Задача A. Треугольники: от A до C

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

-Докажи, что этот треугольник прямоугольный.
-Мамай клянус.

Антону на выходные дали задание по аналитической геометрии — по заданным трём сторонам треугольника a, b и c определить, является ли данный треугольник прямоугольным. Антон, конечно же, все выходные гулял с друзьями и пил . . . молоко. Теперь Антон просит вас помочь и написать программу, которая отвечает на заданный ему вопрос.

Формат входного файла

В единственной строке входного файла заданы три целых числа — a, b, c ($1 \le a, b, c \le 100$).

Формат выходного файла

В единственную строку выходного файла выведите `"YES"` (без кавычек), если данный треугольник является прямоугольным, и `"NO"` (без кавычек) иначе.

Примеры

abc.in	abc.out
3 4 5	YES
1 1 1	NO
3 5 4	YES

Задача B. Баян

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

> Баян (боян, байан и др.) — обозначение повторно опубликованной шутки или информации. При этом, в классическом понимании, информация должна быть повторно опубликована в том же самом источнике (например в том же самом форуме, или даже в том же самом разделе форума). Иными словами, репост (копирование информации из одного источника в другой), бояном в классическом понимании не является.
>
> Lurkmore

Фёдор администрирует юмористический сайт. Каждый день на сайт пользователи выкладывают сотни и тысячи анекдотов. В иные дни анекдотов выкладывается даже больше чем девять тысяч. К сожалению, не все анекдоты уникальны, среди них часто встречаются «баяны». «Баяном» Фёдор называет анекдот, который совпадает с каким-нибудь анекдотом, уже выложенным на сайт. Однако пользователи зачастую заносят анекдоты в базу вручную, по памяти, потому в одном и том же анекдоте они могут наставить лишних пробелов или пропустить какие-то знаки пунктуации.

Более строго, перед сравнением Фёдор с каждым анекдотом проделывает следующие операции:

1. Все знаки препинания заменяются на пробелы.
2. Все заглавные латинские буквы заменяются на соответствующие им строчные.
3. Все последовательно идущие пробелы заменяются на один пробел.

Фёдор считает два анекдота одинаковыми, если после выполнения этих операций они совпадают. При этом пробелы в начале и конце каждого анекдота Фёдор игнорирует. Однако он устал уже проверять анекдоты вручную и просит Вас написать программу, которая ему поможет.

Формат входного файла

В первой строке входного файла записан первый анекдот, во второй строке — второй. Каждый анекдот - это строка не более чем из 10^4 символов: заглавных и строчных латинских букв, пробелов и знаков препинания (`.,:;!?`).

Формат выходного файла

Выведите `"YES"`, если анекдоты из входного файла совпадают (в понимании Фёдора) и `"NO"` иначе.

Примеры

accordeon.in	accordeon.out
I successfully stopped a print job once. AM I GOD? I successfully stopped a print job once. Am i god?	YES
I successfully stopped a print job once. AM I GOD? I successfully stopped a print job once. AM I a GOD?	NO
I successfully stopped a print job once. AM I GOD? I successfully stopped a print job once AM I GOD	YES
I successfully stopped a print job once. AM I GOD? I successfully stopped a print job once . AM I GOD?	YES

Задача C. Ромашка

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

— Тут помню, а тут не помню!
— Слушайте! Заткнитесь, пожалуйста! Устроили тут ромашка: помню, не помню... Дайте спать.

к/ф Джентльмены удачи

Алиса очень любит гадать на ромашке. Для гадания она берёт ромашку и начинает приговаривать «Любит-не любит-любит-не любит-...», отрывая на каждое «любит» и «не любит» по одному лепестку. Если последний лепесток отрывается на слове «любит», Алиса очень радуется, если же на «не любит» – очень расстраивается.

Боб не хочет, чтобы Алиса расстраивалась, потому он хочет дать ей для гадания такую ромашку, последний лепесток которой оторвётся на «любит». Помогите Бобу не расстроить Алису.

Формат входного файла

В первой строке входного файла находится единственное число n — количество ромашек у Боба ($1 \le n \le 100$). Во второй строке находится n чисел a_i — количество лепестков у каждой из ромашек Боба ($1 \le a_i \le 100$).

Формат выходного файла

Выведите единственное число — количество лепестков у ромашки, которую Боб должен дать Алисе. Естественно, такая ромашка должна у него быть. Если требуемой ромашки у Боба нет, выведите единственное число -1.

