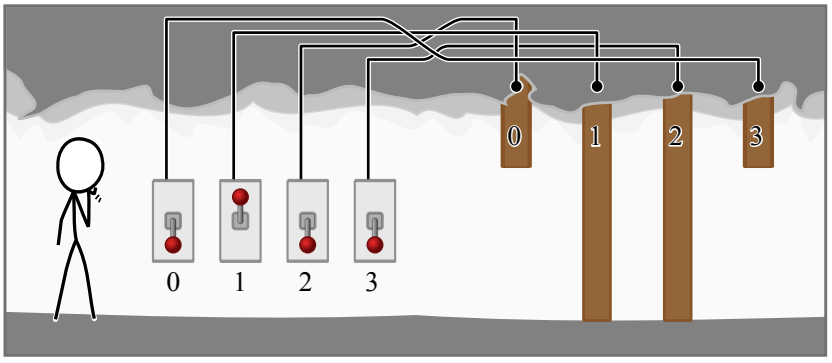
International Olympiad in Informatics 2013
6-13 July 2013
Brisbane, Australia
Day 2 tasks

pećina

Crnogorski — 1.0

Tokom duge šetnje kroz UQ koledž, upali ste u sakriveni sistem podzemnih pećina ispod univerzitetskog kompleksa. Ulaz pećine blokiran je sigurnosnim sistemom koji se sastoji od `N` uzastopnih vrata. Vrata su poređana jedna iza drugih. Postoji `N` prekidača, jedan prekidač za jedna vrata.



Vrata su numerisana sa `0, 1, ..., (N - 1)`, pri čemu su vrata 0 vama najbliža. Prekidači su numerisani sa `0, 1, ..., (N - 1)`, pri čemu ne znate koji prekidač je povezan na koja vrata.

Svi prekidači su postavljeni na ulaz u pećinu. Prekidači mogu biti u dva položaja *upaljen* (gore) ili *ugašen* (dolje). Za svaki prekidač samo je jedna pozicija ispravna. Ako je pozicija prekidača ispravna, tada su njemu odgovarajuća vrata otvorena, inače, ako je položaj prekidača pogrešan tada su vrata sa kojima je povezan zatvorena. Ispravan položaj razlikuje se za različite prekidače, i unaprijed se ne zna koji je položaj ispravan.

Zadatak je da razradite opisani sigurnosni sistem. Dozvoljeno je da postavite prekidače u bilo koju kombinaciju, a onda uđete u pećinu da bi naišli na prva zatvorena vrata. Vrata nijesu providna, tj. kada prvi put naiđete na zatvorena vrata, ne možete vidjeti preostala vrata iza njih.

Imate vremana da probate `70,000` kombinacija za položaje prekidača, ali ne više od tog broja. Vaš zadatak je da odredite ispravni položaj za svaki prekidač i takođe da odredite na koji način su prekidači povezani sa vratima.

Implementacija

Potrebno je da implementirate proceduru `exploreCave()`. Ova procedura može da poziva funkciju `tryCombination()` iz grejdera najviše 70000 puta i mora da završi pozivom procedure `answer()` iz grejdera. Ove funkcije i procedure opisane su u nastavku.

Funkcija iz grejdera: `tryCombination()`

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

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

Opis

Ova funkcija implementirana je unutar grejdera. Omogućava vam da provjerite kombinaciju položaja prekidača, i ulaskom u pećinu otkrijete prva zatvorena vrata. Ako su za datu kombinaciju prekidača sva vrata otvorena onda funkcija vraća `-1`. Vrijeme izvršavanja ove funkcije je `O(N)`, tj. u najgorem slučaju je proporcionalno sa `N`.

Ova funkcija poziva se najviše 70000 puta.

Parametri

- `S`: Niz dužine `N` koji specifikuje položaj svakog prekidača. Element `S[i]` odgovara prekidaču `i`. Vrijednost `0` označava da je prekidač u položaju GORE, dok vrijednost `1` označava da je prekidač u položaju DOLJE.
- *Vraća*: Broj vrata koja su prva zatvorena, ili `-1` ako su sva vrata otvorena.

Funkcija iz grejdera: `answer()`

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

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

Opis

Ovu funkciju pozivate kada otkrijete kombinaciju položaja prekidača koja otvara sva vrata i kada otkrijete koji prekidač je povezan na koja vrata.

Parametri

- `S`: Niz dužine `N`, koji sadrži ispravne položaje za svaki prekidač. Argument se poklapa sa argumentom funkcije `tryCombination()` opisanom gore.
- `D`: Niz dužine `N`, koji sadrži vezu svakog prekidača sa odgovarajućim vratima. Preciznije, elemenat `D[i]` sadrži broj vrata sa kojima je povezan prekidač `i`.
- *Vraća*: Ova procedura ne vraća vrijednost, ali uzrokuje završetak programa.

Vaša procedura: `exploreCave()`

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

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

Opis

Vaše rješenje mora da implementira ovu funkciju.

Ova funkcija koristi rutinu `tryCombination()` iz grejdera da bi odredila ispravnu poziciju svakog prekidača i vezu prekidača sa odgovarajućim vratima. Po određivanju ovih informacija obavezno poziva funkciju `answer()`.

Parametri

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

Primjeri

Pretpostavite da su vrata i prekidači organizovani kao na slici sa početka:

Poziv funkcije	Vraća	Objašnjenje
`tryCombination([1, 0, 1, 1])`	`1`	Ova situacija odgovara slici na početku. Prekidači 0, 2 i 3 su DOLJE, dok je prekidač 1 GORE. Funkcija vraća `1`, što znači da su vrata 1 prva vrata sa 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`	Mijenjanjem položaja prekidača 0 na DOLJE postiže se da su sva vrata otvorena, na što ukazuje povratna vrijednost `-1`.
`answer([1, 1, 1, 0], [3, 1, 0, 2])`	*(Program exits)*	Pogađamo da je ispravna kombinacija `[1, 1, 1, 0]`, i da su prekidači 0, 1, 2 i 3 povezani sa vratima 3, 1, 0 i 2 respektivno.

Ograničenja

- Vrijeme: 2 sekunde
- Memorija: 32 MB
- $1 \le N \le 5{,}000$

Podzadaci

Podzadaci	Bodovi	Dodatna ograničenja ulazanih podataka
1	12	Za svako `i`, prekidač `i` je povezan sa vratima `i`. Vaš zadatak je da samo odredite ispravnu kombinaciju za položaje prekidača.
2	13	Ispravna kombinacija položaja je uvijek `[0, 0, 0, ..., 0]`. Vaš zadatak je da samo odredite koji prekidač je povezan na koja vrata.
3	21	$N \le 100$
4	30	$N \le 2{,}000$
5	24	*(Nema ograničenja)*

Eksperimenti

Grejder na vašem računaru učitava podtke iz datoteke `cave.in`, koja mora biti u sljedećem formatu:

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

Ovdje je `N` broj vrata i prekidača, `S[i]` je ispravan položaj za prekidač `i` i `D[i]` je broj vrata na koja je povezan prekidač `i`.

Prethodni primjer bio bi zapisan u sljedećem formatu:

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

Napomene vezane za jezik implementacije

C/C++	Morate napisati `#include "cave.h"`.
Pascal	Morate definisati `unit Cave`, i morate uključiti funkcije iz grejdera pomoću `uses GraderHelpLib`. Indeksiranje nizova je od `0` (ne od `1`).

Proučite primjere na vašem računaru za dodatna pojašnjenja.