Задача A. Проблема сапожника

Имя входного файла:	`cobbler.in`
Имя выходного файла:	`cobbler.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В некоей воинской части есть сапожник. Рабочий день сапожника длится K минут. Заведующий складом оценивает работу сапожника по количеству починенной обуви, независимо от того, насколько сложный ремонт требовался в каждом случае. Дано n сапог, нуждающихся в починке. Определите, какое максимальное количество из них сапожник сможет починить за один рабочий день.

Формат входного файла

В первой строке вводятся числа K (натуральное, не превышает 1000) и n (натуральное, не превышает 500). Затем идет n чисел — количество минут, которые требуются, чтобы починить i-й сапог (времена — натуральные числа, не превосходят 100).

Формат выходного файла

Выведите одно число — максимальное количество сапог, которые можно починить за один рабочий день.

Примеры

cobbler.in	cobbler.out
10 3 6 2 8	2
3 2 10 20	0

Задача B. Выбор заявок

Имя входного файла:	`request.in`
Имя выходного файла:	`request.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вы прекрасно знаете, что в ЛКШ.Зима 2016 лекции читают лучшие преподаватели мира. К сожалению, лекционных аудиторий у нас не так уж и много, поэтому каждый преподаватель составил список лекций, которые он хочет прочитать ЛКШатам. Чтобы ЛКШата, утром идя на завтрак, увидели расписание лекций, необходимо его составить прямо сейчас. И без вас нам здесь не справиться.

У нас есть список заявок от преподавателей на лекции для одной из аудиторий. Каждая заявка представлена в виде временного интервала $[s_i, f_i)$ — время начала и конца лекции. Лекция считается открытым интервалом, то есть какая-то лекция может начаться в момент окончания другой, без перерыва. Необходимо выбрать из этих заявок такое подмножество, чтобы суммарно выполнить максимальное количество заявок. Учтите, что одновременно в лекционной аудитории, конечно же, может читаться лишь одна лекция.

Формат входного файла

В первой строке вводится натуральное число N, не более 1000 — общее количество заявок на лекции. Затем вводится N строк с описаниями заявок — по два числа в каждом s_i и f_i. Гарантируется, что $s_i < f_i$. Время начала и окончания лекции — натуральные числа, не превышают 1440 (в минутах с начала суток).

Формат выходного файла

Выведите одно число — максимальное количество заявок, которые можно выполнить.

Примеры

request.in	request.out
1 5 10	1
3 1 5 2 3 3 4	2

Задача C. Распечатка условий

Имя входного файла:	`printing.in`
Имя выходного файла:	`printing.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Популярность окружной олимпиады по информатике растёт год от года. При этом организаторы должны заранее распечатать как условия задач, так и другие материалы олимпиады (анкеты, памятки и т.п.). В этом году они оценили объём печатной продукции в N листов.

Фирма, готовая размножить печатные материалы, предлагает следующие финансовые условия. Один лист она печатает за A_1 рублей, 10 листов — за A_2 рублей, 100 листов — за A_3 рублей, 1000 листов — за A_4 рублей, 10000 листов — за A_5 рублей, 100000 листов — за A_6 рублей, 1000000 листов — за A_7 рублей. При этом не гарантируется, что один лист в более крупном заказе обойдется дешевле, чем в более мелком. И даже может оказаться, что для любой партии будет выгодно воспользоваться тарифом для одного листа.

Печать конкретного заказа производится или путем комбинации нескольких тарифов, или путем заказа более крупной партии. Например, 980 листов можно распечатать, заказав печать 9 партий по 100 листов плюс 8 партий по 10 листов, сделав 98 заказов по 10 листов, 980 заказов по 1 листу или заказав печать 1000 (или даже 10000 и более) листов, если это окажется выгоднее. Требуется по заданному объему заказа в листах N определить минимальную сумму денег в рублях, которой будет достаточно для выполнения заказа.

