Содержание

Простые задачи

Задача A. Пути на доске

Рассмотрим бесконечную клетчатую доску.

Назовём *путём* из одной клетки в другую последовательность клеток, в которой каждые две идущие подряд клетки являются соседними по стороне. Длина пути — это количество клеток в нём, не считая начальную.

Назовём путь *простым*, если в нём не встречается двух одинаковых клеток.

Зафиксируем какую-то клетку на доске. Сколько существует простых путей заданной длины, начинающихся в этой клетке?

Формат входных данных

В первой строке входного файла задано целое число n ($0 \leqslant n \leqslant 22$).

Формат выходных данных

В первой строке выходного файла выведите одно число — количество путей длины n из этой клетки.

Примеры

paths.in	paths.out
0	1
1	4
2	12

Задача B. Покрытие доминошками

Сколько способов покрыть прямоугольник $W \times H$ доминошками?
Каждая клетка должна быть покрыта ровно одной доминошкой.

Формат входных данных

Числа W и H. Ограничения: $W \cdot H \leqslant 50$, кроме этого $1 \leqslant W, H$.

Формат выходных данных

Число способов покрыть прямоугольник доминошками.

Пример

dominoes.in	dominoes.out
3 2	3

Задача C. Количество циклов

Формально, *путь* в графе — это чередующаяся последовательность вершин и рёбер $u_1, e_1, u_2, e_2, u_3, \ldots, u_k$, начинающаяся и заканчивающаяся вершиной и такая, что любые соседние вершина и ребро в ней инцидентны.

Цикл — это путь, начальная и конечная вершины которого совпадают. В цикле должно быть хотя бы одно ребро.

Простой путь отличается от обычного пути тем, что в нём не может быть повторяющихся вершин.

Простой цикл — это цикл, в котором нет повторяющихся вершин и рёбер.

Дан неориентированный граф. Посчитайте, сколько в нём различных простых циклов. Заметим, что циклы считаются одинаковыми, если они обходят одно и то же множество вершин в одном и том же порядке, возможно, начиная при этом из другой вершины, или если порядок обхода противоположный. Например, циклы с порядком обхода вершин $1, 2, 3, 1$, $2, 3, 1, 2$ и $1, 3, 2, 1$ считаются одинаковыми, а циклы $1, 2, 3, 4, 1$ и $1, 3, 4, 2, 1$ — нет, поскольку порядок обхода вершин различен.

Формат входных данных

В первой строке входного файла заданы числа N и M через пробел — количество вершин и рёбер в графе, соответственно ($1 \leqslant N \leqslant 10$). Следующие M строк содержат по два числа u_i и v_i через пробел ($1 \leqslant u_i, v_i \leqslant N$, $u_i \neq v_i$); каждая такая строка означает, что в графе существует ребро между вершинами u_i и v_i. В графе нет кратных рёбер.

Формат выходных данных

Выведите одно число — количество простых циклов в заданном графе.

Примеры

numcycle.in	numcycle.out
3 2 1 2 2 3	0
4 5 1 2 2 3 3 4 4 1 1 3	3

Задача D. Золотой песок

Во время ограбления магазина вор обнаружил N ящичков с золотым песком. В ящичек под номером i песок имеет стоимость v_i и вес w_i. Чтобы унести награбленное, вор использует рюкзак. Требуется определить наибольшую суммарную стоимость песка, который может унести грабитель, если грузоподъемность рюкзака ограничена величиной W.

Из ящичков можно пересыпать любое количество песка, тогда отношение стоимости отсыпанного песка к стоимости всего ящичка будет равна отношению объема пересыпанного песка к объему всего ящичка.

Формат входных данных

В первой строке входного файла записаны два числа — N и W ($1 \leqslant N \leqslant 1000$, $0 \leqslant W \leqslant 10^6$). Далее следует N строк по два целых числа в каждой. В i-ой строке записана стоимость v_i и вес w_i песка в i-ом ящичке. Все числа неотрицательные и не превосходят 10^6.

Формат выходных данных

Выведите искомую максимальную стоимость с точностью до 3 знаков после запятой.

Пример

dust.in	dust.out
3 50 60 20 100 50 120 30	180.000

Задача E. Скалярное произведение

Даны два вектора: $v_1 = (x_1, x_2, \ldots, x_n)$ и $v_2 = (y_1, y_2, \ldots, y_n)$. Скалярным произведением этих векторов называется значение, вычисляемое по формуле: $x_1y_1 + x_2y_2 + \ldots + x_ny_n$.

