

Лора грає в онлайн гру. Вона отримує неорієнтований граф з  $N$  вершинами, пронумерованими від 1 до  $N$ . Граф такий, що між кожними двома різними вершинами є ребро, колір якого або синій, або червоний. Ми називатимемо граф червоно-зв'язним, якщо з будь-якої вершини ми можемо досягти будь-якої іншої вершини, рухаючись лише по червоним ребрам. Аналогічно, граф синьо-зв'язний, якщо з будь-якої вершини ми можемо досягти будь-яку іншу вершину, використовуючи лише сині ребра. Тепер ми визначаємо стан графа як пару чисел  $(A, B)$ , таких, що:

- $A=1$  якщо граф червоно-зв'язний,  $A=0$  інакше
- $B=1$  якщо граф синьо-зв'язний,  $B=0$  інакше

Так, наприклад, стан  $(1, 0)$  описує граф, який є червоно-зв'язним, але не є синьо-зв'язним.

За допомогою одного натискання на певному ребру, Лора може змінити його колір (від синього до червоного або від червоного до синього). Метою гри є досягнення певного стану. Потрібно змінити початковий графік на той, який потрібний, використовуючи мінімальну кількість натискань (дивіться приклади). Ваше завдання полягає в тому, щоб допомогти Лорі, написавши програму **colorgraph**, яка знаходить мінімальну кількість натискань, необхідних для вирішення задачі.

#### Вхідні дані

У першому рядку задано одне число  $N$  - кількість вершин графа.

У наступних  $N$  рядках задано по  $N$  чисел, які описують кольори ребер. Нехай  $G_{ij}$  -  $j$ -тнє число  $i$ -го рядка. Якщо  $G_{ij}=0$ , то ребро між  $i$  та  $j$  червоне, інакше  $G_{ij}=1$  та ребро синє. Гарантується, що  $G_{ij}=G_{ji}$ . Для  $i=j$ , значення  $G_{ij}$  неважливе, оскільки у графа немає петель.

У останньому рядку задано два числа  $A$  та  $B$ , що описують потрібний стан.

#### Вихідні дані

Якщо неможливо досягнути потрібний стан, ви повинні вивести **-1**.

У всіх інших випадках виведіть у першому рядку одне ціле число  $K$  - мінімальну кількість натискань, яку Лора має зробити, щоб перетворити граф у потрібний стан.

У наступних  $K$  рядках виведіть по два числа - номери вершин, по ребру між якими Лорі потрібно натиснути. Якщо є більше одного рішення, виведіть будь-яке з них. Порядок, в якому виведені ребро, а також порядок двох вершин, не мають значення.

#### Обмеження

$3 \leq N \leq 250$

#### Підзадачі на оцінювання

Тести розділені на пари, щоб отримати бали за пару, потрібно, щоб ваше рішення пройшло обидва тести.

Підзадача	% тестів	Додаткові обмеження
1	15 %	$N \leq 7$
2	35 %	Бажаний стан $(1, 1)$
3	50 %	Бажаний стан не $(1,1)$



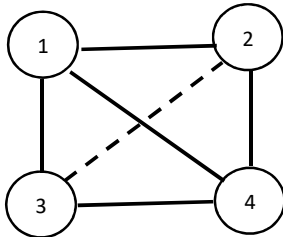
**Приклади**

Вхідні дані #1	Вихідні дані #1	Вхідні дані #2	Вихідні дані #2	Вхідні дані #3	Вихідні дані #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					

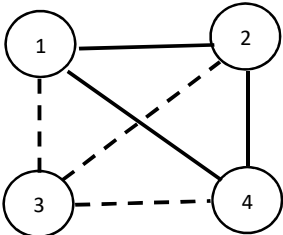
**Пояснення до прикладів**

Червоні ребра зображені відрізками, а сині пунктирами.

У першому прикладі ми маємо, що початковий граф має стан (1, 0):



Після змін ребер 1-3 та 4-3, граф має стан (0, 1) і виглядає так:



У другому прикладі граф зі станом (1, 1) не існує.

У третьому прикладі граф вже в потрібному стані.