Примеры

camomile.in	camomile.out
3 1 2 3	3
2 2 2	-1
4 2 2 1 11	11

Задача D. Клинические проблемы

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

Ваш друг — главный врач маленькой, но очень престижной клиники. Его клиника состоит из n палат, соединённых $n-1$ коридором. При этом клиника, естественно, представляет собой связное здание, то есть из любой палаты можно дойти до любой другой палаты, используя коридоры.

В начале i–го коридора стоит коробка, в которой лежит c_i пар одноразовых бахил. Каждый пациент, входя в коридор, надевает одну пару бахил и только потом идёт по коридору. Пройдя коридор, больной выкидывает бахилы. Если в коридоре не осталось бахил, то по этому коридору больше не пройдёт ни один больной. В каждой палате находится p_i больных, ожидающих осмотра. Главврач вызывает на осмотр сразу всех пациентов палаты разом, при этом пациенты начинают идти из своей палаты в направлении палаты номер 1 (в ней находится главврач). Если в каком-то коридоре на их пути закончились бахилы, то оставшиеся пациенты останутся в начале этого коридора и больше никуда не пойдут. К сожалению, за рабочий день главврач может осмотреть пациентов из не более чем k различных палат. Помогите ему выбрать осматриваемые палаты таким образом, чтобы он осмотрел как можно больше пациентов.

Формат входного файла

В первой строке входного файла дано два целых числа n и k — количество палат в клинике и максимальное число палат, которые может осмотреть доктор соответственно ($1 \le k \le n \le 250$). Во второй строке дано n целых чисел, p_i ($0 \le p_i \le 10^6$) — количество больных в i–ой палате. В следующих $n-1$ строках дано описание коридоров в больнице. Описание коридора — три целых числа u, v, c ($1 \le u < v \le n, 0 \le c \le 10^6$). Они означают, что в больнице есть коридор из палаты с номером v в палату с номером u, и в начале этого коридора стоит коробка, в которой лежит c пар бахил.

Формат выходного файла

В первую строку выходного файла выведите одно число — максимальное количество больных, которых может осмотреть главврач. В следующую строку выходного файла выведите число q — количество палат, которые главврач вызовет на осмотр. В следующую строку выведите q чисел — номера палат, которые главврач вызовет на осмотр. Если оптимальных решений несколько, то выведите любое.

Примеры

clinic.in	clinic.out
4 1 0 10 5 5 1 2 1 1 3 5 1 4 5	5 1 3
4 2 0 0 5 5 1 2 7 2 3 5 2 4 5	7 2 3 4
5 3 10 10 10 10 10 1 2 1 2 3 1 2 4 1 2 5 1	11 2 1 2

Примечание

В первом примере доктору невыгодно вызывать больных из второй палаты, поскольку оттуда до доктора сможет дойти лишь один больной (остальные так и останутся у входа в коридор из второй палаты в первую).

Во втором примере больные из палат 3 и 4 дойдут по коридорам до палаты 2, но в палату 1 сможет пройти только 7 больных из-за нехватки бахил.

В третьем примере доктору нет смысла вызывать больных из ещё какой-либо палаты, поскольку им заведомо не хватит бахил для перехода из палаты 2 в палату 1 (однако вызов больных, которые заведомо не дойдут до доктора, ошибкой в этой задаче не считается).

Задача E. Прикладная флористика

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

Серёжа подарил Ане на день рождения n цветов различных видов. В энциклопедии Аня прочитала, что i–ый цветок вырастает со скоростью v_i сантиметров в день и его изначальный рост равен h_i сантиметров. Ане сразу же посадила все n цветов на грядку и теперь бегает каждый день и проверяет — не «обогнал» ли какой-нибудь цветок другого в росте. Серёже не нравится, что Аня всё своё свободное время проводит с цветами. Теперь его интересует, когда наступит день, после которого ни один другой цветок не сможет обогнать никакой другой цветок. Серёжа просит вас помочь ему определить этот день.

Рассмотрим процесс роста цветов на примере:
Пусть Ане подарили три цветка. Рост первого цветка 5 сантиметров, второго — 7 сантиметров, третьего — 1 сантиметр. Скорость роста первого цветка 2 сантиметра в день, второго — 1 сантиметр в день, третьего — 3 сантиметра в день.

Изначально (в нулевой день) у цветов следующие высоты: 5 7 1.

В первый день: 7 8 4.

Во второй день: 9 9 7.

В третий день: 11 10 10 (первый цветок обогнал второй).

