Задача A. Трисочетание

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Дан трёхдольный полный взвешенный граф. Все три доли имеют одинаковый размер n. Трёхдольный означает, что нет ребер внутри доли. Полный означает, что между долями проведены все возможные ребра, каждое ребро ровно один раз. Нужно найти трисочетание минимального веса. Трисочетание — n троек индексов (a_i, b_i, c_i) такие, что все a_i различны, все b_i различны, все c_i различны. Вес трисочетания равен $\sum\limits_{i=1..n}(w[0, a_i, b_i] + w[1, b_i, c_i] + w[2, c_i, a_i])$, где $w[i, a, b]$ — вес ребра из a-й вершины i-й доли в b-ю вершину $(i + 1) \bmod 3$ доли.

Формат входных данных

На первой строке число $n \geqslant 1$.

Следующие n строк содержат матрицу $w[0]$.

Следующие n строк содержат матрицу $w[1]$.

Следующие n строк содержат матрицу $w[2]$.

Гарантируется, что все веса — случайные числа от k до $2k$ для некоторого k. Все веса — целые числа от 0 до 10^5.

Формат выходных данных

На первой строке выведите суммарный вес найденного трисочетания. На каждой из следующих n строк выведите $a_i b_i c_i$ (вершины нумеруются от 0 до $n - 1$). Тройки можно выводить в любом порядке. Если трисочетаний с минимальным суммарным весом несколько, подойдет любое.

Система оценки

Подзадача 1 (25 баллов) $n \leqslant 5$.

Подзадача 2 (25 баллов) $n \leqslant 9$.

Подзадача 3 (25 баллов) $n \leqslant 11$.

Подзадача 4 (25 баллов) $n \leqslant 13$.

Пример

стандартный ввод	стандартный вывод
3 64 100 96 90 43 50 12 94 23 75 97 45 84 19 45 11 60 28 9 16 79 49 21 54 85 2 74	330 0 2 0 1 1 1 2 0 2

Задача B. Число совершенных паросочетаний

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Совершенным называется паросочетание, покрывающее все вершины графа.

Дан произвольный неориентированный граф.
Найдите количество совершенных паросочетаний по модулю $10^9 + 7$.

Формат входных данных

Число вершин n ($1 \leqslant n \leqslant 30$). Числе рёбер m ($0 \leqslant m \leqslant \frac{n(n-1)}{2}$)

Формат выходных данных

Выведите одно число – количество совершенных паросочетаний по модулю $10^9 + 7$.

Пример

стандартный ввод	стандартный вывод
4 4 1 3 1 4 2 3 2 4	2

Задача C. Дорога, дорога, осталось немного...

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Дан случайный неориентированный граф G из n вершин и m ребер. Ваша задача — найти гамильтонов путь. Гарантируется, что гамильтонов путь в графе есть.

Формат входных данных

На первой строке число вершин $n \geqslant 2$ и число ребер $m \geqslant 1$.

Следующие m строк содержат пары чисел от 1 до n — ребра графа.

В графе нет ни петель, ни кратных ребер.

Поскольку почти полный граф — совсем не интересный тест, $m \leqslant 500$.

Формат выходных данных

На первой строке выведите n различных чисел от 1 до n — вершины гамильтоного пути в порядке прохода по ним. Начинать и заканчивать можно в любой вершине. Если гамильтоновых путей несколько, выведите любой.

Система оценки

Подзадача 1 (20 баллов) $n \leqslant 26$.
Подзадача 2 (20 баллов) $n \leqslant 35$.
Подзадача 3 (20 баллов) $n \leqslant 50$.
Подзадача 4 (20 баллов) $n \leqslant 70$.
Подзадача 5 (20 баллов) $n \leqslant 100$.

Пример

стандартный ввод	стандартный вывод
5 8 3 1 2 5 5 4 3 4 1 4 3 5 3 2 1 2	1 4 3 5 2

Задача D. ЮграНефтеТранс

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Ханты-Мансийский автономный округ — Югра является важнейшим нефтяным регионом России. Добыча нефти составляет 267 млн. т. в год, её транспортировка осуществляется по трубопроводам, общая длина которых превышает длину экватора Земли.

Система транспортировки нефти представляет собой совокупность n распределительных станций и m трубопроводов. Каждый трубопровод соединяет две различные станции. Между любыми двумя станциями проложено не более одного трубопровода.