Разрешено переставлять координаты каждого из векторов любым образом. Выберите такие их перестановки, чтобы скалярное произведение двух полученных векторов было минимальным и выведите его значение.

$1 \leqslant n \leqslant 800$. $-100\,000 \leqslant x_i, y_i \leqslant 100\,000$.

Формат входных данных

Первая строка входного файла содержит единственное целое число t — количество наборов тестовых данных. Далее следуют сами наборы, по три строки в каждом. Первая строка тестового набора содержит единственное целое число n. Две следуюшие строки содержат по n целых чисел, задающих координаты соответствующего вектора, каждая.

Формат выходных данных

Для каждого набора выведите строку с номером этого набора и ответом на задачу — значением минимального скалярного произведения. Следуйте формату, указанному в примере.

Пример

product.in	product.out
2 3 1 3 -5 -2 4 1 5 1 2 3 4 5 1 0 1 0 1	Case #1: -25 Case #2: 6

Задача F. Число

Вася написал на длинной полоске бумаги большое число и решил похвастаться своему старшему брату Пете этим достижением. Но только он вышел из комнаты, чтобы позвать брата, как его сестра Катя вбежала в комнату и разрезала полоску бумаги на несколько частей. В результате на каждой части оказалось одна или несколько идущих подряд цифр.

Теперь Вася не может вспомнить, какое именно число он написал. Только помнит, что оно было очень большое. Чтобы утешить младшего брата, Петя решил выяснить, какое максимальное число могло быть написано на полоске бумаги перед разрезанием. Помогите ему!

Формат входных данных

Входной файл содержит одну или более строк, каждая из которых содержит последовательность цифр. Количество строк во входном файле не превышает 100, каждая строка содержит от 1 до 100 цифр. Гарантируется, что хотя бы в одной строке первая цифра отлична от нуля.

Формат выходных данных

Выведите в выходной файл одну строку — максимальное число, которое могло быть написано на полоске перед разрезанием.

Примеры

number.in	number.out
2 20 004 66	66220004
3	3

Задача G. Белоснежка и n гномов

«Ну не гномы, а наказание какое-то!», — подумала Белоснежка, в очередной раз пытаясь уложить гномов спать. Одного уложишь — другой уже проснулся! И так всю ночь.

У Белоснежки n гномов, и все они очень разные. Она знает, что для того, чтобы уложить спать i-го гнома нужно a_i минут, и после этого он будет спать ровно b_i минут. Помогите Белоснежке узнать, может ли она получить хотя бы минутку отдыха, когда все гномы будут спать, и если да, то в каком порядке для этого нужно укладывать гномов спать.

Например, пусть есть всего два гнома, $a_1 = 1$, $b_1 = 10$, $a_2 = 10$, $b_2 = 20$. Если Белоснежка сначала начнет укладывать первого гнома, то потом ей потребуется целых 10 минут, чтобы уложить второго, а за это время проснется первый. Если же она начнет со второго гнома, то затем она успеет уложить первого и получит целых 9 минут отдыха.

Формат входных данных

Первая строка входного файла содержит число n ($1 \leqslant n \leqslant 10^5$), вторая строка содержит числа $a_1, a_2, \ldots a_n$, третья — числа $b_1, b_2, \ldots b_n$ ($1 \leqslant a_i, b_i \leqslant 10^9$).

Формат выходных данных

Выведите в выходной файл n чисел — порядок, в котором нужно укладывать гномов спать. Если Белоснежке отдохнуть не удастся, выведите число -1.

Если решений несколько, выведите любое.

Пример

dwarfs.in	dwarfs.out
2 1 10 10 20	2 1
2 10 10 10 10	-1
3 1 4 1 5 3 4	2 1 3

Задача H. $F2 \,||\, C_{max}$

Имеется множество из n работ и два станка. Время выполнения i-й работы на первом станке равно a_i, время выполнения i-й работы на втором станке равно b_i. Каждую работу надо выполнить сначала на первом станке, потом на втором. И на первом, и на втором станке работы можно выполнять в произвольном порядке. Каждый станок в каждый момент времени может выполнять только одну работу.

Минимизируйте C_{max} — время выполнения последней работы на втором станке.

Формат входных данных

В первой строке дано одно целое число n ($1 \leqslant n \leqslant 100\,000$) — количество работ. В следующей строке n целых чисел от 0 до 10^6 – время выполнения работ на первом станке. В следующей строке n целых чисел от 0 до 10^6 – время выполнения работ на втором станке.

