Об уровнях олимпиады

В этой интернет-олимпиаде мы продолжаем эксперимент с уровнями сложности.

Вы можете самостоятельно выбрать, задачи какого уровня решать.

Если вы хотите участвовать в олимпиаде базового уровня, решайте задачи A, B, C и D.

Если вы хотите участвовать в олимпиаде усложненного уровня, решайте задачи C, D, E и F.

У базового и усложненного уровней будут отдельные таблицы результатов.

Если вы отправите решение задачи E или F, вы автоматически будете классифицированы в таблице усложненного уровня (даже если ваше решение не пройдет тесты из примера). Иначе вы будете классифицированы в таблице базового уровня.

Задача A. План гостиницы

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

Паша с детства мечтал стать архитектором и рисовал планы строений на листах бумаги, а иногда — на столах и стенах. Конечно, сейчас у него есть высшее образование, и он является известным архитектором.

Однажды Паша перебирал свои детские чертежи и обнаружил интересный план этажа гостиницы. Этаж представлял собой прямоугольник размером $n \times m$ метров. При этом, на чертеже каждый квадрат, соответствующий квадратному метру этажа, был закрашен синим или красным цветом. Немного подумав, Паша понял, что синим цветом он обозначал коридоры, а красным — номера гостиницы.

Паша с детства соблюдает международный стандарт, следуя которому номер гостиницы должен быть прямоугольным, то есть должен изображаться на чертеже полностью закрашенным красным цветом прямоугольником со сторонами, параллельными внешним границам этажа. Кроме этого, вдоль стен всех номеров гостиницы идут коридоры.

В целях пожарной безопасности Паша по краям этажа размещал только коридоры. Таким образом первые и последние строки, а также первые и последние столбцы чертежа Паши закрашены синим.

Рассматривая свой детский план, Паша заинтересовался, какова наибольшая площадь номера на этаже его гостиницы.

Напишите программу для вычисления этой величины.

Формат входного файла

В первой строке входного файла содержатся два целых числа — n и m ($3 \le n, m \le 30$). В следующих n строках содержится по m символов в каждой. Символ «`*`» соответствует клетке, закрашенной в синий цвет (квадратному метру коридора), а «`.`» — в красный цвет (квадратному метру номера).

Формат выходного файла

В выходной файл выведите площадь наибольшего номера. Если же не существует ни одного номера, выведите «`-1`».

Примеры

hotel.in	hotel.out
5 5 ***** *.*** **..* **..* *****	4
4 5 ***** *.*** *.*.* *****	2
3 3 *** *** ***	-1

Задача B. Великая Флатландская Стена

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

Однажды император Флатландии Хи Шуанди, которому от предыдущего правителя досталась лишь горстка разрозненных Воюющих Царств, решил укрепить свою Великую Империю и повелел возвести Великую Флатландскую Стену. Волею судеб все Воюющие Царства не пересекались и имели треугольную форму. По периметру каждого из них были возведены Нерушимые Стены, в каждом углу царства находилось по смотровой башне.

Земли были распределены предыдущим правителем Флатландии Бюй Лувэем таким образом, чтобы из любой башни в ясный день была видна любая другая, а как известно из трудов Великих Флатландских Математиков, такое может быть только в том случае, когда никакие три башни не находятся на одной прямой.

Кроме того, так как императорскому налогосборщику было необходимо собирать налоги со всех Воюющих Царств, и переносить с собой большие суммы денег, страна была построена таким образом, чтобы налогосборщик мог добраться из любого Царства в любое только по Нерушимым Стенам, возможно разных Царств.

Для того чтобы узнать сколько денег необходимо выделить на строительство Великой Флатландской Стены, Хи Шуанди необходимо узнать суммарную длину всех Нерушимых Стен, которые отделяют территорию какого-либо Воюющего Царства, от территории, не принадлежащей Флатландии.

Формат входного файла

В первой строке входного файла содержится число m — количество Воюющих Царств ($1 \le m \le 1000$). В каждой из следующих m строк содержится по шесть чисел x_1, y_1, x_2, y_2, x_3, y_3 — координаты вершин треугольника, составляющего территорию очередного Воюющего Царства. Все координаты целые, по абсолютной величине не превосходят 10^4.

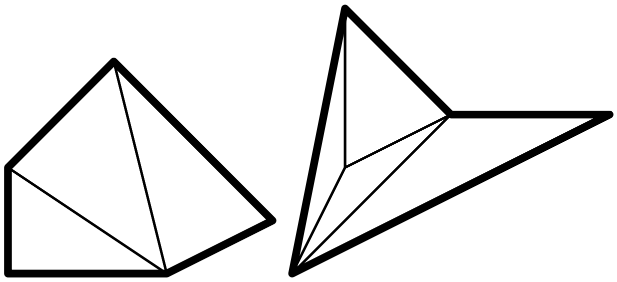
Формат выходного файла

В выходной файл выведите одно число — длину Великой Флатландской Стены. Ответ будет засчитан, если он отличается от правильного не более, чем на 10^{-4}.

