

Задача А. Уральские бифштексы

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

После личного первенства довольные, но голодные программисты зашли в ресторан «Уральские бифштексы» и заказали себе n фирменных бифштексов. Для того чтобы приготовить бифштекс, повар должен прожарить каждую из его сторон на сковороде в течение одной минуты.

К сожалению, у повара есть только одна сковорода, и на ней можно одновременно жарить не более k бифштексов. Определите, сколько минут понадобится повару, чтобы выполнить заказ программистов.

Формат входного файла

В единственной строке через пробел записаны целые числа n и k ($1 \leq n, k \leq 1000$).

Формат выходного файла

Выведите минимальное количество минут, за которое повар сможет приготовить n бифштексов.

Пример

<code>input.txt</code>	<code>output.txt</code>
3 2	3

Задача В. Биатлон

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В биатлонных гонках с отдельным стартом участники стартуют по очереди, с интервалом в 30 секунд, поэтому участник, финишировавший первым, не обязательно будет первым в итоговой таблице результатов. Например, если спортсмен, стартовавший вторым, придёт к финишу на 25 секунд позже стартовавшего первым, то это значит, что он прошёл дистанцию на 5 секунд быстрее, а значит, должен располагаться в итоговой таблице выше.

До Зимних Олимпийских игр 2014 года, которые, как известно, пройдут в городе Екатеринозаводске, осталось всего три года. Строительство трассы для проведения соревнований по биатлону подходит к концу, а стрельбище и трибуны для зрителей уже построены.

Рядом с трибунами планируется установить электронное табло, на котором во время гонки будет гореть фамилия биатлониста, финишировавшего с лучшим на текущий момент результатом. Вам было поручено разработать программу, определяющую этого биатлониста. В качестве исходных данных для тестирования программы вы взяли итоговый протокол одной из гонок недавно завершившегося чемпионата мира по биатлону. Протокол содержит фамилии биатлонистов с указанием времени, за которое они преодолели дистанцию. Фамилии упорядочены по времени старта. Для того чтобы понять, верно ли работает ваша программа, нужно определить всех спортсменов, фамилии которых должны были появиться на табло.

Формат входного файла

В первой строке записано целое число n — количество участников гонки ($1 \leq n \leq 100$). В i -й из следующих n строк через пробел записаны фамилия спортсмена, стартовавшего i -м, и время, за которое он прошёл трассу, в формате «`mm:ss.d`», заданное с точностью до десятых секунды. Гарантируется, что никакие два участника не финишировали одновременно и никакие два участника не показали одинаковое время. Фамилия участника — непустая строка, состоящая из английских букв, длиной не более 20. Первая буква фамилии — заглавная, а все остальные — строчные. Фамилии всех участников различны.

Формат выходного файла

В первой строке выведите количество участников, которые были лидерами гонки на момент их финиша. Далее выведите фамилии этих участников в лексикографическом порядке, по одной в строке.

Пример

<code>input.txt</code>	<code>output.txt</code>
6	3
Zaitseva 21:38.2	Hauswald
Hauswald 21:21.0	Wilhelm
Boulygina 22:04.4	Zaitseva
Henkel 22:06.1	
Wilhelm 21:11.1	
Jonsson 22:05.8	

Задача С. Война Уго II

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Славный король Уго II объявил войну. Священную, победоносную, малокровную, но разрушительную!

Сразу же после объявления войны король начал собирать армию — он решил отправить всем своим непосредственным вассалам приказ о наборе рекрутов. Те, в свою очередь, перешлют приказ своим вассалам, и так далее. Таким образом, даже самый захудалый из дворян королевства окажется вовлечён в подготовку к войне.

Как только дворянин, не имеющий вассалов, получает приказ, он тут же начинает сбор войск и выступает в поход через несколько дней. Если же у дворянина есть вассалы, он ждёт, пока под его знамёнами не окажутся войска по крайней мере $x\%$ из его непосредственных вассалов, затем собирает свои собственные войска и тоже выступает в поход. Славный король Уго II отправится воевать, как только к войне будут готовы хотя бы $x\%$ его непосредственных вассалов.

Число x Уго II указывает в своём приказе о наборе рекрутов. Для успешной кампании королю необходимо собрать как можно больше воинов. С другой стороны, если собирать войска дольше t дней, неприятель может пронюхать о готовящемся вторжении и нанести удар первым! Помогите Уго II найти максимально возможное значение числа x .

