

Задача А. Прямоугольники Калевича

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

Слава Малевича и его картины «Черный квадрат» не дает покоя берляндскому художнику Калевичу. Поэтому Калевич решил изобрести собственный оригинальный стиль написания картин. Отныне каждый мазок его кисти оставляет след в виде черного прямоугольника со сторонами, параллельными сторонам картины. Так мазок за мазком Калевич рисует свои шедевры.

Неожиданно для себя художник заметил, что некоторые его мазки не нужны, так как все точки соответствующего прямоугольника уже закрашены черным цветом на момент его нанесения. Поэтому, рисуя очередной бессмертный шедевр, Калевич решил записывать информацию о всех сделанных мазках, чтобы позже ее проанализировать.

К сожалению, мастер не обладает аналитическим складом ума, и он попросил вас сделать эту работу. Напишите программу, которая находит все лишние мазки, то есть такие, прямоугольники которых уже полностью закрашены черным к моменту их нанесения.

Формат входного файла

В первой строке входного файла содержится число N ($1 \leq N \leq 1500$), где N — количество мазков. Далее входные данные содержат информацию о соответствующих прямоугольниках. Каждый прямоугольник задается четырьмя числами x_1, y_1, x_2, y_2 , где (x_1, y_1) , (x_2, y_2) — координаты двух противоположных углов прямоугольника. Все данные прямоугольники имеют положительную площадь. Все координаты по абсолютной величине не превосходят 10^9 . Мазки заданы в порядке возрастания времени их нанесения. Все входные данные разделяются пробелами и/или переводами строк.

Формат выходного файла

В первую строку выведите Q — количество лишних мазков. Далее во вторую строку выведите возрастающую последовательность номеров лишних мазков. Нумерация мазков определяется их порядком во входных данных.

Пример

<code>rects.in</code>	<code>rects.out</code>
4	1
0 0 10 10	2
1 1 9 9	
0 0 20 10	
0 0 10 20	

Задача В. Пожар

Имя входного файла: `fire.in`
Имя выходного файла: `fire.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

На тушение пожара в Бремле одновременно приехало K пожарных отрядов. Известно, что каждый отряд состоит из N пожарников. Каждый пожарник в отряде имеет звание. В соответствии с Берляндским Кодексом Пожарных все пожарные в отряде должны иметь различные звания. Тушить пожар в один момент времени может только один пожарный, который выбирается из одного из отрядов. Каждый отряд тушит пожар в порядке возрастания званий. Как только пожарник заканчивает свою работу, он отправляется домой. Парадоксально, но факт: заработная плата за пожар не зависит от звания пожарника и равна количеству времени, проведенному пожарником на пожаре. Про каждого пожарного в каждом отряде известно время t_{ij} , за какое он справится со своей работой. Прерывать свою работу пожарный не может, то есть, начав тушить пожар, он работает t_{ij} .

Вам, как министру Автоматизации Тушения Пожаров в Бремле необходимо так организовать работу пожарных, чтобы государство потратило как можно меньше на выплату зарплат.

Формат входного файла

В первой строке записана пара натуральных чисел K, N ($1 \leq K, N \leq 50$). Далее в K строках содержится таблица из чисел t_{ij} , где t_{ij} обозначает время работы j -го по званию пожарного в i -ом отряде. Все числа в таблице целые неотрицательные, не превосходящие 100.

Формат выходного файла

В первую строку выведите наименьшую сумму, которую государство должно потратить на тушение пожара. Далее в $K \cdot N$ строках выведите соответствующее расписание. Каждая строка должна содержать пару натуральных чисел i, j . Такая пара чисел в k -ой строке обозначает, что k -ым по порядку тушит пожар j -ый пожарник из i -го отряда.

