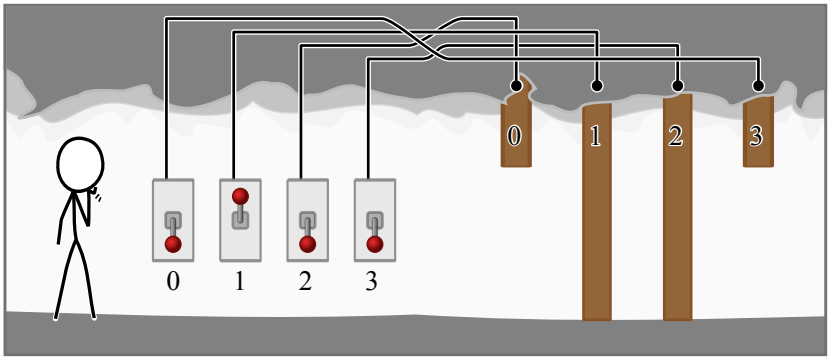
International Olympiad in Informatics 2013
6-13 July 2013
Brisbane, Australia
Day 2 tasks

cave

Estonian — 1.0

Eksinud ära pikal teel kolledžist UQ Centre'isse, leidsid Sa ülikooli all asuva salakoopa sissepääsu. Koopa sissepääsu juures on `N` järjestikusest uksest koosnev turvasüsteem, kus iga uks on eelmise taga, ja `N` lülitit, igaüks ühendatud erineva uksega.



Uksed on nummerdatud järjest `0, 1, …, (N - 1)`, kus uks 0 on Sulle lähim. Ka lülitid on nummerdatud `0, 1, …, (N - 1)`, aga Sa ei tea, millisele lülitile vastab milline uks.

Lülitid asuvad kõik koopa alguses sissepääsu juures. Iga lüliti võib olla *ülemises* või *alumises* asendis. Iga lüliti jaoks on ainult üks asend õige. Kui lüliti on õiges asendis, siis on sellele vastav uks avatud, ja kui lüliti on vales asendis, siis suletud. Õige asend võib eri lülitite jaoks olla erinev ja Sa ei tea, millised asendid on õiged.

Sa soovid sellest turvasüsteemist aru saada. Selleks võid Sa kõik lülitid panna ükskõik millistesse asenditesse ja siis jalutada koopasse vaatama, missugune on esimene suletud uks. Uksed ei paista läbi: Sa ei näe ühtegi ust esimese suletud ukse taga.

Sul on aega proovida vaid `70 000` lülitite kombinatsiooni. Sinu ülesanne on leida iga lüliti jaoks õige asend ja sellele vastav uks.

Realisatsioon

Lahendusena tuleb esitada fail, mis sisaldab protseduuri `exploreCave()`. See võib kasutada hindamissüsteemi funktsiooni `tryCombination()` kuni `70 000` korda ja peab lõpuks kutsuma välja hindamissüsteemi protseduuri `answer()`. Funktsiooni ja protseduure on kirjeldatud järgnevalt.

Hindamissüsteemi funktsiooni `tryCombination()` deklaratsioon:

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

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

Kirjeldus

See funktsioon on olemas hindamissüsteemis. Sellega saad Sa proovida lülitite kombinatsiooni ja seejärel siseneda koopasse esimese suletud ukse tuvastamiseks. Kui kõik uksed on avatud, tagastab funktsioon `-1`. Selle funktsiooni tööaeg on `O(N)`, ehk ei kasva kiiremini kui võrdeliselt `N` väärtusega.

Seda funktsiooni võib kasutada ülimalt `70 000` korda.

Parameetrid

- `S`: massiiv pikkusega `N`, kus on iga lüliti asend. Element `S[i]` vastab lülitile `i`. Väärtus `0` tähendab, et lüliti on ülemises, ja väärtus `1`, et alumises asendis.
- *Tagastab*: esimese suletud ukse numbri või `-1`, kui kõik uksed on avatud.

