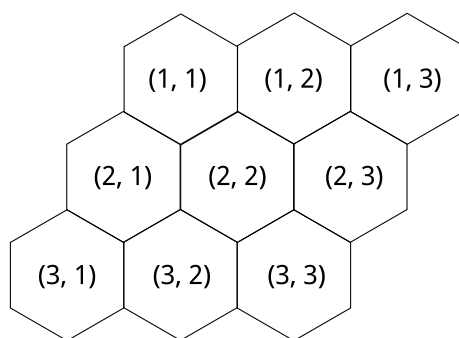


Задача А. Пчелы

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

Ханиленд — небольшое мёдодобывающее государство, населённое пчёлами и имеющее вид, как нетрудно догадаться, медовых сот: n рядов по m правильных шестиугольников в каждом. Каждый из шестиугольников является отдельным и довольно самостоятельным субъектом Медовой Федерации, и вопрос связи между отдельными субъектами стоит очень остро. Для его решения было решено прокопать систему медовых рек, соединяющую все субъекты. Разумеется, отрезок медово-речной системы может соединять только субъекты, граничащие по стороне. Чтобы пустить медовые реки, необходим мёд, а поскольку добывается он непосильным трудом рядовых пчёл, необходимо построить медовый канал так, чтобы для этого потребовалось как можно меньше мёда. К сожалению, пчёлы от природы наделены трудолюбием, а не аналитическим складом ума, поэтому подобная задача им не по силам. Зато вам — в самый раз ;)

Количество мёда, необходимое для заполнения канала между 2 соседними субъектами, вычисляется по довольно интересной формуле: $(x_1x_2 + p_1y_1y_2) \bmod p_2$, где x_1, y_1, x_2, y_2 — координаты первого и второго субъекта соответственно, а p_1, p_2 — простые числа. Первая координата обозначает номер ряда, в котором находится субъект, а вторая — номер субъекта в этом ряду. Ряды нумеруются сверху вниз, а субъекты в них — слева направо, таким образом, левый верхний угол Ханиленда имеет координаты $(1, 1)$. На рисунке изображена карта государства при $n = 3, m = 3$.



Формат входного файла

Входной файл содержит 4 целых числа n, m, p_1 и p_2 ($1 \leq n, m, p_1, p_2 \leq 1000$), причем p_1 и p_2 — простые.

Формат выходного файла

В выходной файл выведите единственно число — минимальное количество мёда, необходимое для осуществления столь грациозного плана.

Примеры

<code>bees.in</code>	<code>bees.out</code>
3 3 11 23	28

Задача В. Максимум в цикле

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

Ничего хорошего не происходит после двух часов ночи. Но порой ночи так прекрасны (или так ужасны) и длинны, что в голову приходят замечательные идеи. Вот и сегодня профессору Фальсификационной и Геокосмической Академии (ФиГА) пришла в голову чудесная идея, не поддающаяся объяснению того, зачем она нужна кому-то, кроме него самого.

Начнём издалека: профессору очень интересно ездить по дорогам, особенно по кольцевым маршрутам, а ещё лучше — когда на дорогах много развязок, перекрёстков, закусочных, пробок и прочего. Каждая дорога характеризуется для профессора своей *ужасностью* — числом, зависящим от этих интересных черт дороги. Так получилось, что ужасности всех дорог различны.

Сеть дорог представляет собой набор двунаправленных дорог, каждая из которых соединяет какие-то две развязки. Из любой развязки можно добраться по дорогам до любой другой развязки. Нет ни одной дороги, соединяющей какую-либо развязку саму с собой.

Теперь профессор решил называть *ужасающими* те дороги, для которых существует маршрут без повторяющихся дорог, начинающийся и заканчивающийся в одной и той же развязке, в котором эта дорога — самая ужасная.

«А не станут ли все дороги ужасающими...» — подумал профессор, но это была настолько грустная мысль, что додумывать её не захотелось. Поэтому он поручил программистам выяснить, какие же дороги всё-таки не являются ужасающими.

