

Лора играет в онлайн-головоломку. У нее есть неориентированный граф с  $N$  вершинами, пронумерованными от 1 до  $N$ . Граф устроен таким образом, что **любые две различные вершины** соединены ребром, которое покрашено либо в синий, либо в красный цвет. Будем называть граф **красно-связным**, если от любой вершины можно добраться до любой другой, используя только красные ребра. Аналогично, граф **сине-связный**, если от любой вершины можно добраться до любой другой, используя только синие ребра. Будем называть **состоянием** графа пару чисел  $(A, B)$ , такую что:

- $A=1$ , если граф является красно-связным,  $A=0$ , если нет
- $B=1$ , если граф является сине-связным,  $B=0$ , если нет

Например, состояние  $(1, 0)$  задает граф, который является красно-связным, но не сине-связным.

За один клик по ребру Лора может изменить его цвет (с синего на красный, либо с красного на синий). Цель игры заключается в том, чтобы по заданному начальному графу и желаемому состоянию добиться того, чтобы граф находился в этом состоянии, за минимальное количество кликов.

Требуется написать программу, которая определяет, какое минимальное количество кликов потребуется Лоре, чтобы решить головоломку.

#### Input

Первая строка входных данных содержит положительное целое число  $N$  – количество вершин в графе. Далее следует  $N$  строк по  $N$  целых чисел, которые задают цвета ребер. Обозначим  $j$ -е число  $i$ -й строки как  $G_{ij}$ . Если  $G_{ij}=0$ , то ребро между вершинами  $i$  и  $j$  красное, если  $G_{ij}=1$ , то ребро между вершинами  $i$  и  $j$  синее. Гарантируется, что  $G_{ij}=G_{ji}$ . Для  $i=j$  значение  $G_{ij}$  не имеет значения, поскольку в графе нет петель. Последняя строка содержит два целых числа, разделенных пробелом –  $A$  и  $B$ , задающих требуемое состояние графа.

#### Output

Если преобразовать граф, чтобы он находился в требуемом состоянии невозможно, выведите **-1** на единственной строке вывода.

Иначе первая строка должна содержать целое число  $K$  – минимальное количество кликов, которое Лора должна сделать, чтобы преобразовать граф к требуемому состоянию. Каждая из следующих  $K$  строк должна содержать два целых числа – концы ребра, которое следует перекрасить очередным кликом. Если есть несколько оптимальных решений, можно вывести любое из них. Ребра можно выводить в любом порядке, аналогично концы ребер можно выводить в любом порядке.

#### Constraints

$3 \leq N \leq 250$

#### Подзадачи и оценка

Тесты разбиты на пары. Чтобы получить баллы за пару тестов программа должна корректно работать на обоих тестах пары.

Подзадача	% тестов	Дополнительные ограничения
1	15 %	$N \leq 7$
2	35 %	Требуемое состояние $(1, 1)$
3	50 %	Требуемое состояние <b>не</b> $(1, 1)$

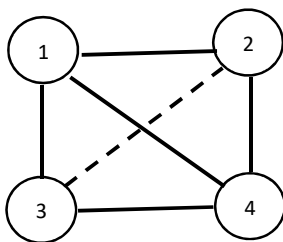
**Examples**

Input #1	Output #1	Input #2	Output #2	Input #3	Output #3
4	2	3	-1	3	0
1 0 0 0	1 3	0 1 1		0 1 1	
0 0 1 0	4 3	1 0 0		1 0 0	
0 1 1 0		1 0 0		1 0 0	
0 0 0 0		1 1		0 1	
0 1					

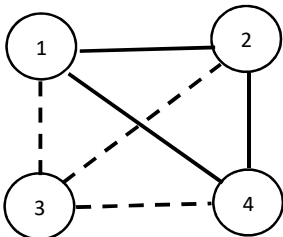
**Пояснение к примерам**

Красные ребра на рисунках изображены сплошными линиями, а синие – штрихованными.

В первом примере исходно граф находится в состоянии (1, 0):



Изменив цвет ребер 1-3 и 4-3, мы преобразуем граф в состояние (0, 1), он выглядит таким образом:



Во втором примере легко понять, что граф с 3 вершинами и состоянием (1, 1) не существует.

В третьем примере граф исходно уже находится в требуемом состоянии.