B - رحلة مظلمة

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

حصلت إريكا مؤخرًا على وظيفة صيفية في مدينة الملاهي فانتازيالاند بالقرب من بون. تم توظيفها للتحكم في إضاءة الغرف التي تمر بها الرحلة المظلمة.

يمر الطريق عبر N غرف، مرقمة من 0 إلى $N-1$. يتم اجتياز الغرف بالترتيب، بدءًا من الغرفة 0 وانتهاءً بالغرفة $N-1$. يتم التحكم في إضاءة الغرف بواسطة N مفاتيح (مرقمة أيضًا من 0 إلى $N-1$)، واحدة لكل غرفة. المفتاح s (حيث $0 \leq s < N$) يتحكم في إضاءة الغرفة p_s.

طلب رئيس إريكا منها تشغيل الإضاءة في الغرفة الأولى والأخيرة وإطفاء جميع الغرف الأخرى. يبدو الأمر سهلاً، أليس كذلك؟ تحتاج فقط إلى تشغيل المفتاحين A وB بحيث $p_A = 0$ و$p_B = N-1$ (أو $p_B = N-1$ و$p_A = 0$). للأسف، لم تنتبه إريكا بشكل كامل عندما شرح رئيسها التحكم في المفاتيح، ولا تتذكر المصفوفة p، أي أي مفتاح يتحكم في أي غرفة.

تحتاج إريكا إلى اكتشاف ذلك قبل أن يلاحظ رئيسها. قبل بدء كل رحلة، تقوم إريكا بإطفاء جميع الأضواء ويمكنها بعد ذلك تشغيل مجموعة جزئية من المفاتيح. مع مرور الرحلة من غرفة إلى أخرى، كلما تنتقل الرحلة من غرفة مضاءة إلى غرفة غير مضاءة أو العكس، ستسمع إريكا الركاب يصرخون من الإثارة. يمكن أن تختلف سرعة الركوب، لذا لا تستطيع إريكا استنتاج الغرف المضاءة مباشرة، لكنها على الأقل ستسمع عدد الصرخات. أي أنها ستعرف عدد المرات التي ينتقل فيها الركوب من غرفة مضاءة إلى غرفة غير مضاءة، أو من غرفة غير مضاءة إلى غرفة مضاءة.

هل يمكنك مساعدة إريكا في معرفة أي مفتاحين يتحكمان في إضاءة الغرفة الأولى والأخيرة قبل أن يلاحظ رئيسها؟ يمكنك استخدام 30 رحلة على الأكثر.

التفاعل

هذه مشكلة تفاعلية.

- يجب أن يبدأ برنامجك بقراءة سطر يحتوي على عدد صحيح N: عدد الغرف في الرحلة المظلمة.
- بعد ذلك، يجب أن يتفاعل برنامجك مع المصحح. لبدء الرحلة، يجب عليك طباعة سطر يبدأ بعلامة استفهام "؟"، ثم سلسلة بطول N تتكون من 0 (مطفأ) و1 (مشغل)، تشير إلى كيفية ضبط المفاتيح الـ N. ثم، يجب على برنامجك قراءة عدد صحيح واحد $l (0 \leq l < N)$، وهو عدد المرات التي تسمع فيها إريكا صرخات الركاب.
- عندما تريد الإجابة، اطبع سطرًا يبدأ بعلامة تعجب "!"، يليه عددين صحيحين A و$B (0 \leq A, B < N)$. ليتم قبول إجابتك، يجب أن تكون هذه هي فهارس المفاتيح التي تتحكم في الغرفتين الطرفيتين، بأي ترتيب. بعد ذلك، يجب أن يتوقف برنامجك.

المصحح غير متكيف، مما يعني أن المصفوفة المخفية p يتم تحديدها قبل بدء التفاعل.

تأكد من تفريغ مخرجات الإخراج القياسي بعد إصدار كل رحلة، وإلا قد يتم تقييم برنامجك بأنه "تجاوز الحد الزمني". في بايثون، يحدث هذا تلقائيًا طالما تستخدم `input()` لقراءة الأسطر. في C++، `cout << endl;` يقوم بالتفريغ بالإضافة إلى طباعة سطر جديد؛ إذا كنت تستخدم `printf`، استخدم `fflush(stdout)`.

القيود والتنقيط

- $3 \leq N \leq 30000$.
- يمكنك إصدار 30 رحلة على الأكثر (طباعة الإجابة النهائية لا تُحسب كرحلة). إذا تجاوزت هذا الحد، ستحصل على الحكم "إجابة خاطئة".

سيتم اختبار حلك على مجموعة من مجموعات الاختبار، كل منها تساوي عددًا من النقاط. تحتوي كل مجموعة اختبار على مجموعة من حالات الاختبار. للحصول على النقاط لمجموعة اختبار، يجب عليك حل جميع حالات الاختبار في المجموعة.

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

أداة الاختبار

لتسهيل اختبار حلك، قمنا بتوفير أداة بسيطة يمكنك تنزيلها. انظر "المرفقات" في أسفل صفحة مشكلة Kattis. الأداة اختيارية للاستخدام. لاحظ أن المصحح الرسمي لـ Kattis يختلف عن أداة الاختبار المقدمة.

