Задача A. Игровой автомат

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Создатели игровых автоматов выпустили на рынок математическую игру. Игра состоит в следующем: в каждом автомате создателями задана уникальная комбинация из $M+1$ числа в P-ичной системе счисления. Каждое число состоит из N разрядов. Первые M чисел во время игры не меняют своего значения, обозначим их a_1, a_2, …, a_M. Во время игры автомат несколько раз случайным образом выбирает из первых M чисел одно число и поразрядно прибавляет его к $M+1$-му числу (будем называть $M+1$-е число счётчиком) по модулю P, тем самым, изменяя его (то есть счётчик поразрядно накапливает сумму всех сложений по модулю P). Игрок получает выигрыш, если в результате игры счётчик обнулится.

Поразрядное сложение по модулю P выполняется следующим образом: если в каком-либо разряде числа получено значение, большее $P-1$, то оно уменьшается на P, например, при $P=5$ и $N=3$ результат сложения чисел 123 и 144 равен 212.

Вам прислали на инспекцию несколько таких автоматов, удостоверьтесь в том, что выигрыш принципиально возможен.

Формат входных данных

Первая строка содержит натуральное число — количество автоматов, присланных на инспекцию. В следующих строках описываются сами игровые автоматы. Каждый автомат описывается отдельно в следующем формате.

Первая строка содержит числа P, N, M ($1 \leqslant N, M \leqslant 100$, $2 \leqslant P \leqslant 255$). Следующие $M+1$ строк содержат по N чисел в P–ичной системе счисления, каждое из которых — значение одного разряда P–ичного числа из уникальной комбинации чисел описываемого автомата. Значения разрядов P–ичного числа задаются как числа в десятичной записи через пробел. Суммарный размер входных данных не превосходит 40 000 байт.

Формат выходных данных

Ответ по каждому автомату должен содержаться в отдельной строке. Ответ — это число 0, если игроку вообще не удастся выиграть. Если же выигрыш возможен, то ответ — это число 1 и далее — M чисел через пробел в этой же строке: k_1, k_2, …, k_M, где значение k_i ($k_i \leqslant P$) указывает, сколько раз нужно прибавить к счетчику число a_i, чтобы в результате всех сложений счетчик обнулился. Если решений несколько, выведите любое из них.

Пример

стандартный ввод	**стандартный вывод**
3 4 2 2 2 2 2 2 3 3 3 2 4 1 0 2 0 0 0 0 1 2 1 14 2 2 12 12 10 10 3 3	0 1 1 0 0 2 0

Задача B. Испорченная матрица

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

У Васи есть секретная квадратная матрица размера n на n. Каждый элемент матрицы является целым числом от 0 до 9. Вася вычислил и запомнил определитель матрицы по модулю $1\,000\,000\,007$.

Однажды Вася обнаружил, что листок с матрицей был в некоторых местах испорчен. Помогите Васе восстановить матрицу.

Формат входных данных

В первой строке заданы целое число n — количество строк и столбцов в матрице — и определитель секретной матрицы d, взятый по модулю $1\,000\,000\,007$ ($1 \leqslant n \leqslant 200$, $0 \leqslant d \leqslant 1\,000\,000\,006$). В следующих n строках задана матрица. Каждая строка матрицы не содержит пробелов и состоит ровно из n символов, причём испорченные элементы обозначаются звёздочками («`*`», ASCII-код 42). Гарантируется, что количество испорченных элементов не превышает 5.

Формат выходных данных

Если не существует матрицы, удовлетворяющей всем условиям, выведите «`error`». Если существует более одного решения, выведите «`ambiguous`». Если существует единственное решение, выведите в первой строке «`unique`», а в следующих n строках выведите саму матрицу в том же формате, что и во вводе.

Примеры

стандартный ввод	стандартный вывод
2 9 12 0*	unique 12 09
2 10 12 0*	error
2 0 ** 12	ambiguous
3 1000000001 579 123 232	unique 579 123 232

Задача C. Числовая пирамида

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Числовой пирамидой размера n называется треугольная таблица, которая обладает следующими свойствами:

- В таблице n рядов.
- В i-м сверху ряду ровно i чисел.
- Каждый следующий ряд расположен так, чтобы числа предыдущего ряда находились над промежутками между соседними числами следующего.
- Каждое число, кроме чисел нижнего ряда, равно сумме двух соседних чисел в следующем ряду, над промежутком между которыми оно находится.