Формат входного файла

На вход программе сначала подается число N ($1 \le N \le 2 \times 10^9$) — количество листов в заказе. В следующих 7 строках ввода находятся натуральные числа $A1, A2, A3, A4, A5, A6, A7$ соответственно ($1 \le A_i \le 10^6$).

Формат выходного файла

Выведите одно число — минимальную сумму денег в рублях, которая нужна для выполнения заказа. Гарантируется, что правильный ответ не будет превышать 2×10^9.

Примеры

printing.in	printing.out
980 1 9 90 900 1000 10000 10000	882
980 1 10 100 1000 900 10000 10000	900

Задача D. Последовательность

Имя входного файла:	`sequence.in`
Имя выходного файла:	`sequence.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Дана последовательность натуральных чисел $a_1, a_2, \ldots, a_n$, и известно, что $a_i \leq i$ для любого $1 \leq i \leq n$. Требуется определить, можно ли разбить элементы последовательности на две части таким образом, что сумма элементов в каждой из частей будет равна половине суммы всех элементов последовательности.

Формат входного файла

В первой строке входного файла находится одно целое число n ($1 \leq n \leq 40\,000$). Во второй строке находится n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq i$).

Формат выходного файла

В первую строку выходного файла выведите количество элементов посделовательности в любой из получившихся двух частей, а во вторую строку через пробел номера этих элементов. Если построить такое разбиение невозможно, выведите -1.

Примеры

sequence.in	sequence.out
3 1 2 3	1 3

Задача E. Алиса и яблоки

Имя входного файла:	`apples.in`
Имя выходного файла:	`apples.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Алисе в стране чудес попались n волшебных яблок. Про каждое яблоко известно, что после того, как его съешь, твой рост сначала уменьшится на a_i сантиметров, а потом увеличится на b_i сантиметров. Алиса очень голодная и хочет съесть все n яблок, но боится, что в какой-то момент ее рост станет равным нулю или еще меньше, и она пропадет совсем. Помогите ей узнать, можно ли съесть яблоки в таком порядке, чтобы в любой момент времени рост Алисы был больше нуля.

Формат входного файла

В первой строке вводятся натуральные числа n и s, не более 1000 — число яблок и начальный рост Алисы. В следующих n строках вводятся пары натуральных чисел a_i, b_i, не больших 1000.

Формат выходного файла

Если яблоки съесть нельзя, выведите число -1. Иначе выведите n чисел — номера яблок, в том порядке, в котором их нужно есть.

Примеры

apples.in	apples.out
3 5 2 3 10 5 5 10	1 3 2
3 5 2 3 10 5 5 6	-1

Задача F. Инверсии и Сережа

Имя входного файла:	`john.in`
Имя выходного файла:	`john.out`
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Недавно Сережа изучил понятие инверсии. Инверсией в последовательности чисел s_k называется пара s_i, s_j, такая что $i < j$ и $s_i > s_j$.

Петя дал Сереже n карточек. На каждой карточке написано два числа: одно красное и одно синее. Сережа хочет применить свои знания об инверсиях к этому набору карточек.

Сережа хочет выложить перед собой карточки в таком порядке, чтобы количество инверсий было как можно меньше. При этом он считает число инверсий между числами одного цвета. Таким образом, он хочет выложить карточки в таком порядке, чтобы если рассмотреть отдельно красные числа в этом порядке и отдельно синие числа в этом порядке, суммарное число инверсий было как можно меньше.

Формат входного файла

Первая строка входного файла содержит число n — число карточек в наборе ($1 \le n \le 100\,000$). Следующие n строк описывают карточки, по одной на строке, i-я строка содержит целые числа r_i и b_i ($1 \le r_i, b_i \le 10^9$) — соответственно красное и синее число на i-й карточке.

Формат выходного файла

Выведите одно число — минимальное возможное число одноцветных инверсий.