لاستخدام الأداة، قم بإنشاء ملف إدخال، مثل "sample1.in"، يجب أن يبدأ بعدد N يليه سطر بـ $p_0, p_1, \ldots, p_{N-1}$ يحدد الترتيب المخفي. على سبيل المثال:

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

لبرامج بايثون، مثل solution.py، التي تعمل عادةً بـ `pypy3 solution.py`. قم بتشغيل:

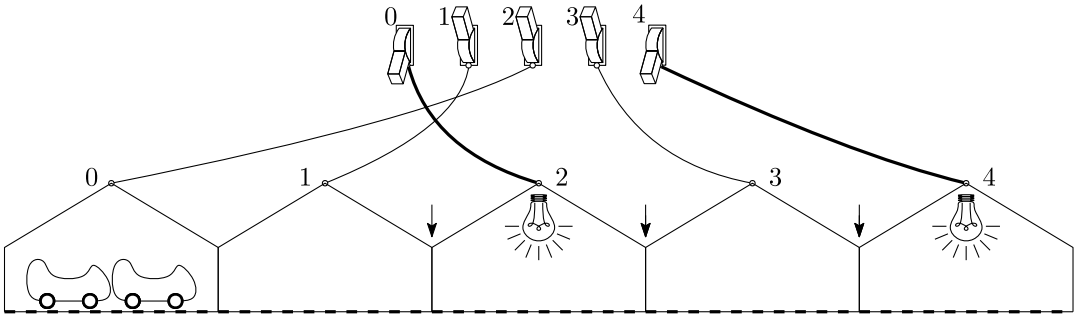
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

لبرامج C++، قم أولاً بتجميعها (مثلًا باستخدام `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) ثم قم بتشغيل:

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

مثال

في العينة الأولى، الترتيب المخفي هو $[p_0, p_1, p_2, p_3, p_4] = [2, 1, 0, 3, 4]$. هذا يفي بقيود مجموعات الاختبار 2، 5، و6. أولاً، يقرأ البرنامج العدد الصحيح $N = 5$. ثم، يطلب البرنامج رحلة مع تشغيل مفتاحين: المفتاح 4 والمفتاح 1. هذان يتحكمان في الغرفتين $p_4 = 4$ و $p_1 = 2$. تسمع إريكا 3 صرخات: الأولى عندما ينتقل الركوب من الغرفة 1 (غير مضاءة) إلى الغرفة 2 (مضاءة)، والثانية من الغرفة 2 (مضاءة) إلى الغرفة 3 (غير مضاءة)، والثالثة عند الانتقال من الغرفة 3 (غير مضاءة) إلى الغرفة 4 (مضاءة). ثم يطلب البرنامج رحلة أخرى حيث تكون الغرف p_0, p_2, p_3 مضاءة، مما يجعل إريكا تسمع 3 صرخات. أخيرًا، يجيب البرنامج بـ $A = 2$ و$B = 4$، وهو بالفعل صحيح حيث يتحكمان في الغرفتين الأولى والأخيرة ($p_2 = 0$ و$p_4 = 4$). لاحظ أن $A = 4$ و$B = 2$ كانتا ستكونان إجابة صحيحة أيضًا.



في العينة الثانية، الترتيب المخفي هو $[p_0, p_1, p_2] = [2, 0, 1]$. هذا يفي بقيود مجموعات الاختبار 1، 2، 5، و6. يطلب البرنامج رحلة حيث يتم تشغيل جميع المفاتيح الثلاثة. بما أن هذا يعني أن جميع الغرف مضاءة، لن تسمع إريكا أي صرخات. في الرحلة الثانية، يتم تشغيل المفتاحين 1 و0، مما يجعل الغرفتين $p_1 = 0$ و$p_0 = 2$ مضاءتين، بينما الغرفة 1 غير مضاءة. تسمع إريكا صرختين: عندما ينتقل الركوب من الغرفة 0 (مضاءة) إلى الغرفة 1 (غير مضاءة)، ومن الغرفة 1 (غير مضاءة) إلى الغرفة 2 (مضاءة). في الرحلة النهائية، لا يتم تشغيل أي مفاتيح، مما يعني أن جميع الغرف الثلاث غير مضاءة، ومرة أخرى لا تسمع إريكا أي صرخات. ثم يجيب البرنامج بالمفتاحين 1 و0، اللذين يتحكمان بالفعل في الغرفتين الأولى والأخيرة. كل من "! 1 0" و"! 0 1" مقبولتان كإجابات.

في العينة الثالثة، الترتيب المخفي هو $[p_0, p_1, p_2, p_3] = [0, 1, 2, 3]$. هذا يفي بقيود مجموعات الاختبار 2، 3، 4، 5، و6. لاحظ أنه ليس بالضرورة إمكانية استنتاج الإجابة بعد هذه الرحلة الواحدة، لكن الحل النموذجي خمن الإجابة وكان محظوظًا.

المثال الأول

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

المثال الثاني

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

المثال الثالث

grader output	your output
4	
	? 1010
3	
	! 0 3