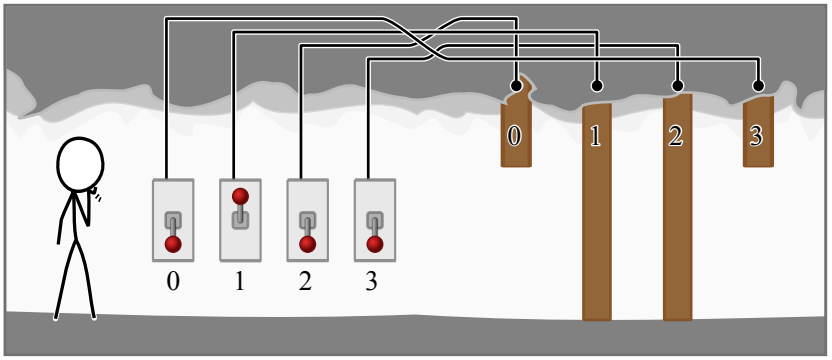
International Olympiad in Informatics 2013
6-13 July 2013
Brisbane, Australia
Day 2 tasks

cave

Croatian — 1.0

Aron i Anton su se opetovano izgubili u UQ kompleksu. Na putu prema dvorani za natjecanje gdje će saznati rezultate naišli na spilju blokiranu sigurnosnim sistemom od `N` uzastopnih vrata koja su spojena na `N` prekidača nepoznatim redoslijedom.



Vrata su označena brojevima `0, 1, …, (N - 1)` na način da su vrata `0` najbliža junacima. Prekidači su označeni brojevima `0, 1, …, (N - 1)`, no nije poznato koji je prekidač spojen s kojim vratima.

Svi prekidači su postavljeni na samom ulazu u sistem. Svaki prekidač može biti u položaju *gore* ili *dolje*. Samo jedna od ovih pozicija je točna za pojedini prekidač. Ako je prekidač u točnoj poziciji tada se vrata spojena na taj prekidač otvaraju, dok se u suprotnom ista vrata zatvaraju. Točne pozicije mogu biti različite za pojedine prekidače i nije poznato koje su točne pozicije za koje prekidače.

Junaci žele shvatiti sigurnosni sistem, no nitko se nije našao u blizini da im ga objasni. Stoga postavljaju prekidače u neke položaje i kreću zajedno šetati kroz sistem do prvih zatvorenih vrata. Vrata su neprozirna: jednom kada junaci dođu do prvih zatvorenih vrata, neće moći vidjeti stanje ostalih vrata nakon tih.

Junaci imaju vremena napraviti samo `70,000` postavljanja prekidača i šetnji kroz sistem prije no što rezultati natjecanja koje ih zanima postanu dostupni. Vaš zadatak je pomoći junacima odrediti točnu poziciju za svaki prekidač i zaključiti koja vrata su spojena na koji prekidač.

Implementacija

Potrebno je priložiti datoteku koja implementira funkciju `exploreCave()`. Ova funkcija može zvati funkciju grejdera `tryCombination()` najviše 70,000 puta i mora završiti pozivom funkcije grejdera `answer()`. Ove funkcije su opisane ispod.

Funkcija grejdera: `tryCombination()`

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

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

Opis

Grejder će implementirati ovu funkciju. Ona omogućuje vašem rješenju da proba neku kombinaciju prekidača i da mu vrati oznaku prvih zatvorenih vrata. Ako su uz ovu kombinaciju sva vrata otvorena, funkcija će vratiti `-1`. Ova funkcija se pokreće u `O(N)` vremenu tj. vrijeme pozivanja je proporcionalno broju vrata - `N`.

Ova funkcija moći će se pozvati najviše `70,000` puta.

Parametri

- `S`: Niz dužine `N` koji predstavlja položaje svakog prekidača. Element `S[i]` predstavlja prekidač s oznakom `i`. Vrijednost `0` predstavlja položaj *gore*, dok vrijednost `1` predstavlja položaj *dolje*.
- *Povratna vrijednost*: Oznaka prvih vrata koja su zatvorena ili `-1` ukoliko su sva vrata otvorena.

