

B. Dark Ride

Problem Name	Dark Ride
Time Limit	1 seconds
Memory Limit	1 gigabyte

Erika hat vor kurzem einen Ferienjob im Freizeitpark Phantasialand bei Bonn bekommen. Sie wurde angeheuert, um die Beleuchtung in den Räumen zu steuern, durch die eine Dark Ride führt.

Die Fahrt führt durch N Räume, nummeriert von 0 bis $N - 1$. Die Räume werden der Reihe nach durchlaufen, beginnend in Raum 0 und endend in Raum $N - 1$. Die Beleuchtung in den Räumen wird durch N Schalter (ebenfalls nummeriert von 0 bis $N - 1$) gesteuert, einen für jeden Raum. Schalter s (wobei $0 \leq s < N$) steuert das Licht in Raum p_s .

Erikas Chef hat sie gebeten, das Licht im ersten und letzten Zimmer einzuschalten und in allen anderen auszuschalten. Klingt einfach, oder? Sie muss nur die beiden Schalter A und B so einschalten, dass $p_A = 0$ und $p_B = N - 1$ (oder $p_B = 0$ und $p_A = N - 1$) ist. Leider hat Erika nicht richtig aufgepasst, als ihr Chef die Steuerung erklärte, und **sie erinnert sich nicht an das Array p – also daran, welcher Schalter welchen Raum steuert.**

Erika muss das herausfinden, bevor ihr Chef es bemerkt. Vor jeder Fahrt schaltet Erika alle Lichter aus und kann dann eine Auswahl der Schalter einschalten. Während die Fahrt von Raum zu Raum fährt, hört Erika jedes Mal, wenn die Fahrt von einem beleuchteten in einen unbeleuchteten Raum oder umgekehrt wechselt, die aufgeregten Schreie der Passagiere. Da die Geschwindigkeit der Fahrt variieren kann, kann Erika nicht direkt erkennen, welche Räume beleuchtet sind, aber sie hört zumindest die Anzahl der Schreie. Das heißt, sie erfährt, wie oft die Fahrt von einem beleuchteten in einen unbeleuchteten Raum oder von einem unbeleuchteten in einen beleuchteten Raum wechselt.

Kannst du Erika helfen, herauszufinden, welche zwei Schalter die Lichter im ersten und letzten Raum steuern, bevor ihr Chef es bemerkt? Du kannst maximal 30 Fahrten nutzen.

Interaction

Dies ist ein interaktives Problem.

- Ihr Programm sollte mit dem Lesen einer Zeile mit einer Ganzzahl N beginnen: der Anzahl der Räume in der Dark Ride.
- Dann sollte Ihr Programm mit dem Bewerter interagieren. Um eine Fahrt zu starten, sollten Sie eine Zeile drucken, die mit einem Fragezeichen „ ? “ beginnt, und dann eine Zeichenfolge der Länge N die aus 0 “ (ausgeschaltet) und 1 (eingeschaltet) besteht und angibt, wie Sie die N -Schalter eingestellt haben. Dann sollte Ihr Programm eine einzelne Ganzzahl ℓ ($0 \leq \ell < N$) lesen, die angibt, wie oft Erika die Schreie der Passagiere hört.
- Wenn Sie antworten möchten, drucken Sie eine Zeile mit einem Ausrufezeichen „ ! “, gefolgt von zwei Ganzzahlen A und B ($0 \leq A, B < N$). Damit Ihre Antwort akzeptiert wird, müssen dies die Indizes der Schalter sein, die die beiden Endräume steuern, in beliebiger Reihenfolge. Danach sollte Ihr Programm beendet werden.

Der Grader ist nicht adaptiv, was bedeutet, dass das versteckte Array p bestimmt wird, bevor die Interaktion beginnt.

Stellen Sie sicher, dass die Standardausgabe nach jeder Fahrt geleert wird, da Ihr Programm sonst möglicherweise als „Zeitlimit überschritten“ eingestuft wird. In Python geschieht dies automatisch, solange Sie `input()` zum Lesen von Zeilen verwenden. In C++ `cout << endl;` zusätzlich zum Drucken einer neuen Zeile einen Leervorgang aus. Bei Verwendung von `printf` verwenden Sie `fflush(stdout)`.

Constraints and Scoring

- $3 \leq N \leq 30\,000$.
- Sie können maximal 30 Fahrten ausgeben (das Ausdrucken der endgültigen Antwort zählt nicht als Fahrt). Wenn Sie dieses Limit überschreiten, erhalten Sie das Urteil „Falsche Antwort“.

Ihre Lösung wird in einer Reihe von Testgruppen getestet, die jeweils eine bestimmte Anzahl von Punkten wert sind. Jede Testgruppe enthält eine Reihe von Testfällen. Um die Punkte für eine Testgruppe zu erhalten, müssen Sie alle Testfälle in der Testgruppe lösen.

Group	Score	Limits
1	9	$N = 3$
2	15	$N \leq 30$
3	17	$p_0 = 0$, i.e., switch 0 controls room 0
4	16	N is even, with the switch for one of the end rooms in the first half ($0 \leq A < \frac{N}{2}$) and the other in the second half ($\frac{N}{2} \leq B < N$)
5	14	$N \leq 1000$
6	29	No additional constraints

Testtool

Um das Testen Ihrer Lösung zu erleichtern, haben wir ein einfaches Tool bereitgestellt, das Sie herunterladen können. Siehe „Anhänge“ unten auf der Kattis-Problemseite. Die Nutzung des Tools ist optional. Beachten Sie, dass sich der offizielle Kattis-Grader vom bereitgestellten Testtool unterscheidet.

