Задача А. Треугольная агрономия

Имя входного файла:	`agronomy.in`
Имя выходного файла:	`agronomy.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Фермер Джон живет в стране Немерика. По соседству с этой страной находится другая страна, Флатландия. Во Флатландии N городов, и некоторые из них соединены прямыми двусторонними дорогами. Никакие две дороги не пересекаются вне городов, и из любого города можно добраться в любой другой город, используя эти дороги.

Фермер Джон занимается выращиванием кукурузы на своем маленьком поле. Точнее, занимался, так как королю Флатландии Никите Первому пришла мысль выращивать кукурузу на всех полях своей страны, и он пригласил Джона, известного своими прогрессивными начинаниями, руководить этим процессом. Теперь Джон — министр сельского хозяйства Флатландии.

Заняв эту ответственную должность, Джон сразу же принялся за дело. Он решил выделить поля на территории Флатландии. При этом, по его замыслу, каждое поле должно представлять собой конечный участок плоскости, ограниченный дорогами. Однако Флатландия — страна большая, поэтому очень скоро Джон понял, что сам с этой задачей ни за что не справится. Поэтому он пригласил Вас, вручил карту страны и попросил помочь ему в этом нелегком деле.

Внимательно рассмотрев карту, Вы с удивлением обнаружили, что каждое поле ограничено ровно тремя дорогами, то есть, является треугольником. Даже знание агрономии не требуется для того, чтобы понять, насколько задача упрощается от этого!

Помогите фермеру... простите, министру сельского хозяйства Флатландии Джону составить список полей.

Формат входного файла

В первой строке файла находятся два целых числа N и M $(3 \le N \le 10000)$, $(3 \le M \le 50000)$ — число городов и число дорог в Флатландии, соответственно.

В последующих N строках находятся пары целых чисел X_i, Y_i — координаты i-го города. Координаты не превосходят 10000 по абсолютному значению. Никакие два города не совпадают.

В следующих M строках находятся пары целых чисел S_j, F_j — номера городов, соединяемых j-той дорогой. Никакие две дороги не соединяют одну и ту же пару городов. Дороги не могут иметь общих точек нигде, за исключением городов. Города нумеруются, начиная с единицы, в том порядке, в котором они были описаны во входном файле.

Гарантируется, что все поля имеют ненулевую площадь.

Формат выходного файла

В первой строке выходного файла выведите число полей K.

В последующих K строках выведите, разделяя пробелами, по три числа A_i, B_i, C_i — номера дорог, которые ограничивают i-тое поле. Дороги нумеруются с единицы в порядке описания их во входном файле. Номера дорог в описании поля, а также сами описания полей можно выводить в произвольном порядке.

Примеры

agronomy.in	agronomy.out
4 5 0 0 1 0 0 1 1 1 1 2 1 3 2 3 2 4 3 4	2 1 2 3 3 4 5
5 8 1 1 2 2 1 3 3 1 3 3 1 3 5 3 3 2 4 5 1 4 4 2 1 2 2 5	4 1 3 7 2 3 8 8 6 4 7 6 5

Задача B. Крестики и крестики — 2

Имя входного файла:	`crosses.in`
Имя выходного файла:	`crosses.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Иннокентий и Емельян играют в одну очень интересную игру. Перед ними поле размера $2 \times n$, причем все клетки изначально пусты. Игроки по очереди ставят крестики в свободные клетки. Выигравшим считается игрок, который первым поставит 3 крестика в квадрат размера 2×2. Игроки решили, что первый ход будет за Емельяном. К играющим присоединился Вениамин и сказал, что знает, кто победит при оптимальных действиях обоих игроков. А вы можете сказать, кто же выиграет?

Формат входного файла

Во входном файле дано одно целое число n ($2 \le n \le 1000$).

Формат выходного файла

В выходной файл выведите «First player wins», если победит Емельян, и «First player loses» иначе.