В четвертый день: 13 11 13 (третий цветок обогнал второй).

В пятый день: 15 12 16 (третий цветок обогнал первый).

Начиная с пятого дня никакие два цветка уже не обгонят друг друга, поэтому ответом является число 5.

Формат входного файла

В первой строке входного файла дано целое число n — количество различных видов цветов ($1 \le n \le 10^5$). В следующих n строках даны по два целых числа h_i и v_i — изначальный рост i–ого цветка и его скорость роста соответственно ($0 \le h_i, v_i \le 10^9$).

Формат выходного файла

В выходной файл выведите единственное число — номер искомого дня.

Примеры

flowers.in	flowers.out
3 5 2 7 1 1 3	5
1 0 0	0
2 5 1 6 1	0

Примечание

Первый пример разобран в условии. Во втором примере есть всего один цветок, ему некого обгонять, поэтому ответ 0. В третьем примере цветки уже не могут обогнать друг друга, поэтому ответ 0.

Задача F. Клавиатура

Имя входного файла:	`keyboard.in`
Имя выходного файла:	`keyboard.out`
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Наверное, каждый программист считает своим долгом изобрести велосипед и наблюдать, как он работает, так и программист Валера решил изобрести робота, который бы печатал заданный текст на любой клавиатуре. Из-за сложностей реализации распознавания образов клавиш, было решено использовать прямоугольные клавиатуры, которые бы имели вид матрицы 3×10, каждая ячейка которой — один из символов следующего алфавита: ‘a’, ‘b’, ‘c’, . . . , ‘x’, ‘y’, ‘z’, ‘ ?’, ‘!’, ‘.’, ‘,’. Помимо этого, в тексте могут встречаться пробелы, поэтому каждая клавиатура имеет клавишу пробела шириной 10 ячеек, которая по существу является четвертым нижним рядом клавиш.

Валера пытается сделать из робота профессионала с 10-пальцевой печатью, но так как робот не резиновый, пришлось ввести следующие ограничения на расположение робопальцев:

1. робот имеет две руки по пять пальцев на каждой,
2. оба больших пальца рук должны всегда располагаться на клавише пробела и, в случае необходимости, нажимать на неё,
3. в каждом столбце клавиш (от 1 до 10) может располагаться максимум один палец,
4. руки не должны перекрещиваться,
5. пальцы каждой из рук не должны перекрещиваться,
6. пальцы разных рук не должны перекрещиваться.

Для тестирования своего робота Валера задает клавиатуры с разным расположением символов. На первых порах робот работал довольно медленно и программисту приходилось ждать порядка 20 минут, пока робот наберёт хотя бы 5 символов текста. Проблема заключалась не только в медленном распознавании символов, но и в перемещении пальцев робота по клавиатуре. Пальцы робота перемещаются исключительно параллельно сторонам клавиатуры, то есть, чтобы переместиться из клавиши на позиции (1, 3) в (2, 4), робот двигает палец на один ряд вниз и затем на один столбец вправо, тратя на это 2 минуты. Более формально: при перемещении пальца из позиции (x_1, y_1) в (x_2, y_2) робот тратит $|x_1 - x_2| + |y_1 - y_2|$ минут, где x_1, x_2 — номера рядов клавиш, y_1, y_2 — номера столбцов.

Перед запуском набора текста мизинец, безымянный, средний и указательный пальцы левой руки робота находятся соответственно на следующих позициях: $(2, 1), (2, 2), (2, 3), (2, 4)$, а соответствующие пальцы правой руки находятся на следующих позициях: $(2, 7), (2, 8), (2, 9), (2, 10)$.

Валера решил написать программу, которая бы соблюдала описанные требования к расположению пальцев во время набора текста, а также позволяла набрать текст роботу как можно быстрее.

Формат входного файла

Первые три строки файла содержат по 10 разделенных пробелом символов из описанного в условии алфавита (без пробела). Четвертая строка файла содержит текст, составленный из того же набора символов и пробела длиной не более 20 символов. В данной задаче вы можете считать, что робот тратит одну минуту на распознавание любого символа, даже пробела, несмотря на то, что два больших пальца находятся на них постоянно.

Формат выходного файла

В выходной файл выведите целое положительное число — время в минутах, затрачиваемое на набор роботом заданного текста.

