

## Задача А. Кантонский язык

Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Поразмыслив над тем, где провести свой очередной отпуск, Вова решил совершить путешествие по Южному Китаю, посетив Гуанчжоу, Шэньчжэнь, Макао и Гонконг.

Вова слышал, что на юге Китая говорят на кантонском диалекте китайского языка, поэтому перед поездкой выучил несколько простых фраз на кантонском. Во время своей первой прогулки по центру Гуанчжоу Вова поздоровался с  $n$  прохожими, и  $m$  из них ответили на его приветствие. Вова сделал вывод, что остальные  $(n - m)$  прохожих, очевидно, говорят на мандаринском диалекте китайского языка.

После этой прогулки Вова решил оценить число  $M$  жителей Гуанчжоу, говорящих на кантонском диалекте. Википедия подсказала ему, что население города составляет  $N$  человек. Помогите Вове найти такое  $M$ , при котором вероятность того, что среди  $n$  случайных прохожих ровно  $m$  говорят на кантонском, максимальна.

### Формат входных данных

В единственной строке записаны целые числа  $n$ ,  $m$  и  $N$  ( $1 \leq n \leq N \leq 10^8$ ;  $0 \leq m \leq n$ ). Каждый из  $n$  прохожих был жителем Гуанчжоу и встретился с Вовой ровно один раз за прогулку.

### Формат выходных данных

Выведите искомое  $M$ . Если при нескольких значениях  $M$  достигается максимальное значение вероятности, выведите наибольшее из них.

### Пример

<i>Входные данные</i>	<i>Выходные данные</i>
10 1 200	20

## Задача В. В китайском ресторане

Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Когда Вова приехал в Гуанчжоу, его китайские друзья первым делом пригласили его в ресторан. Всего в ресторан пришло  $n$  человек, включая Вову. Официант предложил разместиться всей компании за традиционным большим круглым столом с вращающейся подставкой для блюд в центре.

Вову, как гостя, сразу усадили на почётное место около двери. После чего  $m$  человек из присутствующих заявили, что непременно хотят сидеть рядом с определённым человеком. Требуется определить количество допустимых рассадок друзей Вовы за столом.

### Формат входных данных

В первой строке записаны целые числа  $n$  и  $m$  ( $2 \leq n \leq 100$ ;  $0 \leq m \leq n$ ). В следующих  $m$  строках записаны целые числа  $k_1, \dots, k_m$ , где  $k_i$  — номер человека, рядом с которым изъявил желание сидеть человек с номером  $i$  ( $1 \leq k_i \leq n$ ;  $k_i \neq i$ ). Вова, как почётный гость, имеет номер 1. А его друзья пронумерованы целыми числами от 2 до  $n$ .

### Формат выходных данных

Выведите количество допустимых вариантов рассадки, взятое по модулю  $10^9 + 7$ .

### Примеры

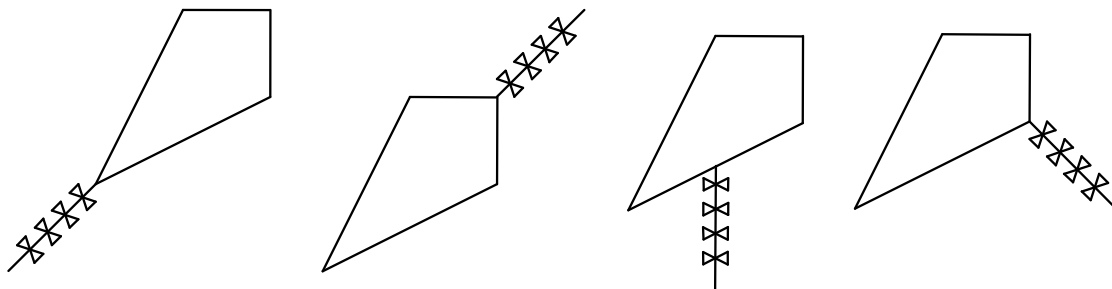
<i>Входные данные</i>	<i>Выходные данные</i>
6 6 2 1 1 5 6 5	4
4 3 2 3 1	0

## Задача С. Воздушный змей

Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Вова купил себе на рынке в Гуанчжоу набор для изготовления воздушного змея. На следующий день выдалась хорошая погода, и он решил склеить змея и запустить его. Инструкция по изготовлению, конечно, была только на китайском языке, поэтому Вова решил, что разберётся и без неё. Немного повозившись, он соорудил змея в форме плоского четырёхугольника, оставалось только приклеить к нему хвост.