Примеры

<table>
<tr><th>greatwall.in</th><th>greatwall.out</th></tr>
<tr><td>3
0 0 3 0 0 2
3 0 0 2 2 4
3 0 2 4 5 1</td><td>14.3071357893652</td></tr>
<tr><td>4
0 0 1 5 1 2
1 5 1 2 3 3
0 0 3 3 1 2
0 0 3 3 6 3</td><td>17.6356505708383</td></tr>
</table>

Иллюстрации к примерам:



Задача C. Загадочное число

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

Толя — младший научный сотрудник в *Отделе Изучения Смысла Жизни*. Недавно он обнаружил, что в десятичной записи числа N скрыта важная истина, проливающая свет на некоторые из вопросов жизни человечества. Однако по своей натуре Толя — человек эгоистичный и не хочет с кем-либо делиться своим открытием. Поэтому он решил запомнить это число, а все записи, связанные с ним, съесть.

К сожалению, Толя немного рассеян и забывчив, поэтому для запоминания числа N он выбрал следующий путь. Есть некоторый набор из K чисел, так или иначе связанных с жизнью нашего ученого, которые он помнит очень хорошо, даже лучше, чем дату своего рождения. Все эти числа не более чем трехзначные.

Толя пытается представить число N в виде конкатенации чисел из этого набора (например, конкатенация чисел 1 и 2 может дать 12 или 21, в зависимости от порядка). Никакое число нельзя использовать дважды, с другой стороны все числа использовать необязательно. При этом Толя хочет придумать (для упрощения запоминания) такое представление, в котором использовалось бы как можно меньше чисел.

Напишите программу, которая будет находить такое представление.

Формат входного файла

Первая строка входного файла содержит число N ($1 \le N < 10^{45}$). Вторая строка содержит число K — количество чисел в наборе ($1 \le K \le 1000$). Третья строка содержит K памятных Толе чисел ($0 \le a_i < 1000$). Все числа в наборе различны и не содержат ведущих нулей.

Гарантируется, что существует хотя бы одно требуемое представление числа N.

Формат выходного файла

В первой строке выходного файла выведите число M — количество чисел в искомом представлении. В последующих M строках выведите числа в том порядке, в котором их необходимо конкатенировать для получения числа N. Если искомых представлений несколько, выведите любое из них.

Примеры

enigmatic.in	enigmatic.out
123 4 1 3 12 23	2 12 3
123 4 1 2 3 123	1 123

Решения, правильно работающие для случая $K \le 8$, будут оцениваться исходя из 40 баллов.

Задача D. Память подводит

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

Широко известный в узких кругах режиссер Женя Кэмерофф ведет подготовку к съемкам своего нового фильма «Смайл». Этот фильм будет широко использовать новейшие трехмерные спецэффекты, что приведет к небывалому успеху — за первую неделю проката планируется собрать меньше 1000000000 миллиграмм помидоров и тухлых яиц, брошенных в экран кинотеатра возмущенными зрителями.

Женя все продумал и рассчитал. Для качественного обсчета спецэффектов ему необходимо N компьютеров, которые он планирует купить в ближайшем компьютерном ларьке. Чтобы удостовериться в том, что он покупает качественный товар, он решил протестировать оперативную память каждого из приобретаемых им компьютеров. Для этого он купил M специальных устройств, подсоединяемых к компьютеру через *USB*.

Каждое из этих устройств, будучи подключенным к компьютеру, проверяет все новые и новые области оперативной памяти. На то, чтобы проверить всю память одного компьютера, у него уходит K часов. Если устройство в некоторый момент вытащить из разъема, то проверка памяти этого компьютера приостановится. Если снова подсоединить устройство к компьютеру, то проверка продолжится с этого же места. Заметим, что мы можем продолжить проверку как тем же самым устройством, так и любым другим. Если по окончании проверки устройство останется в компьютере, ничего не произойдет.

Женя Кэмерофф ценит время. Поэтому он хочет знать, за какое минимальное время все N компьютеров могут быть протестированы. Более того, он хочет знать последовательность действий, которая приведет к этому результату. Ваша задача — помочь ему в этом.

Формат входного файла

В первой строке входного файла находятся три целых числа N, M и K ($1 \le N, M, K \le 1000$).

Формат выходного файла

В первой строке выходного файла выведите минимальное время T_{min}, за которое все компьютеры могут быть протестированы, в виде несократимой дроби в формате `A/B`, где A — числитель дроби, а B — ее знаменатель. Для чисел A и B должны выполняться следующие неравенства: $0 \le A \le 10^9$, $1 \le B \le 10^9$.

Во второй строке выведите число действий C. В каждой из последующих C строк выведите информацию о требуемом действии в следующем формате:

`N_i/D_i: Connect V_i to K_i`

Это означает, что спустя $\frac{N_i}{D_i}$ часов после начала тестирования тестирующее устройство с номером V_i ($1 \le V_i \le M$) должно быть подсоединено к компьютеру с номером K_i ($1 \le K_i \le N$). При этом, если устройство V_i было подключено к какому-либо компьютеру, то оно от него отсоединяется. Если к компьютеру K_i было подключено какое-либо устройство, то оно также отсоединяется.