Примеры

crosses.in	crosses.out
2	First player wins
4	First player loses

Задача C. Последовательность Хэмминга

Имя входного файла:	`hamming.in`
Имя выходного файла:	`hamming.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Последовательностью Хэмминга называется упорядоченная по возрастанию последовательность натуральных чисел, все простые делители которых не превосходят 5.

Например, число $10 = 2 \cdot 5$ входит в эту последовательность, а число $26 = 2 \cdot 13$ не входит. Начало этой последовательности выглядит так: 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, ...

Ваша задача состоит в том, чтобы для заданного числа x найти номер его позиции в последовательности Хэмминга.

Формат входного файла

Входной файл содержит несколько наборов входных данных. В первой строке входного файла находится число t — количество наборов входных данных ($1 \le t \le 10^5$).

Каждая из последующих t строк содержит целое число x, позицию которого в последовательности Хэмминга необходимо определить ($1 \le x \le 10^{18}$).

Формат выходного файла

Для каждого набора входных данных выведите номер позиции (позиции нумеруются с единицы) соответствующего числа в последовательности Хэмминга или фразу `Not in sequence`, если число не принадлежит этой последовательности.

Примеры

hamming.in	hamming.out
11 1 2 6 7 8 9 10 11 12 13 14	1 2 6 Not in sequence 7 8 9 Not in sequence 10 Not in sequence Not in sequence

Задача D. Пересекающий отрезок

Имя входного файла:	`interfering.in`
Имя выходного файла:	`interfering.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Рассмотрим простой многоугольник P (т.е. многоугольник без самопересечений и самокасаний) и отрезок S, находящийся строго внутри него.

Триангуляцией многоугольника P называется его разбиение на неперекрывающиеся треугольники, объединение которых совпадает с P. В данной задаче мы дополнительно потребуем от триангуляции следующего:

- каждая вершина треугольника из триангуляции должна совпадать с некоторой вершиной P;
- никакая вершина многоугольника P не должна лежать на внутренности никакого ребра треугольника из триангуляции.

Назовем отрезок *пересекающим* триангуляцию, если он имеет хотя бы одну общую точку с границей какого-либо треугольника из триангуляции.

Даны многоугольник P и отрезок S. Ваша задача состоит в том, чтобы определить, существует ли такая триангуляция многоугольника P, что S не пересекает эту триангуляцию.

Известно, что любой простой многоугольник можно триангулировать. Поэтому Вам не нужно выводить всю триангуляцию, если таковая существует. Вам следует лишь вывести такой треугольник, который принадлежит какой-либо триангуляции многоугольника P и содержит отрезок S строго внутри.

Формат входного файла

В первой строке файла находится числе N ($3 \le N \le 100$) — число вершин многоугольника P.

Следующие N строк содержат пары целых чисел (X_i, Y_i) — координаты вершин P. Все вершины различны, никакие три последовательные вершины не лежат на одной прямой. Вершины занумерованы, начиная с 1, в порядке появления их во входном файле.

В последней строке находятся четыре целых числа X_s, Y_s, X_f, Y_f — координаты концов отрезка S.

Все координаты вершин не превосходят по модулю 10^4. Гарантируется, что отрезок S лежит строго внутри многоугольника P, а также имеет ненулевую длину.

Формат выходного файла

Если решение задачи существует, выведите индексы вершин многоугольника, которые составляют треугольник, являющегося решением. Индексы должны быть отделены друг от друга одиночными пробелами.

Если решения не существует, выведите слово «`Impossible`» (без кавычек).

Выходной файл должен содержать ровно одну строку.

Пример

interfering.in	interfering.out
3 0 0 0 3 4 3 1 2 2 2	1 2 3
4 0 0 2 0 2 3 0 3 1 1 1 2	Impossible

Задача E. Новые дороги

Имя входного файла:	`newroads.in`
Имя выходного файла:	`newroads.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

В королевстве Многоземье правит мудрый и справедливый король Август IV. Одним из направлений его внутренней политики является развитие торговых отношений между городами королевства.