Примеры

john.in	john.out
3 10 3 20 2 30 1	3
4 2 2 5 25 2 1 10 9	1

Задача G. Красивые перестановки

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

Будем называть ценностью перестановки $\langle a_1, a_2, \ldots, a_n \rangle$ величину $(a_1a_2 + a_2a_3 + \ldots + a_{n-1}a_n) \bmod r$.

Петя считает число красивым, если оно либо равно 0, либо число его делителей кратно 3. Например, 9 — красивое число (у него три делителя: 1, 3 и 9), а 6 — нет (у него 4 делителя: 1, 2, 3 и 6).

Вам даны n и r, найдите число перестановок, ценность которых является красивым числом.

Формат входного файла

Во входном файле заданы числа n и r ($2 \le n \le 10$, $2 \le r \le 1000$).

Формат выходного файла

Выведите в выходной файл число перестановок, ценность которых является красивым числом.

Пример

beautiful.in	beautiful.out
3 10	2

Комментарий

В примере искомые перестановки — $\langle 1, 3, 2 \rangle$ и $\langle 2, 3, 1 \rangle$

Система оценки:

В этой задаче вы получите 70 баллов, если ваша программа будет правильно работать на тестах, где $n \le 8$. И, кстати, в этой задаче проблема со временем может быть не только из-за питона. Будьте осторожны!

Задача H. Двоичные вектора 2

Имя входного файла:	`vectors2.in`
Имя выходного файла:	`vectors2.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Выведите в выходной файл все двоичные вектора, в которых нет двух единиц подряд.

Формат входного файла

Во входном файле задано число n ($1 \le n \le 16$).

Формат выходного файла

В первой строке выходного файла выведите количество двоичных векторов длины n в которых нет двух единиц подряд. В следующих строках выведите сами эти вектора в лексикографическом порядке по одному в строке.

Пример

vectors2.in	vectors2.out
3	5 000 001 010 100 101

Система оценки:

Злые члены жюри не хотят оценивать частичные решения в этой задаче. Так что только 100.

Задача I. Квадратный корень

Имя входного файла:	`sqroot.in`
Имя выходного файла:	`sqroot.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайта

Введем в рассмотрение так называемые 0-1 матрицы размером 4 на 4. Такая матрица — это квадратная таблица, содержащая 16 чисел $a_{i,j}$ ($i = 1 \dots 4$, $j = 1 \dots 4$), каждое из которых равно 0 или 1.

Произведением двух матриц A и B называется матрица $A \cdot B = C$, элементы которой вычисляются по формуле

$$c_{i,j} = \left(\sum_{k=1}^{4} a_{i,k} b_{k,j} \right) \bmod 2.$$

Квадратным корнем из матрицы A называется 0-1 матрица B, такая что $B \cdot B = A$.

Задана некоторая 0-1 матрица размера 4 на 4. Вычислите ее квадратный корень или установите, что его не существует.

Формат входного файла

Входной файл содержит четыре строки, каждая из которых содержит четыре числа, каждое из этих чисел — либо 0, либо 1, j-е число i-й строки соответствует элементу $a_{i,j}$ заданной матрицы A.

Формат выходного файла

Выведите в выходной файл квадратный корень из заданной матрицы в формате, аналогичном входному файлу. Если квадратного корня не существует — выведите в выходной файл `NO SOLUTION`. Если решений несколько, выведите любое.

Примеры

sqroot.in	sqroot.out
1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1	1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1
0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 0	NO SOLUTION

Система оценки:

Вы уже видели эту задачу, но мало кто ее решил. Решите ее теперь и получите 100 баллов.

Задача J. Останки Юрского периода

Имя входного файла:	`jurassic.in`
Имя выходного файла:	`jurassic.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 мегабайта

Археологи недавно нашли фрагменты динозавра Юрского периода. Археологи решили, что они отправят кости динозавра в музей. Но, к сожалению, кости такие большие, что они не смогли положить их в один ящик. Поэтому они разделили скелет на отдельные кости и отправили каждый из них по-отдельности. Теперь сотрудникам музея предстоит собрать скелет. Для того, чтобы они могли это сделать, археологи отметили точки, в которых кости должны были быть соединены, специальными пометками. А именно — в каждой точке, где соединялись две кости, археологи написали на каждой из них одинаковые заглавные латинские буквы.

