

B. Temačna vožnja

Ime naloge	Temačna vožnja (Dark Ride)
Omejitev časa	1 sekunda
Omejitev spomina	1 gigabyte

Erika je pred kratkim začela opravljati počitniško delo v zabaviščnem parku Fantazijaland v bližini Bonna. Najeli so jo, da upravlja z lučmi v sobah, skozi katere poteka temačna vožnja.

Temačna vožnja poteka skozi N sob, ki so oštevilčene od 0 do $N - 1$. Skozi sobe se vlakec premika po vrsti - začne v sobi 0, zaključi pa v sobi $N - 1$. Luči v sobah upravlja N stikal (ravno tako so oštevilčena od 0 do $N - 1$), eno za vsako sobo. Stikalo s (za kateri velja $0 \leq s < N$) upravlja z lučmi v sobi p_s .

Erikin šef jo je prosil, naj prižge luči v prvi in zadnji sobi ter ugasne vse ostale. Sliši se preprosto, kajne? Samo vklopiti mora stikali A in B tako da je $p_A = 0$ in $p_B = N - 1$ (ali $p_B = 0$ in $p_A = N - 1$). Žal Erika ni bila povsem pozorna, ko ji je šef opisoval upravljanje, in **se ne spomni seznama p - torej, katero stikalo upravlja s katero sobo.**

Erika mora to ugotoviti, preden to opazi njen šef. Pred začetkom vsake vožnje Erika ugasne vse luči in nato lahko prižge nekaj stikal. Ko se vlak premika iz sobe v sobo, kadar koli se vožnja premika iz osvetljene v neosvetljeno sobo ali obratno, bo Erika slišala potnike, ki vznemirjeno kričijo. Hitrost vožnje se lahko spreminja, zato Erika ne more neposredno sklepati, katere sobe so osvetljene, bo pa vsaj slišala število krikov. To pomeni, da se bo naučila, kolikokrat se je vlak premaknil iz osvetljene v neosvetljeno sobo ali iz neosvetljene v osvetljeno sobo.

Ali lahko pomagaš Eriki ugotoviti, kateri dve stikali upravljata luči v prvi in zadnji sobi, preden to opazi njen šef? Uporabiš lahko največ 30 voženj.

Interakcija

To je interaktivna naloga.

- Tvoj program naj za začetek prebere vrstico, ki vsebuje N : število sob, ki sestavljajo temačno vožnjo.
- Potem naj tvoj program interagira z ocenjevalnikom. Za začetek vožnje moraš izpisati vrstico, ki se začne z vprašajem " ? ", nato pa niz dolžine N ki ga sestavljajo 0 (izklopljeno) in 1

(vklopljeno), kar označuje, kako nastaviš stikala N . Nato naj tvoj program prebere eno samo celo število ℓ ($0 \leq \ell < N$), ki pove kolikokrat Erika sliši krike potnikov.

- Ko želiš odgovoriti, izpiši vrstico s klicajem " ! ", ki mu sledita dve celi števili A in B ($0 \leq A, B < N$). Da bo tvoj odgovor sprejet, morata biti to indeksa stikal, ki krmilita obe krajni sobi, v poljubnem vrstnem redu. Po tem se mora tvoj program zaključiti.

Ocenjevalnik se ne prilagaja, kar pomeni, da je skrito zaporedje p določeno preden se interakcija začne.

Po vsaki vožnji se prepričaj, da si izpraznila standardni izhod, sicer se lahko zgodi, da tvoj program prekorači časovno omejitev. V Pythonu se to zgodi samodejno, če za branje vrstic uporabiš `input()`. V C++ `cout << endl;` poleg izpisa nove vrstice tudi izprazni vrstico; če uporabljaš `printf`, uporabi `fflush(stdout)`.

Omejitve in točkovanje

- $3 \leq N \leq 30\,000$.
- Uporabiš lahko največ 30 voženj (izpis na koncu ne šteje kot vožnja). Če prekoračiš to omejitev, boš dobila oceno "Wrong Answer".

Tvoja rešitev bo preizkušena na nizu testnih skupin, od katerih je vsaka vredna določeno število točk. Vsaka testna skupina vsebuje niz testnih primerov. Da bi dobila točke za testno skupino, mora tvoj program pravilno rešiti vse testne primere v testni skupini.

Skupina	Točke	Omejitve
1	9	$N = 3$
2	15	$N \leq 30$
3	17	$p_0 = 0$, to pomeni, da stikalo 0 upravlja z lučmi v sobi 0
4	16	N je sodo, in stikalo za eno krajno sobo je v prvi polovici ($0 \leq a < \frac{N}{2}$) in stikalo za drugo krajno sobo je v drugi polovici ($\frac{N}{2} \leq b < N$)
5	14	$N \leq 1000$
6	29	Brez dodatnih omejitev

Testno orodje

Za lažje testiranje svoje rešitve imaš na voljo preprosto orodje, ki ga lahko preneš. Poglej »priloge« na dnu strani s problemom Kattis. Uporaba orodja ni obvezna. Upoštevaj, da se uradni ocenjevalec Kattis razlikuje od priloženega orodja za testiranje.