Всего в королевстве N городов, и на данный момент налажено M торговых путей. Каждый такой путь проходит от одного города к другому по дороге, соединяющей непосредственно эти города. Любой торговый путь является односторонним, но может существовать другой, обратный ему.

Торговцы используют торговые пути, чтобы добраться из своего родного города до какого-либо другого и продавать там свои товары. Однако торговля идет успешнее, если, продав весь свой товар, торговец может закупиться новым, местным товаром и вернуться домой не с пустыми руками. В этом случае, для безопасности также следует использовать какие-либо торговые пути.

Поэтому исторически сложилось, что если из города A в город B, а затем из B в A можно добраться, используя только торговые пути, то эти города поддерживают особенно развитые торговые отношения. Будем в этом случае говорить, что города A и B являются «дружественными».

Чтобы облегчить тяжелый труд простого торговца, Август IV постановил обустроить дороги и проложить новые торговые пути между некоторыми городами. Однако денег у короля не очень много, поэтому он решил проложить только те из них, которые заведомо себя оправдают, то есть, если они соединяют дружественные города и не дублируют уже существующие пути.

Теперь королю необходимо оценить финансовые затраты на свое начинание. Помогите ему и посчитайте, какое минимальное количество торговых путей надо проложить, чтобы все дружественные города были соединены торговыми путями напрямую.

Формат входного файла

В первой строке входного файла находятся два целых числа N ($1 \le N \le 100000$) и M ($0 \le M \le 200000$) — количество городов и существующих торговых путей соответственно. Города занумерованы от 1 до N включительно.

В каждой из последующих M строк находятся два целых числа A_i и B_i, обозначающие торговый путь, идущий из города с номером A_i в город с номером B_i. Никакой путь не соединяет город с самим собой.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

newroads.in	newroads.out
3 3 1 2 2 3 3 1	3
3 3 1 2 2 3 1 3	0

Задача F. Пиксельная графика

Имя входного файла:	`pixel.in`
Имя выходного файла:	`pixel.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

На плоскости задан многоугольник, все стороны которого параллельны осям координат, а все вершины находятся в точках с целыми координатами.

Если провести все возможные вертикальные и горизонтальными прямыми через точки с целыми координатами, то плоскость будет разбита на единичные квадраты. Необходимо найти все единичные квадраты, которые находятся внутри заданного многоугольника.

Формат входного файла

Первая строка входного файла содержит три целых числа: число вершин многоугольника n $(4 \le n \le 100)$, w и h $(1 \le w, h \le 200)$. Каждая из последующих n строк описывает одну вершину многоугольника и содержит два целых числа x_i и y_i — ее координаты $(0 \le x_i \le w, 0 \le y_i \le h)$.

Вершины заданы в порядке обхода против часовой стрелки. Многоугольник не содержит самопересечений и самокасаний.

Формат выходного файла

В выходной файл выведите прямоугольник размером w на h символов, каждый из которых соответствует одному единичному квадрату. Единичный квадрат, находящийся снаружи многоугольника обозначайте символом `.` («точка»), находящийся внутри — символом `#`. Левый нижний угол прямоугольника должен соответствовать точке $(0,\ 0)$, правый верхний — точке $(w,\ h)$.

Примеры

pixel.in	pixel.out
4 3 4 0 0 2 0 2 1 1 0	... ... ##.
12 3 3 1 0 2 0 2 1 3 1 3 2 2 2 2 3 1 3 1 2 0 2 0 1 1 1	.#. ### .#.

Задача G. Сумма? Это просто!

Имя входного файла:	`prime.in`
Имя выходного файла:	`prime.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

> Сумма всех простых чисел от 2 до 1000000 равна 37550402023 и действительно является простым числом.
>
> из блога http://blog.sjinks.org.ua

Вася занимается в математическом кружке и недавно узнал о *простых* числах. Напомним, что целое число x называется простым, если $x \geq 2$ и x не делится ни на одно число от 2 до $x - 1$.

