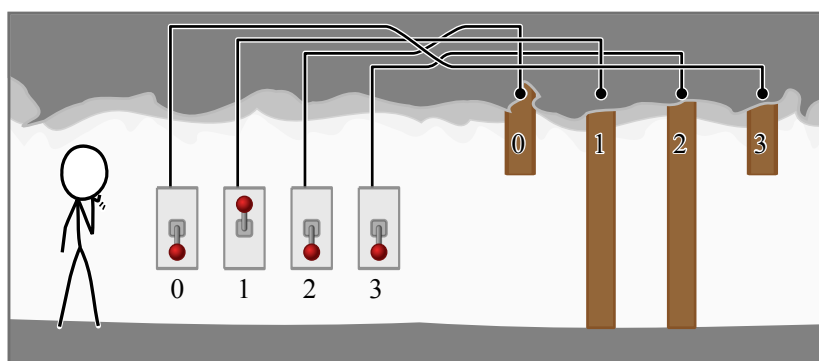


Ejot no koledžas uz UQ Centru, jūs esat apmaldījušies un uzgājuši ieeju slepenā alā, kas stiepjas dziļi zem universitātes. Alas ieejai priekšā ir N vienas otrām sekojošas durvis un N slēdži. Katram slēdzim atbilst savas durvis.



Durvis ir sanumurētas pēc kārtas ar skaitļiem $0, 1, \dots, (N - 1)$, Jums tuvākās ir 0-tās durvis. Arī slēdži ir sanumurēti ar skaitļiem $0, 1, \dots, (N - 1)$, bet jūs nezināt kurš slēdzis atbilst kurām durvīm.

Visi slēdži atrodas pie alas ieejas. Katram slēdzim ir iespējamās divas pozīcijas: *augšā* vai *lejā*. Tikai viena no pozīcijām katram slēdzim ir īstā. Ja slēdzis ir īstajā pozīcijā, tad tam atbilstošās durvis būs atvērtas, bet ja nebūs, tad durvis būs ciet. Katram slēdzim īstā pozīcija var būt sava un jūs nezināt kura tā ir.

Jūs gribat saprast šo drošības sistēmu. Lai to izdarītu, jūs varat ieslēgt slēdžus kaut kādos stāvokļos un tad iet iekšā alā un apskatīties kuras ir pirmās durvis, kas ir ciet. Durvīm nevar redzēt cauri: līdzko jūs nonākat pie ciet esošām durvīm, jūs neredzat tālākās durvis.

Jums ir laiks izmēģināt ne vairāk kā 70,000 slēdžu kombinācijas. Jūsu uzdevums ir noskaidrot katra slēdža īsto pozīciju un kuras durvis atbilst katram slēdzim.

Implementācija

Jums jāiesūta fails, kurā realizēta procedūra `exploreCave()`. Tā var izsaukt testēšanas funkciju `tryCombination()` ne vairāk kā 70,000 reizes, un beigās jāizsauc testēšanas procedūra `answer()`. Šīs funkcijas un procedūras aprakstītas zemāk.

Testēšanas funkcija: `tryCombination()`

C/C++ `int tryCombination(int S[]);`

Pascal `function tryCombination(var S: array of LongInt) : LongInt;`

Apraksts

Šī testētāja funkcija ir dota un tā ļauj izmēģināt slēdžu kombināciju un noskaidrot pirmās durvis kuras ir ciet. Ja visas durvis ir atvērtas, funkcija atgriezīs `-1`. Funkcijas laika sarežģītība ir $O(N)$; tas ir, tās izpildes laiks sliktākajā gadījumā ir proporcionāls N .

Šo funkciju var izsaukt ne vairāk kā `70,000` reizes.

Parametri

- `S`: Masīvs ar garumu N , aprakstot katra slēdža stāvokli. Masīva elements `S[i]` atbilst i -tajam slēdzim. Vērtība `0` nozīmē, ka slēdzis ir augšā, un vērtība `1`, ka lejā.
- *Rezultāts*: Pirmo durvju, kuras ir ciet, numurs, vai `-1`, ja visas durvis ir vaļā.

