Задача A. Автомойка

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

Петя занялся бизнесом — он открыл автомойку. Петина автомойка предоставляет два типа услуг: мойка автомобиля и химчистка салона. Недавно Петя выиграл тендер на мойку и химчистку n машин правительства Флатландии.

Изучив вопрос, Петя выяснил, что на мойку i-й машины требуется a_i минут, а на химчистку ее салона — b_i минут. Мойка и химчистка не могут проводится одновременно. Петина мойка пока довольно небольшая, поэтому у него работает всего два человека — Ваня и Гриша. Ваня занимается мойкой машин, а Гриша — химчисткой салона. К сожалению, ни Ваня, ни Гриша не могут одновременно заниматься двумя машинами — в каждый момент времени каждый работает не более чем с одной. При этом автомобили могут обрабатываться в произвольном порядке и каждая конкретная машина может быть либо сначала помыта, а затем почищена, либо наоборот.

Теперь Петю интересует, какое минимальное время ему потребуется, чтобы помыть и почистить все машины. Помогите ему составить расписание для работников, чтобы добиться оптимального результата.

Формат входного файла

Первая строка входного файла содержит число n ($1 \le n \le 10\,000$). Следующие n строк содержит по два целых числа: a_i и b_i ($1 \le a_i, b_i \le 10^5$).

Формат выходного файла

Первая строка выходного файла должна содержать t — количество минут от момента начала работы с первой машиной до момента окончания работ с последней. Следующие n строк должны содержать по два числа — время от начала процесса до начала мойки данной машины и время от начала процесса до начала химчистки салона в ней.

Если оптимальных решений несколько, можно вывести любое.

Пример

carwash.in	carwash.out
6 10 6 7 9 3 8 1 2 12 7 6 6	39 11 26 4 17 0 7 3 15 27 0 21 32

Задача B. Сегодня твоя очередь готовить!

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

Группа из k студентов Университета Вкусной и Здоровой Пищи решила, что каждый день семестра один из них будет готовить обед для всех. Известно, что семестр в этом университете содержит n дней.

Чтобы все было честно, решено было, что каждый из студентов должен готовить обед хотя бы один раз в течение семестра. Теперь им интересно, сколько различных способов существует для определения того, кто будет готовить в какой день. Поскольку лекции по математике поварам не читают, они не справляются. Помогите им!

Формат входного файла

Входной файл содержит два целых числа k и n ($1 \le k \le n \le 100$).

Формат выходного файла

Выведите одно число — искомое количество способов.

Пример

dinner.in	dinner.out
2 3	6

В примере есть шесть следующих способов: $(1, 1, 2)$, $(1, 2, 1)$, $(2, 1, 1)$, $(1, 2, 2)$, $(2, 1, 2)$, $(2, 2, 1)$.

Задача C. Система неравенств

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

Задана система неравенств, в которой каждое неравенство имеет вид `x > y`, `x > a`, `x < a` где x и y — некоторые переменные, а a — некоторое целое число от 0 до 100. В этой системе неравенств используются только переменные x, y и z.

Ваша задача — найти число решений этой системы, в которых x, y и z являются целыми числами от 0 до 100.

Формат входного файла

Первая строка входного файла содержит целое число n ($0 \le n \le 50$) — число неравенств в системе. Каждая из последующих строк содержит одно из неравенств.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

ineq.in	ineq.out

`6`
`x > 2`
`x < 5`
`y > 2`
`y < 5`
`z > 2`
`z < 5`
	`8`

`1` `x > y`	`510050`

Задача D. k-я порядковая статистика

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

Задано n целых чисел $a_1, a_2, \ldots, a_n$. Назовем k-й порядковой статистикой среди них такое число $x = a_j$ (для некоторого j), что среди $a_1, \ldots, a_n$ существует ровно $k - 1$ число, меньшее x.

Напишите программу, которая по заданному набору чисел находит k-ю порядковую статистку в нем.

Формат входного файла

Первая строка входного файла содержит три целых числа n, k и m ($1 \le n \le 30\,000\,000$, $1 \le k \le n$, $min(n, 2) \le m \le \min(n, 1000)$). Вторая строка содержит три целых числа a, b и c ($-2^{31} \le a, b, c \le 2^{31} - 1$). Третья строка содержит m целых чисел: $a_1, a_2, \ldots, a_m$ ($-2^{31} \le a_i \le 2^{31} - 1$).

Остальные элементы могут быть вычислены по формуле: $a_i = f(a \cdot a_{i-2} + b \cdot a_{i-1} + c)$. Здесь $f(y)$ возвращает такое число z, где $-2^{31} \le z \le 2^{31} - 1$, что $y - z$ делится без остатка на 2^{32}.

Формат выходного файла

В выходной файл выведите k-ю порядковую статистку среди $a_1, a_2, \ldots, a_n$.