Примеры

keyboard.in	keyboard.out
a b c d e f g h i j k l m n o p q r s t u v w x y z , . ! ? some .	9

Задача G. Нефть

Имя входного файла:	`oil.in`
Имя выходного файла:	`oil.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Однажды в городе П. решили построить нефтяную сеть. Для этого было построено n транспортных узлов, которые соединялись между собой m нефтепроводами. Стоит помнить, что нефть по трубопроводу течет только в одну сторону. Из-за приближающегося срока сдачи проекта в эксплуатацию подрядная организация начала очень быстро сооружать узлы и трубопроводы. Для каждого нефтепровода известно направление движения и объем протекающей нефти, измеренный в литрах в секунду. Не трудно заметить, что при этом в некоторых узлах может возникнуть избыток/недостаток нефти в том смысле, что втекает/вытекает нефти больше, чем вытекает/втекает. Требуется определить в каких вершинах необходимо докачивать нефть, а в каких сливать неиспользуемый избыток для того, чтобы соблюдался баланс (количество втекающей нефти должно быть равно количеству вытекающей с учето дополнительной откачки/долива).

Формат входного файла

В первой строке содержатся целые числа n и m ($1 \le n, m \le 100000$). В следующих m строках находятся описания нефтепроводов: тройка чисел u, v, c, где u — номер узла, из которого вытекает нефть в узел v со скоростью c литров в секунду ($1 \le u, v \le n, u \ne v, 1 \le c \le 10000$). Узлы нефтесети пронумерованы целыми числами от 1 до n, между двумя узлами может быть более одного нефтепровода.

Формат выходного файла

В выходной файл выведите n чисел — информацию об изменениях в каждом узле: i-ое число равно c_i и удовлетворяет одному из следующих условий:

1. если c_i отрицательное, то из узла i необходимо дополнительно слить $-c_i$ (л/сек),
2. если c_i положительное, то в узел i необходимо дополнительно закачать c_i (л/сек),
3. если c_i равно нулю, то производить изменения с узлом i не нужно, так как соблюдается баланс объема втекающей и вытекающей нефти.

Примеры

oil.in	oil.out
4 4 1 2 3 2 3 4 3 4 4 4 1 2	1 1 0 -2

Задача H. Криптография

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

На занятии по криптографии Вовочке задали расшифровать хитрым образом зашифрованный текст. Вовочка быстро определил алгоритм шифрования, написал алгоритм дешифрования, но, к сожалению, столкнулся с проблемой.

Как известно, во многих криптографических алгоритмах, помимо текста, еще используется некий секретный ключ. Для успешной дешифровки задания нашему герою осталось отыскать этот ключ.

Проведя много часов за поиском этого ключа, Вовочка наконец узнал, как его определить. Оказалось, что ключ — это количество раз, которое первые два символа зашифрованного текста встречаются далее в этом тексте подряд.

Поиски ключа так измотали нашего героя, что он просит Вас помочь ему справиться с этой несложной задачей!

Формат входного файла

В первой строке входного файла находятся первые два символа зашифрованной строки. Во второй строке находится остаток зашифрованного текста длиной N символов ($1 \le N \le 10^5$). В конце второй строки обязательно присутствует символ перевода строки. Все символы в строках — строчные и прописные символы латинского алфавита, а также цифры.

Формат выходного файла

В выходной файл выведите одно целое число — количество раз, которое два символа из первой строки входного файла встречаются во второй строке входного файла подряд.

Примеры

pair.in	pair.out
ab cdef0	0
ab 1Rabab2Z	2
xx JiMxxxMDg3OyxxxYjMTA4xx	5

Примечание

В первом примере пара символов `"ab"` ни разу не встречается в оставшейся строке, потому ответ равен нулю.

Во втором примере пара `"ab"` встречается в остатке текста начиная с 3 и 5 позиций.

В третьем примере пара `"xx"` встречается пять раз — на 4, 5, 13, 14 и 22 позициях.

Задача I. Турникет

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

В одном IT-Парке государственного университета города P поставили турникет для учета входящих и выходящих сотрудников. Каждый работник имеет ключ-карту с уникальным пятизначным номером, приложив которую к турникету можно пройти через него в обе стороны. При прикладывании ключ-карты, турникет сохраняет информацию о времени, номере ключ-карты и действии (вход или выход сотрудника).

Вахтеру было поручено вести ежедневный учет пребывания сотрудников на территории IT-Парка. Так как в IT-Парке работают только программисты (стало быть и вахтер тоже программист), то вахтер решил написать простенькую программу, которая по данным с турникета за день, выводила бы для каждого сотрудника, время его пребывания в стенах IT-Парка на момент наступления полуночи.