Мирон нарисовал на доске числовую пирамиду размера n из целых чисел. После этого Никифор подошёл к доске и молча стёр некоторые числа. Мирон возмутился, но Никифор в ответ сказал, что ничего непоправимого не произошло: не существует двух различных пирамид размера n, в которых оставшиеся числа были бы такими же, даже если разрешить использовать не целые числа, а вещественные.

Помогите Мирону восстановить числовую пирамиду.

Формат входных данных

В первой строке задано целое число n — размер пирамиды ($1 \leqslant n \leqslant 9$).

Далее следует n строк. В i-й из этих строк задан i-й ряд числовой пирамиды: он задаётся как i целых чисел, разделённых пробелами. Строка может начинаться с одного или нескольких пробелов. Каждое число лежит в пределах от 1 до 9, если оно известно. Неизвестные числа обозначаются нулём.

Гарантируется, что существует единственная числовая пирамида из вещественных чисел, в которой известные числа совпадают с заданными. Также гарантируется, что в этой единственной числовой пирамиде все числа целые.

Формат выходных данных

Выведите n строк. В i-й из этих строк выведите i-й ряд числовой пирамиды: он задаётся как i целых чисел, разделённых пробелами. В позициях, где числа были известны, должны стоять те же числа в том же порядке. Разрешается выводить пробелы в начале и в конце строки, а также несколько пробелов подряд.

Примеры

стандартный ввод	***стандартный вывод***
2 3 0 1	3 2 1
2 0 5 5	10 5 5

Задача D. Головоломка

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Вася очень любит играть в квесты. Он любит кликать тут и там мышкой и наслаждается шутками, которых много в квестах. Но там есть ещё и много разных головоломок, и Вася проводит много времени, решая их. Однажды он столкнулся с головоломкой, которую никак не мог решить. К счастью, Вася отличный программист, и он смог написать программу, которая решила головоломку и помогла ему закончить квест.

Васина головоломка представляет собой матрицу 3×3, каждая клетка которой окрашена в чёрный или белый цвет. Если кликнуть на клетку, то она и её соседи меняют свой цвет на противоположный. Цель — сделать все клетки одного цвета.

Ваша задача чуть более сложная. Пусть есть N клеток, занумерованных от 1 до N. Каждая клетка имеет множество клеток, связанных с ней. Когда игрок кликает на клетку, все клетки из множества, связанного с ней, меняют свой цвет. По данным связанным множествам и начальной раскраске выведите последовательность клеток, на которые нужно кликать, чтобы все клетки приобрели один цвет.

Формат входных данных

Первая строка содержит целое число N ($1 \leqslant N \leqslant 200$). Следующие N строк описывают множества, связанные с клетками $1, 2, \ldots, N$. Каждое описание начинается с целого числа k — количества клеток в множестве, а затем идут k различных целых чисел (номера клеток). Последняя строка содержит N нулей и единиц и задаёт начальную раскраску клеток.

Формат выходных данных

Если невозможно окрасить все клетки в один цвет, выведите число -1. Иначе в первой строке выведите целое число L — количество кликов, необходимых для решения головоломки, а во второй строке L чисел — номера клеток, на которые необходимо кликать, в любом порядке. Если есть несколько различных решений, выведите любое.

Пример

стандартный ввод	**стандартный вывод**
9 3 1 2 4 4 1 2 3 5 3 2 3 6 4 1 4 5 7 5 2 4 5 6 8 4 3 5 6 9 3 4 7 8 4 5 7 8 9 3 6 8 9 1 0 1 0 1 0 1 0 1	9 1 2 3 4 5 6 7 8 9

Задача E. Линейные уравнения

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Система линейных уравнений, как всем известно, есть множество уравнений

$$\begin{aligned} a_{11}x_1 + \ldots + a_{1n}x_n &= b_1 \\ &\ldots \\ a_{n1}x_1 + \ldots + a_{nn}x_n &= b_n \end{aligned}$$

Ваша задача — решить её.

