

## B. Geisterbahn

Problemname	Geisterbahn
Time Limit	1 Sekunde
Memory Limit	1 GB

Erika hat vor kurzem einen Ferienjob im Freizeitpark Phantasialand in der Nähe von Bonn bekommen. Sie wurde eingestellt, um die Beleuchtung in den Räumen einer Geisterbahn zu steuern.

Die Geisterbahnfahrt verläuft durch  $N$  Räume, nummeriert von  $0$  bis  $N - 1$ . Die Räume werden der Reihe nach durchfahren, beginnend in Raum  $0$  und endend in Raum  $N - 1$ . Die Beleuchtung in den Räumen wird durch  $N$  Schalter (ebenfalls von  $0$  bis  $N - 1$  nummeriert) 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 Raum einzuschalten und in allen anderen auszuschalten. Klingt einfach, oder? Sie muss nur die beiden Schalter  $A$  und  $B$  einschalten, so 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 dies herausfinden, bevor ihr Chef es bemerkt. Zu Beginn jeder Fahrt schaltet Erika alle Lichter aus. Anschließend kann sie einen Teil der Schalter wieder einschalten. Während die Fahrt von Raum zu Raum geht, hört Erika jedes Mal die aufgeregten Schreie der Passagiere, wenn die Fahrt von einem beleuchteten in einen unbeleuchteten Raum oder umgekehrt wechselt. Die Geschwindigkeit der Fahrt kann variieren, sodass Erika nicht erkennen kann, welche Räume beleuchtet sind, aber sie hört die Anzahl der Schreie. Das heißt, sie erfährt, wie oft die Fahrt von einem beleuchteten in einen unbeleuchteten 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.

### Interaktion

Dies ist ein interaktives Problem.

- Dein Programm muss mit dem Lesen einer Zeile mit einer Ganzzahl  $N$  beginnen: der Anzahl der Räume in der Geisterfahrt.
- Dann muss dein Programm mit dem Grader interagieren. Um eine Fahrt zu starten, musst du eine Zeile ausgeben, die mit einem Fragezeichen „ ? “ beginnt, und dann eine Zeichenfolge der Länge  $N$  die aus 0 (ausgeschaltet) und 1 (eingeschaltet) besteht. Diese gibt an, wie die  $N$ -Schalter eingestellt sind. Danach muss dein Programm eine einzelne Ganzzahl  $\ell$  ( $0 \leq \ell < N$ ) einlesen, die angibt, wie oft Erika die Schreie der Passagiere hört.
- Wenn du antworten möchtest, gebe eine Zeile mit einem Ausrufezeichen „ ! “, gefolgt von zwei Ganzzahlen  $A$  und  $B$  ( $0 \leq A, B < N$ ) aus. Damit deine Antwort akzeptiert wird, müssen dies die Indizes der Schalter sein, die die beiden Endräume steuern, in beliebiger Reihenfolge. Danach muss sich dein Programm beenden.

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

Stelle sicher, dass der Ausgabepuffer der Standardausgabe nach jeder Fahrt geleert wird („ flush “), da das Programm sonst möglicherweise als „Time Limit Exceeded“ bewertet wird. In Python geschieht dies automatisch, solange du `input()` zum Lesen von Zeilen verwendest. In C++ löst dies `cout << endl;` zusätzlich zum Drucken einer neuen Zeile aus. Bei Verwendung von `printf` musst du zusätzlich `fflush(stdout)` benutzen.

## Einschränkungen und Bewertung

- $3 \leq N \leq 30\,000$ .
- Du kannst maximal 30 Fahrten benutzen (das Ausgeben der endgültigen Antwort zählt nicht als Fahrt). Wenn du dieses Limit überschreitest, wird dein Programm mit „Wrong Answer“ bewertet.

Deine Lösung wird an 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, musst du alle Testfälle in der Testgruppe lösen.