Za uporabo orodja ustvari vhodno datoteko, kot je »sample1.in«, ki se mora začeti s številko N ki ji sledi vrstica z zaporedjem p_0, p_1, \dots, p_{N-1} , ki določa skrito permutacijo. Na primer:

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

Program v Pythonu, recimo `solution.py` (običajno se zažene kot `pypy3 solution.py`), zaženi z:

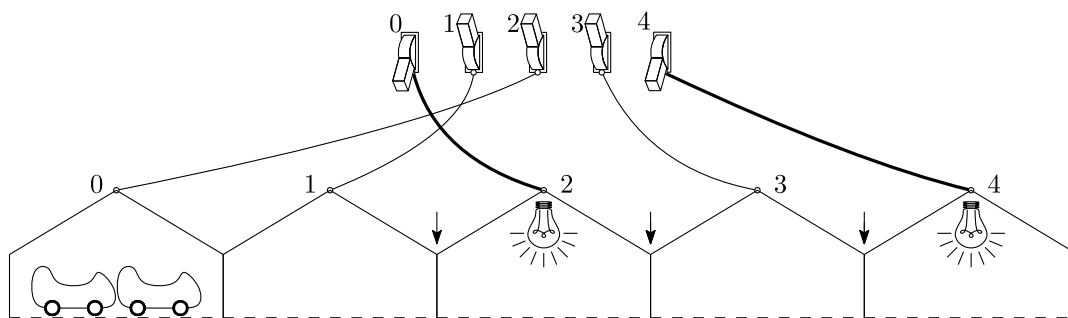
```
python3 orodje_z_testing.py pypy3 rešitev.py < vzorec1.in
```

Program v C++ najprej prevedi (npr. `g++ -g -O2 -std=gnu++23 -static solution.cpp -o solution.out`) in nato zaženi z:

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

Primeri

V prvem primeru je skrita permutacija $[p_0, p_1, p_2, p_3, p_4] = [2, 1, 0, 3, 4]$. To zaporedje izpolnjuje omejitve testnih skupin 2, 5 in 6. Najprej program prebere celo število $N = 5$, nato pa program zahteva vožnjo z vklopljenima dvema stikaloma: stikalom 4 in stikalom 0. Ti kontrolni sobi sta $p_4 = 4$ in $p_0 = 2$; glej spodnjo ilustracijo. Erika sliši 3 krike (označeni so s puščicami na sliki): prvič, ko se vožnja pelje iz neosvetljene sobe 1 v osvetljeno sobo 2; drugič iz osvetljene sobe 2 v neosvetljeno sobo 3; in tretjič, ko se pelje iz neosvetljene sobe 3 v osvetljeno sobo 4. Program nato zahteva še eno vožnjo, kjer so sobe p_0, p_2 in p_3 osvetljene, zaradi česar Erika sliši 3 krike. Nazadnje program odgovori z $A = 2$ in $B = 4$, kar je res pravilno, saj ti dve sobi nadzorujeta prvo in zadnjo sobo ($p_2 = 0$ in $p_4 = 4$). Upoštevaj, da bi bila tudi $A = 4$ in $B = 2$ pravi odgovor.



V drugem primeru je skrita permutacija $[p_0, p_1, p_2] = [2, 0, 1]$. To izpolnjuje omejitve testnih skupin 1, 2, 5 in 6. Program zahteva vožnjo, pri kateri so vsa tri stikala vklopljena. Ker to pomeni, da so vse sobe osvetljene, Erika ne bo slišala krikov. Med drugo vožnjo sta stikali 1 in 0 vklopljeni, zaradi česar sta sobi $p_1 = 0$ in $p_0 = 2$ osvetljeni, medtem ko soba 1 ni osvetljena. Erika sliši dva krika: ko se vožnja pelje iz sobe 0 (osvetljena) v sobo 1 (neosvetljena) in iz sobe 1 (neosvetljena) v sobo 2

(osvetljena). Med zadnjo vožnjo ni vklopljeno nobeno stikalo, kar pomeni, da so vse tri sobe neosvetljene in da Erika spet ne sliši krikov. Program nato odgovori s stikaloma 1 in 0 , ki dejansko upravljata prvo in zadnjo sobo. Sprejeta odgovora: » ! 0 1 « in ! 1 0 «.

V tretjem vzorcu je skrita permutacija $[p_0, p_1, p_2, p_3] = [0, 1, 2, 3]$. To izpolnjuje omejitve testnih skupin 2, 3, 4, 5 in 6. Upoštevajte, da po tej eni vožnji ni nujno mogoče sklepati na odgovor, vendar je vzorčna rešitev uganila odgovor in imela srečo.

Prvi primer

Izhod ocenjevalnika	Tvoj izhod
5	
	? 10001
3	
	? 10110
3	
	! 2 4

Drugi primer

Izhod ocenjevalnika	Tvoj izhod
3	
	? 111
0	
	? 110
2	
	? 000
0	
	! 1 0

Tretji primer

Izhod ocenjevalnika	Tvoj izhod
4	
	? 1010
3	
	! 0 3