B. ダークライド(Dark Ride)

問題名	ダークライド(Dark Ride)
実行時間制限	1 秒
メモリ制限	1 GB

この夏，エリカはPhantasialandというボンの近くにある遊園地で働くことになった．彼女の仕事は，ダークライドという乗り物が通過する部屋の明かりを管理することである．

この乗り物は N 個の部屋を通過する．部屋には 0 から $N-1$ まで番号がついており，ダークライドは部屋 0 から出発し部屋 $N-1$ で終わるよう，番号順に部屋を通過する．部屋の明かりは N 個のスイッチにより管理されており，スイッチには 0 から $N-1$ まで番号がついている．各部屋の明かりを変更するスイッチはちょうど 1 つある．スイッチ s $(0 \leq s < N)$ は部屋 p_s の明かりを変更する．

エリカは上司から，最初と最後の部屋の明かりをつけ，その他の部屋の明かりをすべて消すよう頼まれた．簡単に思えるでしょう？2 つのスイッチ A, B であって，$p_A = 0$ かつ $p_B = N-1$ (もしくは $p_B = 0$ かつ $p_A = N-1$) となるスイッチをオンにすればよいだけである．しかし，エリカはあまり注意せずに上司の説明を聞いていたため，**彼女は数列 p を覚えていない，つまり，どのスイッチがどの部屋を管理しているのか覚えていない**．

エリカは上司に気づかれる前に，2 つのスイッチを見つける必要がある．ダークライドを一回行う前に，エリカはすべての部屋の明かりを消した後，いくつかのスイッチを選んでスイッチをオンにすることができる．ダークライドが順に部屋を通過している間，明かりがついた部屋からついていない部屋に移動する，もしくは明かりがついていない部屋からついている部屋に移動するときに，エリカは乗客の興奮した叫び声を聞く．ダークライドの速さは変化しうるので，エリカはどの部屋の明かりがついているのか直接推測することはできないが，少なくとも叫び声のした回数は分かる．つまり，エリカは明かりのついた部屋からついていない部屋に移動した，もしくは，明かりのついていない部屋からついている部屋に移動した回数を知ることができる．

上司に気づかれる前に，エリカが最初と最後の部屋を管理する 2 つのスイッチを見つけるのを手伝ってくれないだろうか．あなたは，高々 30 回までダークライドを行うことができる．

インタラクション

この問題はインタラクティブ問題である．

- あなたのプログラムは最初に整数 N を読み込む．N はダークライドが通る部屋の個数である．
- その後，あなたのプログラムは採点プログラムとやり取りを行う．ダークライドを行うには，“`?`”の後に続き `0` と `1` からなる長さ N の文字列を一行に出力しなければならない．この文字列は各スイッチをオンにするかどうかを表しており，`0` はオフに，`1` はオンにすることを表す．その後，エリカが聞いた乗客の叫び声の回数を表す 1 つの整数 ℓ $(0 \leq \ell < N)$ を読み込まなければならない．
- 最後に答えを出力したいときは，“`!`”の後に続き 2 つの整数 A, B $(0 \leq A, B < N)$ を一行に出力しなければならない．あなたの解答が正答となるには，これらの整数が端の 2 つの部屋を管理しているスイッチの番号である必要がある．番号の順序は問わない．その後，あなたのプログラムは終了しなければいけない．

採点プログラムはアダプティブではない．すなわち，数列 p はやり取りが始まる前に決まっている．

各出力の最後には，必ず標準出力を flush せよ．さもなくば，Time Limit Exceeded と判定される可能性がある．Pythonの場合，`input()` により自動的に flush される．C++の場合，`cout << endl;` によって，改行されるとともに自動的に flush される．もし `printf` を使用する場合，`fflush(stdout)` を用いよ．

制約・採点形式

- $3 \leq N \leq 30\,000$.
- あなたのプログラムは最大 30 回ダークライドを行うことができる．答えの出力はこの回数に含まれない．もし超過した場合，Wrong Answerと判定される．

あなたの解答は各小課題ごとに評価され，小課題にはそれぞれ配点が割り当てられている．各小課題は複数のテストケースからなる．各小課題について得点を得るためには，その小課題に含まれるすべてのテストケースに正解する必要がある．

小課題	配点	制約
1	9	$N = 3$.
2	15	$N \leq 30$.
3	17	$p_0 = 0$, つまりスイッチ 0 が部屋 0 を管理している．
4	16	N は偶数であり，一方の端の部屋に対応するスイッチは前半の番号 $(0 \leq A < \frac{N}{2})$，もう一方の端の部屋に対応するスイッチは後半の番号 $(\frac{N}{2} \leq B < N)$ である．
5	14	$N \leq 1000$.
6	29	追加の制約はない．

テストのためのツール

あなたが解法をテストすることを容易にするため，簡単なツールがダウンロードできるようになっている．Kattisの問題ページ下部の “attachments” の項を見よ．ここで，本ツールを必ずしも使う必要はない．なお，Kattisで使用される実際の採点プログラムは，本ツールとは異なる．