Однако пока археологи разбирали и упаковывали скелет, были обнаружены еще кости и они тоже были отправлены в музей. Причем они тоже могли быть соединены друг с другом, поэтому на них также могли быть пометки. Более того, археологи всегда использовали одинаковые пометки в точках на костях, которые должны были быть соединены, но они могли использовать одну и ту же пометку для различных соединений. Правда на каждой кости было не более одной пометки конкретной буквой.

Теперь работники музея пытаются разобраться с этой ситуацией и найти хотя бы какое-нибудь теоретически возможное множество костей исходного скелета динозавра. А именно, они хотят найти множество костей, которое удовлетворяет следующим условиям.

- Кости, помеченные некоторой буквой, можно попарно соединить друг с другом. Иначе говоря, каждая пометка встречается четное число раз.
- Число костей в наборе максимально.

Формат входного файла

Первая строка входного файла содержит N — число костей ($1 \le N \le 24$). Следующие N строк содержат описание костей: каждая строка содержит непустую последовательность различных заглавных букв латинского алфавита — метки на костях.

Формат выходного файла

На первой строке выходного файла выведите L — максимальное вомзожное число костей, из которых можно собрать скелет. Вторая строка должна содержать L чисел — номера костей. Кости пронумерованы от 1 до N в порядке, в котором они заданы во входном файле.

Пример

jurassic.in	jurassic.out
6 ABD EG GE ABE AC BCD	5 1 2 3 5 6
1 ABC	0

Система оценки:

Первая группа содержит тесты, в которых $N \le 12$. Она оценивается в 60 баллов.

Вторая группа содержит тесты, в которых $N \le 24$. Она оценивается в 40 баллов.

Задача K. Сокровища

Имя входного файла:	`dowry.in`
Имя выходного файла:	`dowry.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Дочь короля Флатландии собирается выйти за прекрасного принца. Принц хочет подарить принцессе сокровища, но он не уверен какие именно бриллианты из своей коллекции выбрать.

В коллекции принца n бриллиантов, каждый характеризуется весом w_i и стоимостью v_i. Принц хочет подарить наиболее дорогие бриллианты, однако король умен и не примет бриллиантов суммарного веса больше R. С другой стороны, принц будет считать себя жадным всю оставшуюся жизнь, если подарит бриллиантов суммарным весом меньше L.

Помогите принцу выбрать набор бриллиантов наибольшей суммарной стоимости, чтобы суммарный вес был в отрезке $[L, R]$.

Формат входного файла

Первая строка содержит число n ($1 \leqslant n \leqslant 32$), L и R ($0 \leqslant L \leqslant R \leqslant 10^{18}$). Следующие n строк описывают бриллианты и содержит по два числа — вес и стоимость соответствующего бриллианта ($1 \leqslant w_i, v_i \leqslant 10^{15}$).

Формат выходного файла

Первая строка вывода должна содержать k — количество бриллиантов, которые нужно подарить принцессе. Вторая строка должна содержать номера даримых бриллиантов.

Бриллианты нумеруются от 1 до n в порядке появление во входных данных.

Если составить подарок принцессе невозможно, то выведите 0 в первой строке вывода.

Примеры

dowry.in	dowry.out
3 6 8 3 10 7 3 8 2	1 2

Система оценки:

В этой задаче вы получите 100 баллов, если ваша программа будет правильно работать на всех тестах. Такие дела.

Задача L. Set

Имя входного файла:	`set.in`
Имя выходного файла:	`set.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Реализуйте множество с использованием хеш таблицы.