Funkcija grejdera: `answer()`

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

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

Opis

Vaše rješenje poziva ovu funkciju nakon što je uspjelo zaključiti koja su vrata spojena na koje prekidače te koje su točne pozicije svakog prekidača.

Parametri

- `S`: Niz duljine `N` koji predstavlja točan položaj svakog prekidača. Vrijednosti položaja su iste kao i kod funkcije `tryCombination()` koja je opisane gore.
- `D`: Niz duljine `N` koji predstavlja vrata koja su spojena na svaki prekidač. Element `D[i]` treba predstavljati oznaku vrata na koje je spojen prekidač s oznakom `i`.
- *Povratna vrijednost*: Ova funkcija neće imati povratnu vrijednost, odnosno uzrokovati će završetak programa.

Vaša funkcija: `exploreCave()`

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

Pascal
```
procedure exploreCave(N: longint);
```

Opis

Vaše rješenje mora implementirati ovu funkciju.

Ova funkcija treba koristiti funkciju grejdera `tryCombination()` kako bi zaključila točne pozicije svakog prekidača i oznake vrata koje su spojene na svaki prekidač. Nakon što to napravi treba pozvati funkciju grejdera `answer()`.

Parametri

- `N`: Broj prekidača i vrata u sistemu.

Testni podaci

Pretpostavimo da su vrata i prekidači spojeni kao na slici iznad:

Poziv funkcije	Povratna vrijednost	Objašnjenje
`tryCombination([1, 0, 1, 1])`	`1`	Ovaj poziv odgovara slici. Prekidači 0, 2 i 3 su dolje, dok je prekidač 1 gore. Funkcija vraća `1` što znači da su vrata s oznakom 1 prva vrata s lijeva koja su zatvorena.
`tryCombination([0, 1, 1, 0])`	`3`	Vrata 0, 1 i 2 su otvorena, dok su vrata 3 zatvorena.
`tryCombination([1, 1, 1, 0])`	`-1`	Mijenjajući prekidač 0 prema dolje uzrokuje otvaranje svih vrata što je predstavljeno povratnom vrijednosti `-1`.
`answer([1, 1, 1, 0], [3, 1, 0, 2])`	*(Program završava)*	Zaključujemo da je točna kombinacija `[1, 1, 1, 0]`, i da su prekidači 0, 1, 2 i 3 spojeni na vrata 3, 1, 0 i 2 respektivno.

Ograničenja

- Vremensko ograničenje: 2 sekunde
- Memorijsko ograničenje: 32 MB
- $1 \le N \le 5{,}000$

Bodovanje

Podzadatak	Bodovi	Dodatna ograničenja
1	12	Za svaki `i`, prekidač `i` je spojen na vrata `i`. Vaš zadatak je samo otkriti točne položaje.
2	13	Točni položaji će uvijek biti `[0, 0, 0, …, 0]`. Vaš zadatak je samo otkriti koji su prekidači spojeni na koja vrata.
3	21	$N \le 100$
4	30	$N \le 2{,}000$
5	24	*(n/a)*

Lokalno testiranje

Grejder na vašem računalu će pročitati ulaz iz datoteke `cave.in` koja mora biti u sljedećem obliku:

- redak 1: `N`
- redak 2: `S[0] S[1] … S[N - 1]`
- redak 3: `D[0] D[1] … D[N - 1]`

Ovdje je `N` broj vrata i prekidača, `S[i]` je točan položaj za prekidač `i`, a `D[i]` je oznaka vrata koja su spojena na prekidač `i`.

Npr., prvi primjer bi bio predstavljen:

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

Napomene

C/C++	Potrebno je dodati `#include "cave.h"`.
Pascal	Potrebno je definirati `unit Cave` te je potrebno uključiti `uses GraderHelpLib`. Svi nizovi su numerirani od `0` (a ne od `1`).

Pogledajte predloške rješenja na vašem računalu.