Gruppe	Punkte	Einschränkungen
1	9	$N = 3$
2	15	$N \leq 30$
3	17	$p_0 = 0$ , das heißt, Schalter 0 kontrolliert Raum 0
4	16	$N$ ist gerade. Der Schalter für einen der Endräume ist in der ersten Hälfte ( $0 \leq A < \frac{N}{2}$ ) und der andere in der zweiten Hälfte ( $\frac{N}{2} \leq B < N$ )
5	14	$N \leq 1000$
6	29	Keine weiteren Einschränkungen

# Testtool

Um das Testen deiner Lösung zu erleichtern, haben wir ein einfaches Tool bereitgestellt, das du herunterladen kannst. Siehe „attachments“ unten auf der Kattis-Problemseite. Die Nutzung des Tools ist optional. Beachte auch, dass sich der offizielle Kattis-Grader vom bereitgestellten Testtool unterscheidet.

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

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

Python Programme, wie z.B. `solution.py` (normalerweise ausgeführt mit `python3 solution.py`), werden folgendermaßen ausgeführt:

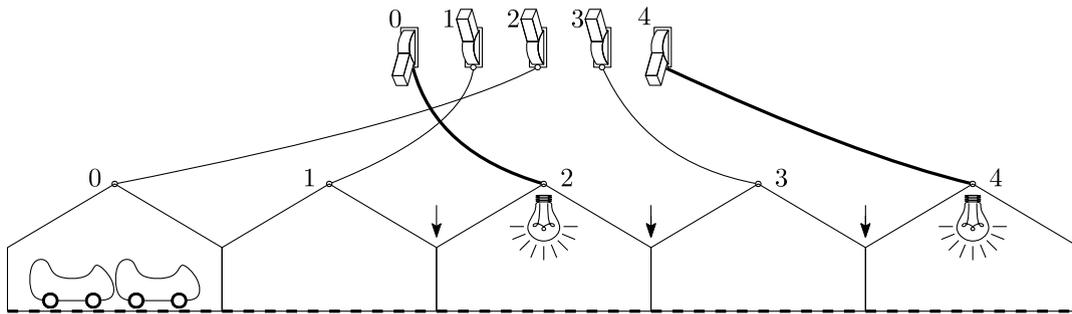
```
python3 testing_tool.py python3 solution.py < sample1.in
```

Für C++-Programme: kompiliere es zuerst (z.B. mit `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) und führe es dann folgendermaßen aus:

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

## Beispiel

Im ersten Beispiel 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 verlangt das Programm eine Fahrt mit  $K = 2$  eingeschalteten Schaltern: Schalter 4 und Schalter 0. Diese kontrollieren Räume  $p_4 = 4$  und  $p_0 = 2$ ; siehe Abbildung unten. Erika hört 3 Schreie (in der Abbildung durch die Pfeile markiert): zuerst, als die Fahrt vom unbeleuchteten Raum 1 zum beleuchteten Raum 2 geht; ein zweiter Schrei beim Übergang vom beleuchteten Raum 2 zum unbeleuchteten Raum 3; und einen dritten, als sie vom unbeleuchteten Raum 3 zum beleuchteten Raum 4 geht. Das Programm verlangt dann eine weitere Fahrt, 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  $p_4 = 4$ ). Beachte, 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 verlangt eine Fahrt, bei der alle drei Schalter eingeschaltet sind. Da dies bedeutet, dass alle Räume beleuchtet sind, hört Erika keine Schreie. Bei der zweiten Fahrt sind die Schalter 1 und 0 eingeschaltet, wodurch die Räume  $p_1 = 0$  und  $p_0 = 2$  beleuchtet sind, 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 akzeptable Antworten.

Im dritten Beispiel 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. Beachte, dass es nicht möglich ist, nach dieser einen Fahrt auf die Antwort zu schließen, aber die Beispiellösung hatte Glück und hat die Antwort erraten.

### Erstes Beispiel

Grader Ausgabe	Deine Ausgabe
5	
	? 10001
3	
	? 10110
3	
	! 2 4

## Zweites Beispiel

Grader Ausgabe	Deine Ausgabe
3	
	? 111
0	
	? 110
2	
	? 000
0	
	! 1 0

## Drittes Beispiel

Grader Ausgabe	Deine Ausgabe
4	
	? 1010
3	
	! 0 3