Формат входного файла

В первой строке ввода даны два целых неотрицательных числа n и m — количество развязок и количество дорог ($n, m \leq 10^5$). В следующих m строках даны дороги в виде пар чисел — номеров развязок, которые эти дороги соединяют. Развязки нумеруются целыми числами от 1 до n . Дороги перечислены в порядке увеличения ужасности.

Гарантируется, что из любой развязки можно добраться по дорогам до любой другой развязки, а также что нет ни одной дороги, соединяющей какую-либо развязку саму с собой.

Формат выходного файла

В первой строке выведите количество дорог, которые не являются ужасающими. В следующей строке выведите номера этих дорог через пробел в порядке возрастания. Дороги нумеруются с единицы в том порядке, в котором они заданы во вводе.

Примеры

<code>maxincycle.in</code>	<code>maxincycle.out</code>
4 6 1 2 2 1 2 3 2 3 3 4 4 3	3 1 3 5
4 4 1 2 2 3 3 4 4 1	3 1 2 3

Пояснение

В первом тесте дорога 2 является ужасающей, потому что она самая ужасная в маршруте из дорог 1 и 2. Аналогично дорога 4 — самая ужасная в маршруте из 3 и 4, а дорога 6 — самая ужасная в маршруте из 5 и 6. Есть и другие кольцевые маршруты без повторяющихся дорог, например, маршрут из дорог 1, 3, 4 и 2, но самой ужасной в них снова является одна из дорог 2, 4 и 6. Дороги 1, 3 и 5 не являются ужасающими.

Во втором тесте есть всего один кольцевой маршрут — из дорог 1, 2, 3 и 4. Самая ужасная дорога в нём имеет номер 4.

Задача С. Соединение точек

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

Даны N точек на плоскости. Требуется провести отрезки между некоторыми парами точек таким образом, чтобы, во-первых, из любой данной точки в любую можно было пройти по этим отрезкам, а во-вторых, суммарная длина проведённых отрезков была минимальна.

Формат входного файла

В первой строке входного файла задано число N — количество точек ($1 \leq N \leq 200$). Следующие N строк содержат по два числа X_i Y_i каждая через пробел — координаты i -ой точки ($-1000 \leq X_i, Y_i \leq 1000$). Никакие две данные точки не совпадают, никакие три не лежат на одной прямой. Все числа во входном файле целые.

Формат выходного файла

В первой строке выходного файла выведите L — суммарную длину проведённых отрезков с точностью не менее шести десятичных знаков после запятой. Во второй строке выведите K — их количество. В следующих K строках выведите по два числа A_j B_j через пробел в каждой — номера точек, соединённых j -ым отрезком ($1 \leq A_j, B_j \leq N$, $A_j \neq B_j$). Точки нумеруются с единицы в том порядке, в котором они даны во входном файле. Если ответов с минимальным L несколько, разрешается выводить любой из них.

Примеры

<code>connect.in</code>	<code>connect.out</code>
4	3
0 0	3
0 1	1 2
1 0	2 4
1 1	4 3
5	7.064495
0 0	4
0 2	3 1
1 1	3 2
3 0	3 4
3 2	4 5

Задача D. День объединения

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

В Бейтландии есть целых n городов, но нет ни одной дороги. Король страны, Вальдемар Де Беар, решил исправить эту ситуацию и соединить некоторые города дорогами так, чтобы по этим дорогам можно было бы добраться от любого города до любого другого. Когда строительство будет завершено, Король планирует отпраздновать День Объединения. К сожалению, казна Бейтландии почти пуста, поэтому Король требует сэкономить деньги, минимизировав суммарную длину всех построенных дорог.

Формат входного файла

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 5000$) — количество городов в Бейтландии. Каждая из следующих n строк содержит два целых числа x_i, y_i — координаты i -го города ($-10\,000 \leq x_i, y_i \leq 10\,000$). Никакие два города не расположены в одной точке.

Формат выходного файла

Первая строка выходного файла должна содержать минимальную суммарную длину дорог. Выведите число с точностью не менее 10^{-3} .

Примеры

unionday.in	unionday.out
6 1 1 7 1 2 2 6 2 1 3 7 3	9.65685