

Задача А. Задача Читериуса

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Читериус — знаменитый на всю Берляндию астролог, маг, волшебник, а также обманщик и пройдоха. Одно из последних его изобретений — амулеты Читериуса! Они приносят удачу и здоровье, однако стоят немалых денег. Читериус изготавливает их сам. Секрет изготовления хранится в глубокой тайне. Но мы то знаем, что долгими ночами Читериус склеивает супер-клеем доминошки по двое, чтобы получать квадратики 2×2 — волшебные амулеты Читериуса!



Так выглядит один из амулетов Читериуса

После тяжелой трудовой ночи Читериус изготовил n амулетов. Каждый из них представляет собой квадрат 2×2 , каждая четвертинка содержит от 1 до 6 точек. Теперь он хочет разложить их в стопки, каждая стопка должна содержать одинаковые амулеты. Два амулета называются одинаковыми, если их можно повернуть так, что после наложения их друг на друга количества точек в соответствующих четвертинках окажется равным. Переворачивать амулеты нельзя.

Напишите программу, которая по заданным амулетам найдет количество стопок на столе Читериуса.

Формат входного файла

В первой строке записано целое число n ($1 \leq n \leq 1000$), n — количество амулетов. Далее содержатся описания амулетов. Каждое описание занимает две строки и содержит по две цифры (от 1 до 6) в каждой из строк. Между каждой парой амулетов находится строка «**».

Формат выходного файла

Выведите искомое количество стопок.

Примеры

stdin	stdout
4 31 23 ** 31 23 ** 13 32 ** 32 13	1
4 51 26 ** 54 35 ** 25 61 ** 45 53	2

Задача В. Анализ таблиц bHTML

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

В этой задаче используется крайне упрощенный вариант разметки таблиц средствами HTML. Пожалуйста, используйте условие как формальный документ.

Строка определяет таблицу bHTML, если имеет вид:

```
TABLE ::= <table>ROWS</table>
ROWS ::= ROW | ROW ROWS
ROW ::= <tr>CELLS</tr>
CELLS ::= CELL | CELL CELLS
CELL ::= <td></td> | <td>TABLE</td>
```

Пробелы в грамматике заданы исключительно для наглядности, при записи таблиц они не используются. Иными словами, таблицы в bHTML очень похожи на обычные таблицы HTML в которых только участвуют теги «table», «tr», «td», все теги парные и таблица содержит хотя бы одну строку и хотя бы одну ячейку в каждой строке. Посмотрите тесты из условия в качестве примеров таблиц.

Как видно, таблицы могут вкладываться друг в друга. Вам задана одна таблица (в которую, возможно, вложены другие). Вам надо написать программу, которая проанализирует все таблицы и найдет для каждой из них количество ее ячеек. Таблицы не обязаны быть прямоугольными.

Формат входного файла

Для удобства входные данные могут быть разбиты на непустые строки произвольным образом. Входные данные состоят из не более чем 10 строк. Объедините все строки входных данных в одну, чтобы получить текстовое представление s заданной таблицы. Текстовое s представление соответствует грамматике из условия (корневой элемент грамматики TABLE), длина s не превосходит 5000. Используются только строчные буквы при записи тегов, пробелов в строках нет.

Формат выходного файла

Выведите размеры всех таблиц во входных данных в порядке неубывания.

Примеры

stdin
<code><table><tr><td></td></tr></table></code>
stdout
1
stdin
<code><table> <tr> <td> <table><tr><td></td></tr><tr><td></ td ></tr><tr> ><td></td></tr><tr><td></td></tr></table> </td> </tr> </table></code>
stdout
1 4

stdin
<pre><table><tr><td> <table><tr><td> <table><tr><td> <table><tr><td></td><td></td> </tr><tr><td></td></tr></table> </td></tr></table> </td></tr></table> </td></tr></table></pre>
stdout
<pre>1 1 1 3</pre>

Задача С. Три базовых станции

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Деревня Нью-Васюки вытянута вдоль автомобильной трассы и поэтому каждый город на ней характеризуется своим смещением относительно некоторой фиксированной точки — координатой x_i . Деревня состоит из n домов, i -ый дом расположен в точке с координатой x_i .

