

B. Train fantôme

Nom du problème	Train fantôme
Limite de temps	1 seconde
Limite de mémoire	1 gigabyte

Erika a récemment décroché un job d'été dans le parc d'attraction Phantasialand près de Bonn. Elle a été embauchée pour contrôler les lumières dans les pièces traversées par un train fantôme.

Le trajet passe à travers N pièces, numérotées de 0 à $N - 1$. Les pièces sont traversées dans l'ordre, en commençant par la pièce 0 et en finissant par la pièce $N - 1$. Les lumières dans les pièces sont contrôlées par N interrupteurs (aussi numérotés de 0 à $N - 1$), un par pièce. L'interrupteur s (où $0 \leq s < N$) contrôle la lumière dans la pièce p_s .

Le patron d'Erika lui a demandé d'allumer les lumières dans la première et la dernière pièce et d'éteindre toutes les autres. Cela paraît facile, n'est-ce pas ? Il lui suffit d'allumer les deux interrupteurs A et B tels que $p_A = 0$ et $p_B = N - 1$ (ou $p_B = 0$ et $p_A = N - 1$). Malheureusement, Erika n'a pas fait entièrement attention lorsque son patron a décrit les commandes, et **elle ne se souvient pas du tableau électrique p – c'est à dire quel interrupteur contrôle quelle pièce.**

Erika doit comprendre cela avant que son patron ne le remarque. Avant chaque tour de manège, Erika éteint toutes les lumières. Elle peut alors en allumer un sous-ensemble en actionnant les interrupteurs. Alors que le train fantôme passe d'une pièce à l'autre, chaque fois que le train passe d'une pièce éclairée à une pièce non éclairée ou vice versa, Erika entend les passagers crier d'excitation. La vitesse du train peut varier, donc Erika ne peut pas déduire directement quelles pièces sont éclairées, mais au moins elle entendra le nombre de cris. C'est-à-dire qu'elle apprendra le nombre de fois que le train passe d'une pièce éclairée à une pièce non éclairée, ou d'une pièce non éclairée à une pièce éclairée.

Pouvez-vous aider Erika à déterminer quels sont les deux interrupteurs qui contrôlent les lumières de la première et de la dernière pièce avant que son patron ne remarque quelque chose ? Vous pouvez utiliser au plus 30 trajets.

Interaction

Il s'agit d'un problème interactif.

- Votre programme doit commencer par lire une ligne avec un entier N : le nombre de pièces du manège.
- Ensuite, votre programme doit interagir avec le grader. Pour démarrer un trajet, vous devez afficher une ligne commençant par un point d'interrogation "?", et ensuite une chaîne de caractères de longueur N comprenant des 0 (éteint) et des 1 (allumé), indiquant comment vous disposez les N interrupteurs. Ensuite, votre programme doit lire un unique entier ℓ ($0 \leq \ell < N$), le nombre de fois qu'Erika entend les passagers crier.
- Lorsque vous voulez répondre, affichez une ligne avec un point d'exclamation "!", suivi de deux entiers A et B ($0 \leq A, B < N$). Pour que votre réponse soit acceptée, ceux-ci doivent être les indices des interrupteurs commandant les salles aux deux extrémités du trajet, dans n'importe quel ordre. Après cela, votre programme doit se terminer.

Le grader est non-adaptatif, ce qui veut dire que le tableau p est fixé avant que l'interaction commence.

Assurez-vous de vider la sortie standard après chaque trajet, sinon votre programme pourrait être évalué comme dépassant la limite de temps. En Python, cela se produit automatiquement lorsque vous utilisez `input()` pour lire des lignes. En C++, `cout << endl`; vide en plus d'afficher une nouvelle ligne ; si vous utilisez `printf`, utilisez `fflush(stdout)`.

Contraintes et scores

- $3 \leq N \leq 30\,000$.
- Vous pouvez utiliser au maximum 30 trajets (afficher la réponse finale ne compte pas comme un trajet). Si vous dépassez cette limite, vous aurez comme verdict "Wrong Answer".

Votre solution sera testée sur un ensemble de sous-tâches, chacune valant un certain nombre de points. Chaque sous-tâche contient un ensemble de tests. Pour obtenir les points d'une sous-tâche, vous devez réussir tous les tests de cette sous-tâche.

Sous-tâche	Score	Limites
1	9	$N = 3$
2	15	$N \leq 30$
3	17	$p_0 = 0$, i.e. l'interrupteur 0 contrôle la pièce 0
4	16	N est impair, avec l'interrupteur pour l'une des salles d'extrémité dans la première moitié ($0 \leq A < \frac{N}{2}$) et l'autre dans la seconde moitié ($\frac{N}{2} \leq B < N$)
5	14	$N \leq 1000$
6	29	Pas de contrainte supplémentaire

Outil de test

Pour faciliter l'évaluation de votre solution, nous avons fourni un outil simple que vous pouvez télécharger. Consulter "attachments" en bas de la page du problème sur Kattis. L'utilisation de l'outil est optionnelle. Notez que l'évaluateur officiel sur Kattis diffère de l'outil de test fourni.