Формат входного файла

Входной файл содержит описание операций. В каждой строке находится одна из следующих операций:

- `insert` x — добавить элемент x в множество. Если элемент уже есть в множестве, то ничего делать не надо.
- `delete` x — удалить элемент x. Если элемента x нет, то ничего делать не надо.
- `exists` x — если ключ x есть в множестве выведите «true», если нет «false».

В множество помещаются и извлекаются только целые числа, не превышающие по модулю 10^9.

Формат выходного файла

Выведите последовательно результат выполнения всех операций `exists`. Следуйте формату выходного файла из примера.

Пример

set.in	set.out
insert 2 insert 5 insert 3 exists 2 exists 4 insert 2 delete 2 exists 2	true false false

Задача M. Map

Имя входного файла:	`map.in`
Имя выходного файла:	`map.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Реализуйте ассоциативный массив с использованием хеш таблицы.

Формат входного файла

Входной файл содержит описание операций. В каждой строке находится одна из следующих операций:

- `put` x y — поставить в соответствие ключу x значение y. Если ключ уже есть, то значение необходимо изменить.
- `delete` x — удалить ключ x. Если элемента x нет, то ничего делать не надо.
- `get` x — если ключ x есть в ассоциативном массиве, то выведите соответствующее ему значение, иначе выведите «none».

Ключи и значения — строки из латинских букв длинной не более 20 символов.

Формат выходного файла

Выведите последовательно результат выполнения всех операций `get`. Следуйте формату выходного файла из примера.

Пример

map.in	map.out
put hello privet put bye poka get hello get bye delete hello get hello	privet poka none

Задача N. LinkedMap

Имя входного файла:	`linkedmap.in`
Имя выходного файла:	`linkedmap.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Реализуйте прошитый ассоциативный массив с использованием хеш таблицы.

Формат входного файла

Входной файл содержит описание операций. В каждой строке находится одна из следующих операций:

- `put` x y — поставить в соответствие ключу x значение y. Если элемент уже есть, то значение необходимо изменить.
- `delete` x — удалить ключ x. Если элемента x нет, то ничего делать не надо.
- `get` x — если ключ x есть в множестве выведите соответствующее ему значение, если нет выведите «none».
- `prev` x — вывести значение соответствующее ключу находящемуся в ассоциативном массиве, который был вставлен позже всех, но до x или «none», если такого нет или в массиве нет x.
- `next` x — вывести значение соответствующее ключу находящемуся в ассоциативном массиве, который был вставлен раньше всех, но после x или «none», если такого нет или в массиве нет x.

Ключи и значения — строки из латинских букв длинной не более 20 символов.

Формат выходного файла

Выведите последовательно результат выполнения всех операций `get`, `prev`, `next`. Следуйте формату выходного файла из примера.

Пример

linkedmap.in	linkedmap.out
put zero a put one b put two c put three d put four e get two prev two next two delete one delete three get two prev two next two next four	c b d c a e none

Задача O. MultiMap

Имя входного файла:	`multimap.in`
Имя выходного файла:	`multimap.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Реализуйте множественное отображение с использованием хеш таблиц.

Формат входного файла

Входной файл содержит описание операций. В каждой строке находится одна из следующих операций:

- `put` x y — добавить пару (x, y). Если пара уже есть, то второй раз её добавлять не надо.
- `delete` x y — удалить пару (x, y). Если пары нет, то ничего делать не надо.
- `deleteall` x — удалить все пары с первым элементом x.
- `get` x — вывести количество пар с первым элементом x, а затем вторые элементы всех этих пар в произвольном порядке.

Ключи и значения — строки из латинских букв длинной не более 20 символов.

Формат выходного файла

Выведите последовательно результат выполнения всех операций `get`. Следуйте формату выходного файла из примера. Гарантируется, что размер выходного файла не превысит 10 мегабайт.

Пример

multimap.in	multimap.out
put a a put a b put a c get a delete a b get a deleteall a get a	3 b c a 2 c a 0