

B. Dark Ride

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

Erika recently got a summer job at the amusement park Phantasialand near Bonn. She was hired to control the lights in the rooms through which a dark ride passes.

The ride passes through N rooms, numbered from 0 to $N - 1$. The rooms are traversed in order, starting in room 0 and ending in room $N - 1$. The lights in the rooms are controlled by N switches (also numbered from 0 to $N - 1$), one for each room. Switch s (where $0 \leq s < N$) controls the light in room p_s .

Erika's boss has asked her to turn on the lights in the first and last rooms and turn off all the others. Sounds easy, right? She just needs to turn on the two switches A and B such that $p_A = 0$ and $p_B = N - 1$ (or $p_B = 0$ and $p_A = N - 1$). Unfortunately, Erika did not fully pay attention when her boss described the controls, and **she does not remember the array p – that is, which switch controls which room.**

Erika needs to figure this out before her boss notices. At the start of each ride, Erika turns off all the lights. She can then turn on a subset of the switches. As the ride goes from room to room, whenever the ride goes from a lit room to an unlit room or vice versa, Erika will hear the passengers scream in excitement. The speed of the ride can vary, so Erika cannot directly infer which rooms are lit but at least she will hear the number of screams. That is, she will learn the number of times the ride passes from lit to unlit room, or unlit to lit room.

Can you help Erika figure out which two switches control the lights for the first and last rooms before her boss notices? You can use at most 30 rides.

Interaction

This is an interactive problem.

- Your program should start by reading a line with an integer N : the number of rooms in the dark ride.

- Then, your program should interact with the grader. To start a ride, you should print a line beginning with a question mark “?”, and then a string of length N consisting of 0's (turned off) and 1's (turned on), indicating how you set the N switches. Then, your program should read a single integer ℓ ($0 \leq \ell < N$), the number of times Erika hears the passengers scream.
- When you want to answer, print a line with an exclamation mark “!”, followed by two integers A and B ($0 \leq A, B < N$). For your answer to be accepted, these must be the indices of the switches controlling the two end rooms, in any order. After this, your program should exit.

The grader is non-adaptive, meaning that the hidden array p is determined before the interaction begins.

Make sure to flush standard output after issuing each ride, or else your program might get judged as Time Limit Exceeded. In Python, this happens automatically as long as you use `input()` to read lines. In C++, `cout << endl;` flushes in addition to printing a newline; if using `printf`, use `fflush(stdout)`.

Constraints and Scoring

- $3 \leq N \leq 30\,000$.
- You can issue at most 30 rides (printing the final answer does not count as a ride). If you exceed this limit, you will get the verdict “Wrong Answer”.

Your solution will be tested on a set of test groups, each worth a number of points. Each test group contains a set of test cases. To get the points for a test group, you need to solve all test cases in the test group.

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

Testing Tool

To facilitate the testing of your solution, we have provided a simple tool that you can download. See “attachments” at the bottom of the Kattis problem page. The tool is optional to use. Note that

the official Kattis grader differs from the provided testing tool.

To use the tool, create an input file, such as "sample1.in", which should start with a number N followed by a line with p_0, p_1, \dots, p_{N-1} specifying the hidden permutation. For example:

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

For Python programs, say `solution.py` (normally run as `pypy3 solution.py`), run:

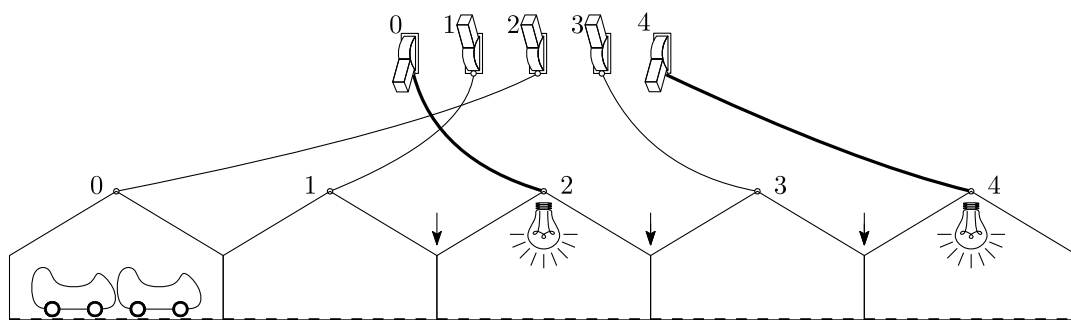
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

For C++ programs, first compile it (e.g. with `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) and then run:

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

Example

In the first sample, the hidden permutation is $[p_0, p_1, p_2, p_3, p_4] = [2, 1, 0, 3, 4]$. This satisfies the constraints of test groups 2, 5, and 6. First, the program reads the integer $N = 5$. Then, the program requests a ride with $K = 2$ turned on switches: switch 4 and switch 0. These control rooms $p_4 = 4$ and $p_0 = 2$; see the illustration below. Erika hears 3 screams (marked by arrows in the figure): first when the ride passes from unlit room 1 to lit room 2; second from lit room 2 to unlit room 3; and thirdly when passing from unlit room 3 to lit room 4. The program then requests another ride where rooms p_0, p_2 , and p_3 are lit, making Erika hear 3 screams. Finally, the program answers with $A = 2$ and $B = 4$, which is indeed correct as these control the first and last rooms ($p_2 = 0$ and $p_4 = 4$). Note that $A = 4$ and $B = 2$ would also have been a correct answer.



In the second sample, the hidden permutation is $[p_0, p_1, p_2] = [2, 0, 1]$. This satisfies the constraints of test groups 1, 2, 5, and 6. The program requests a ride where all three switches are turned on. Since this means all the rooms are lit, Erika will hear no screams. In the second ride, switches 1 and 0 are turned on, making rooms $p_1 = 0$ and $p_0 = 2$ be lit, while room 1 is unlit. Erika hears two

screams: when the ride goes from room 0 (lit) to room 1 (unlit), and from room 1 (unlit) to room 2 (lit). In the final ride, no switches are turned on, meaning that all three rooms are unlit, and again that Erika hears no screams. The program then answers with switches 1 and 0, which indeed control the first and last rooms. Both “! 0 1” and “! 1 0” are accepted answers.

In the third sample, the hidden permutation is $[p_0, p_1, p_2, p_3] = [0, 1, 2, 3]$. This satisfies the constraints of test groups 2, 3, 4, 5, and 6.

First Sample

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

Second Sample

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

Third Sample

grader output	your output
4	
	? 1010
3	
	! 0 3