Формат выходных данных

В первой строке выведите единственное число — C_{max}. Во второй строке выведите перестановку – порядок выполнения работ на первом станке. В третьей строке выведите перестановку – порядок выполнения работ на втором станке.

Если решений несколько, выведите любое.

Примеры

f2cmax.in	f2cmax.out
3 1 2 3 5 5 5	16 1 3 2 1 2 3
2 3 2 1 3	6 2 1 2 1

Задачи

Задача I. Японский компьютер

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.

Если решений несколько, выведите любое.

Примеры

computer.in	computer.out
3 3 5 18	5 1<<1 1<<4 1+2 2+3 2+16
1 29	4 1<<1 1<<5 1+2 32-3
4 12 19 41 42	8 1<<1 1<<4 1+2 3<<2 3+16 19<<1 3+38 1+41

Задача J. Кривые зеркала

За один залп BFG-9000 уничтожает монстров на трех соседних балкончиках. Балкончики при этом не разрушаются. А монстры погибают. (N-й балкончик соседствует с первым). После залпа оставшиеся в живых монстры наносят Леониду (главному герою романа) повреждения — по одной единице каждый. Далее следует новый залп и так до тех пор, пока все монстры не погибнут. Требуется определить минимальные повреждения, которые может понести Леонид.

Формат входных данных

Первая строка содержит целое число N, количество балкончиков, на которых монстры заняли круговую оборону ($3 \leqslant N \leqslant 30$). Во второй строке даны N целых чисел — количество монстров на каждом балкончике (на каждом не менее 1 и не более 100).

Формат выходных данных

Выведите минимальное количество единиц повреждений.

Пример

mirror.in	mirror.out
7 3 4 2 2 1 4 1	9

Задача K. Conway

В комнате есть M лампочек и N переключателей. Для удобства гарантируется, что N **нечётное**. Каждая лампочка имеет собственную мощность p_i и может быть только в двух состояниях: полностью гореть, или полностью не гореть. Если лампочка включена, освещённость в комнате увеличивается на p_i единиц.

Каждый переключатель соединён с некоторым множеством лампочек. Смена состояния переключателя меняет состояние всех лампочек, соединённых с ним. Никаких ограничений на соединения нет, это значит, что любой переключатель может быть соединён с любым множеством лампочек, а каждая лампочка может быть соединёна с любым множеством переключателей.

Джон изобрёл новую игру и пригласил своих друзей Роланда и Патрика сыграть разок. Роланд любит свет и пытается к концу игры сделать комнату максимально освещённой. У Патрика ровно обратная цель – уменьшить освещённость комнаты настолько, насколько возможно.

Джон выбирает величину K, чтобы определить победителя. В конце игры, если суммарная мощность всех включённых лампочек хотя бы K, побеждает Роланд, иначе Патрик. Игра устроена следующим образом.

- Переключатели занумерованы числами от 1 до N.
- Игроки делают ровно по $\frac{N-1}{2}$ ходов. Ходят они по очереди. Первым ходит Роланд.
- Когда Роланд совершает свой i-й ход, он может использовать переключатели с номерами $2 \cdot i-1$ и $2 \cdot i$. Это первый и второй переключатели на его первом ходу, третий и четвёртый переключатели на его втором ходу, и так далее. Заметьте, что Роланд на каждом ходу может выбрать любое из 4 подмножеств доступных переключателей (какой-то один, оба, ни одного). Если какая-то лампочка подключена сразу к двум переключателям, которые использует на своём ходу Роланд, лампочка меняет своё состояние дважды.
- Правила для Патрика абсолютно такие же, за исключением индексов переключателей, которыми он может пользоваться. На своём i-м ходу он может пользоваться переключателями с номерами $2 \cdot i$ и $2 \cdot i+1$. Это второй и третий на его первом ходу, четвёртый и пятый на его втором ходу, и так далее.

Джону нравится смотреть за игрой своих друзей. Особенно если он заранее знает результат игры при оптимальных действиях обоих игроков. Он просит вас написать программу, которая определит победителя в каждой из T игр. Следует предполагать, что и Роланд, и Патрик играют оптимально.

