Задача A. Кракозябрики

Имя входного файла:	`bfs.in`
Имя выходного файла:	`bfs.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	16 мегабайт

... мораль сей басни такова — у нас хорошая трава.

Парад параллелей ЛКШ.Зима

Кракозябрик Пушистик живёт в столице Кракозябрии — Финденляндии, а его лучший друг детства — в деревне Менделеевка. Кракозябрики очень дружнелюбные существа, поэтому в большие праздники у них принято поздравлять старых-добрых друзей лично.

Наступает День Рождения Короля Кракозябрии — Мирикундина Первого, праздник будет отмечаться по всей стране, а Пушистик, возможно, даже не встретится со своим другом. Всё дело в том, что Пушистик — известный музыкант, он играет на волшебных доремифасольчиках, и его график гастролей расписан буквально по часам. Но он никогда себе не простит, если он не встретится с другом детства в такой день!

Пушистик внимательно изучил, какие самолёты летают в Кракозябрии и теперь на основании этих данных хочет узнать, какое минимальное количество раз ему придется взлетать на самолёте, чтобы в итоге увидеться с лучшим другом детства. Пожалуйста, помогите ему в этом.

Формат входного файла

В первой строке входного файла содержится три натуральных числа N, S и F ($1 \leqslant S, F \leqslant N \leqslant 100$) — количество городов в Кракозябрии и номера, которыми обозначены Финденляндия и Менделеевка. Далее в N строках заданы все самолётные рейсы страны. Если значение в j-м элементе i-й строки равно 1, то в Пушистик может напрямую лететь из города номер i в город номер j.

Формат выходного файла

Выведите одно целое число — минимальное количество самолётов при полёте из Финденляндии в Менделеевка. Если пути не существует и старым добрым друзьям так и не суждено увидеться, выведите 0.

Пример

bfs.in	bfs.out
4 4 3 0 1 1 1 1 0 1 0 1 1 0 0 1 0 0 0	2

Задача B. Максимум по минимуму

Имя входного файла:	`maxmin.in`
Имя выходного файла:	`maxmin.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Дан ориентированный невзвешенный граф. В нём необходимо найти вершину, кратчайшее расстояние от которой до заданной максимально.

Формат входного файла

В первой строке входного файла содержится три натуральных числа N, M и S ($1 \leqslant S \leqslant N \leqslant 1000$, $1 \leqslant M \leqslant 10000$) — количество вершин и рёбер в графе и номер заданной вершины соответственно. Далее в M строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

Формат выходного файла

Вывести одно целое число — искомое кратчайшее расстояние.

Пример

maxmin.in	maxmin.out
3 5 3 1 2 2 1 3 1 2 3 3 3	2

Задача C. Снова про коней

Имя входного файла:	knight3.in
Имя выходного файла:	knight3.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

На шахматной доске 8×8 указаны две несовпадающие клетки. Найдите кратчайший маршрут коня из первой клетки во вторую.

Формат входного файла

Во входном файле записаны координаты двух клеток. Каждая координата представлена двумя символами, где сначала указана одна строчная буква от `a` до `h`, а после буквы (без пробела) цифра от `1` до `8`, например `h8`. Каждая клетка записана в отдельной строке.

Формат выходного файла

Программа должна вывести последовательность клеток, первая из которых совпадает с первой данной, а последняя совпадает со второй данной. Две соседние клетки должны быть соединены ходом коня, при этом количество клеток в последовательности должно быть минимально возможным.

Пример

knight3.in	knight3.out
a1 b1	a1 b3 d2 b1

Задача D. TopSort

Имя входного файла:	topsort.in
Имя выходного файла:	topsort.out
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

Формат входного файла

В первой строке входного файла два натуральных числа N и M ($1 \leqslant N \leqslant 10^5, 1 \leqslant M \leqslant 10^5$) — количество вершин и рёбер в графе соответственно. Далее в M строках перечислены рёбра графа. Каждое ребро задается парой чисел — номерами начальной и конечной вершин соответственно.

Формат выходного файла

Вывести любую топологическую сортировку графа в виде последовательности номеров вершин. Если граф невозможно топологически отсортировать, вывести -1.

Пример

topsort.in	topsort.out
6 6 1 2 3 2 4 2 2 5 6 5 4 6	4 6 3 1 2 5
3 3 1 2 2 3 3 1	-1