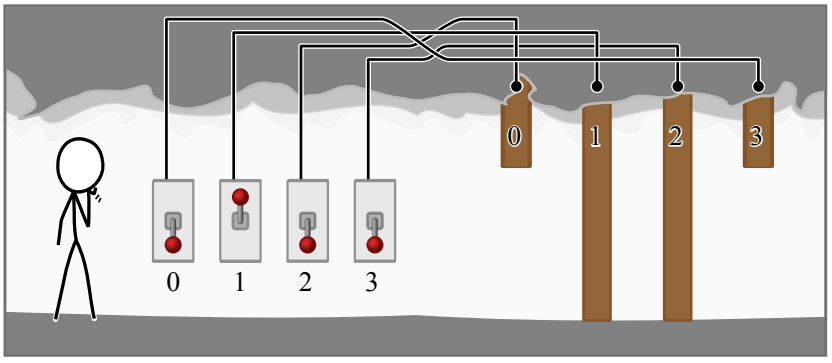
International Olympiad in Informatics 2013
6-13 July 2013
Brisbane, Australia
Day 2 tasks

агуй
Англи — 1.0

Та Квинслэндийн их сургуулийн төвөөс коллеж руу явж байхдаа төөрсөн тул уг их сургуулийн доор газрын гүнд байрлах агуйн системийн үүдэнд ирсэн. Энэ үүдийг ар араасаа дараалан байрласан N тооны хаалга болон тус бүр нь ялгаатай хаалгатай холбогдсон N тооны унтраалганаас тогтох хамгаалалтын системээр тоноглосон.



Хаалгануудыг $0, 1, \ldots, (N - 1)$ гэсэн эрэмбээр дугаарласан ба танд хамгийн ойр байгаа хаалга 0 гэсэн дугаартай. Унтраалгануудыг мөн $0, 1, \ldots, (N - 1)$ гэж дугаарласан боловч та аль унтраалга нь аль хаалганых болохыг мэдэхгүй байгаа.

Бүх унтраалгануд агуйн үүдэнд байрлана. Унтраалга бүр *дээш* эсвэл *доош* гэсэн байрлалын аль нэгэнд байна. Унтраалга бүрийн хувьд энэ хоёр байрлалын зөвхөн нэг нь л зөв байна. Хэрэв ямар нэг унтраалга зөв байрлалдаа байгаа бол түүнтэй холбогдсон хаалга нээгдэх ба хэрэв уг унтраалга буруу байрлалд байвал түүнтэй холбогдсон хаалга хаагдана. Ялгаатай унтраалгуудын хувьд зөв байрлалууд нь ялгаатай байж болох ба та аль байрлалууд нь зөв болохыг мэдэхгүй байгаа.

Та энэ хамгаалалтын системийн учрыг олохыг хүсч байгаа. Ингэхийн тулд та унтраалгуудын ямар нэг хослолыг (комбинацийг) тохируулаад агуй руу орж, хамгийн эхний түгжээтэй хаалга аль нь байгааг олж харна. Хаалгануд нэвт харагддаггүй: та хамгийн эхний түгжээтэй хаалган дээр ирээд цаана нь байгаа хаалгануудыг харж чадахгүй.

Танд унтраалгануудын $70,000$ хослолыг турших л хугацаа байгаа ба үүнээс илүү хугацаа байхгүй. Таны даалгавар бол унтраалга бүрийн зөв байрлалыг болон унтраалга бүр аль хаалганд холбогдсон байгааг тодорхойлох явдал юм.

Хэрэгжүүлэлт

Та `exploreCave()` функцийн хэрэгжүүлэлтийг агуулсан файлыг илгээх хэрэгтэй. Энэ нь шалгагчийн `tryCombination()` функцийг 70,000 хүртлэх удаа дуудаж болох ба шалгагчийн `answer()` функцийг дуудаад дуусах ёстой. Эдгээр функцуудийг доор тайлбарласан.

Шалгагчийн Функц: `tryCombination()`

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

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

Тайлбар

Энэ функц шалгагч дотор байгаа. Уг функц нь танд унтраалгуудын ямар нэг хослолыг туршиж, агуйд орсны дараа түгжээтэй байгаа хамгийн эхний хаалгыг тодорхойлох боломж олгоно. Хэрэв бүх хаалга нээлттэй бол уг функц `-1` утга буцаана. Энэ функц нь `O(N)` хугацаанд ажиллах ба өөрөөр хэлбэл ажиллах хугацаа нь хамгийн муудаа `N`-тэй пропорциональ байна.

Уг функцийг дээд тал нь `70,000` удаа дуудаж болно.