Провайдер сотовой связи TELE3 запланировал разместить три базовых станции таким образом, чтобы покрыть связью каждый дом в деревне. Базовая станция мощности d , размещенная в точке t , покрывает связью дома в отрезке $[t - d, t + d]$ (включая границы).

Для упрощения монтажа (да и просто «чтобы не перепутать») планируется, что все три станции будут иметь одинаковую мощность d . Какое минимальное значение d достаточно, чтобы покрыть связью все дома в деревне.

Формат входного файла

В первой строке записано целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество домов в деревне. Вторая строка содержит координаты домов — последовательность x_1, x_2, \dots, x_n целых чисел ($1 \leq x_i \leq 10^9$). Возможно, что два или более города расположены в одной точке. Координаты заданы в произвольном порядке.

Формат выходного файла

Выведите искомую минимальную мощность d . Во вторую строку выведите три числа — возможные координаты расположения базовых станций. Выводите координаты с 6 цифрами после десятичной точки. Позиции станций могут быть любыми от 0 до $2 \cdot 10^9$ включительно. Допускается, чтобы базовые станции имели совпадающие координаты. Если решений несколько, выведите любое.

Примеры

stdin	stdout
4 1 2 3 4	0.500000 1.500000 2.500000 3.500000
3 10 20 30	0 10.000000 20.000000 30.000000
5 10003 10004 10001 10002 1	0.500000 1.000000 10001.500000 10003.500000

Задача D. Геометрическая задача

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 second
Ограничение по памяти: 256 megabytes

Поликарп очень любит геометрические прогрессии — он их коллекционер. Но так как такие прогрессии попадают крайне редко, то ему очень нравятся и такие последовательности чисел, в которых можно удалить один элемент, что в результате получается геометрическая прогрессия.

В этой задаче геометрическими прогрессиями будем называть такие конечные последовательности чисел a_1, a_2, \dots, a_k , что $a_i = c \cdot b^{i-1}$ для некоторых вещественных чисел c и b . Например, последовательности $[2, -4, 8]$, $[0, 0, 0, 0]$, $[199]$ — геометрические прогрессии, а $[0, 1, 2, 3]$ — нет.

Недавно Поликарпу попала последовательность и он никак не может ее классифицировать. Помогите ему в этом. Определите, является ли она геометрической прогрессией. Если нет, то проверьте, можно ли удалить из нее один элемент так, чтобы в результате она стала геометрической прогрессией.

Формат входного файла

В первой строке записано целое число n ($1 \leq n \leq 10^5$) — количество чисел в заданной последовательности. Вторая строка содержит заданную последовательность. Числа разделяются пробелами. Все элементы заданной последовательности — целые, не превосходящие по абсолютной величине 10^4 .

Формат выходного файла

Выведите 0, если заданная последовательность является геометрической прогрессией. В противном случае проверьте, верно ли, что можно удалить один элемент так, чтобы она стала геометрической прогрессией. Если это так, то выведите 1. Если таких позиций несколько, то выведите любую. В случае, если это сделать невозможно, выведите -1.

Примеры

stdin	stdout
4 3 6 12 24	0
4 -8 -16 24 -32	1
4 0 1 2 3	2

Задача E. Пентагон

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 10 seconds
Ограничение по памяти: 256 megabytes

В соответствии с последним указом Президента Берляндии каждый город страны должен иметь собственное здание Министерства Обороны (свой собственный Пентагон). Мегполис Бербург не стал исключением. В этом городе n перекрестков, некоторые пары из которых соединены двусторонними дорогами. Всего в городе m дорог, между каждой парой перекрестков не более одной.

В настоящий момент обсуждается выбор места расположения Пентагона в Бербурге. Решено, что Пентагон должен занимать территорию пяти различных перекрестков, которые соединены в цикл дорогами. В порядке цикла по дорогам будет построена специальная стена (как и положено — с высоким напряжением, колючей проволокой и т.п.). Таким образом, количество возможных способов построить Пентагон в городе равно количеству различных циклов длины 5, составленных из перекрестков и дорог.