Формат входного файла

В первой строке записаны целые числа n и t — количество дворян в королевстве и максимальное количество дней, которое можно потратить на сбор армии ($1 \leq n \leq 10^4$; $0 \leq t \leq 10^6$). Дворяне занумерованы числами от 1 до n , король Уго II имеет номер 1. В следующей $n - 1$ строке описаны дворяне со второго по n -й, по одному в строке. В каждой из этих строк записаны целые числа p_i и t_i , где p_i — номер сюзерена, в подчинении которого находится i -й вассал, а t_i — количество дней, которое понадобится i -му вассалу на сбор своих войск ($1 \leq p_i \leq n$; $0 \leq t_i \leq 100$).

Формат выходного файла

Выведите максимально возможное значение x с абсолютной погрешностью не более 10^{-4} .

Пример

input.txt	output.txt
6 3 1 2 2 2 2 1 1 2 1 4	50.00000000

Задача D. Идеальный газ

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Многие из вас знакомы с универсальным способом решения простых задач по физике: нужно найти в учебнике равенство, в котором известны значения всех величин, кроме одной, подставить в это равенство числа и вычислить неизвестную величину.

Эта задача настолько проста, что вам сразу известно, что для её решения нужно использовать уравнение состояния идеального газа Клапейрона — Менделеева. Данное уравнение связывает давление идеального газа p , количество вещества n , занимаемый газом объём V и его температуру T . Вы должны по трём из этих величин определить четвёртую. Учтите, что температура газа и занимаемый им объём всегда должны быть положительными.

Формат входного файла

Каждая из трёх строк входных данных имеет вид « $X = \text{value}$ », где X — обозначение физической величины, а value — неотрицательное целое число, не превосходящее 1000. Три строки задают значения трёх разных величин. Давление задаётся в паскалях, количество вещества — в молях, объём — в кубических метрах, температура — в кельвинах. Гарантируется, что температура и объём положительны. Универсальную газовую постоянную R следует считать равной $8.314 \text{ Дж}/(\text{моль} \cdot \text{К})$.

Формат выходного файла

Если входные данные противоречивы, выведите в единственной строке «`error`». Если значение неизвестной величины X можно определить однозначно, выведите его в формате « $X = \text{value}$ », с абсолютной или относительной погрешностью не более 10^{-6} . Если же значение X нельзя определить однозначно, в единственной строке выведите «`undefined`».

Пример

<code>input.txt</code>	<code>output.txt</code>
<code>p = 1</code> <code>n = 1</code> <code>V = 1</code>	<code>T = 0.120279</code>

Напоминаем, что $\text{Па} = \text{Н}/\text{м}^2$, а $\text{Дж} = \text{Н} \cdot \text{м}$.

Задача Е. Ифрит-бомбардировки

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

За шесть с половиной лет, прошедших после первых экспериментальных джинн-бомбардировок, Питирим Шварц достиг значительных успехов. Его новейшая разработка — бутылка с ифритом — предназначена специально для бомбардировки крупных городов. С помощью нескольких таких бутылок можно, например, стереть с лица земли все n крупных городов соседнего государства. . . Увы, бутылок с ифритами у Питирима совсем немного, поэтому он поручил программисту Привалову вычислить минимальное количество бутылок, необходимое для уничтожения всех n городов.

Для удобства прицеливания планируется сбрасывать бутылки только непосредственно на города. Ифрит разрушает все города, находящиеся на расстоянии не более r от точки падения бутылки. Радиус поражения столь велик, что города можно считать точками на плоскости.

Формат входного файла

В первой строке записаны целые числа n и r ($1 \leq n \leq 30$; $1 \leq r \leq 1\,000$). В i -й из следующих n строк записаны координаты i -го города — два целых числа в пределах от 0 до 1 000.

Формат выходного файла

В первой строке выведите минимальное количество бутылок с ифритами, необходимых для разрушения всех городов. В следующей строке через пробел выведите номера городов, на которые нужно сбросить эти бутылки.