Формат входных данных

Первая строка входных данных содержит целое число T — количество тестовых примеров ($1 \leqslant T \leqslant 5$). Далее каждый тестовый пример описывает отдельную игру следующим образом. Сперва строка с тремя целыми числами N, M и K ($3 \leq N \leq 33$, N нечётно, $1 \leq M \leq 32$, $0 \leq K \leq 2 \cdot 10^9$) — количество переключателей, количество лампочек, уровень освещённости для определения победителя. На следующей строке идут M чисел p_i ($1 \leq p_i \leq 5 \cdot 10^7$) — мощности лампочек. Далее N строк описывают соединения между переключателями и лампочками. Каждая из строк содержит M символов. Если i-й переключатель соединён с j-й лампочкой, то j-й символ i-й строки равен '`1`', иначе символ равен '`0`'.

Формат выходных данных

Для каждого тестового примера на отдельной строке выведите имя победителя – или "`Roland`", или "`Patrick`".

Примеры

conway.in	conway.out
2 3 2 10 10 10 01 00 11 3 5 1 1 2 3 4 5 01011 11000 10011	Roland Patrick

Замечание

В первой игре у Роланда есть следующая оптимальная стратегия: на своём первом ходу он использует и первый, и второй переключатель. Рассмотрим любой возможный ход Патрика: к концу игры будет ровно одна лампочка, которая останется включённой, поэтому Роланд по любому выиграет.

Во второй игре, не зависимо от хода Роланда, Патрик может выключить все лампочки.

Задача L. Конфеты

Мальчик Костя очень любит конфеты, но мама не разрешает ему брать их слишком много. Поэтому каждый раз, когда Костя хочет съесть конфету, мама предлагает ему сыграть в игру.

Изначально у Кости нет конфет, а у мамы их N (они пронумерованы от 1 до N). На каждой конфете мама написала два числа A_i и C_i. Мама очень следит за *уровнем вредности* конфет, который получает ее сын. Изначально этот уровень равен 0. На каждом ходу игры Костя может взять одну конфету. Если Костя возьмет конфету с номером i, то *уровень вредности* увеличивается на A_i. Если сразу после этого *уровень вредности* становится большей C_i, то брать эту конфету запрещается.

Брать конфеты можно в произвольном порядке, но одну и ту же можно брать не более одного раза.

Помогите Косте взять как можно больше конфет (вне зависимости от финального *уровеня вредности*).

Формат входных данных

В первой сроке входных данных записано целое число N ($1 \leqslant N \leqslant 1000$) — количество видов конфет. Во второй строке записаны N целых чисел A_i ($1 \leqslant A_i \leqslant 10^6$). В третей строке записаны N целых чисел C_i ($1 \leqslant C_i \leqslant 10^9$).

Формат выходных данных

В единственной строке выведите целое число, равное максимальному количеству конфет, которые может взять Костя.

Примеры

stdin	stdout
3 3 9 5 2 1 3	0
3 5 4 5 14 14 14	3
7 1 5 5 1 3 4 1 6 7 6 6 11 7 4	5

Задача M. Коробки

У Васи в комнате очень много коробок, которые валяются в разных местах. Васина мама хочет, чтобы он прибрался. Свободного места в комнате мало и поэтому Вася решил собрать все коробки и поставить их одну на другую.

К сожалению, это может быть невозможно. Например, если на картонную коробку с елочными украшениями положить что-то железное и тяжелое, то вероятно следующий Новый год придется встречать с новыми игрушками.

Вася взвесил каждую коробку и оценил максимальный вес который она может выдержать. Помогите ему определить какое наибольшее количество коробок m он сможет поставить одну на другую так, чтобы для каждой коробки было верно, что суммарный вес коробок сверху не превышает максимальный вес, который она может выдержать.

Формат входных данных

Первая строка входного файла содержит целое число n ($1 \leqslant n \leqslant 10^5$) — количество коробок в комнате. Каждая следующая из n строк содержит два целых числа w_i и c_i ($1 \leqslant w_i \leqslant 10^5, 1 \leqslant c_i \leqslant 10^9$), где w_i – это вес коробки с номером i, а c_i – это вес который она может выдержать.

Формат выходных данных

В выходной файл выведите одно число — ответ на задачу.

Пример

boxes.in	boxes.out
3 10 11 20 100 30 10	3
3 11 11 20 100 30 10	2

Дополнительные задачи

Задача N. Авторитеты

Толик придумал новую технологию программирования. Он хочет уговорить друзей использовать ее. Однако все не так просто. i-й друг согласится использовать технологию Толика, если его авторитет будет не меньше a_i (авторитет выражается целым числом). Как только i-й друг начнет ее использовать, к авторитету Толика прибавится число b_i (попадаются люди, у которых $b_i < 0$). Помогите Толику наставить на путь истинный как можно больше своих друзей.

