

Лора играе логическа игра в интернет. Тя получава неориентиран граф с  $N$  върха, номерирани от 1 до  $N$ . Графът е такъв, че **между всеки два различни** върха има ребро, оцветено или в синьо, или в червено. Ще казваме, че графът е червено-свързан, ако можем от всеки връх да достигнем до всеки друг, различен от него, движейки се само по червени ребра. Аналогично, графът е синьо-свързан, ако можем от всеки връх да достигнем всеки друг, различен от него, движейки се само по сини ребра. Дефинираме *състояние* на графа като двойка числа  $(A, B)$ , като:

- $A=1$  ако графът е червено-свързан и  $A=0$  ако не е;
- $B=1$  ако графът е синьо-свързан и  $B=0$  ако не е

Така например, състояние  $(1, 0)$  описва граф, който е червено-свързан, но не е синьо-свързан.

С едно кликване върху дадено ребро Лора може да промени цвета му (от син в червен или от червен в син). Целта на играта е по зададен начален граф и желано състояние, с минимален брой кликания, началният граф да се промени до такъв, който е в желаното състояние (вижте примера за повече информация). Помогнете на Лора, като напишете програма **colorgraph**, която намира минималния брой необходими кликания.

#### Вход

От първия ред на стандартния вход се въвежда  $N$  – броят върхове в графа. Следват  $N$  реда с по  $N$  числа, разделени с по един интервал, описващи цветовете на ребрата. Нека с  $G_{ij}$  бележим  $j$ -тото число на  $i$ -тия от тези редове. Ако  $G_{ij}=0$ , то реброто между  $i$  и  $j$  е червено, а ако  $G_{ij}=1$ , то реброто между  $i$  и  $j$  е синьо. Гарантирано е, че  $G_{ij}=G_{ji}$ . Ако  $i=j$ , стойността на  $G_{ij}$  няма значение, тъй като в графа няма примки. На последния ред се въвеждат две числа –  $A$  и  $B$ , разделени с интервал, описващи желаното крайно състояние  $(A, B)$ .

#### Изход

Ако не е възможно да получим граф в желаното състояние, изведете **-1** на единствен ред на стандартния изход. В противен случай, на първия ред на стандартния изход изведете едно цяло число  $K$  - минималния брой кликания, които са нужни на Лора за да трансформира графа в желаното състояние. На всеки от следващите  $K$  реда изведете двойка числа – двата края на ребро, върху което Лора трябва да кликне. Ако има повече от едно решение, изведете което и да е от тях. Редът, в който се извеждат ребрата, и редът, в който се извеждат двата края на дадено ребро, са без значение.

#### Ограничения

$$3 \leq N \leq 250$$

#### Подзадачи и оценяване

Тестовите са групирани по двойки. Точките за двойка тестове се получават само, ако и двата теста преминат успешно.

Подзадача	% тестове	Допълнителни ограничения
1	15 %	$N \leq 7$
2	35 %	Желаното състояние е $(1, 1)$
3	50 %	Желаното състояние не е $(1, 1)$

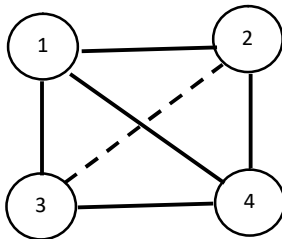
**Примери**

Вход #1	Изход #1	Вход #2	Изход #2	Вход #3	Изход #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					

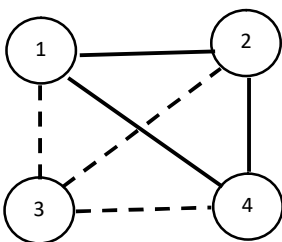
**Обяснение на примерите**

Червените ребра са дадени с плътни линии, а сините – с пунктирани.

В първия пример имаме следния начален граф, който е в състояние (1,0):



След промяната на ребра 1-3 и 4-3, графът е в желаното състояние (0, 1) и изглежда по следния начин:



Във втория пример не съществува граф с 3 върха и състояние (1, 1).

В третия пример графът е в състояние (0,1) още в началото.