Эффективность работы станций существенно зависит от вязкости нефти. Поэтому компания «ЮграНефтеТранс», в ведении которой находится сеть трубопроводов, заказала инновационному исследовательскому предприятию разработку и изготовление новых сверхточных датчиков вязкости на основе самых современных технологий.

Изготовление датчиков — процесс трудоёмкий и дорогостоящий, поэтому было решено изготовить k датчиков ($k \leqslant 40$) и выбрать k различных станций, на которых датчики будут установлены. Необходимо осуществить выбор станций так, чтобы датчики контролировали все трубопроводы: для каждого трубопровода хотя бы один датчик должен быть установлен на станции, где начинается или заканчивается этот трубопровод.

Напишите программу, которая проверяет, существует ли требуемое расположение датчиков, и в случае положительного ответа находит это расположение.

Формат входных данных

В первой строке входного файла записаны три натуральных числа — n, m и k ($k \leqslant n \leqslant 2\,000, 1 \leqslant m \leqslant 10^5, 1 \leqslant k \leqslant 40$).

Далее следуют m строк, каждая из которых описывает один трубопровод. Трубопровод задаётся двумя целыми числами — порядковыми номерами станций, которые он соединяет. Станции пронумерованы от 1 до n. Гарантируется, что к любой станции подведён хотя бы один трубопровод и между любыми двумя станциями проложено не более одного трубопровода. Числа в каждой строке разделены пробелами.

Формат выходных данных

В первую строку выходного файла выведите слово «Yes», если требуемое расположение датчиков существует, в противном случае — слово «No». В случае положительного ответа выведите во вторую строку выходного файла k различных целых чисел — номера станций, на которых необходимо установить датчики. Номера можно выводить в любом порядке. Если существует несколько подходящих расположений датчиков, выведите любое из них. Разделяйте числа во второй строке пробелами.

Примеры

стандартный ввод	стандартный вывод
2 1 2 1 2	Yes 1 2
3 3 1 1 2 2 3 3 1	No
7 6 2 1 2 1 3 1 4 2 5 2 6 2 7	Yes 1 2
5 5 2 1 2 1 3 1 4 1 5 4 5	Yes 1 5

Задача E. Японский компьютер

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.25 секунд
Ограничение по памяти:	256 мегабайт

Как известно, для обороны границ японские инженеры разрабатывают огромных боевых человекоподобных роботов. Каждый такой робот управляется японским компьютером. Понятно, что для повышения эффективности робота программа в компьютере должна быть как можно более оптимальной, чтобы компьютер мог выполнять как можно больше программ за как можно меньшее время.

На данный момент японским программистам задали следующую задачу (её смысл секретен, поэтому здесь его описывать нельзя): изначально в памяти компьютера находится единственное число x. Требуется получить в его памяти следующие числа: $a_1x, a_2x, \ldots, a_nx$. При этом компьютер может выполнять следующие операции:

1. Сложение двух чисел
2. Вычитание двух чисел
3. Побитовый сдвиг влево (сдвиг на k бит эквивалентен умножению на 2^k)

Все полученные промежуточные значения сохраняются в памяти, так что ими можно пользоваться при вычислении других значений.

При вычислениях никогда не должно получаться значение большее, чем $42x$. Гарантируется, что при выполнении этого ограничения, в компьютере не происходит переполнений. Также, компьютер не может работать с отрицательными числами, так что вычитать большее число из меньшего также запрещено.

Порядок, в котором в памяти будут появляться числа $a_1x, a_2x, \ldots, a_nx$, не имеет значения.

Формат входных данных

В первой строке находится число n — количество требуемых значений ($1 \leqslant n \leqslant 41$). Во второй строке находится n чисел a_i ($2 \leqslant a_i \leqslant 42$). Все a_i различны. Само число x вам не дано, так что ваша последовательность операций должна быть верной для любого x.

Формат выходных данных

В первой строке выведите единственное число — минимальное количество требуемых операций. Далее выведите требуемые операции в следующем формате:

1. Сдвиг влево ax на k бит: “`a<<k`”
2. Сложение ax и bx: “`a+b`”
3. Вычитание ax из bx: “`b-a`”

Запись операций не должна содержать пробелов.

