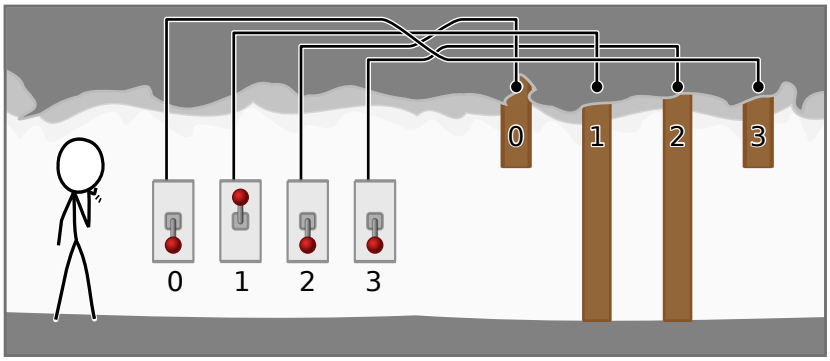
International Olympiad in Informatics 2013
6-13 July 2013
Brisbane, Australia
Day 2 tasks

cave

Russian — 1.0

При поиске места проведения соревнования — UQ Центра, вы заблудились и оказались в секретной группе пещер под университетом. Вход в группу пещер преграждает система безопасности, включающая в себя `N` дверей, расположенных друг за другом, и `N` переключателей, каждый из которых соединён только с одной из дверей. Различные переключатели соединены с различными дверями.



Двери пронумерованы от входа в группу пещер от `0` до `(N - 1)` в порядке от ближних к дальним. Переключатели также пронумерованы от `0` до `(N - 1)`, однако, вы не знаете, с какой дверью соединен каждый переключатель.

Все переключатели расположены у входа в группу пещер. Каждый переключатель может быть в одном из двух положений: "вверх" или "вниз". Одно из этих положений является правильным. Если он находится в правильном положении, то соединенная с переключателем дверь будет открыта, иначе эта дверь будет закрыта. Правильное положение может различаться для разных переключателей, и вы не знаете, какое из положений для какого переключателя является правильным.

Вы хотите понять, как устроена система безопасности. Для этого вы можете задать комбинацию положений переключателей, то есть установить каждый из переключателей в любое из положений; после чего войти в пещеру и узнать номер первой закрытой двери. Двери, находящиеся за ней, не видны.

У вас есть время на то, чтобы попробовать не более `70 000` комбинаций положений переключателей. Ваша задача — определить правильное положение для каждого переключателя, а также для каждого переключателя найти дверь, с которой он соединен.

Детали реализации

Ваше решение должно реализовывать функцию `exploreCave()`. Она может вызывать функцию проверяющего модуля `tryCombination()` не более 70 000 раз и должна завершаться вызовом функции `answer()`. Эти функции описаны ниже.

Функция проверяющего модуля: `tryCombination()`

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

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

Описание

Эту функцию реализует проверяющий модуль. Она позволяет вам задать комбинацию положений переключателей для того, чтобы войти в группу пещер и узнать номер первой закрытой двери. Если все двери открыты, то функция возвращает значение `-1`. Эта функция работает за время `O(N)`, то есть время работы в худшем случае пропорционально `N`.

Эта функция может быть вызвана не более `70 000` раз.

Параметры

- `S`: массив длины `N`, задающий комбинацию положений переключателей. Элемент массива `S[i]` соответствует переключателю с номером `i`. Значение `0` означает, что переключатель находится в положении "вверх". Значение `1` означает, что переключатель находится в положении "вниз".
- *Возвращаемое значение*: номер первой закрытой двери или `-1`, если все двери открыты.

Функция проверяющего модуля: `answer()`

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

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

Описание

Вызовите эту функцию, когда определена комбинация положений переключателей, при которой все двери открыты, а также для каждого переключателя определена дверь, с которой он соединён.

Параметры

- `S`: массив длины `N`, содержащий правильное положение каждого переключателя. Формат данного массива такой же, как и у параметра вышеописанной функции `tryCombination()`.
- `D`: массив длины `N`, элементы которого для каждого переключателя задают, с какой дверью он соединён. Элемент `D[i]` должен содержать номер двери, с которой соединён переключатель с номером `i`.
- *Возвращаемое значение*: эта функция не возвращает управление вашей программе, то есть приводит к завершению вашей программы.

Ваша функция: `exploreCave()`

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

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

Описание

Ваше решение должно реализовывать эту функцию.

Эта функция должна использовать функцию проверяющего модуля `tryCombination()`, чтобы определить правильное положение для каждого переключателя, а также для каждого переключателя определить дверь, с которой он соединен. Когда ваша функция получит эту информацию, она должна вызвать функцию `answer()`.

Параметры

- `N`: количество переключателей и дверей в группе пещер.

Пример

Предположим, что двери и переключатели расположены, как показано на рисунке выше. Возможный порядок вызова функций представлен ниже.

Вызов функции	Возвращаемое значение	Объяснение
`tryCombination([1, 0, 1, 1])`	`1`	Ситуация соответствует показанной на рисунке. Переключатели 0, 2 и 3 находятся в положении "вниз", а переключатель 1 — в положении "вверх". Функция возвращает значение `1`, что значит, что дверь 1 — первая дверь слева, которая закрыта.
`tryCombination([0, 1, 1, 0])`	`3`	Двери 0, 1 и 2 открыты, а дверь 3 — закрыта.
`tryCombination([1, 1, 1, 0])`	`-1`	Перевод переключателя 0 в положение "вниз" приводит к тому, что все двери открылись, что обозначается возвращаемым значением `-1`.
`answer([1, 1, 1, 0], [3, 1, 0, 2])`	*(Программа завершается)*	Ваша программа определила, что комбинация `[1, 1, 1, 0]` задаёт правильное положение каждого из переключателей, и переключатели 0, 1, 2 и 3 соединены с дверями 3, 1, 0 и 2, соответственно.

Ограничения

- Ограничение по времени: 2 секунды.
- Ограничение по памяти: 32 МиБ.
- $1 \le N \le 5000$

Подзадачи

Подзадача	Баллы	Дополнительные ограничения на входные данные
1	12	Для каждого i, переключатель с номером i соединён с дверью с номером i. Ваша задача — определить правильное положение для каждого переключателя.
2	13	Правильное положение всех переключателей всегда `[0, 0, 0, …, 0]`. Ваша задача — определить, какой переключатель соединён с какой дверью.
3	21	$N \le 100$
4	30	$N \le 2000$
5	24	*(нет)*

Взаимодействие с проверяющим модулем

Проверяющий модуль на вашем компьютере будет считывать входные данные из файла `cave.in`, который должен иметь следующий формат:

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

Здесь `N` — количество дверей и переключателей, `S[i]` — правильное положение переключателя с номером `i`, а `D[i]` — дверь, с которой соединён переключатель с номером `i`.

В частности, вышеописанный пример должен быть задан в таком формате:

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

Особенности конкретных языков программирования

C/C++	Вы должны подключить заголовочный файл с помощью `#include "cave.h"`.
Pascal	Вы должны написать модуль с заголовком `unit Cave`, а также импортировать процедуры проверяющего модуля с помощью `uses GraderHelpLib`. Все массивы нумеруются, начиная с `0` (а не с `1`).

Для примера посмотрите шаблоны решений на вашем компьютере.