И тут Вова пришлось задуматься над тем, к какой точке границы четырёхугольника должен быть приклеен хвост змея. Интуиция подсказала ему, что для того, чтобы полёт змея был устойчивым, его хвост должен лежать на некоторой оси симметрии четырёхугольника. На двух рисунках слева изображены воздушные змеи, летающие устойчиво, а на двух рисунках справа — змеи, летающие неустойчиво.



Сколько существует точек на границе четырёхугольника, таких что, если приклеить к ним хвост, в результате получится воздушный змей, летающий устойчиво?

### Формат входных данных

В четырёх строках записаны координаты вершин четырёхугольника в порядке обхода. Все координаты — целые числа, по модулю не превышающие 1 000. Никакие три последовательные вершины четырёхугольника не лежат на одной прямой. Противоположные стороны четырёхугольника не пересекаются.

### Формат выходных данных

Выведите количество точек на границе четырёхугольника, к которым можно приклеить хвост змея.

### Примеры

<i>Входные данные</i>	<i>Выходные данные</i>
0 0 1 2 2 2 2 1	2
0 0 2 1 2 2 0 2	0

Осью симметрии плоской фигуры называется прямая, лежащая в плоскости фигуры и такая, что при отражении относительно неё фигура переходит сама в себя.

## Задача D. Китайский язык

Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Вова переехал из Гуанчжоу в Шэньчжэнь. Тут же выяснилось, что местные жители также не понимают его фразы на кантонском. Вова попытался общаться с ними на мандаринском диалекте, но и тут его постигла неудача.

Тогда Вова решил узнать больше о диалектах китайского языка. Оказалось, что в Китае говорят на  $k$  различных диалектах, причём не менее  $a_1$  китайцев говорят на первом диалекте, не менее  $a_2$  китайцев — на втором диалекте, ..., не менее  $a_k$  китайцев — на  $k$ -м диалекте. Сколько минимум китайцев говорят на всех  $k$  диалектах, если население Китая составляет  $n$  человек?

### Формат входных данных

В первой строке записаны целые числа  $n$  и  $k$  ( $2 \leq k \leq 20$ ;  $1 \leq n \leq 10^9$ ). Во второй строке через пробел записаны целые числа  $a_1, \dots, a_k$  ( $1 \leq a_k \leq n$ ).

### Формат выходных данных

Выведите минимальное количество жителей Китая, говорящих на всех  $k$  диалектах китайского языка.

### Примеры

<i>Входные данные</i>	<i>Выходные данные</i>
1000000000 2 800000000 800000000	600000000
1000000000 2 500000000 500000000	0

## Задача Е. Саженцы груши

Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Когда Вова шёл по одной из центральных улиц Шэньчжэня, он обратил внимание на саженцы грушевых деревьев, растущие вдоль тротуара. К каждому саженцу была прикреплена табличка, на которой было написано некоторое число. Обойдя все  $n$  саженцев, Вова обнаружил, что числа на табличках попарно различны и лежат в пределах от 1 до  $n$ . Очевидно, сначала предполагалось посадить саженцы в указанном порядке, но сейчас они были перемешаны странным образом: после шестого шёл четвёртый, затем третий, затем пятый...

Рядом стояла тележка с мороженым. Мороженщик, заметив, что Вова с недоумением разглядывает саженцы груши, рассказал ему, что видел, как неделю назад их высаживали в землю. Бригадир поручил это дело двум рабочим, сказав, чтобы те посадили саженцы по порядку номеров, указанных на табличках. После этого бригадир уехал, а рабочие поделили между собой саженцы и приступили к работе. Поскольку бригадир не уточнил, по возрастанию или убыванию должны следовать номера на табличках, каждый рабочий принял на этот счёт самостоятельное решение, не посоветовавшись с напарником. Оба рабочих, сажая деревья, следовали в одном направлении с одного и того же конца улицы.