本ツールを使うには，まず “sample1.in” のような入力ファイルを生成しなければならない．この入力ファイルでは，最初の行に整数 N，次の行に隠された順列 $p_0, p_1, ..., p_{N-1}$ を指定しなければならない．具体例を以下に示す．

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

Python の場合，たとえば `solution.py` の場合，(通常は `pypy3 solution.py` とすれば実行できるが) 以下を実行しなければならない．

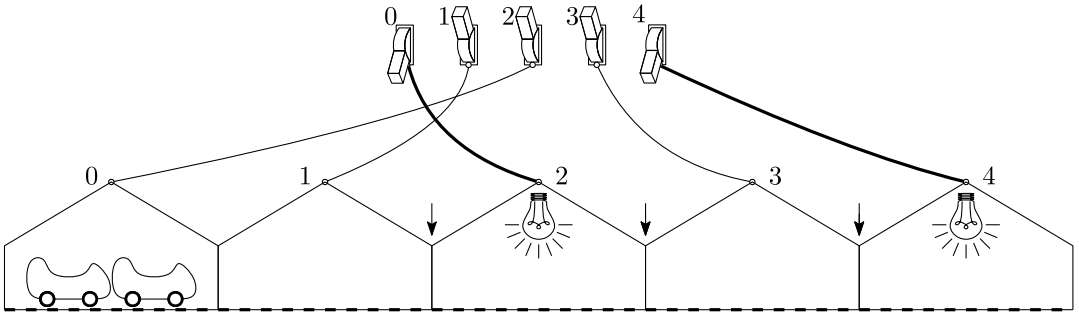
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

C++ の場合，まずコンパイルを行い (例: `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`)，その後以下を実行しなければならない．

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

例

サンプルテストケース 1 では，隠された順列は $[p_0, p_1, p_2, p_3, p_4] = [2, 1, 0, 3, 4]$ である．このテストケースは小課題 $2, 5, 6$ の制約を満たす．まず，正解例のプログラムは整数 $N = 5$ を読み込む．その後，このプログラムは $K = 2$ 個のスイッチ，スイッチ 4 とスイッチ 0，をオンにして行うダークライドを指定する．これらのスイッチは部屋 $p_4 = 4$ と部屋 $p_0 = 2$ を管理している (下図を見よ)．エリカは (図の矢印で記されているように) 3 回叫び声を聞く．1 回目は明かりのついていない部屋 1 からついている部屋 2 に移動するとき，2 回目は明かりのついている部屋 2 からついていない部屋 3 に移動するとき，3 回目は明かりのついていない部屋 3 からついている部屋 4 に移動するときである．その後，このプログラムは部屋 p_0, p_2, p_3 の明かりをつけて行うダークライドを指定する．このとき，エリカは 3 回叫び声を聞く．最後に，このプログラムは $A = 2, B = 4$ と答える．これらのスイッチは最初と最後の部屋を管理している ($p_2 = 0$ および $p_4 = 4$) ので，正答となる．$A = 4, B = 2$ としても正答となることに注意せよ．



サンプルテストケース 2 では，隠された順列は $[p_0, p_1, p_2] = [2, 0, 1]$ である．このテストケースは小課題 $1, 2, 5, 6$ の制約を満たす．正解例のプログラムは 3 つのスイッチすべてをオンにして行うダークライドを指定する．この場合すべての部屋の明かりがついているため，エリカが叫び声を聞くことはない．2 回目のダークライドでは，スイッチ 1 とスイッチ 0 をオンにする．このとき，部屋 $p_1 = 0$ と部屋 $p_0 = 2$ の明かりはついており，部屋 1 の明かりはついていない．エリカは部屋 0 (オン) から部屋 1 (オフ) に向かうときと部屋 1 (オフ) から部屋 2 (オン) に向かうときの 2 回叫び声を聞く．最後のダークライドでは，どのスイッチもオフであり，すべての部屋の明かりがついていない．この場合も，エリカが叫び声を聞くことはない．その後，このプログラムはスイッチ 1 とスイッチ 0 を答える．実際にこれらは最初と最後の部屋を管理している．“`! 0 1`” および “`! 1 0`” のいずれも正答となる．

サンプルテストケース 3 では，隠された順列は $[p_0, p_1, p_2, p_3] = [0, 1, 2, 3]$ である．このテストケースは小課題 $2, 3, 4, 5, 6$ の制約を満たす．正解例にある 1 回のダークライドでは答えを推測することができるとは限らないが，正解例のプログラムは運よく正しい答えを推測している．

サンプルテストケース 1

採点プログラムの出力	あなたのプログラムの出力
5	
	? 10001
3	
	? 10110
3	
	! 2 4

サンプルテストケース 2

採点プログラムの出力	あなたのプログラムの出力
3	
	? 111
0	
	? 110
2	
	? 000
0	
	! 1 0

サンプルテストケース 3

採点プログラムの出力	あなたのプログラムの出力
4	
	? 1010
3	
	! 0 3