

B. Passeio Sombrio

Nome do problema	Passeio sombrio
Tempo limite	1 segundo
Memória limite	1 gigabyte

Erika conseguiu recentemente um emprego de verão no parque de diversões Phantasialand, perto de Bonn. Ela foi contratada para controlar as luzes nas salas de um passeio por onde passa um carrinho em um caminho escuro.

O carrinho passa por N salas, numeradas de 0 a $N - 1$. As salas são percorridas em ordem, começando na sala 0 e terminando na sala $N - 1$. As luzes nas salas são controladas por N interruptores (também numerados de 0 a $N - 1$), um para cada sala. O interruptor s (onde $0 \leq s < N$) controla a luz na sala p_s .

O chefe de Erika pediu que ela acendesse as luzes da primeira e da última sala e apagasse todas as outras. Parece fácil, não é? Ela só precisa ligar os dois interruptores A e B de forma que $p_A = 0$ e $p_B = N - 1$ (ou $p_B = 0$ e $p_A = N - 1$). Infelizmente, Erika não prestou muita atenção quando seu chefe descreveu os controles, e **ela não se lembra do vetor p - ou seja, qual interruptor controla qual cômodo.**

Erika precisa descobrir isso antes que seu chefe perceba. Antes do início de cada passeio, Erika apaga todas as luzes e pode, então, ligar um subconjunto de interruptores. À medida que o carrinho passa de sala em sala, sempre que o carrinho passa de uma sala iluminada para uma sala apagada ou vice-versa, Erika ouvirá os passageiros gritarem de excitação. A velocidade do carrinho pode variar, então Erika não pode inferir diretamente quais salas estão iluminadas, mas pelo menos ouvirá o número de gritos. Ou seja, ela aprenderá o número de vezes que o carrinho passa de uma sala iluminada para uma apagada, ou de uma apagada para uma iluminada.

Você consegue ajudar a Erika a descobrir quais dois interruptores controlam as luzes do primeiro e do último cômodo antes que o chefe dela perceba? Você pode usar no máximo 30 voltas.

Interação

Este é um problema interativo.

- Seu programa deve começar lendo uma linha com um inteiro N : o número de salas no caminho escuro.
- Então, seu programa deve interagir com o classificador. Para iniciar um passeio, você deve imprimir uma linha começando com um ponto de interrogação " ? ", e então uma string de comprimento N consistindo de 0 (desligado) e 1 (ligado), indicando como você configurou os N interruptores. Então, seu programa deve ler um único inteiro ℓ ($0 \leq \ell < N$), o número de vezes que Erika ouve os passageiros gritarem.
- Quando quiser responder, imprima uma linha com um ponto de exclamação " ! ", seguido por dois inteiros A e B ($0 \leq A, B < N$). Para que sua resposta seja aceita, estes devem ser os índices dos interruptores que controlam as duas salas finais, em qualquer ordem. Depois disso, seu programa deverá fechar.

O classificador não é adaptativo, o que significa que o vetor oculto p é determinado antes do início da interação.

Certifique-se de liberar a saída padrão após emitir cada passeio, caso contrário seu programa poderá ser considerado como Tempo Limite Excedido. Em Python, isso acontece automaticamente, desde que você use `input()` para ler linhas. Em C++, `cout << endl;` limpa, além de imprimir uma nova linha; se estiver usando `printf`, use `fflush(stdout)`.

Restrições e pontuação

- $3 \leq N \leq 30\,000$.
- Você pode emitir no máximo 30 corridas (imprimir a resposta final não conta como corrida). Se exceder esse limite, você receberá o veredito "Resposta Errada".

Sua solução será testada em um conjunto de grupos de teste, cada um valendo um certo número de pontos. Cada grupo de teste contém um conjunto de casos de teste. Para obter os pontos de um grupo de teste, você precisa resolver todos os casos de teste do grupo.

Grupo	Pontuação	Limites
1	9	$N = 3$
2	15	$N \leq 30$
3	17	$p_0 = 0$, ou seja, o interruptor 0 controla a sala 0
4	16	N é par, com a troca para uma das salas finais na primeira metade ($0 \leq a < \frac{N}{2}$) e a outra na segunda metade ($\frac{N}{2} \leq b < N$)
5	14	$N \leq 1000$
6	29	Sem restrições adicionais

Ferramenta de teste

Para facilitar o teste da sua solução, fornecemos uma ferramenta simples que você pode baixar. Veja os "anexos" na parte inferior da página de problemas do Kattis. O uso da ferramenta é opcional. Observe que o corretor oficial do Kattis é diferente da ferramenta de teste fornecida.