Примеры

стандартный ввод	стандартный вывод
3 3 5 18	5 1+1 2+1 3+2 1<<4 16+2
1 29	4 1+1 2+1 1<<5 32-3
4 12 19 41 42	8 1+1 2+1 3<<2 12+12 24-2 22-3 19+22 41+1

Задача F. SO-SAT

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Найдите решение `3-SAT`. Гарантируется, что оно существует.

Формулировка `3-SAT`: нужно подобрать значения n булевых переменных так, чтобы все m утверждений вида $x_{i_1} = e_1 \vee x_{i_2} = e_2 \vee x_{i_3} = e_3$ обратились в истину.

Формат входных данных

На первой строке число переменных n и число утверждений m $(1 \leqslant n \leqslant 90, 1 \leqslant m \leqslant \min(n^2, 1000))$.

Каждая из следующих m строк содержит числа $i_1, e_1, i_2, e_2, i_3, e_3$ и задает утверждение $x_{i_1} = e_1 \vee x_{i_2} = e_2 \vee x_{i_3} = e_3$.

Все тесты случайны, тем не менее гарантируется, что решение существует.

Формат выходных данных

Выведите строку из n нулей и единиц — значения переменных.

Если у данной задачи `3-SAT` есть несколько решений, выведите любое.

Пример

стандартный ввод	стандартный вывод

`2 3`
`1 0 1 0 1 0`
`2 0 2 1 1 1`
`1 1 2 1 1 1`
	`01`

Задача G. Макс клика

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.25 секунд
Ограничение по памяти:	256 мегабайт

Макс, опытнейший игрок в доту, постоянно кликал.

Дан случайный неориентированный граф G из n вершин и m ребер.

Подкликой называется такое подмножество вершин A: $\forall a, b \in A, a \neq b \quad \exists$ ребро (a, b).

Ваша задача — найти подклику A: $|A|$ максимально.

Формат входных данных

На первой строке число вершин $n \geqslant 1$ и число ребер $m \geqslant 1$.

Следующие m строк содержат пары чисел от 1 до n — ребра графа.

В графе нет ни петель, ни кратных ребер.

Формат выходных данных

На первой строке выведите k — количество вершин в максимальной подклике. На следующей строке k целых чисел от 1 до n — номера вершин в подклике. Вершины можно выводить в любом порядке. Если максимальных подклик несколько, выведите любую.

Система оценки

Подзадача 1 (10 баллов) $n \leqslant 20$.
Подзадача 2 (10 баллов) $n \leqslant 25$.
Подзадача 3 (20 баллов) $n \leqslant 40$.
Подзадача 4 (20 баллов) $n \leqslant 60$.
Подзадача 5 (20 баллов) $n \leqslant 70$.
Подзадача 6 (20 баллов) $n \leqslant 80$.

Пример

стандартный ввод	стандартный вывод
5 8 5 4 3 5 1 5 1 3 2 3 1 4 5 2 3 4	4 3 5 1 4

Задача H. Множество множеств

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Даны несколько случайных подмножеств множества $A = \{1, 2, \ldots, n\}$. Нужно выбрать минимальное количество подмножеств, которые в объединении дают A. Гарантируется, что объединение всех подмножеств равно A.

Формат входных данных

На первой строке число $n \geqslant 1$ и число множеств $m \geqslant 1$. Следующие m строк задают множества. Каждое множество задается числом элементов $k_i \geqslant 5$ и k_i различными числами от 1 до n. Гарантируется, что множества генерировались следующим образом: фиксируем c, далее m раз сперва выбираем случайный размер s от 1 до c, далее равновероятно из всех C_n^s вариантов выбираем множество размера s.

Формат выходных данных

На первой строке выведите x — минимальное количество множеств. На следующей строке выведите номера выбранных множеств, x различных чисел от 1 до m. Номера можно выводить в произвольном порядке. Если оптимальных ответов несколько, выведите любой.

Система оценки

Подзадача 1 (25 баллов) $n \leqslant 20, m \leqslant 50$.
Подзадача 2 (25 баллов) $n \leqslant 30, m \leqslant 100$.
Подзадача 3 (25 баллов) $n \leqslant 40, m \leqslant 100$.
Подзадача 4 (25 баллов) $n \leqslant 50, m \leqslant 100$.