Пример

<code>input.txt</code>	<code>output.txt</code>
4 50	3
10 10	1 3 4
500 500	
501 501	
999 999	

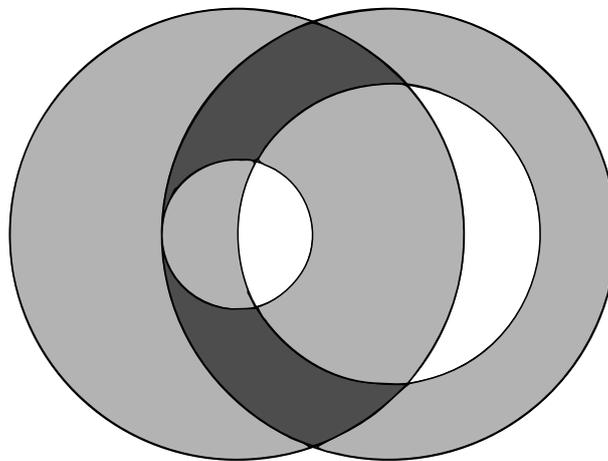
Задача F. Ифрит-бомбардировки 2

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Первое же применение бутылок с ифритами вызвало массовые протесты со стороны мирового сообщества. ЮНЕСКО заявила, что при ифрит-бомбардировках полностью уничтожаются исторические центры крупных городов, обладающие огромной культурной значимостью.

Питирим Шварц смог подкорректировать поведение ифритов таким образом, чтобы центры городов оставались невредимыми. Теперь область поражения бутылки представляет собой не круг, а кольцо — ифриты уничтожают всё на расстоянии не меньше r и не больше R от места падения бутылки.

Чтобы оценить эффективность ковровых ифрит-бомбардировок, Питирим хочет для начала вычислить общую площадь поражения при сбрасывании двух бутылок с ифритами.



Формат входного файла

В единственной строке записаны целые числа d , r_1 , R_1 , r_2 , R_2 — расстояние между точками падения бутылок, внутренний и внешний радиус области поражения первой бутылки, внутренний и внешний радиус области поражения второй бутылки, соответственно ($0 \leq d \leq 15\,000$; $1 \leq r_i < R_i \leq 15\,000$).

Формат выходного файла

Выведите общую площадь поражения с абсолютной или относительной погрешностью не более 10^{-6} .

Пример

<code>input.txt</code>	<code>output.txt</code>
200 100 300 200 300	353992.933435

Задача G. Минное поле

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Во время выполнения задания разведгруппа из n человек переходит минные заграждения противника. Поскольку миноискатель у группы всего один, перемещаться решено следующим образом: в сторону противника переходят парой, затем один человек возвращается, чтобы принести оставшейся группе миноискатель.

Каждый человек переходит минные заграждения со своей скоростью. Скорость пары определяется скоростью более медленного её члена.

Найдите минимальное время, за которое вся группа сможет преодолеть минные заграждения.

Формат входного файла

В первой строке записано целое число n ($2 \leq n \leq 100$). В i -й из следующих n строк записано положительное целое число, не превосходящее 600 — время, необходимое i -му члену группы на преодоление минных заграждений.

Формат выходного файла

Выведите минимальное суммарное время, которое потребуется разведгруппе для преодоления заграждений противника.

Пример

<code>input.txt</code>	<code>output.txt</code>
4 1 10 5 2	17

Задача Н. Войны туземцев

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Туземные племена островного государства, расположенного в Тихом океане между 35 и 36 градусами южной широты, часто воюют друг с другом. Недавно учёные узнали, что туземцы описывают все междоусобные конфликты в специальной книге, и даже смогли эту книгу прочитать.

К сожалению, у туземцев нет календаря, поэтому они запоминают сутки по количеству звёзд, которые были видны на небе ночью. Для каждой прошедшей междоусобицы туземцы записывали в книгу три числа: x — количество звёзд на небе в первые сутки конфликта, y — количество звёзд на небе в последние сутки конфликта и d — продолжительность конфликта в сутках, включая первые и последние сутки.

Учёные смогли вычислить, сколько звёзд было видно туземцам на небе в каждые из n последних суток. Помогите им определить по этой информации, когда туземцы могли воевать друг с другом.

Формат входного файла

В первой строке записано целое число n ($2 \leq n \leq 10^5$). Во второй строке через пробел записаны целые числа a_1, \dots, a_n , где a_i — количество звёзд, которые туземцы видели на небе i суток назад ($0 \leq a_i \leq 2 \cdot 10^8$). В третьей строке записано целое число m — количество описанных в книге междоусобиц ($1 \leq m \leq 10^5$). В i -й из следующих m строк записаны целые числа x_i, y_i и d_i — описание i -й междоусобицы ($0 \leq x_i, y_i \leq 2 \cdot 10^8$; $2 \leq d_i \leq 50$).