Примеры

kth.in	kth.out
3 2 3 0 0 0 1 2 3	2
1000 10 3 1 1 1 1 2 3	-2081387834

Задача E. Снова опоздал

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

Сережа снова опоздал в школу на урок информатики. Его учитель, Владимир Иванович Пряников, сказал, что все дело в том, что он неправильно выбирает дорогу из школы до дома, всегда ходит каким-то странным путем. В качестве наказания за опоздания Владимир Иванович дал Сереже задание нарисовать карту города, отметить на ней время прохода по каждой улице и найти кратчайший путь от дома до школы. И убедиться, наконец, что путь, которым он обычно ходит — не оптимальный!

Сережа представил карту города в виде неориентированного графа, вершинами графа он сделал перекрестки, а ребрами — фрагменты улиц. Около каждого ребра он написал время прохода по нему, а также отметил свой обычный путь в школу. Да, путь не оптимальный. Но такой интересный! Надо доказать учителю, что на самом деле путь оптимален. Но как это сделать...

После некоторых размышлений Сережа решил, что он изменит некоторые времена прохода по улицам так, чтобы его путь стал оптимальным. Разумеется, если изменить время слишком сильно, учитель может заподозрить неладное. Поэтому надо, чтобы максимальное изменение было как можно меньше. Кроме того, вряд ли учитель поверит, что Сережа может пройти от одного перекрестка до другого быстрее чем за 1 минуту.

Помогите Сереже защитить его путь в школу.

Формат входного файла

Первая строка входного файла содержит n и m — количество перекрестков и количество улиц в городе ($2 \le n \le 1000$, $1 \le m \le 20\,000$). Следующие m строк описывают улицы: каждая улица описывается тремя целыми числами — номерами перекрестков, которые она соединяет и временем, которое требуется для прохода по ней (все времена целые и лежат в диапазоне от 1 до 10^4, включительно). По каждой улице можно ходить в любом направлении.

Следующая строка содержит число l — количество улиц, по которым Сережа проходит по пути из дома до школы. Следующая строка содержит l чисел — номера улиц в том порядке, в котором Сережа по ним проходит. Улицы пронумерованы от 1 до m в том порядке, в котором они заданы во входном файле. Сережа не проходит через один перекресток (и следовательно одну улицу) дважды.

Дом Сережи находится около перекрестка номер 1, а школа — около перкрестка номер n.

Формат выходного файла

Первая строка выходного файла должна содержать v — минимальную величину, такую, что изменив время прохода по каждой улице не больше чем на v, Сережа сделает свой путь одним из возможных кратчайших путей из дома в школу. Вторая строка должна содержать m вещественных чисел — новые времена прохода по улицам. Выводите числа с точностью не менее 10^{-5}.

Пример

lateagain.in	lateagain.out
4 4 1 2 1 2 4 2 1 3 1 3 4 3 2 3 4	0.333333333 1.333333333 2.333333333 1 2.666666667

Задача F. Снукер

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

На турнире по снукеру (одна из разновидностей бильярда) очень любят подсчитывать различную статистику. Наиболее популярным является подсчет количества сотенных серий — подходов к столу, когда игрок набирает более ста очков.

У организаторов накопилось порядочное количество данных о таких сериях и теперь они просят вас составить таблицу результатов по этим данным. Для этого был разработан метод сравнения двух игроков.

Для начала сравниваются максимальные по количеству очков серии этих игроков, и тот, у кого очков было получено больше, считается лучше другого игрока. Если же максимальные серии совпадают, то лучше считается игрок, вторая по величине серия которого больше. Если же и они совпадают, то сравнеие происходит по третьим сериям и т.д. Если же статистики игроков полностью совпадают, то лучше считается игрок, сделавший сотенную серию первым.

Формат входного файла

В первой строке одно целое число n — количество зафиксированных на турнире сотенных серий ($1 \le n \le 1000$). Далее следуют n строк вида “a_i s_i”, где a_i — количество очков набранных в i-ой сотенной серии ($100 \le a_i \le 155$), а s_i — имя игрока совершившего серию, состоящие из не более чем 30 заглавных и строчных латинских букв, а также пробелов. Имя игрока не может начинаться с пробела или им заканчиваться. Строчные и заглавные буквы считаются различными.

Формат выходного файла

В выходной файл следует вывести таблицу результатов. В первой строке должна быть статистика по лучшему игроку, во второй — по второму и т.д. Статистика по каждому игроку должна содержать имя игрока и далее все его сотенные серии, перечисленные в порядке невозрастания.

Примеры

snooker.in	snooker.out
5 140 Allister Carter 147 Andrew Higginson 147 Allister Carter 140 Andrew Higginson 144 Neil Robertson	Allister Carter 147 140 Andrew Higginson 147 140 Neil Robertson 144

Задача G. Поиск подстроки

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

У Андрюши есть маленькая сестричка Аня. Она любит писать сообщения своему другу Гоше. Она хочет, чтобы никто не мог прочитать ее сообщения, поэтому она шифрует их подстановочным шифром. Подстановочный шифр заменяет каждый символ в сообщении на какой-либо еще, при этом равные символы заменяются на равные, а различные — на различные.

