Огледална табла

Иванчо научи нова игра. Тя се играе върху две квадратни дъски долепени една над друга. Всяка от дъските предстаблява таблица разделена на N реда и N стълба. Върху всяка клетка от таблицата е поставен пул - бял или черен.

По време на играта всеки ход може да се вземат пул от клетка от първата дъска и пул от огледалната на нея клетка от втората дъска, и да се разменят. Огледалната клетка е тази, която се намира в същата колона като първата, но огледалния ред.
На първи ред от първата дъска е огледален последният ред от втората дъска, на втори ред от първата дъска е огледален предпоследният ред от втората дъска и т.н.

Целта на играта е всички черни пулове да се преместят върху една от дъските. Иванчо се старае да стане добър в играта, затова иска да постигне това с възможно най-малко ходове.

Помогнете му като напишете програма **chips**, която по зададено начално състояние на дъските определя дали е възможно да се преместят всички черни пулове върху една дъска и ако да намира минималния брой ходове с които може да стане това.

Вход: От първия ред на входния файл **chips.in** се въвежда едно число **N** - броят редове и колони във всяка дъска.
След това следват N реда, всеки от който се състои от N на брой нули и единици, които описват поредния ред от първата дъска. 0 означава бял чип, а 1 - черен.
След това следват N реда, отново съдържащи по N нули и единици. Те описват пуловете върху втората дъска.

Изход: В изходния файл **chips.out** трябва да се изведе едно единствено число - минималният брой ходове, необходими за да се преместят всички черни пулове в една дъска, или -1, ако това е невъзможно.

Ограничения:
2 <= N <= 30

Ограничение за време: 0.2 сек.
Ограничение за памет: 256 MiB.

Забележка: Четенето от файл и печатането във файл става като добавите оператор за това. Можете да използвате оператор *freopen* като включите библиотека *cstdio* и добавите следните два реда в началото на *main* функцията си:

```
freopen ("chips.in", "r", stdin);
freopen ("chips.out", "w" , stdout);
```

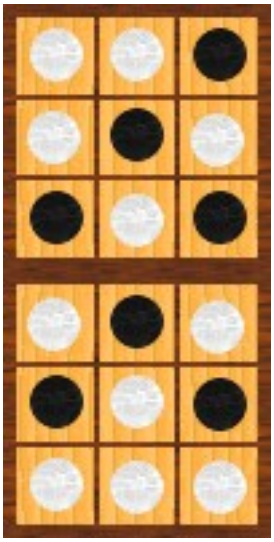
Брой предварителни тестове: 4
Брой финални тестове: 10

Примерен тест:

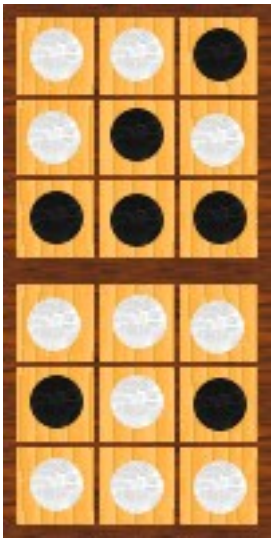
chips.in	chips.out
3 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 0	3

Обяснение на примера:
Едно примерно решение е следното:

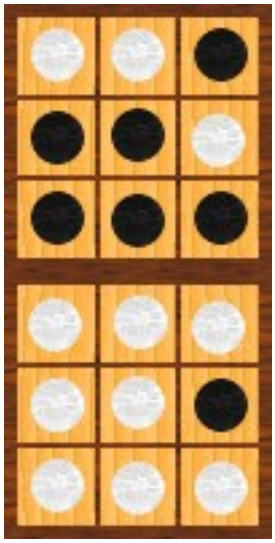
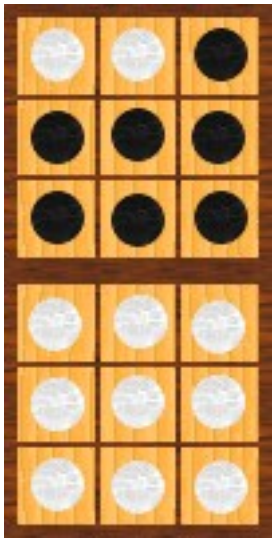
Първоначално



След ход 1



След ход 2



След ход 3