Pour utiliser l'outil, créez un fichier d'entrée, tel que "sample1.in", qui doit commencer par un nombre N suivi par une ligne avec p_0, p_1, \dots, p_{N-1} qui représente la permutation cachée. Par exemple :

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

Pour les programmes Python, utilisez `solution.py` (normalement exécuté comme `pypy3 solution.py`), exécutez :

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

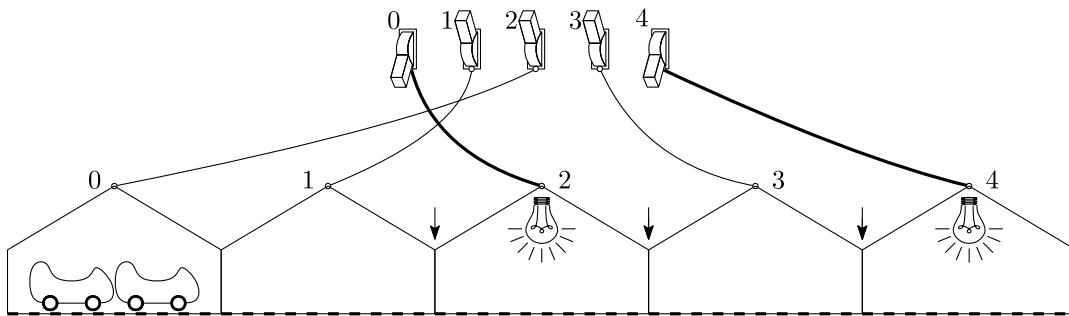
Pour les programme en C++, commencez par compiler (i.e. avec `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) puis exécutez-le :

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

Exemple

Dans le premier exemple, la permutation cachée est $[p_0, p_1, p_2, p_3, p_4] = [2, 1, 0, 3, 4]$. Cela satisfait les contraintes des sous-tâches 2, 5, et 6. D'abord, le programme lit l'entier $N = 5$. Puis le

programme demande un trajet avec deux interrupteurs allumés : l'interrupteur 4 et l'interrupteur 0. Ceux-ci contrôlent les pièces $p_4 = 4$ and $p_0 = 2$; consultez l'illustration ci-dessous. Erika entend 3 cris (marqués par une flèche sur la figure) : le premier lorsque le train passe de la pièce non éclairée 1 à la pièce éclairée 2 ; le second quand il passe de la pièce éclairée 2 à la pièce non éclairée 3 ; et le troisième quand il passe de la pièce non éclairée 3 à la pièce éclairée 4. Le programme demande alors un autre trajet où les pièces p_0, p_2 , et p_3 sont éclairées, permettant à Erika d'entendre 3 cris. Finalement, le programme répond avec $A = 2$ et $B = 4$, ce qui est en effet correct, ces interrupteurs contrôlant la première et la dernière pièce ($p_2 = 0$ et $p_4 = 4$). Notez que $A = 4$ et $B = 2$ aurait aussi été une réponse correcte.



Dans le second exemple, la permutation cachée est $[p_0, p_1, p_2] = [2, 0, 1]$. Cela satisfait les contraintes des sous-tâches 1, 2, 5, et 6. Le programme demande un trajet où les trois interrupteurs sont allumés. Puisque cela signifie que toutes les pièces sont éclairées, Erika ne va entendre aucun cri. Lors du deuxième trajet, les interrupteurs 1 et 0 sont allumés, éclairant les pièces $p_1 = 0$ et $p_0 = 2$, alors que la pièce 1 est non éclairée. Erika entend deux cris : quand le train va de la pièce 0 (éclairée) à la pièce 1 (non éclairée), et de la pièce 1 (non éclairée) à la pièce 2 (éclairée). Pour le dernier trajet, aucun interrupteur n'est allumé, ce qui veut dire que toutes les pièces sont non éclairées, et qu'à nouveau Erika n'entend aucun cri. Le programme répond alors avec les interrupteurs 1 et 0, qui en effet contrôlent la première et la dernière pièce. À la fois "1 0" et "1 1" sont des réponses acceptées.

Dans le troisième exemple, la permutation cachée est $[p_0, p_1, p_2, p_3] = [0, 1, 2, 3]$. Cela satisfait les contraintes des sous-tâches 2, 3, 4, 5, and 6. Notez qu'il n'est pas nécessairement possible de déduire la réponse après ce trajet, mais la solution d'exemple a deviné la réponse et a eu de la chance.

Premier exemple

sortie du grader	votre sortie
5	
	? 10001
3	
	? 10110
3	
	! 2 4

Deuxième exemple

sortie du grader	votre sortie
3	
	? 111
0	
	? 110
2	
	? 000
0	
	! 1 0

Troisième exemple

sortie du grader	votre sortie
4	
	? 1010
3	
	! 0 3