Формат выходного файла

Выведите строку длины n , состоящую из нулей и единиц. На i -й позиции должна стоять единица, если i суток назад туземцы могли воевать.

Пример

input.txt	output.txt
6	011110
6 5 4 3 2 1	
3	
2 4 3	
4 5 2	
1 6 5	

Задача I. Приближение прогрессией

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Дана последовательность целых чисел $\{a_1, \dots, a_n\}$. Найдите арифметическую прогрессию $\{b_1, \dots, b_n\}$, для которой величина $\sum_{i=1}^n (a_i - b_i)^2$ принимает минимально возможное значение. При этом элементы прогрессии не обязательно должны быть целыми.

Формат входного файла

В первой строке записано целое число n — количество элементов в последовательности ($2 \leq n \leq 10^4$). Во второй строке записаны целые числа a_1, \dots, a_n , не превосходящие по модулю 10^4 .

Формат выходного файла

В единственной строке выведите через пробел два числа — первый член искомой арифметической прогрессии и её разность, с абсолютной или относительной погрешностью не более 10^{-6} . Гарантируется, что для всех входных данных ответ определяется однозначно.

Примеры

input.txt	output.txt
4 0 6 10 15	0.400 4.900
4 -2 -2 -2 -2	-2 0

Задача J. Таблицы маршрутизации

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Чтобы управлять движением сетевых пакетов, используются специальные устройства — маршрутизаторы. Поведение маршрутизатора определяется таблицей маршрутизации. Эта таблица состоит из нескольких строк, каждая из которых содержит IP-адрес сети назначения d , маску m и IP-адрес шлюза g . Например, строка «192.168.24.0 255.255.255.0 192.168.14.1» означает, что пакет, адресованный в сеть 192.168.24.0 с маской 255.255.255.0, нужно отправить на шлюз 192.168.14.1.

IP-адрес представляет собой 32-битное число. Для удобства он разбивается на четыре байта, каждый из них записывается в десятичном виде, а между байтами ставятся точки. Так, IP-адрес 11000000101010000001100000000000 записывается как 192.168.24.0. Маски имеют такой же вид, при этом в двоичном представлении маски сначала идут только единицы, а потом — только нули.

Когда на маршрутизатор приходит пакет, отправленный на адрес a , маршрутизатор находит те строки таблицы, для которых выполняется условие $d \text{ and } m = a \text{ and } m$ (and — операция побитового «и»). Затем он выбирает из них строку, количество единичных битов в маске которой максимально, и отправляет пакет на записанный в этой строке шлюз. Гарантируется, что для любого адреса назначения таких строк всегда будет не более одной.

Назовём две таблицы маршрутизации *эквивалентными*, если при их использовании пакет с любым адресом назначения будет отправлен на один и тот же шлюз (либо не отправлен ни на какой шлюз). Следующие таблицы маршрутизации эквивалентны:

192.168.0.0 255.255.255.0 192.168.14.1	192.168.0.0 255.255.252.0 192.168.14.1
192.168.1.0 255.255.255.0 192.168.14.1	192.168.2.0 255.255.254.0 192.168.14.2
192.168.2.0 255.255.255.0 192.168.14.2	
192.168.3.0 255.255.255.0 192.168.14.2	

Напишите программу для сравнения двух таблиц маршрутизации.

Формат входного файла

В первой строке записано целое число — количество строк в первой таблице маршрутизации. Далее идёт эта таблица в описанном выше формате. Затем аналогичным образом описывается вторая таблица маршрутизации. Общее количество строк в таблицах не превосходит 65 536.

Формат выходного файла

Если таблицы эквивалентны, выведите «YES», в противном случае выведите «NO».

Примеры

input.txt	output.txt
4 192.168.0.0 255.255.255.0 192.168.14.1 192.168.1.0 255.255.255.0 192.168.14.1 192.168.2.0 255.255.255.0 192.168.14.2 192.168.3.0 255.255.255.0 192.168.14.2	YES
2 192.168.0.0 255.255.252.0 192.168.14.1 192.168.2.0 255.255.254.0 192.168.14.2	
1 192.168.0.0 255.255.255.0 192.168.14.1	NO
2 192.168.0.0 255.255.255.0 192.168.14.1 172.16.0.0 255.255.0.0 172.16.0.1	