Вася — очень любознательный мальчик, поэтому он решил найти как можно больше информации о простых числах. Для этого он решил воспользоваться интернетом и ввел в строку поиска одной популярной поисковой системы словосочетание `простые числа`. На одной из найденных страниц он прочитал, что сумма всех простых чисел, не превосходящих миллион, является простым числом, и заинтересовался, насколько часто случаются подобные ситуации.

Помогите Васе — напишите программу, которая по числу n, вычислит сумму всех простых чисел, не превосходящих n и определит, является ли эта сумма простым числом.

Формат входного файла

Первая строка входного файла содержит число n ($2 \leq n \leq 10^4$).

Формат выходного файла

В первой строке выходного файла выведите сумму S всех простых чисел, не превосходящих n. Во второй строке выходного файла выведите слово `YES`, если S является простым числом, и слово `NO` иначе.

Примеры

prime.in	prime.out
3	5 YES
6	10 NO

Задача H. Метро Санкт-Петербурга

Имя входного файла:	`spbmetro.in`
Имя выходного файла:	`spbmetro.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Многие большие города на нашей планете имеют разветвленную сеть метро. Зачастую система метрополитена состоит более чем из десяти линий и содержит несколько сотен станций. В таком случае добраться от одной станции до другой становится достаточно сложно, особенно если требуется сделать это как можно быстрее. В Санкт-Петербурге метро намного меньше, чем, например, в Лондоне, однако описанная проблема актуальна и там.

Ваша задача состоит в том, чтобы написать программу, которая по названиям двух станций петербургского метрополитена определит длину кратчайшего пути между ними (длина измеряется числом перегонов между станциями, входящими в путь).

Формат входного файла

Первая строка входного файла содержит название начальной станции пути, вторая строка входного файла содержит название конечной станции пути.

Входной файл задан в кодировке Win-1251. Все числа в названиях станций записаны арабскими цифрами. В названиях станций не используются сокращения (например, вместо «Пл.» используется «Площадь»). Все тире в названиях станций отделены от предшествующих и последующих слов пробелами. Все слова записаны заглавными буквами.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

spbmetro.in	spbmetro.out
`ПЛОЩАДЬ АЛЕКСАНДРА НЕВСКОГО - 1` `ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ - 2`	`4`
`ЛЕНИНСКИЙ ПРОСПЕКТ` `ПЛОЩАДЬ ЛЕНИНА`	`10`

Задача I. Подстрока

Имя входного файла:	`string.in`
Имя выходного файла:	`string.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

В этой задаче Вам требуется найти максимальную по длине подстроку данной строки, такую что каждый символ встречается в ней не более k раз.

Формат входного файла

В первой строке даны два целых числа n и k ($1 \leq n \leq 100000$, $1 \leq k \leq n$), где n — количество символов в строке. Во второй строке n символов — данная строка, состоящая только из строчных латинских букв.

Формат выходного файла

В выходной файл выведите два числа — длину искомой подстроки и номер ее первого символа. Если решений несколько, выведите любое.

Примеры

string.in	string.out
3 1 abb	2 1
5 2 ababa	4 2

Задача J. Объединение параллелепипедов — 2

Имя входного файла:	`union2.in`
Имя выходного файла:	`union2.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

В трехмерном пространстве задано n прямоугольных параллелепипедов. Найдите объем их объединения.

Формат входного файла

Первая строка входного файла содержит количество параллелепипедов n ($1 \le n \le 15$), Каждая из последующих n строк описывает один из параллелепипедов. Описание параллелепипеда состоит из шести целых чисел: x_1, y_1, z_1, x_2, y_2, z_2 ($0 \le x_1 \le x_2 \le 10^5$, $0 \le y_1 \le y_2 \le 10^5$, $0 \le z_1 \le z_2 \le 10^5$). Точка (x_1, y_1, z_1) является одним из углов параллелепипеда, а (x_2, y_2, z_2) — ему противоположным.

Формат выходного файла

В выходной файл выведите объем объединения заданных во входном файле прямоугольных параллелепипедов.

Примеры

union2.in	union2.out
1 0 0 0 1 1 1	1
2 0 0 0 1 1 1 0 0 0 2 2 2	8