Дроби $\frac{N_i}{D_i}$ должны быть несократимыми. Для чисел N_i, D_i должны выполняться следующие неравенства: $0 \le N_i \le 10^9$, $1 \le D_i \le 10^9$.

Действия должны быть выведены в порядке неубывания времени. Число действий C не должно превышать 10000.

Если существует несколько возможных последовательностей действий, выполняющихся за минимальное время и удовлетворяющих ограничению на число действий, то выведите любую из них.

Примеры

memory.in	memory.out
2 2 1	1/1 2 0/1: Connect 2 to 1 0/1: Connect 1 to 2
3 2 1	3/2 4 0/1: Connect 1 to 1 0/1: Connect 2 to 2 1/2: Connect 1 to 3 1/1: Connect 2 to 1

Задача E. Волшебные лекарства

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

Известный волшебник Ла-Шэто работает в своей лаборатории над набором волшебных лекарств для ближайшей волшебной ярмарки. Он уже закупил большое количество пузырьков, каждый из которых содержит одну из n магических жидкостей, из которых он планирует делать лекарства, и готов к работе.

Каждое лекарство получается в результате смешивания двух целых пузырьков, содержащих различные магические жидкости. Маг смешивает содержимое двух пузырьков и произносит магическое заклинание, в результате получается волшебное лекарство.

Ла-Шэто хочет изготовить как можно больше порций лекарств, при этом он хочет изготовить как можно более качественные лекарства. Ла-Шэто занумеровал магические жидкости от 1 до n в порядке убывания их качества, при этом наилучшая жидкость получила номер 1, а наихудшая — номер n. Ла-Шэто хочет изготовить как можно больше лекарств, если есть несколько способов это сделать, он хочет изготовить как можно больше лекарств из первой и второй магических жидкостей. Если по прежнему есть несколько способов, он хочет максимизировать количество лекарств из жидкостей 1 и 3, затем 1 и 4, и т. д., 1 и n, затем 2 и 3, и т. д.

Помогите ему выяснить, сколько лекарство какого типа изготовить.

Формат входного файла

Первая строка входного файла содержит n — количество жидкостей, которые закупил Ла-Шэто ($2 \le n \le 100\,000$). Вторая строка содержит n целых чисел: $a_1, a_2, \ldots, a_n$ — количество пузырьков с 1 жидкостью, количество пузырьков со 2 жидкостью, и т. д. ($1 \le a_i \le 10^9$).

Формат выходного файла

Первая строка выходного файла должна содержать m — количество различных типов лекарств, которые следует изготовить Ла-Шэто. Следующие m строк должны содержать по три целых числа: i, j, c_{ij} — номера жидкостей, из которых следует изготовить лекарство и количество порций лекарства, которое следует изготовить. В каждом описании должно выполняться $i < j$, выводите описание лекарств в порядке возрастания i, при равном i в порядке возрастания j.

Примеры

magic.in	magic.out
4 1 1 1 1	2 1 2 1 3 4 1

Задача F. Дороги

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

После изобретения колеса, жители Флатландии решили соединить свои n городов дорогами. Поскольку строительство дорог оказалось довольно дорогим, они решили построить минимальное количество дорог, чтобы из каждого города до каждого можно было добраться по дорогам. Поскольку правила дорожного движения в те времена еще не были изобретены, по каждой дороге можно было двигаться в любом направлении. Флатландия расположена на плато, поэтому дороги должны были быть проложены по отрезкам прямых, соединяющих города.

Жители Флатландии немедленно начали спорить, какие города соединить дорогами. Разумеется, жители каждого города хотели, чтобы как можно больше дорог выходило из их города. Чтобы прекратить бессмысленные обсуждения, король Флатландии издал указ: для каждого города было назначено количество дорог, которое должно было выходить из этого города. Король Флатландии был очень мудрым, поэтому из каждого города выходило от 1 до $n-1$, а их сумма была $2n-2$.

Но министр дорожного строительства не был таким же мудрым. Получив от короля эти числа, он не мог придумать, как построить дороги. Проблема в том, что поскольку туннели, мосты и светофоры, а также знак «главная дорога» не были изобретены, дороги не должны были пересекаться.

Помогите министру решить, какие города соединить дорогами.

Формат входного файла

Первая строка входного файла содержит n — количество городов во Флатландии ($2 \le n \le 1000$). Следующие n строк содержат по три целых числа: x_i, y_i и d_i. Здесь x_i и y_i — координаты соответствующего города, а d_i — количество дорог, которые должны из него выходить. Координаты не превышают 10^4 по абсолютной величине. Никакие три города не лежат на одной прямой.

Формат выходного файла

Если построить дороги невозможно, выведите -1 на первой строке выходного файла.

В противном случае выведите $n-1$ строку. Каждая строка должна содержать два целых числа — номера городов, которые следует соединить соответствующей дорогой. Города нумеруются от 1 до n в том порядке, в котором они заданы во входном файле.

Примеры

roads.in	roads.out
7 0 6 1 0 0 1 2 3 1 3 2 3 4 4 4 7 0 1 7 6 1	1 5 2 4 3 4 4 5 5 6 5 7