Testētāja procedūra: `answer()`

C/C++ `void answer(int S[], int D[]);`

Pascal `procedure answer(var S, D: array of LongInt);`

Apraksts

Izsauciet šo procedūru, kad esat noskaidrojuši slēdžu kombināciju, kas atver visas durvis, kā arī kuras durvis atbilst kuram slēdzim.

Parametri

- `S`: Masīvs ar garumu N , kurā aprakstītas visu slēdžu īstās pozīcijas. Formāts ir tāds pats kā funkcijā `tryCombination()`, kas aprakstīts iepriekš.
- `D`: Masīvs ar garumu N , kurā aprakstīts kuras durvis atbilst katram slēdzim. Elementā `D[i]` jāieraksta to durvju, kuras atbilst i -tajam slēdzim, numurs.
- *Rezultāts*: Šī procedūra neatgriež nekādu vērtību, bet liks programmai beigties.

Jūsu procedūra: `exploreCave()`

C/C++ `void exploreCave(int N);`

Pascal `procedure exploreCave(N: longint);`

Apraksts

Jūsu iesūtījumam jārealizē šī procedūra.

Šai procedūrai ir jālieto testētāja funkcija `tryCombination()` lai noskaidrotu īsto katru slēdža stāvokli un kuras durvis atbilst katram slēdzim. Kad visa šī informācija ir noskaidrota procedūrai vienu reizi jāizsauc `answer()`.

Parametri

- `N`: Slēdžu un durvju skaits alā.

Piemēra sesija

Pieņemsim, ka durvju un slēdžu atbilstība ir kā augstāk esošajā attēlā:

Funkcijas izsaukums	Atgriež	Paskaidrojums
<code>tryCombination([1, 0, 1, 1])</code>	1	Slēdžu stāvoklis atbilst attēlam: slēdži 0, 2 un 3 ir lejā, bet slēdzis 1 ir augšā. Funkcija atgriež 1, kas nozīmē, ka durvis 1 ir pirmās no kreisās puses, kuras ir ciet.
<code>tryCombination([0, 1, 1, 0])</code>	3	Durvis 0, 1 un 2 ir vaļā, bet durvis 3 ir ciet.
<code>tryCombination([1, 1, 1, 0])</code>	-1	Pārslēdzot slēdzi 0 uz leju visas durvis būs vaļā - uz to norāda atgrieztā vērtība -1.
<code>answer([1, 1, 1, 0], [3, 1, 0, 2])</code>	<i>(Programma beidzas)</i>	Mēs minam, ka īstā kombinācija ir [1, 1, 1, 0], un slēdži 0, 1, 2 un 3 atbilst attiecīgi durvīm 3, 1, 0 un 2.

Ierobežojumi

- Laika ierobežojums: 2 sekundes
- Atmiņas ierobežojums: 32 MiB
- $1 \leq N \leq 5,000$

Apakšuzdevumi

Apakšuzdevums	Punkti	Papildus ievada ierobežojumi
1	12	Katram i , i -tais slēdzis atbilst i -tajām durvīm. Jūsu uzdevums ir noskaidrot īsto kombināciju.
2	13	Īstā kombinācija vienmēr būs $[0, 0, 0, \dots, 0]$. Jūsu uzdevums ir noskaidrot kuras durvis atbilst katram slēdzim.
3	21	$N \leq 100$
4	30	$N \leq 2,000$
5	24	(Nav papildus ierobežojumu)

Eksperimentēšana

Jūsu datorā esošais piemēru vērtētājs ielasīs ievaddatus no `cave.in` faila šādā formātā:

- 1. rinda: `N`
- 2. rinda: `S[0] S[1] ... S[N - 1]`
- 3. rinda: `D[0] D[1] ... D[N - 1]`

,kur `N` ir slēdžu un durvju skaits, `S[i]` īstā pozīcija i -tajam slēdzim, un `D[i]` ir durvju, kuras atbilst i -tajam slēdzim, numurs.

Augstāk esošais piemērs jāapraksta šādā veidā:

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

Piezīmes par valodām

C/C++ Jums jāiekļauj `#include "cave.h"`.

Pascal Jums jādefinē `unit Cave`, un jums jāimportē testētāja metodes ar `uses GraderHelpLib`. Visi masīvi tiek numurēti sākot no `0` (nevis `1`).

Iepazīstieties ar risinājumu piemēru šabloniem uz jūsu datora.