Рассмотрим пример. Пусть  $n = 8$ , и рабочие поделили саженцы так, что первому достались саженцы с номерами 1, 4 и 5, а второму — 2, 3, 6, 7 и 8. В результате получилась такая последовательность номеров на табличках: 8, 7, 1, 6, 4, 3, 5, 2 (первый сажал деревья по возрастанию номеров, а второй — по убыванию).

Вова записал все номера на табличках в том порядке, в котором деревья были посажены, и захотел определить, какие саженцы были высажены первым рабочим, а какие — вторым. Помогите ему сделать это.

### Формат входных данных

В первой строке записано целое число  $n$  — количество саженцев ( $3 \leq n \leq 100\,000$ ). Во второй строке записаны  $n$  различных целых чисел в пределах от 1 до  $n$  — перестановка номеров, записанная Вовой.

### Формат выходных данных

Выведите в первой строке два целых числа в пределах от 1 до  $(n - 1)$  — сколько деревьев посадил первый и второй рабочий соответственно. Во второй строке выведите номера деревьев, посаженных первым рабочим, в третьей — номера деревьев, посаженных вторым рабочим. Номера должны следовать в том порядке, в котором рабочий сажал эти деревья. Если существует несколько решений, можно вывести любое из них. Если решения не существует, выведите «Fail».

### Примеры

<i>Входные данные</i>	<i>Выходные данные</i>
8 8 7 1 6 4 3 5 2	3 5 1 4 5 8 7 6 3 2
6 3 5 1 2 6 4	3 3 3 5 6 1 2 4
6 3 5 2 1 6 4	Fail

## Задача F. Велодорожки

Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

В Шэньчжэне Вова взял напрокат велосипед и большую часть времени ездил по городу на нём. Подъезжая к одному из городских парков, Вова обратил внимание на то, что на плане парка, висящем около центрального входа, отмечены несколько каменных статуй, находящихся в этом парке. Одна из таких статуй стояла тут же, прямо у входа в парк. Вова захотел заехать в парк на велосипеде и сфотографировать все статуи. На территории парка есть несколько двусторонних велодорожек. Каждая велодорожка начинается и заканчивается у одной из каменных статуй и представляет собой отрезок на плоскости. Если две велодорожки имеют общую точку, то в этой точке можно свернуть с одной из них на другую. Если статуя находится непосредственно на велодорожке, то она не мешает передвижению по ней и может быть с неё сфотографирована.

Сможет ли Вова добраться на велосипеде до всех статуй в парке, не съезжая с велодорожек?

### Формат входных данных

В первой строке записаны целые числа  $n$  и  $m$  — количество статуй и велодорожек в парке ( $1 \leq m < n \leq 200$ ). Далее идут  $n$  строк, каждая из которых содержит координаты одной статуи на плане парка. Координаты — целые числа, по модулю не превосходящие 30 000. Координаты любых двух статуй различны. Каждая из следующих  $m$  строк содержит два различных целых числа в пределах от 1 до  $n$  — номера статуй, между которыми проложена велодорожка.

### Формат выходных данных

Выведите «YES», если Вова сможет доехать по велодорожкам от входа в парк до всех статуй, находящихся в нём, и «NO» в противном случае.

### Примеры

<i>Входные данные</i>	<i>Выходные данные</i>
4 2 0 0 1 0 1 1 0 1 1 3 4 2	YES
4 3 0 0 1 0 1 1 0 1 1 2 2 1 3 4	NO
3 2 0 0 1 0 1 1 1 3 3 2	YES

## Задача G. Казино для программистов

Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Город Макао называют «Монте-Карло Востока». В Макао работают более тридцати казино, приманивающих ежегодно миллионы игроков как из Китая, так и из других стран. Индустрия азартных игр формирует около 40% ВВП города.

От своих друзей из Гуанчжоу Вова узнал, что в Макао есть специальные казино, в которых играют только программисты. Это показалось Вова очень интересным, и он решил непременно сходить в такое казино. Одно из подобных заведений было расположено неподалёку от отеля, где остановился Вова, поэтому вечером он, пополнив запас наличных денег, пошёл в казино.