Формат входных данных

В первой строке записано целое число n ($1 \leqslant n \leqslant 20$). В следующих n строках записано по $n+1$ целых чисел: $a_{i1}, \ldots, a_{in}, b_i$. Все эти числа не превышают 100 по абсолютному значению.

Формат выходных данных

Первая строка должна содержать одно из следующих сообщений:

- `impossible` — решений нет
- `infinity` — бесконечно много решений
- `single` — единственное решение. В этом случае вторая строка должна содержать n чисел $x_1, \ldots, x_n$, разделённых пробелами. Решение должно быть выведено ровно с тремя знаками после десятичной точки.

Примеры

стандартный ввод	**стандартный вывод**
2 1 1 1 2 2 2	infinity
2 1 2 0 1 2 1	impossible
2 1 2 1 2 1 0	single -0.333 0.667

Замечание

Если решение получает неправильный ответ на тесте 34, это проблемы с точностью.

Задача F. Уравнение 1

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Многочленом $P(x)$ над полем $\mathbb{F}_2$ называется выражение вида $\sum_{i=0}^{n} c_i \cdot x^i$, в котором каждое из чисел c_i равно либо 0, либо 1, а $c_n = 1$. Число $n = \deg(P)$ называется степенью многочлена. Все коэффициенты c_i при $i > n$ считаются равными нулю.

Произведением многочленов $P(x) = \sum_{i=0}^{n} p_i \cdot x^i$ и $Q(x) = \sum_{j=0}^{m} q_j \cdot x^j$ называется многчлен

$$S(x) = \sum_{i=0}^{n} \sum_{j=0}^{m} p_i \cdot q_j \cdot x^{i+j}.$$

Помните, что в $\mathbb{F}_2$ результат каждого сложения и умножения коэффициентов рассматривается исключительно по модулю 2.

Утверждение $P = 0$ считается верным, если все коэффициенты многочлена $P(x)$ равны нулю.

Даны два многочлена P и Q над полем $\mathbb{F}_2$. Необходимо найти такие два многочлена A и B, чтобы выполнялось утверждение $AP + BQ \neq 0$ и при этом степень многочлена $AP + BQ$ была минимальна.

Формат входных данных

В первой строке задано число тестовых случаев T ($1 \leqslant T \leqslant 100$). В каждой из следующих T пар строк задан один тестовый случай — два многочлена. Многочлен записывается в следующей форме: $n\ c_0\ c_1\ c_2\ \ldots\ c_n$. Все c_i равны 0 или 1, а $c_n = 1$.

Все заданные многочлены имеют положительную степень. Сумма степеней всех многочленов в одном тесте не превосходит 1000.

Формат выходных данных

Для каждого теста выведите два многочлена — сначала A, а затем B — в формате, аналогичном вводу. Если ответов несколько, вы можете вывести любой. Степени выведенных многочленов не должны превосходить $\max(10, 2(\deg(P) + \deg(Q)))$. Гарантируется, что для любых P и Q существует ответ, минимизирующий степень $AP + BQ$ и удовлетворяющий этим ограничениям.

Пример

стандартный ввод	**стандартный вывод**
3 2 1 1 1 1 1 1 4 1 0 0 0 1 2 1 0 1 4 1 0 1 0 1 3 1 1 0 1	0 1 1 0 1 0 0 0 1 2 0 1 1 3 1 0 1 1

Задача G. Уравнение 2

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Многочленом $P(x)$ над полем $\mathbb{F}_2$ называется выражение вида $\sum\limits_{i=0}^{n} c_i \cdot x^i$, в котором каждое из чисел c_i равно либо 0, либо 1, а $c_n = 1$. Число $n = \deg(P)$ называется степенью многочлена. Все коэффициенты c_i при $i > n$ считаются равными нулю.

Произведением многочленов $P(x) = \sum\limits_{i=0}^{n} p_i \cdot x^i$ и $Q(x) = \sum\limits_{j=0}^{m} q_j \cdot x^j$ называется многчлен

$$S(x) = \sum_{i=0}^{n} \sum_{j=0}^{m} p_i \cdot q_j \cdot x^{i+j}.$$

Помните, что в $\mathbb{F}_2$ результат каждого сложения и умножения коэффициентов рассматривается исключительно по модулю 2.