Um das Tool zu verwenden, erstellen Sie eine Eingabedatei, z. B. „sample1.in“, die mit der Zahl N beginnt, gefolgt von einer Zeile mit p_0, p_1, \dots, p_{N-1} die die versteckte Permutation angibt. Beispiel:

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

Für Python Programme schreibe: `solution.py` (üblicherweise `pypy3 solution.py`), daher:

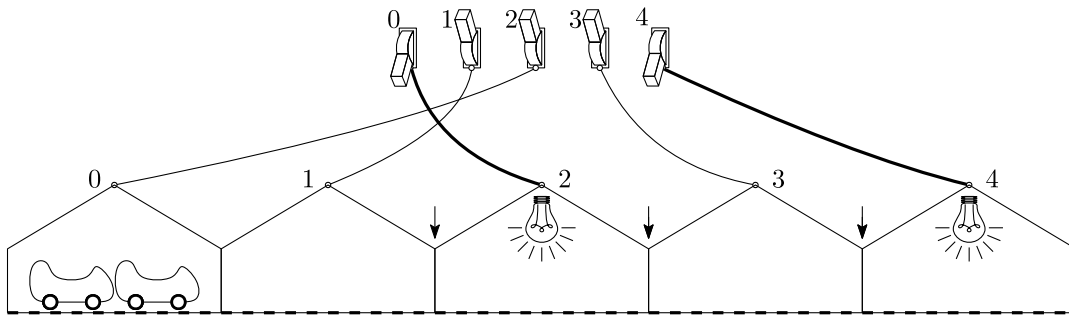
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

Für C++ Programme: zuerst compilieren: (z.B. mit: `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) und dann laufen lassen mit::

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

Beispiel

I In der ersten Stichprobe ist die versteckte Permutation $[p_0, p_1, p_2, p_3, p_4] = [2, 1, 0, 3, 4]$. Dies erfüllt die Einschränkungen der Testgruppen 2, 5 und 6. Zuerst liest das Programm die Ganzzahl $N = 5$. Dann fordert das Programm eine Fahrt mit zwei eingeschalteten Schaltern an: Schalter 4 und Schalter 0. Diese Kontrollräume $p_4 = 4$ und $p_0 = 2$; siehe Abbildung unten. Erika hört 3 Schreie (in der Abbildung durch Pfeile markiert): zuerst, als die Fahrt vom unbeleuchteten Raum 1 zum beleuchteten Raum 2 führt; zweitens vom beleuchteten Raum 2 zum unbeleuchteten Raum 3; und drittens, als sie vom unbeleuchteten Raum 3 zum beleuchteten Raum 4 führt. Das Programm fordert dann eine weitere Fahrt an, bei der die Räume p_0, p_2 und p_3 beleuchtet sind, wodurch Erika 3 Schreie hört. Schließlich antwortet das Programm mit $A = 2$ und $B = 4$, was tatsächlich richtig ist, da diese den ersten und letzten Raum steuern ($p_2 = 0$ und $p_4 = 4$). Beachten Sie, dass $A = 4$ und $B = 2$ auch eine richtige Antwort gewesen wären.



Im zweiten Beispiel ist die versteckte Permutation $[p_0, p_1, p_2] = [2, 0, 1]$. Dies erfüllt die Einschränkungen der Testgruppen 1, 2, 5 und 6. Das Programm fordert eine Fahrt an, bei der alle drei Schalter eingeschaltet sind. Da dies bedeutet, dass alle Räume beleuchtet sind, hört Erika keine Schreie. Bei der zweiten Fahrt werden die Schalter 1 und 0 eingeschaltet, wodurch die Räume $p_1 = 0$ und $p_0 = 2$ beleuchtet werden, während Raum 1 unbeleuchtet ist. Erika hört zwei Schreie: als die Fahrt von Raum 0 (beleuchtet) nach Raum 1 (unbeleuchtet) und von Raum 1 (unbeleuchtet) nach Raum 2 (beleuchtet) geht. Bei der letzten Fahrt werden keine Schalter eingeschaltet, was bedeutet, dass alle drei Räume unbeleuchtet sind und Erika auch hier keine Schreie hört. Das Programm antwortet dann mit den Schaltern 1 und 0, die tatsächlich den ersten und letzten Raum steuern. Sowohl „! 0 1“ als auch „! 1 0“ sind akzeptierte Antworten.

In der dritten Stichprobe lautet die versteckte Permutation $[p_0, p_1, p_2, p_3] = [0, 1, 2, 3]$. Dies erfüllt die Bedingungen der Testgruppen 2, 3, 4, 5 und 6.

Erstes Beispiel

grader output	dein output
5	
	? 10001
3	
	? 10110
3	
	! 2 4

Zweites Beispiel

grader output	dein output
3	
	? 111
0	
	? 110
2	
	? 000
0	
	! 1 0

Drittes Beispiel

grader output	dein output
4	
	? 1010
3	
	! 0 3