За игровыми столами в казино, по-видимому, и вправду сидели программисты, потому что ни карт, ни фишек на столах не было, но перед каждым игроком лежал блокнот, в который тот записывал какие-то числа. Как пояснили Вова, смысл игры заключался в следующем. Сначала каждый игрок записывал на чистом листе блокнота одну единицу. После этого крупье начинал зачитывать последовательность из нулей и единиц, называя по одной цифре. Если крупье называл ноль, все игроки дописывали ноль в конец последнего числа, выписанного на их листе. Если же крупье называл единицу, то игрок мог либо дописать эту единицу в конец последнего числа, либо записать её в новой строке, начав тем самым с неё новое число. После того как крупье прекращал зачитывать цифры, каждый игрок переводил все записанные им двоичные числа в десятичный вид, затем склеивал их все в одно десятичное число в том порядке, в котором они были записаны. Выигрывал тот игрок, который получал в результате наименьшее число.

Например, если крупье зачитал последовательность 1011100101, первый игрок записал двоичные числа 1, 1011 и 100101 (десятичные 1, 11 и 37), а второй игрок записал 110, 11100 и 101 (десятичные 6, 28 и 5), то победу одержал второй игрок, поскольку 6285 меньше 11137.

Перед тем как сесть за игровой стол, Вова решил научиться оптимально играть для простого случая, когда последовательность крупье представляет собой запись в двоичном виде всех целых чисел подряд от 2 до  $n$  (в разобранный выше примере  $n$  равнялось пяти). Но быстро придумать оптимальную стратегию Вова не смог, поэтому он обратился за помощью к вам.

### Формат входных данных

В единственной строке записано целое число  $n$  ( $2 \leq n \leq 100\,000$ ).

### Формат выходных данных

В первой строке выведите количество чисел, которые должен записать Вова в случае его оптимальной игры. Во второй строке выведите длины этих чисел в битах в том порядке, в котором числа нужно записать. Если есть несколько решений, дающих в результате наименьшее десятичное число, выведите любое из них.

### Пример

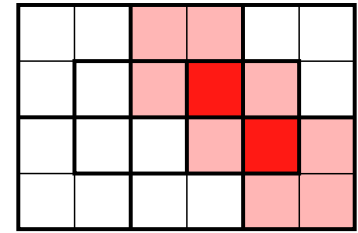
<i>Входные данные</i>	<i>Выходные данные</i>
5	2 1 10

В примере нужно записать все цифры, зачитанные крупье, во второе число. В этом случае после перевода чисел в десятичную систему и их склейки получится число 1741.

## Задача Н. Стеклянная пирамида

Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Вова — фанат небоскрёбов. Когда он приезжает в город, в котором есть небоскрёбы, он всегда старается попасть на смотровую площадку, расположенную на самом высоком из них, чтобы сделать оттуда несколько фотографий. Этот раз не стал исключением. Как только Вова добрался до Гонконга, сразу после заселения в отель он поехал в Международный коммерческий центр — 118-этажное здание, строительство которого завершилось пару лет назад.



Перед главным входом в Международный коммерческий центр расположена пирамида из больших стеклянных блоков. Каждый блок имеет форму прямоугольного параллелепипеда с высотой один метр и квадратным основанием  $2 \times 2$  метра. Нижний слой пирамиды имеет размеры  $m \times n$  метров. Второй слой имеет размеры  $(m - 2) \times (n - 2)$  метров и расположен так, что вершины нижней грани каждого блока в нём совпадают с центрами верхних граней блоков предыдущего слоя. Аналогично построен третий и последующие слои. Верхний слой состоит из одного ряда блоков (в частности, при  $m = n$  он состоит из одного блока). Часть блоков, из которых сложена пирамида, изготовлены из бесцветного стекла, а часть — из стекла, имеющего красноватый оттенок.

Когда Вова забрался на смотровую площадку, расположенную на сотом этаже небоскрёба, пирамида показалась ему оттуда прямоугольным полем, разбитым на ячейки  $1 \times 1$  метр. По оттенку красного каждой ячейки можно было определить, сколько блоков из красного стекла лежало над этой ячейкой. Вова сфотографировал эту пирамиду сверху и хочет, чтобы по этой фотографии вы определили, какие блоки пирамиды сделаны из красного стекла, а какие — из бесцветного. Вова также добавил, что ни один блок из красного стекла не лежит точно над блоком из бесцветного стекла.