Para usar a ferramenta, crie um arquivo de entrada, como "sample1.in", que deve começar com um número N seguido por uma linha com p_0, p_1, \dots, p_{N-1} especificando a permutação oculta. Por exemplo:

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

Para programas em Python, digamos `solution.py` (normalmente executado como `pypy3 solution.py`), execute:

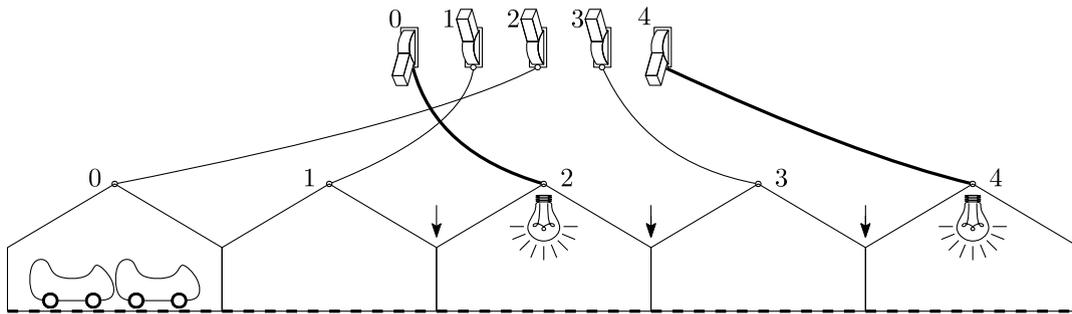
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

Para programas em C++, primeiro compile-o (por exemplo, com `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) e então execute:

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

Exemplo

No primeiro exemplo, a permutação oculta é $[p_0, p_1, p_2, p_3, p_4] = [2, 1, 0, 3, 4]$. Isso satisfaz as restrições dos grupos de teste 2, 5 e 6. Primeiro, o programa lê o inteiro $N = 5$. Em seguida, o programa solicita um passeio com dois interruptores ligados: interruptor 4 e interruptor 0. Essas salas de controle $p_4 = 4$ e $p_0 = 2$; veja a ilustração abaixo. Erika ouve 3 gritos (marcados por setas na figura): primeiro quando o carrinho passa do quarto 1 sem iluminação para o quarto 2 iluminado; segundo, do quarto 2 iluminado para o quarto 3 sem iluminação; e terceiro, ao passar do quarto 3 sem iluminação para o quarto 4 iluminado. O programa então solicita outro passeio onde os quartos p_0, p_2 e p_3 estão iluminados, fazendo Erika ouvir 3 gritos. Finalmente, o programa responde com $A = 2$ e $B = 4$, o que é de fato correto, pois eles controlam o primeiro e o último quartos ($p_2 = 0$ e $p_4 = 4$). Observe que $A = 4$ e $B = 2$ também teriam sido uma resposta correta.



No segundo exemplo, a permutação oculta é $[p_0, p_1, p_2] = [2, 0, 1]$. Isso satisfaz as restrições dos grupos de teste 1, 2, 5 e 6. O programa solicita um passeio onde todos os três interruptores estão ligados. Como isso significa que todos os quartos estão iluminados, Erika não ouvirá gritos. No segundo passeio, os interruptores 1 e 0 estão ligados, fazendo com que os quartos $p_1 = 0$ e $p_0 = 2$ estejam iluminados, enquanto o quarto 1 está apagado. Erika ouve dois gritos: quando o carrinho vai do quarto 0 (aceso) para o quarto 1 (apagado), e do quarto 1 (apagado) para o quarto 2 (aceso). No passeio final, nenhum interruptor está ligado, o que significa que todos os três quartos estão apagados, e novamente que Erika não ouve gritos. O programa então responde com os interruptores 1 e 0, que de fato controlam o primeiro e o último quartos. Tanto " ! 0 1 " quanto " ! 1 0 " são respostas aceitas.

Na terceira amostra, a permutação oculta é $[p_0, p_1, p_2, p_3] = [0, 1, 2, 3]$. Isso satisfaz as restrições dos grupos de teste 2, 3, 4, 5 e 6.

Primeiro Exemplo

saída do classificador	sua saída
5	
	? 10001
3	
	? 10110
3	
	! 2 4

Segundo Exemplo

saída do classificador	sua saída
3	
	? 111
0	
	? 110
2	
	? 000
0	
	! 1 0

Terceiro Exemplo

saída do classificador	sua saída
4	
	? 1010
3	
	! 0 3