Пример

стандартный ввод	стандартный вывод
5 4 4 4 5 1 2 5 3 5 1 2 4 1 2 2 3 5	1 2

Задача I. Видеонаблюдение

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.3 секунд
Ограничение по памяти:	256 мегабайт

Город-Таз известен нереально высоким уровнем криминала. Полиция не видит вариантов развития ситуации, кроме как усилить меры безопасности. Они хотят установить движующихся дронов-беспилотников на некоторых перекрёстках города, чтобы знать, кто проезжает перекрёстки на красный свет. Если машина проедет на красный свет, дрон погонится за ней и остоновит машину, чтобы вручить водителю соответствующий талон. Дроны-беспилотники довольно глупы, поэтому остановятся обязательно до следующего перекрёстка, иначе они рискуют заблудиться и потерять дорогу домой. Домом дрона считается перекрёсток, к которому тот прикреплён. Дроны не умеют определять присутствие других дронов (отличать их от машин-нарушителей), поэтому полицейский технический департамент принял решение, что никакие два дрона нельзя прикреплять к соседним перекрёсткам. Как и в большинстве городов, в Городе-Тазе нет перекрёстков с более чем четырьмя соседними перекрёстками.

Дроны выделяются государством (бесплатно!), поэтому полиция хочет получить настолько много дронов, насколько это возможно. Вас попросили определить, возможно ли в городе расположить заданное число дронов, не нарушая правило, что ни на каких двух соседних перекрёстках не должно одновременно быть дронов.

Формат входных данных

Первая строка содержит число k ($0 \leqslant k \leqslant 15$) – количество дронов, которых нужно разместить в городе. На второй строке n ($1 \leqslant n \leqslant 100\,000$) – число перекрёстков в Городе-Тазе. Следующие n строк описывают перекрёстки в i-й строке сперва дано число d ($0 \leqslant d \leqslant 4$), количество соседних перекрёстков с i-м, затем d номеров соседних перекрёстков. Все эти d чисел различны и отличны от i. Отношение “соседние перекрёстки” симметрично – если i соседний с j, то и j соседний с i. Номера перекрёстков – числа от 1 до n.

Формат выходных данных

На первой строке `possible` или `impossible`. Если ответ – `possible`, выведите на второй строке k целых чисел от 1 до n, номера вершин.

Система оценки

Потестовая оценка.

Примеры

стандартный ввод	стандартный вывод
4 7 2 2 4 3 1 3 5 1 2 2 1 5 4 2 6 4 7 2 5 7 2 6 5	impossible
4 8 2 2 4 3 1 3 5 1 2 2 1 5 4 2 6 4 7 2 5 8 2 8 5 2 7 6	possible 5 3 1 8

Задача J. Коробки

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дано N коробок в пространстве. Про каждую коробку известны её три измерения. Коробки можно вращать — то есть выбирать, какое измерение будет длиной, какое — шириной, а какое — высотой. Из этих коробок необходимо построить башню максимальной высоты. Коробки разрешается ставить друг на друга, только если длина коробки сверху не превосходит длины коробки снизу, а ширина коробки сверху — ширины коробки снизу.

Формат входных данных

В первой строке задано целое число N ($1 \leqslant N \leqslant 30$). Следующие N строк содержат по три целых числа a_i, b_i и c_i ($1 \leqslant a_i, b_i, c_i \leqslant 10^6$) — три измерения i-ой коробки.

Формат выходных данных

В первой строке выведите максимальную высоту башни H. Во второй строке выведите целое число k — количество коробок в искомой башне. В последующих k строках выведите по четыре числа для каждой коробки — её номер m_i, длину l_i, ширину d_i и высоту h_i (коробки нужно выводить в порядке снизу вверх). Коробки нумеруются с единицы в порядке, заданном во входном файле.

Пример

стандартный ввод	стандартный вывод
3 3 1 3 2 2 2 1 2 1	5 2 1 1 3 3 3 1 1 2

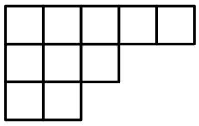
Задача K. Увидеть Юнга и умереть

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

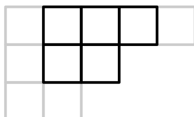
Диаграммы Юнга используются для того, чтобы изобразить разбиение числа на слагаемые. Разбиение числа n на слагаемые представляет собой сумму вида $n = m_1 + m_2 + \ldots + m_k$, где $m_1 \geqslant m_2 \geqslant \ldots \geqslant m_k$.