Hindamissüsteemi protseduuri `answer()` deklaratsioon:

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

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

Kirjeldus

Kutsu seda protseduuri välja üks kord, kui oled tuvastanud lülitite kombinatsiooni, mis avab kõik uksed, ja iga lüliti jaoks ukse, mille see avab.

Parameetrid

- `S`: massiiv pikkusega `N`, kus on iga lüliti jaoks õige asend. Formaat on sama, mis funktsiooni `tryCombination()` puhul.
- `D`: massiiv pikkusega `N`, kus on iga lüliti jaoks sellele vastav ukse number. Täpsemalt peab element `D[i]` sisaldama selle ukse numbrit, millega on ühendatud lüliti `i`.
- *Tagastab*: see protseduur ei tagasta midagi, kuid lõpetab programmi töö.

Sinu protseduuri `exploreCave()` deklaratsioon:

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

Pascal `procedure exploreCave(N: longint);`

Kirjeldus

Sinu lahendus peab sisaldama seda protseduuri.

See protseduur peaks kasutama hindamissüsteemi funktsiooni `tryCombination()`, et tuvastada iga lüliti jaoks õige asend ja vastav uks, ja peab kutsuma välja protseduuri `answer()`, kui on need andmed tuvastanud.

Parameetrid

- `N`: lülitite ja uste arv koopas.

Interaktsiooni näide

Olgu uksed ja lülitid seadistatud nii, nagu näidatud eeloleval joonisel:

Väljakutse	Tagastab	Selgitus
`tryCombination([1, 0, 1, 1])`	`1`	See vastab joonisele. Lülitid 0, 2 ja 3 on alumises ning lüliti 1 ülemises asendis. Funktsioon tagastab `1`, näidates, et uks number 1 on vasakult vaadates esimene suletud uks.
`tryCombination([0, 1, 1, 0])`	`3`	Uksed 0, 1 ja 2 on avatud, uks 3 suletud.
`tryCombination([1, 1, 1, 0])`	`-1`	Lüliti 0 alumisse asendisse viimine põhjustab kõigi uste avanemise, nagu näitab ka tagastatud väärtus `-1`.
`answer([1, 1, 1, 0], [3, 1, 0, 2])`	*Programm lõpetab töö*	Me arvame, et õige kombinatsioon on `[1, 1, 1, 0]` ning lülitid 0, 1, 2 ja 3 on ühendatud vastavalt ustega 3, 1, 0 ja 2.

Piirangud

- Ajalimiit: 2 sekundit
- Mälulimiit: 32 MiB
- $1 \le N \le 5\,000$

Alamülesanded

Alamülesanne	Punkte	Täiendavad sisendi kitsendused
1	12	Iga `i` puhul on lüliti `i` ühendatud uksega `i`. Sinu ülesanne on tuvastada õige kombinatsioon.
2	13	Õige kombinatsioon on alati `[0, 0, 0, …, 0]`. Sinu ülesanne on tuvastada, millisele lülitile vastab milline uks.
3	21	$N \leq 100$
4	30	$N \leq 2\,000$
5	24	*Puuduvad*

Katsetamine

Sinu arvutis ülesande materjalide hulgas olev hindamisprogramm loeb sisendi failist `cave.in`, mis peab olema järgmises vormingus:

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

`N` on lülitite ja uste arv, `S[i]` lüliti `i` õige asend ja `D[i]` lülitile `i` vastav uks.

Eelnev näide kirjeldatud vormingus:

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

Keelespetsiifilised märkused

C/C++	Sa pead kaasama: `#include "cave.h"`.
Pascal	Sa pead defineerima: `unit Cave`, ja kaasama hindamisüsteemi funktsionaalsuse: `uses GraderHelpLib`. Kõigi massiivide indeksid algavad `0` st (mitte `1` st).

Vaata näidetena ka oma arvutis olevaid programmipõhju.