### Формат входных данных

Первая строка содержит целые числа  $m$  и  $n$  — размеры основания пирамиды ( $2 \leq m, n \leq 40$ ;  $m$  и  $n$  чётные). Далее следуют  $m$  строк по  $n$  неотрицательных чисел в каждой, описывающих фотографию, которую сделал Вова. Каждое число показывает количество блоков из красного стекла, лежащих над соответствующей ячейкой. Гарантируется, что входные данные задают некоторое возможное расположение блоков в пирамиде.

### Формат выходных данных

Выведите описание слоёв — от самого нижнего до самого верхнего. Для вывода информации об  $i$ -м слое используется  $k_i$  строк по  $l_i$  символов в каждой ( $k_1 = m/2$ ;  $l_1 = n/2$ ;  $k_i = k_{i-1} - 1$ ;  $l_i = l_{i-1} - 1$ ). Символ «R» обозначает блок из красного стекла, «W» — блок из бесцветного стекла. Каждый последующий слой отделяется от предыдущего пустой строкой. Если существует несколько решений, можно вывести любое из них.

### Пример

Входные данные	Выходные данные
4 6	WRW
0 0 1 1 0 0	WWR
0 0 1 2 1 0	WR
0 0 0 1 2 1	
0 0 0 0 1 1	



## Задача I. Гонконгский трамвай

Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Трамвайная линия — одна из достопримечательностей Гонконга. Она обслуживается двухэтажными вагонами 1940-х–50-х годов без стёкол и с деревянными сидениями. Трамваи ходят строго по расписанию, так что жители домов, расположенных вдоль трамвайной линии, могут сверять свои часы по грохоту трамваев, проходящих мимо их дома.

Мимо отеля, в котором остановился Вова, ходят трамваи по маршруту Саукэйвань–Сёнвань. Вчера Вова купил на рынке мешок редких китайских монет, поэтому весь сегодняшний день он провёл в гостиничном номере, сортируя эти монеты по годам и достоинству. В течение дня по улице несколько раз проезжали трамваи. По шуму трамвая можно точно определить момент времени, в который тот проезжает мимо отеля, но нельзя определить, в какую сторону он в этот момент едет. Вова записал в блокнот все моменты прохождения трамваев мимо его отеля и хочет по этим данным составить расписание их движения. Помогите ему в этом.

Скорость движения трамвая постоянна и одинакова для всех трамваев. Временем остановок трамвая и его стоянок на конечных станциях следует пренебречь. Все трамваи за день совершают одинаковое количество рейсов, одним рейсом считается маршрут Саукэйвань–Сёнвань–Саукэйвань. Первый трамвай уезжает со станции Саукэйвань не раньше 00:00:00, последний трамвай возвращается туда не позже 23:59:59. Последний трамвай уезжает со станции Саукэйвань до того, как первый трамвай выполнит свой первый рейс. От Вовинога отеля до каждой из двух конечных станций трамвай едет целое положительное число секунд. Никакие два трамвая не проезжали мимо отеля в один и тот же момент времени.

### Формат входных данных

В первой строке записано время выхода первого трамвая со станции Саукэйвань. В следующих  $n$  строках записаны времена прохождения трамваев мимо Вовинога отеля ( $2 \leq n \leq 5\,000$ ;  $n$  чётно). Все времена приведены в формате «чч:мм:сс» и лежат в пределах от 00:00:00 до 23:59:58. Времена прохождения трамваев различны и упорядочены по возрастанию. Время выхода первого трамвая со станции Саукэйвань строго меньше времени его первого прохождения мимо отеля.

### Формат выходных данных

Для каждого трамвая нужно вывести одну строку, содержащую времена прохождения конечных станций этим трамваем в порядке возрастания в формате «чч:мм:сс». Первой должна идти строка с данными для трамвая, вышедшего на линию первым, второй — для трамвая, вышедшего вторым, и т.д. Если возможных вариантов несколько, то можно вывести любой из них. Гарантируется, что хотя бы одно решение существует.