Диаграмма состоит из n квадратиков, организованных в виде k рядов, где k количество слагаемых в разбиении. Ряд, соответствующий числу m_i, содержит m_i квадратиков. Все ряды выровнены по левому краю и упорядочены от более длинного к более короткому.

Например, диаграмма Юнга, приведенная на рисунке, соответствует разбиению $10 = 5 + 3 + 2$.



Иногда можно вписать одну диаграмму Юнга в другую. Диаграмму X можно вписать в диаграмму Y, если можно удалить некоторые квадратики из диаграммы Y так, чтобы получилась диаграмма X. Отметим, что разрешается только удалять некоторые квадратики, вращать или отражать диаграмму не разрешается. Например, диаграмма для разбиения $5 = 3 + 2$ может быть вписана в диаграмму для разбиения $10 = 5 + 3 + 2$, как показано на рисунке.



С другой стороны, диаграмму для разбиения $8 = 4+4$ нельзя вписать в диаграмму для разбиения $10 = 5 + 3 + 2$.

Для заданного n найдите такое разбиение n на слагаемые, что в соответствующую ему диаграмму Юнга можно вписать максимальное количество различных диаграмм.

Например, в диаграмму для разбиения $10 = 5 + 3 + 2$ можно вписать 36 различных диаграмм. Однако это не максимальное значение. В диаграмму для разбиения $10 = 4 + 2 + 2 + 1 + 1$ можно вписать 41 диаграмму Юнга.

Формат входных данных

Входной файл содержит целое число n ($1 \leqslant n \leqslant 100$).

Формат выходных данных

На первой строке выходного файла выведите максимальное число диаграмм Юнга, которые можно вписать в некоторую диаграмму, соответствующую разбиению на слагаемые числа n.

На второй строке выведите одно или более целых чисел — количество квадратиков в каждом из рядов оптимальной диаграммы.

Пример

стандартный ввод	стандартный вывод
10	41 4 3 2 1

Задача L. Мать драконов

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.3 секунд
Ограничение по памяти:	256 мегабайт

В Королевстве Ланнистеров n замков и несколько стен, соединяющих два замка, никакие два замка не соединены более, чем одной стеной, ни одна стена не соединяет замок с собой.

Сир Джейме Ланнистер узнал, что Дейенерис Таргариен собирается атаковать его королевство. Он хочет защитить свои владения. У него есть k литров странной жидкости. Он хочет распределить эту жидкость между замками так, чтобы каждый замок содержал некоторое количество жидкости (возможно, нулевое или нецелое количество литров). После этого стабильность стены, соединяющей замки a и b, содержащие x и y литров жидкости, соответственно, равна $x \cdot y$.

Ваша задача — найти максимальную возможную сумму стабильностей стен, которую Сир Джейме Ланнистер сможет достичь

Формат входных данных

Первая строка содержит два целых числа n и k ($1 \leqslant n \leqslant 40$, $1 \leqslant k \leqslant 1000$).

Далее следует n строк. В i-й из них содержится n целых чисел $a_{i,1}, a_{i,2}, \ldots, a_{i,n}$ ($a_{i,j} \in \{0, 1\}$). Если замки i и j соединены стеной, $a_{i,j} = 1$. В противном случае оно равно 0.

Гарантируется, что $a_{i,j} = a_{j,i}$ и $a_{i,i} = 0$ для всех $1 \leqslant i, j \leqslant n$.

Формат выходных данных

Выведите одно число — максимальную возможную сумму стабильностей стен, которую Сир Джейме Ланнистер сможет достичь.

Ваш ответ будет считаться правильным, если его абсолютная или относительная точность не превосходит 10^{-6}.

А именно, если ваш ответ равен a, а ответ жюри равен b, то ваш ответ будет зачтен, если $\frac{|a-b|}{max(1,b)} \leqslant 10^{-6}$.

Примеры

стандартный ввод	стандартный вывод
3 1 0 1 0 1 0 0 0 0 0	0.250000000000
4 4 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0	4.000000000000

Замечание

В первом примере, если замки $1, 2, 3$ содержат $0.5, 0.5, 0$ литров жидкости, соответственно, ответ равен 0.25.

Во втором примере, если замки $1, 2, 3, 4$ содержат $1.0, 1.0, 1.0, 1.0$ литров жидкости, ответ равен 4.0.