Утверждение $P = 0$ считается верным, если все коэффициенты многочлена $P(x)$ равны нулю.

Даны два многочлена P и Q над полем $\mathbb{F}_2$. Необходимо найти такие два многочлена A и B, чтобы выполнялось утверждение $AP + BQ \neq 0$ и при этом степень многочлена $AP + BQ$ была минимальна.

Формат входных данных

В первой строке задано число тестовых случаев T ($1 \leqslant T \leqslant 100$). В каждой из следующих T пар строк задан один тестовый случай — два многочлена. Многочлен записывается в следующей форме: $n\ c_0\ c_1\ c_2\ \ldots\ c_n$. Все c_i равны 0 или 1, а $c_n = 1$.

Все заданные многочлены имеют положительную степень. Сумма степеней всех многочленов в одном тесте не превосходит 10^5.

Формат выходных данных

Для каждого теста выведите два многочлена — сначала A, а затем B — в формате, аналогичном вводу. Если ответов несколько, вы можете вывести любой. Степени выведенных многочленов не должны превосходить $\max(10, 2(\deg(P) + \deg(Q)))$. Гарантируется, что для любых P и Q существует ответ, минимизирующий степень $AP + BQ$ и удовлетворяющий этим ограничениям.

Пример

стандартный ввод	**стандартный вывод**
3 2 1 1 1 1 1 1 4 1 0 0 0 1 2 1 0 1 4 1 0 1 0 1 3 1 1 0 1	0 1 1 0 1 0 0 0 1 2 0 1 1 3 1 0 1 1

Задача H. Баянический квадрат

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Будем рассматривать квадраты размера $(n+1)\times(n+1)$, клетки которых нумеруются от $(0,0)$ до (n,n) (на рисунке изображён такой квадрат при $n=2$):

	0	1	2
0	$(0,0)$	$(1,0)$	$(2,0)$
1	$(0,1)$	$(1,1)$	$(2,1)$
2	$(0,2)$	$(1,2)$	$(2,2)$

В клетках этого квадрата расставляются вещественные числа от -10 до 10. Квадрат с расставлеными в нём числами назовём *баяническим*, если для каждой его внутренней (то есть имеющей ровно четырёх соседей) клетки верно, что значение в ней равно среднему арифметическому значений этих соседей, то есть справедлива формула $C_{xy} = (C_{x-1,y} + C_{x+1,y} + C_{x,y+1} + C_{x,y-1})/4$. К примеру, квадрат

1	2	5	6
2	3	4	5
5	4	3	2
6	5	2	1

является баяническим. Напишите программу, которая бы по известным числам в граничных клетках баянического квадрата строила бы сам квадрат (гарантируется, что решение существует и единственно).

Формат входных данных

В первой строке задано число n ($2 \leqslant n \leqslant 50$). В последующих строках идут значения чисел на границе квадрата в порядке обхода по часовой стрелке, начиная с клетки $(0,0)$. Например, для $n=2$ числа будут идти в таком порядке: $(0,0)$, $(1,0)$, $(2,0)$, $(2,1)$, $(2,2)$, $(1,2)$, $(0,2)$, $(0,1)$. Все числа разделены произвольным количеством пробелов и/или переводов строки.

Формат выходных данных

Выведите искомый баянический квадрат, по $n+1$ числу на строку. Каждое число необходимо выводить с пятью десятичными знаками после запятой. Числа должны быть расположены в том же порядке, что и в квадратах на рисунке. Строки расположены в порядке возрастания значения y. Соседние числа в строке отделяйте друг от друга одним пробелом.

Пример

стандартный ввод

```
3
1 2 5 6
5  2
1 2 5 6    5 2
```

стандартный вывод

```
1.00000 2.00000 5.00000 6.00000
2.00000 3.00000 4.00000 5.00000
5.00000 4.00000 3.00000 2.00000
6.00000 5.00000 2.00000 1.00000
```

Задача I. Переключатели

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Маленькая Зина мечтает стать робототехником. Она уже скачала себе программу-симулятор, и теперь погрузилась в мир компьютерных задачек для роботов.

В очередной задаче робот стоит перед бесконечной стеной, на которой ровной линией один за другим идут слева направо переключатели. Каждый переключатель имеет ширину в один сантиметр. Изначально все переключатели выключены.

