Lora gra online w puzzle. Dziewczynka otrzymała nieskierowany graf, który ma **N** wierzchołków ponumerowanych od 1 do N. **Pomiędzy każdą parą różnych wierzchołków** istnieje krawędź, która jest albo czerwona albo niebieska. Mówimy, że graf jest czerwono-spójny, jeśli z każdego wierzchołka można osiągnąć wszystkie inne wierzchołki używając tylko czerwonych krawędzi. Podobnie, mówimy, że graf jest niebiesko-spójny, jeśli z każdego wierzchołka możemy osiągnąć wszystkie inne wierzchołki, używając tylko niebieskich krawędzi. Niech *stan* grafu będzie taką parą liczb **(A,B)**, że:

- ***A***=1, jeśli graf jest czerwono-spójny i **A**=0, jeśli nie jest czerwono-spójny

- ***B***=1, jeśli graf jest niebiesko-spójny i **B**=0, jeśli nie jest niebiesko-spójny

Przykładowo, stan (1, 0) opisuje graf, który jest czerwono-spójny i nie jest niebiesko-spójny.

Lora może kliknąć w dowolną krawędź i zmienić jej kolor na przeciwny (z niebieskiego na czerwony lub z czerwonego na niebieski). Celem gry jest wykonanie minimalnej liczby ruchów na dostarczonym grafie tak, aby graf znalazł się w stanie docelowym, który został podany na wejściu (przeanalizowanie przykładów powinno wszystko wyjaśnić). Twoim zadaniem jest napisanie programu **colorgraph**, który obliczy minimalną liczbę kliknięć potrzebną do rozwiązania problemu.

Wejście

W pierwszym wierszu standarowego wejścia zapisano jedną liczbę naturalną **N** – liczbę wierzchołków w grafie. W każdy z kolejnych **N** wierszy zapisano po **N** liczb, oddzielonych spacjami, opisujących kolory krawędzi. Niech G_{ij} oznacza j-tą liczbę w i-tym wierszu. Jeśli G_{ij}.=0, wtedy krawędź pomiędzy krawędziami *i* i *j* jest czerwona. Jeśli zaś G_{ij}=1, wtedy krawędź pomiędzy wierzchołkami *i* i *j* jest niebieska. Mamy zagwarantowane, że $G_{ij}=G_{ji}$. Dla *i=j*, wartość G_{ij} jest nieistotna, ponieważ grał nie zawiera pętli. W ostatnim wierszu zapisano dwie liczby **A** i **B**, oddzielone spacją, opisujące stan docelowy.

Wyjście

W pierwszym wierszu standardowego wyjścia należy wypisać **-1**, jeśli nie jest możliwe przekształcenie grafu do stanu docelowego. W przeciwnym przypadku należy wypisać jedną liczbę całkowitą **K** – minimalną liczbę kliknięć potrzebną do otrzymania grafu w stanie docelowym. W każdym z kolejnych **K** wierszy należy wypisać po dwie liczby całkowite – końce krawędzi, które Lora powinna kliknąć. Jeśli istnieje wiele poprawnych odpowiedzi, Twój program powinien wypisać dowolną z nich. Kolejność krawędzi nie ma znaczenia.

Ograniczenia

3≤***N***≤250

Podzadania i ocenianie

Testy są połączone w pary. Aby otrzymać punkty za parę testów, program musi dać dobry wynik w obu przypadkach (normalna sprawa).

Podzadanie	% testów	Ograniczenia
1	15 %	N≤7
2	35 %	Docelowy stan to (1, 1)
3	50 %	Docelowym stanem nie jest (1,1)

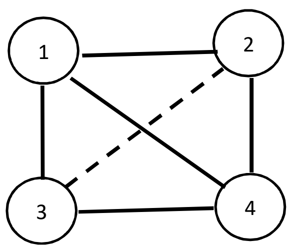
Przykłady

Wejście #1	Wyjście #1	Wejście #2	Wyjście #2	Wejście #3	Wyjście #3
4 1 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 1	2 1 3 4 3	3 0 1 1 1 0 0 1 0 0 1 1	-1	3 0 1 1 1 0 0 1 0 0 0 1	0

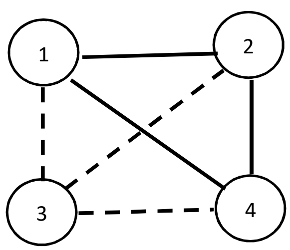
Wyjaśnienie do przykładów

Czerwone krawędzie narysowano linią ciągłą, zaś niebieskie linią przerywaną.

W pierwszym przykładzie mamy następujący graf w stanie (1, 0):



Po zmianie krawędzi 1-3 i 4-3, otrzymujemy graf w stanie (0, 1), który wygląda następująco:



W drugim teście przykładowym graf zawiera 3 wierzchołki i doprowadzenie go do stanu (1, 1) nie jest możliwe.

W trzecim teście przykładowym, graf podany na wejściu jest już w stanie docelowym.