (normalmente eseguito come `pypy3 solution.py`), eseguire:

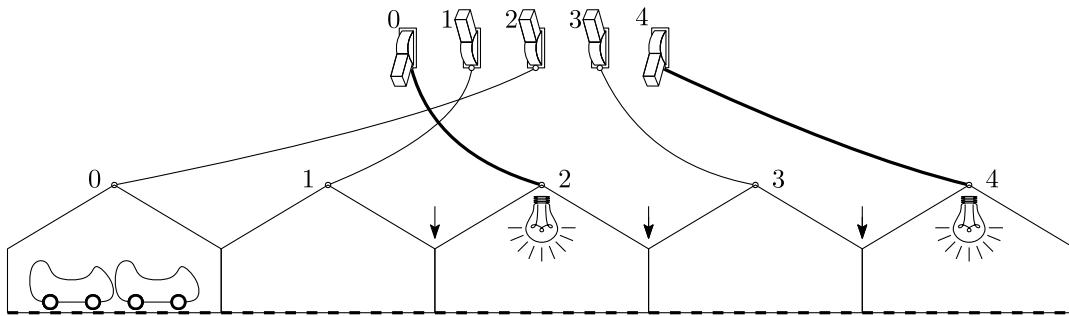
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

Per i programmi C++, prima compilalo (ad esempio con `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) e poi esegui:

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

## Esempi

Nel primo esempio, la permutazione nascosta è  $[p_0, p_1, p_2, p_3, p_4] = [2, 1, 0, 3, 4]$ . Ciò soddisfa le assunzioni dei subtask 2, 5 e 6. Innanzitutto, il programma legge l'intero  $N = 5$ . Quindi, richiede una corsa con  $K = 2$  gli interruttori accesi sono il 4 e lo 0. che controllano le luci delle sale  $p_4 = 4$  e  $p_0 = 2$ ; vedere l'illustrazione sottostante. Erika sente 3 urla (indicate dalle frecce nella figura): la prima quando l'attrazione passa dalla stanza 1 non illuminata alla stanza 2 illuminata; la seconda dalla stanza 2 illuminata alla stanza 3 non illuminata; e la terza quando passa dalla stanza 3 non illuminata alla stanza 4 illuminata. Il programma richiede quindi un'altra corsa in cui le stanze  $p_0, p_2$  e  $p_3$  sono illuminate, facendo sì che Erika senta 3 urla. Infine, il programma risponde con  $A = 2$  e  $B = 4$ , il che è effettivamente corretto poiché controllano la prima e l'ultima stanza ( $p_2 = 0$  e  $p_4 = 4$ ). Si noti che anche  $A = 4$  e  $B = 2$  sarebbero state risposte corrette.



Nel secondo esempio, la permutazione nascosta è  $[p_0, p_1, p_2] = [2, 0, 1]$ . Ciò soddisfa le assunzioni dei subtask 1, 2, 5 e 6. Il programma richiede una corsa in cui tutti e tre gli interruttori sono accesi. Poiché questo significa che tutte le stanze sono illuminate, Erika non sentirà urla. Nel secondo giro, gli interruttori 1 e 0 sono accesi, facendo sì che le stanze  $p_1 = 0$  e  $p_0 = 2$  siano illuminate, mentre la stanza 1 è spenta. Erika sente due urla: quando il giro passa dalla stanza 0 (illuminata) alla stanza 1 (spenta) e dalla stanza 1 (spenta) alla stanza 2 (illuminata). Nell'ultimo giro, nessun interruttore è acceso, il che significa che tutte e tre le stanze sono spente e, di nuovo, che Erika non sente urla. Il programma risponde quindi con gli interruttori 1 e 0, che in effetti controllano la prima e l'ultima stanza. Sia "! 0 1" che "! 1 0" sono risposte accettate.

Nel terzo caso di esempio, la permutazione nascosta è  $[p_0, p_1, p_2, p_3] = [0, 1, 2, 3]$ . Ciò soddisfa le assunzioni dei subtask 2, 3, 4, 5 e 6.

### Primo caso di esempio

output del grader	il tuo output
5	
	? 10001
3	
	? 10110
3	
	! 2 4

### Secondo caso di esempio

output del grader	il tuo output
3	
	? 111
0	
	? 110
2	
	? 000
0	
	! 1 0

### Terzo caso di esempio

output del grader	il tuo output
4	
	? 1010
3	
	! 0 3