

Задача А Поле чудес

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Максимальное время работы на одном тесте: 1 секунда
Максимальный объем используемой памяти: 64 мегабайта

Для игры в "Поле чудес" используется круглый барабан, разделенный на сектора, и стрелка. В каждом секторе записано некоторое число. В различных секторах может быть записано одно и то же число.

Однажды ведущий решил изменить правила игры. Он сам стал вращать барабан и называть игроку (который барабана не видел) все числа подряд в том порядке, в котором на них указывала стрелка в процессе вращения барабана. Получилось так, что барабан сделал целое число оборотов, то есть последний сектор совпал с первым.

После этого ведущий задал участнику вопрос: какое наименьшее число секторов может быть на барабане? Напишите программу, отвечающую на этот вопрос.

Формат входных данных

Во входном файле записано сначала число N — количество чисел, которое назвал ведущий ($2 \leq N \leq 30000$). Затем записано N чисел, на которые указывала стрелка в процессе вращения барабана. Первое число всегда совпадает с последним (в конце стрелка указывает на тот же сектор, что и в начале). Числа, записанные в секторах барабана, — натуральные, не превышающие 32000.

Формат выходных данных

Выведите минимальное число секторов, которое может быть на барабане.

Примеры

Входные данные	Выходные данные
13 5 3 1 3 5 2 5 3 1 3 5 2 5	6
4 1 1 1 1	1
4 1 2 3 1	3

Задача В Триангуляция

Имя входного файла: `input.txt`
 Имя выходного файла: `output.txt`
 Максимальное время работы на одном тесте: 2 секунды
 Максимальный объем используемой памяти: 64 мегабайта

Вася нарисовал выпуклый N -угольник и провел в нем несколько диагоналей таким образом, что никакие две диагонали не пересекаются внутри N -угольника.

Теперь он утверждает, что весь N -угольник оказался разбит на треугольники.

Напишите программу, которая проверяет истинность Васиного утверждения.

Формат входных данных

Во входном файле записано сначала число N — количество вершин N -угольника ($3 \leq N \leq 1000$). Далее записано число M — количество диагоналей, проведенных Васей. Далее записано M пар чисел, задающих диагонали (каждая диагональ задается парой номеров вершин, которые она соединяет). Гарантируется, что каждая пара чисел задает диагональ (то есть две вершины различны, и не являются соседними), а также что никакие две пары не задают одну и ту же диагональ. Никакие две диагонали не пересекаются внутри N -угольника.

Вершины N -угольника нумеруются числами от 1 до N .

Формат выходных данных

Если Васиное утверждение верно, то выходной файл должен содержать единственное число 0.

В противном случае быть выведено сначала число K — количество вершин в какой-нибудь не треугольной части. Далее должно быть выведено K чисел — номера вершин исходного N -угольника, которые являются вершинами этой K -угольной части в порядке обхода этой части.

Примеры

Входные данные	Выходные данные
3 0	0
4 1 1 3	0
6 2 1 3 5 3	4 1 3 5 6

Задача С Соревнования по программированию

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Максимальное время работы на одном тесте:	2 секунды
Максимальный объем используемой памяти:	256 мегабайт

Вова проводит соревнования и тренировки по программированию в своей школе. Для этого он скачал из Интернета много архивов разных соревнований и сборов по программированию. Он разархивировал все, что скачал, на жесткий диск своего компьютера, и теперь не может разобраться в получившемся наборе файлов. Вова хочет понять, сколько описаний соревнований по программированию он скачал.

Пара файлов называется *тестом*, если они находятся в одном каталоге и имеют полные имена вида «XY» и «XY.a», где «XY» — номер теста (дополненный ведущим нулем, если он меньше десяти). В первом из указанных файлов хранятся входные данные, а во втором — эталонный ответ.

Каталог называется *каталогом с тестами*, если в нем есть тесты со всеми номерами от 1 до N , где $1 \leq N \leq 99$, а других файлов нет (но могут быть подкаталоги).

Каталог называется *задачей*, если в нем есть файл с именем «check» и любым (возможно пустым) расширением и подкаталог «tests», который является каталогом с тестами. В каталоге-задаче помимо этого могут быть другие файлы и подкаталоги.

Каталог называется *описанием соревнования*, если в нем есть хотя бы один подкаталог, и все его подкаталоги являются задачами.

Задано описание всех файлов, хранящихся на жестком диске Вовиноного компьютера. Необходимо найти, сколько описаний соревнований содержится на его жестком диске.

Формат входных данных

Первая строка входного файла содержит n — число файлов ($1 \leq n \leq 1000$). Каждая из последующих n строк содержит полный путь к файлу. Каждая из этих строк содержит от одного до 200 символов.

Элементы пути разделены символами «\». В начале элемента пути идет буква диска (от «A» до «Z»), затем следует двоеточие, затем «\». Имена каталогов в пути и имена файлов состоят из символов с кодами от 33 до 126, за исключением символа «\». Последний элемент пути является полным именем файла. Полное имя файла содержит не более одной точки, при этом до и после точки идет хотя бы один символ. Если имя файла содержит точку, то часть имени после точки называется расширением, а часть до точки — именем файла. Иначе считается, что файл имеет пустое расширение, а имя файла совпадает с его полным именем.