## Пример

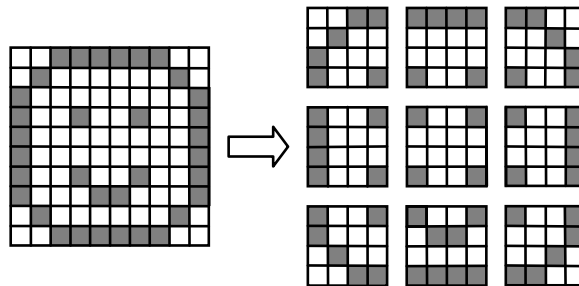
<i>Входные данные</i>				
08:00:00				
08:07:30				
08:38:30				
08:44:49				
08:51:32				
08:53:30				
09:15:49				
09:22:32				
09:24:30				
09:30:49				
09:37:32				
10:01:49				
10:08:32				
<i>Выходные данные</i>				
08:00:00	08:23:00	08:46:00	09:09:00	09:32:00
08:37:19	09:00:19	09:23:19	09:46:19	10:09:19
08:44:02	09:07:02	09:30:02	09:53:02	10:16:02

## Задача J. 皇后像廣場

Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Гуляя по Statue Square (皇后像廣場) в Гонконге, Вова обратил внимание на то, что квадратные плиты, которыми вымощена площадь, имеют разный цвет. Поизучав эти плиты, Вова понял, что они образуют некоторый рисунок. Плиты были большие, и охватить их взглядом все сразу не выходило. Вероятно, рисунок из плит должен был наблюдаться с высоты птичьего полёта.

Вова решил сфотографировать часть рисунка размером  $10 \times 10$  плит, но и она никак не влезала в один кадр. Тогда Вова сделал девять кадров, каждый из которых охватывал участок рисунка размером  $4 \times 4$  (см. иллюстрацию ниже). Если правильно разложить девять фотографий, по ним можно было восстановить исходный узор размера  $10 \times 10$ .



К сожалению, вскоре после того, как сделал фотографии, Вова забыл не только порядок, в котором их нужно разложить, но и то, как правильно нужно повернуть каждую из них. Изображение на фотографии может быть повернуто на произвольный угол, кратный 90 градусам. Помогите Вова восстановить по девяти фотографиям исходный узор размера  $10 \times 10$ .

### Формат входных данных

Входные данные содержат девять фотографий, которые сделал Вова. Фотографии задаются матрицами размера  $4 \times 4$  из целых чисел в пределах от 0 до 99, обозначающих цвета соответствующих плит. Числа в строке разделены пробелами. Каждая матрица отделена от следующей пустой строкой.

### Формат выходных данных

Выведите исходный узор в виде матрицы размера  $10 \times 10$ . Элементы матрицы в строке следует разделять пробелами. Если существует несколько решений, можно вывести любое из них. Гарантируется, что хотя бы одно решение существует.

**Пример**

<i>Входные данные</i>	<i>Выходные данные</i>
1 1 9 9	1 1 9 9 9 9 9 9 1 1
1 9 1 1	1 9 1 1 1 1 1 1 9 1
9 1 1 1	9 1 1 1 1 1 1 1 1 9
9 1 1 9	9 1 1 9 1 1 9 1 1 9
9 1 1 9	9 1 1 1 1 1 1 1 1 9
9 1 1 1	9 1 1 1 1 1 1 1 1 9
9 1 1 1	9 1 1 9 1 1 9 1 1 9
9 1 1 9	9 1 1 1 9 9 1 1 1 9
9 1 1 9	1 9 1 1 1 1 1 1 9 1
9 1 1 1	1 1 9 9 9 9 9 9 1 1
1 9 1 1	
1 1 9 9	
9 1 1 9	
1 1 1 1	
1 1 1 1	
9 1 1 9	
9 1 1 9	
1 1 1 9	
1 1 9 1	
9 9 1 1	
9 1 1 9	
1 1 1 9	
1 1 1 9	
9 1 1 9	
9 9 1 1	
1 1 9 1	
1 1 1 9	
9 1 1 9	
9 9 9 9	
1 1 1 1	
1 1 1 1	
9 1 1 9	
9 1 1 9	
1 9 9 1	
1 1 1 1	
9 9 9 9	