Параметрүүд

- `S`: Унтраалга бүрийн байрлалыг заах `N` урттай массив. `S[i]` элемент нь `i`-р унтраалгад харгалзана. `0` утга нь тухайн унтраалга дээш байрлалтай, `1` утга нь тухайн унтраалга доош байрлалтай байгааг илэрхийлнэ.
- *Буцаах утга*: Түгжээтэй байгаа хамгийн эхний хаалганы дугаарыг, харин бүх хаалга нээлттэй бол `-1` утгыг буцаана.

Шалгагчийн Функц: `answer()`

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

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

Тайлбар

Та бүх хаалгыг нээх унтраалгуудын хослолыг болон аль хаалга аль унтраалгад холбогдсоныг олсны дараа энэ функцийг дуудна.

Параметрүүд

- `S`: Унтраалга бүрийн зөв байрлалыг хадгалах `N` урттай массив. Формат нь дээр өгүүлсэн `tryCombination()` функцийнхтэй адил.
- `D`: Унтраалга бүр аль хаалгатай холбогдсоныг хадгалах `N` урттай массив. Өөрөөр хэлбэл, `D[i]` элементэд `i`-р унтраалгатай холбоотой хаалганы дугаар байна.
- *Буцаах утга*: Энэ функц ямар нэг утга буцаахгүй ба харин програмыг төгсгөнө.

Таны Функц: `exploreCave()`

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

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

Тайлбар

Таны илгээсэн бодолтод энэ функцийн хэрэгжүүлэлт байна.

Уг функц нь шалгагчийн `tryCombination()` функцийг ашиглан унтраалга бүрийн зөв байрлал болон унтраалга бүрт холбогдсон хаалгыг олох ба мэдээллийг олсны дараа `answer()` функцийг нэг удаа дуудах ёстой.

Параметрүүд

- `N`: Агуй дахь унтраалга болон хаалганы тоо.

Жишээ Өгөгдөл

Хаалга болон унтраалгууд дээрх зурагт үзүүлсэн байдлаар байгаа гэж үзье:

Функцийн Дуудалт	Буцаах утга	Тайлбар
`tryCombination([1, 0, 1, 1])`	`1`	Энэ нь зураг дээр байгаатай тохирно. 0, 2 болон 3-р унтраалгууд доош, 1-р унтраалга дээш байрлалтай байна. Уг функц `1` утгыг буцаах ба энэ нь түгжээтэй байгаа хамгийн эхний хаалга нь 1-р хаалга юм гэдгийг харуулж байна.
`tryCombination([0, 1, 1, 0])`	`3`	0, 1 болон 2-р хаалгууд бүгд нээлттэй ба 3-р хаалга хаалттай.
`tryCombination([1, 1, 1, 0])`	`-1`	0-р унтраалгыг доош болгосноор бүх хаалга нээгдэх ба `-1` утгыг буцаана.
`answer([1, 1, 1, 0], [3, 1, 0, 2])`	*(Програм төгснө)*	Бид зөв байрлал нь `[1, 1, 1, 0]` ба 0, 1, 2 болон 3-р унтраалгууд харгалзан 3, 1, 0 ба 2-р хаалгуудтай холбоотой гэдгийг тааж байна.

Хязгаарлалтууд

- Хугацааны хязгаарлалт: 2 секунд
- Санах ойн хязгаарлалт: 32 Мб
- $1 \le N \le 5,000$

Дэд бодлогууд

Дэд бодлого	Оноо	Оролтын Нэмэлт Хязгаарлалтууд
1	12	`i` бүрийн хувьд `i`-р унтраалга `i`-р хаалгатай холбогдсон. Таны даалгавар бол зөвхөн зөв хослолыг тодорхойлох явдал юм.
2	13	Зөв хослол нь үргэлж `[0, 0, 0, …, 0]` байна. Таны даалгавар бол зөвхөн аль унтраалга аль хаалганд холбогдсоныг олох явдал юм.
3	21	$N \le 100$
4	30	$N \le 2,000$
5	24	*(Байхгүй)*

Туршилт

Таны компьютер дээр байгаа жишээ шалгагч нь өгөгдлийг `cave.in` файлаас уншина. Энэ файл нь дараах форматтай байна:

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

Энд, `N` нь хаалга болон унтраалгуудын тоо, `S[i]` нь `i`-р унтраалганы зөв байрлал ба `D[i]` нь `i`-р унтраалгатай холбогдсон хаалганы дугаар юм.

Дээрх жишээ дараах форматтай байна:

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

Хэлнүүдийн тайлбар

C/C++	Таны програм `#include "cave.h"` мөрийг агуулсан байх ёстой.
Pascal	Та `unit Cave`-г зарлах ёстой ба `uses GraderHelpLib`-г ашиглан шалгагчийн функцуудыг оруулж ирэх ёстой. Бүх массивууд `0`-ээс эхлэн дугаарлагдсан (`1`-ээс эхлэхгүй).

Жишээ болгон компьютер дээрээ байгаа бодолтын загваруудыг харна уу.