У робота может быть несколько манипуляторов. Каждый манипулятор — это линия из одного или нескольких гнёзд для пальцев, ширина каждого гнезда — один сантиметр, и в каждом либо установлен палец, либо пусто. Например, у нынешнего робота Зины два манипулятора: один из пяти гнёзд с пальцами в первом, среднем и последнем гнезде, а второй из трёх гнёзд с пальцами во всех трёх.

Если робот прикасается манипулятором к стене — так, чтобы положения пальцев совпали с положениями каких-то переключателей, — то переключатели, до которых дотронулись пальцы, меняют своё положение (выключенные включаются, включённые выключаются). Каждый манипулятор может свободно двигаться направо и налево, но не может поворачиваться. Каждым манипулятором можно прикоснуться к стене сколько угодно раз в произвольных позициях.

Как только переключатели приходят в такое положение, что между самым левым краем включённого переключателя и самым правым таким краем ровно k сантиметров, в стене открывается проход, и робот может перейти к следующему заданию. Положение переключателей между этими двумя не имеет значения. Например, для стены, перед которой стоит робот Зины, $k = 5$. Если условие про k сантиметров перестаёт выполняться, проход вновь закрывается.

Зина решила эту и несколько похожих задачек, но одна из них никак не решалась. Девочка отправилась читать форум и выяснила, что, действительно, из-за ошибки в программе-симуляторе некоторые задачки не имеют ни одного решения, зато у некоторых других существует больше одной итоговой конфигурации переключателей между самым левым и самым правым включённым. Зина задумалась: как по конкретной задачке узнать, сколько у неё решений?

Решите обобщённый вариант Зининой задачи. По заданному списку манипуляторов, а также числу k, найдите количество различных итоговых конфигураций переключателей, при которых в стене открыт проход.

Формат входных данных

Первая строка входных данных содержит два целых числа n и k — количество манипуляторов у робота и требуемое расстояние между краями включённых переключателей ($1 \leqslant n, k \leqslant 50$).

Следующие n строк описывают манипуляторы. Каждая из них состоит из нулей и единиц и перечисляет гнёзда манипулятора слева направо: нулю соответствует пустое гнездо, а единице — гнездо с пальцем. Гарантируется, что длина каждой из этих строк лежит в пределах от 1 до 50 символов, а также что первый и последний символы каждой такой строки — это единицы.

Формат выходных данных

Выведите одно целое число: количество различных конфигураций переключателей, которые можно получить при помощи доступных манипуляторов и при которых расстояние между самым левым краем включённого переключателя и самым правым таким краем в точности равно k сантиметрам. Если две конфигурации получены разными способами, но положения переключателей в них совпадают, они считаются одинаковыми. Конфигурации, получающиеся друг из друга сдвигом влево или вправо, также считаются одинаковыми.

Примеры

стандартный ввод	стандартный вывод	конфигурации
2 5 10101 111	2	1: 10101 => ...*.*.*... 2: 10101 111 => ...**.**...
2 3 1001 100011	1	1: 1001 100011 => ......***...

Задача J. Векторы

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Задано m векторов из нулей и единиц, длина каждого вектора равна n. Сложение векторов осуществляется покомпонентно по модулю 2.

Требуется для каждого вектора определить, можно ли его получить, сложив некоторое подмножество предыдущих, а для последнего вектора определить, какие вектора следует сложить, чтобы его получить.

Формат входных данных

Первая строка содержит два целых числа n и m — длину векторов и их количество ($1 \leqslant n, m \leqslant 2000$). Следующие m строк содержат по n целых чисел, каждое из которых равно 0 или 1.

Формат выходных данных

Для каждого вектора выведите "`Yes`" или "`No`" в отдельной строке. Если ответ для последнего вектора "`Yes`", в следующей строке выведите $m - 1$ число, каждое из которых должно быть равно 0 или 1. Выведите 1 для тех векторов, которые следует сложить, чтобы получить последний вектор.

Пример

стандартный ввод	*стандартный вывод*
4 6 0 0 0 0 1 0 0 1 1 0 1 0 0 0 1 1 0 1 0 0 0 1 1 1	Yes No No Yes No Yes 0 1 1 0 1