Формат входных данных

На первой строке содержатся два целых числа: Количество друзей у Толика n ($1 \leqslant n \leqslant 10^5$) и первоначальный авторитет Толика a_0 ($-10^9 \leqslant a_0 \leqslant 10^9$). Следующие n строк содержат пары целых чисел a_i и b_i ($-10^9 \leqslant a_i, b_i \leqslant 10^9$).

Формат выходных данных

На первой строке выведите число m — максимальное число друзей, которых может увлечь Толик. На второй строке выведите m чисел — номера друзей в том порядке, в котором их нужно агитировать.

Если решений несколько, выведите любое.

Пример

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

Задача O. Видеонаблюдение

Город-Таз известен нереально высоким уровнем криминала. Полиция не видит вариантов развития ситуации, кроме как усилить меры безопасности. Они хотят установить движущихся 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

Только одно словао – `possible` или `impossible`.

Примеры

basin.in	basin.out
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

Замечание

Проще говоря, проверьте, есть ли в неорграфе независимое множество размера хотя бы k.

Задача P. Верификация моделей

Сережа работает над новым проектом по верификации недетерминированных программных моделей. Первая версия программы будет работать с ациклическими программами. Ациклическая программа в виде, пригодном для верификации, представляется в виде ориентированного ациклического взвешенного графа, в котором есть вершина s, из которой достижимы все остальные.

Результатом верификации ациклической программы является подмножество её рёбер, верификационное дерево. *Верификационным деревом* называется ориентированное корневое дерево с корнем в вершине s, по дугам которого можно добраться из вершины s до любой другой вершины. Изучаемой характеристикой верификационного дерева является его характеристика Бабёнко–Копелиовича (ХБК) — количество единиц в двоичной записи суммы весов дуг, из которых оно состоит. Поскольку у ациклической программы может быть несколько верификационных деревьев, Сережу интересует среднее значение ХБК по всем возможным верификационным деревьям.

Формат входных данных

Первая строка входного файла содержит два целых числа n и m — количество вершин и дуг графа, соответственно ($2 \leqslant n \leqslant 20$, $1 \leqslant m \leqslant 50$). Вершина s имеет номер 1.

Следующие m строк содержат по три целых числа a_i, b_i и c_i — номер вершины, из которой выходит дуга, номер вершины, в которую она входит и веса этой дуги. Веса дуг неотрицательны и не превышают 10^7. В графе нет параллельных дуг. В графе нет циклов.

Формат выходных данных

Выведите в выходной файл одно вещественное число — среднее количество единиц в двоичной записи веса верификационного дерева. Ответ должен отличаться от правильного не более чем на 10^{-6}.

Примеры

model.in	model.out
4 4 1 2 1 1 3 1 2 4 1 3 4 2	1.5

Задача Q. $O2 \,||\, C_{max}$

Дано два станка и n деталей. Каждую деталь нужно обработать на каждом станке. Детали на каждом из станков можно обрабатывать в любом порядке. Каждую деталь можно сперва обработать на первом, потом на втором, а можно наоборот. Единственное ограничение – в каждый момент времени станок может обрабатывать только одну деталь, и, начав, обарабатывать деталь, нужно сперва закончить, а уже потом переходить к следующей.

Время обработки i-й детали на первом станке — a_i, на втором — b_i.

Нужно минимизировать время завершения всех работ.

Формат входных данных

В первой строке дано целое число n ($1 \leqslant n \leqslant 200\,000$) — количество работ. Во второй строке заданы целые числа a_i ($0 < a_i \leqslant 10^9$) — времена, необходимые для обработки деталей на первом станке. В третьей строке заданы целые числа b_i ($0 < b_i \leqslant 10^9$) — времена, необходимые для обработки деталей на втором станке.

Формат выходных данных

В первой строке выведите единственное целое число — минимальное время завершения всех работ. Во второй строке выведите n целых чисел $t_{1,i}$ — момент времени, в который нужно начать обрабатывать i-ю деталь на первом станке. В третьей строке аналогично выведите числа $t_{2,i}$.

Если решений несколько, выведите любое.

Примеры

o2cmax.in	o2cmax.out
3 1 2 3 2 1 3	6 0 1 3 3 5 0