Строчные и заглавные буквы в путях не различаются. Ни в каком каталоге нет файла и подкаталога, имеющих одинаковые имена.

Формат выходных данных

В выходной файл выведите количество описаний соревнований по программированию, которые содержатся в описанном наборе файлов.

Пример

Входные данные	Выходные данные
22 C:\olymp\roi2005\apusb\tests\01 C:\olymp\roi2005\apusb\tests\01.a C:\olymp\roi2005\apusb\tests\02 C:\olymp\roi2005\apusb\tests\02.a C:\olymp\roi2005\apusb\check.exe C:\olymp\roi2005\gcd\tests\01 C:\olymp\roi2005\gcd\tests\01.a C:\olymp\roi2005\gcd\tests\02 C:\olymp\roi2005\gcd\tests\02.a C:\olymp\roi2005\gcd\check.cpp C:\olymp\roi2005\gcd\solution.exe C:\olymp\roi2006\apusb\tests\01 C:\olymp\roi2006\apusb\tests\01.a C:\olymp\roi2006\apusb\tests\03 C:\olymp\roi2006\apusb\tests\03.a C:\olymp\roi2006\apusb\check.exe C:\olymp\roi2006\gcd\tests\01 C:\olymp\roi2006\gcd\tests\01.a C:\olymp\roi2006\gcd\tests\03 C:\olymp\roi2006\gcd\tests\02.a C:\olymp\roi2006\gcd\check.cpp C:\olymp\roi2006\gcd\solution.exe	1

Задача D Формула-3

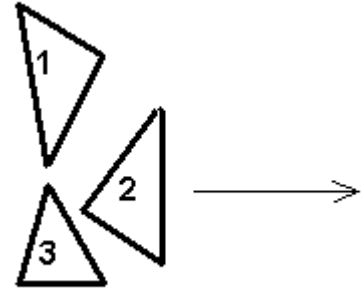
Имя входного файла: `input.txt`
 Имя выходного файла: `output.txt`
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 мегабайта

В ежегодном чемпионате Флатландии (которая, естественно, является плоским миром) по космическим гонкам "Формула-3" участвуют N космических скутеров, имеющие форму треугольников. До начала гонок скутеры занимают положение в стартовой зоне согласно результатам жеребьевки.

Скутеры стартуют строго по порядку. Каждый скутер, получив команду «старт», уезжает в положительном направлении оси Ox . Следующий скутер стартует лишь тогда, когда предыдущий покинет стартовую зону. Скутеры уезжают строго параллельно оси Ox , скутеры в стартовой зоне не поворачивают и не разворачиваются.

Естественно, что если в момент старта на пути скутера окажется другой скутер, то произойдет авария (даже если скутер заденет лишь угол другого скутера своим углом).

Для уменьшения опасности столкновения скутеров на старте строго соблюдается следующее правило: прямые, параллельные оси Ox и пересекающие какой-то скутер, должны *в совокупности* пересекать не более 100 других скутеров (прямая, проходящая через одну точку скутера также считается прямой, пересекающей скутер). Например, на приведенном рисунке прямые, параллельные Ox и пересекающие скутер 2, проходят через 2 других скутера (1 и 3), а прямые, проходящие через скутер 1, проходят только через один другой скутер (номер 2).



Главный Судья гонок хочет определить порядок, в котором должны стартовать скутеры, чтобы аварии не произошло. Например, в ситуации, приведенной на рисунке, сначала должен стартовать скутер номер 2 (если попытается стартовать скутер номер 1 или 3, то он столкнется со скутером номер 2). После этого скутеры 1 и 3 могут стартовать в любом порядке (они друг другу не мешают).

Помогите Главному Судье — напишите программу, которая определит какой-нибудь порядок старта скутеров, чтобы аварии не произошло.

Формат входных данных

В первой строке входного файла записано натуральное число N ($1 \leq N \leq 30\,000$).

В каждой из следующих N строк записано по 6 чисел: $x_1, y_1, x_2, y_2, x_3, y_3$ — координаты трех вершин скутера на старте, целые числа, не превосходящие по модулю 10^6 . В начальный момент скутеры не задевают друг друга.

Формат выходных данных

Выведите через пробел N чисел — номера скутеров в том порядке, в котором они могут стартовать. Если решений несколько, выведите одно любое из них. Если решений нет, выведите одно число -1 .

Примеры

Входные данные	Выходные данные	Комментарий
3 1 19 3 9 6 15 5 6 10 2 10 12 1 1 6 1 3 7	2 1 3	Данный тест соответствует приведенному рисунку. Ответ 2 3 1 в этом тесте также является правильным
3 0 1 -2 1 -1 -1 5 6 10 2 10 12 1 1 6 1 3 7	2 3 1	