Ваша задача вывести количество способов построить Пентагон в Бербурге. Учтите, что в этой задаче пройдет только достаточно оптимизированное решение, так что обязательно протестируйте свой код на максимальном тесте.

Формат входного файла

В первой строке записаны два целых числа n и m ($1 \leq n \leq 700; 0 \leq m \leq n \cdot (n-1)/2$), n — количество перекрестков, а m — количество дорог в городе. Далее в m строках содержатся описания дорог, по одной дороге в строке. Каждая дорога задана парой целых чисел a_i, b_i ($1 \leq a_i, b_i \leq n; a_i \neq b_i$), где a_i и b_i — номера соединяемых дорогой перекрестков. Перекрестки занумерованы от 1 до n . Не гарантируется, что из любого перекрестка можно добраться в любой другой, двигаясь по дорогам.

Формат выходного файла

Выведите единственное число — искомое количество способов. Пожалуйста, не используйте спецификатор `%lld` для чтения или записи 64-х битовых чисел на C++. Рекомендуется использовать поток `cout` (также вы можете использовать спецификатор `%I64d`).

Примеры

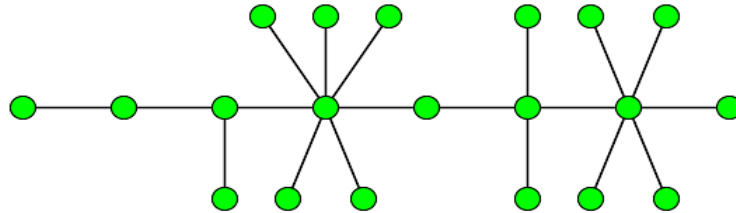
stdin	stdout
5 5 1 2 2 3 3 4 4 5 5 1	1
5 10 1 2 1 3 1 4 1 5 2 3 2 4 2 5 3 4 3 5 4 5	12

Задача F. Гусеница

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Неориентированный граф называется *гусеницей*, если он является связным графом без циклов, и в нем существует такой путь, что любая вершина находится на расстоянии не более 1 от него. Гусеница может содержать петли, но не может содержать кратных ребер.

На рисунке изображен пример гусеницы:



Вам задан неориентированный граф G с которым можно производить операцию сжатия двух вершин в одну. Для этого выбираются любые две вершины графа a и b ($a \neq b$). Из графа удаляются обе эти вершины вместе с их ребрами (инцидентными хотя бы одной из вершин a или b), но добавляется новая вершина w вместе с ребрами вида (x, w) для каждого ребра (a, w) и (b, w) . Если в графе существовало ребро (a, b) , то оно преобразуется в петлю (w, w) . В результате операции слияния могут появляться кратные ребра и петли. Заметим, что эта операция уменьшает количество вершин графа на 1, но оставляет неизменным количество ребер в графе.

Операция сжатия двух вершин может быть неформально описана как объединение двух вершин графа в одну, с естественным преобразованием ребер графа.

С помощью последовательного применения описанной операции можно любой заданный неориентированный граф сделать гусеницей. Напишите программу, которая выведет наименьшее количество операций сжатия, чтобы заданный граф сделать гусеницей.

Формат входного файла

В первой строке содержится пара целых чисел n, m ($1 \leq n \leq 2000; 0 \leq m \leq 10^5$), n — количество вершин в графе, а m — количество ребер в нем. Далее в m строках заданы ребра графа, по одному ребру в строке. Каждая строка содержит пару целых чисел a_i, b_i ($1 \leq a_i, b_i \leq n; a_i \neq b_i$), a_i, b_i — номера соединяемых ребром вершин. Вершины пронумерованы от 1 до n . Между каждой парой вершин может быть не более одного ребра. Заданный граф не обязательно является связным.

Формат выходного файла

Выведите искомое наименьшее число операций.

Примеры

stdin	stdout
4 4 1 2 2 3 3 4 4 2	2
6 3 1 2 3 4 5 6	2
7 6 1 2 2 3 1 4 4 5 1 6 6 7	1