Стоит отметить, что программисты, по большей части, народ ленивый, и иногда они могут забывать уходить на ночь домой с работы, или наоборот, могут рано утром, проработав всю ночь, уйти домой.

Вахтер успешно написал программу, однако, как любой уважающий себя программист, он решил проверить правильность ее работы. Ваша задача — написать программу, которая по входных данным рассчитывает время пребывания в стенах IT-Парка каждого упомянутого в этих данных сотрудника, и помочь вахтеру проверить его навыки программирования.

Формат входного файла

В первой строке входного файла находится одно целое число n ($0 \le n \le 86400$) — количество записей турникета за день. В последующих n строках находится отсортированная по возрастанию времени информация о действиях ключ-карт. В каждой строке сначала записано время в формате $HH : MM : SS$ ($00 \le HH \le 23, 00 \le MM, SS \le 59$), где HH , MM, SS — часы, минуты и секунды, с ведущими нулями, соответственно. Все времена различны. Далее, через пробел записано целое число — пятизначный номер ключ-карты сотрудника (номер никогда не начинается с нуля). Завершает строку записанное через пробел целое число t. Если $t = 0$, то это означает, что сотрудник вышел из здания, если $t = 1$ — вошел. Никакой сотрудник не может войти или выйти более одного раза подряд. Секунда, в которую выходит сотрудник, засчитывается, как пребывание в стенах IT-Парка.

Формат выходного файла

В первой строке выходного файла выведите число m — количество различных сотрудников, прошедших в течение дня через турникет. Далее в m строках выведите информацию по каждому сотруднику — номер ключ-карты сотрудника и через пробел его время пребывания в IT-Парке, считая от полуночи ($00 : 00 : 00$) и до полуночи ($23 : 59 : 59$) включительно, в формате $HH : MM : SS$ ($00 \le HH \le 24, 00 \le MM, SS \le 59$). Номера ключ-карт сотрудников выведите в возрастающем порядке.

Примеры

pass.in	pass.out
4 12:32:44 10001 1 17:13:32 10004 0 17:31:36 10003 1 18:28:32 10001 0	3 10001 05:55:49 10003 06:28:24 10004 17:13:33
7 08:14:08 10003 1 10:31:36 10005 1 11:33:42 10000 1 14:31:28 10001 1 18:26:56 10003 0 19:07:59 10008 1 20:20:53 10002 1	6 10000 12:26:18 10001 09:28:32 10002 03:39:07 10003 10:12:49 10005 13:28:24 10008 04:52:01
8 01:42:52 12345 1 03:19:56 12345 0 05:42:32 42042 1 07:31:34 42042 0 13:56:05 12345 1 14:49:40 42042 1 19:50:05 12345 0 20:21:27 42042 0	2 12345 07:31:06 42042 07:20:51
2 12:00:01 12345 0 12:00:02 12345 1	1 12345 24:00:00

Задача J. Ступенечки

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

В связи с тем, что в городе P был открыт уже N–й торговый центр, указом градоначальника было решено один из торговых центров преобразовать в спортивный комплекс. Эскалаторы было решено использовать для занятий легкой атлетикой.

На очередной тренировке в торговом центре Вася и Петя решили устроить соревнования по скоростному преодолению ступенек эскалатора. Их целью было пробежать как можно большее расстояние по эскалатору.

В торговом центре выбран один эскалатор для забегов. Он имеет длину L ступенек и двигается вверх с постоянной скоростью x ступенек в минуту. Все ступеньки эскалатора одинаковой длины.

Известно, что Вася всегда бегает по ступенькам эскалатора со скоростью v_1 ступенек в минуту, а Петя — v_2 ступенек в минуту ($v_1, v_2 > x$).

Вас, как главного наставника легкоатлетической сборной, интересует вопрос, кто же пробежал большее расстояние по эскалатору? Вася и Петя всегда бегают снизу вверх.

Формат входного файла

В первой строке входного файла находятся два целых числа L ($10 \le L \le 10^3$) и x ($1 \le x \le 10^5$). Во второй строке находятся два целых числа v_1 ($x < v_1 < 10^6$) и v_2 ($x < v_2 < 10^6$), скорости Васи и Пети соответственно.

Формат выходного файла

В выходной файл выведите '1', если Вася пробежал большее расстояние, '2', если Петя пробежал большее расстояние, или '12', если они пробежали одинаковое расстояние. (Числа выведите без кавычек).

Примеры

stair.in	stair.out
10 2 5 7	2
100 15 35 25	1
11 11 42 42	12