Например, при шифровании с помощью подстановочного шифра $e \to a$, $l \to b$, $o \to w$, $v \to c$ слово "*love*" оказывается зашифровано как "*bwca*".

Андрюша недавно перехватил одно из Аниных сообщений t и хочет выяснить, встречается ли там текст p. А именно, он хочет найти все позиции i, такие что существует подстановочный шифр, такой что $t[i..i+|p|-1]$ представляет собой зашифрованную версию p. Будем называть такие позиции *потенциальными вхождениями* p в t.

Помогите Андрюше найти все потенциальные вхождения p в t.

Формат входного файла

Первая строка входного файла содержит t. Вторая строка входного файла содержит p. Каждая строка состоит из символов с ASCII кодами от 33 до 126. Длина p не превышает длины t. Длина t не превышает 200 000. Обе строки непусты.

Формат выходного файла

Первая строка выходного файла должна содержать k — количество потенциальных вхождений p в t. Вторая строка должна содержать k целых чисел — позиции потенциальных вхождений. Позиции в строке нумеруются, начиная с 1. Позиции следует перечислить в возрастающем порядке.

Пример

substring.in	substring.out
abacabadabacaba aba	7 1 3 5 7 9 11 13
abacabadabacaba love	0

Задача H. Два шланга

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

Иван Петрович — заядлый садовод. Он настолько любит свои посевы, что для их полива использует два шланга, а не один, как это делает его сосед Петр Иванович. Естественным образом эти шланги часто запутываются. Вот и сейчас они, естественно, переплелись.

Чтобы распутать получившуюся конфигурацию шлангов, Иван Петрович вытягивает их вдоль дорожки таким образом, что, идя вдоль дорожки, всё время можно наблюдать два параллельно идущих шланга, но в некоторых местах можно наблюдать, как шланги "меняются местами", то есть один шланг проходит над другим. Иван Петрович классифицировал два вида таких наложений: 1 — дальний шланг проходит сверху, 2 — ближний проходит сверху.

Теперь Ивана Петровича интересует вопрос, если он попросит Петра Ивановича подержать шланги с одного конца, а сам потянет за два других конца, то распутаются ли шланги.

Формат входного файла

В первой строке целое число n — количество наложений шлангов ($1 \le n \le 1000$). Во второй строке n чисел — виды наложений в порядке прохода по дорожке от Петра Ивановича в Ивану Петровичу.

Формат выходного файла

В выходной файл выведите «`YES`», если данным образом удастся распутать шланги, и «`NO`» иначе.

Примеры

twohoses.in	twohoses.out
3 1 1 1	NO
2 1 2	YES

Задача I. Видео по запросу

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

Антон владеет компанией, которая предоставляет сервис «видео по запросу» через интернет. Пользователь выбирает видео и может смотреть его прямо через браузер. К сожалению, у многих пользователей слишком маленькая пропускная способность канала, поэтому они не могут смотреть видео в режиме реального времени. Федя — один из таких пользователей.

Пропускная способность Фединого канала позволяет ему скачивать d байт в секунду. Но видеопоток файла, который он хотел бы посмотреть — b байт в секунду. Видеоплеер, который использует Федя, позволяет буферизовать видеоданные, смягчая проблему недостаточной пропускной способности канала. Если размер буфера установлен в s, плеер ведет себя следующим образом.

Сначала он скачивает s байтов видеоданных. Затем он начинает проигрывать видео, скачивая новые фрагменты видео в буфер в фоновом режиме. Когда буфер заканчивается, проигрывание видео останавливается и буфер снова полностью заполняется. Для простоты будем считать информацию и время непрерывными в этой задаче (то есть будем рассматривать произвольные доли байта и секунды).

Если оставшегося материала недостаточно для заполнения буфера, плеер скачивает весь оставшийся материал.

Федя хотел бы просмотреть программу длиной t секунд. Он понимает, что в процессе просмотра будут возникать паузы, но хотел бы, чтобы каждая пауза, включая ту, которая будет перед началом проигрывания, не превышала p секунд. Теперь он хочет установить размер буфера, чтобы количество пауз было минимально. Помогите ему.

Формат входного файла

Входной файл содержит четыре целых числа: d, b, t и p ($1024 \le d \le 1073741824$, $d < b \le 1073741824$, $1 \le t \le 14400$, $1 \le p \le 3600$).

Формат выходного файла

Выведите одно целое число — размер буфера, которые следует установить. Если несколько вариантов приводят к одному и тому же количеству пауз, выведите минимальный размер буфера.

Пример

video.in	video.out
1024 2048 100 20	20480
1024 2048 90 20	18432
1024 1025 1 1	1