Пример

fire.in	fire.out
2 2	11
1 2	1 1
1 1	2 1
	2 2
	1 2

Задача С. Шоколадная игра

Имя входного файла: `choco.in`
Имя выходного файла: `choco.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

У молодого доцента Берляндского Государственного Университета Лопухова П.П. есть прямоугольная плитка шоколада, размером $N \times M$ долек. Все дольки одинаковы по размеру, но не одинаковы по массе. Зав. кафедрой профессор Палецвротнеклади предложил доценту поиграть в интересную игру. Они будут ходить по очереди (первым, разумеется, начинает профессор). За ход участник должен отломать какую-то часть шоколадки, разломав ее вдоль горизонтальной или вертикальной линии раздела между дольками, и отломанную часть съесть. Так как начинает игру зав. кафедрой, то Лопухов сказал, что это нечестные правила, так как первым ходом профессор может съесть почти всю шоколадку, и добавил правило, что нельзя отламывать (и, значит, съедать) кусок шоколадки массой больше W грамм. Это правило несколько огорчило профессора, однако, не теряя врожденной бодрости духа, он согласился с этим правилом, но добавил свое: если игрок не может сделать ход (то есть отломать кусочек массы меньшей или равной W грамм), то остаток доедает тот, кто начинал игру, и игра заканчивается. Заметим, что ничего не отламывать нельзя (это подразумевает то, что шоколадка любым разломом делится на части положительной массы).

Вам предлагается написать программу, которая определит, сколько грамм шоколада съест профессор, и сколько — доцент, если они оба будут играть наилучшим образом.

Формат входного файла

В первой строке входного файла записано три целых числа N, M, W ($1 \leq N, M \leq 20, 1 \leq W \leq 10^6$). Далее в N строках содержится по M чисел — массы соответствующих долек шоколадки. Все массы являются натуральными числами, не превосходящими 100.

Формат выходного файла

Выведите два целых числа — массу шоколада, съеденного профессором Палецвротнеклади, и массу шоколада, съеденного доцентом Лопуховым, при оптимальной игре обоих игроков.

Пример

<code>choco.in</code>	<code>choco.out</code>
2 2 10 6 5 5 5	15 6
2 2 10 6 5 5 6	22 0

Задача D. Реорганизация завода

Имя входного файла: `factory.in`
Имя выходного файла: `factory.out`
Ограничение по времени: 1.5 секунды
Ограничение по памяти: 64 мегабайта

Основная рабочая линия берляндского завода по производству промышленных отходов состоит из N станков, некоторые пары из которых соединены однонаправленными транспортерами. Каждый транспортер характеризуется своей длиной. В целях сокращения рабочих мест правление завода решило приостановить работу некоторых транспортеров. Главный инженер высказал мысль, что если два станка до реорганизации были связаны хотя бы одной направленной последовательностью транспортеров, то нарушение этого свойства может привести к падению курса акций завода. Правление завода признало мысль главного инженера здравой и поручило ему предоставить список набора транспортеров, которые можно приостановить без риска для курса акций. Так как стоимость обслуживания транспортера пропорциональна его длине, искомый список должен содержать множество транспортеров наибольшей суммарной длины.

Эту работу главный инженер перепоручил вам, сообщив, что завод устроен таким образом, что никакой набор транспортеров не образует направленный цикл.

Формат входного файла

Во входном файле записана пара натуральных чисел N, M ($1 \leq N \leq 1000$, $1 \leq M \leq 80000$), где N — количество станков на заводе, а M — количество транспортеров. Далее в M строках заданы транспортеры тройками целых чисел A, B, L ($1 \leq A, B \leq N$, $1 \leq L \leq 10000$), где A, B — номера соединяемых станков в порядке направления работы транспортера, а L — длина транспортера. Между парой станков может быть несколько транспортеров.

Формат выходного файла

Выведите в первую строку число Q — количество транспортеров в списке. Далее выведите Q чисел — номера транспортеров, включенных в список. Номера определяются порядком следования транспортеров во входных данных. Если ответов несколько, выведите любой. Номера в списке выводите в порядке возрастания.

Пример

factory.in	factory.out
4 4	1
1 3 2	2
1 2 5	
3 2 2	
1 4 1	

Задача Е. Карта

Имя входного файла: `map.in`
Имя выходного файла: `map.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

В **** году полковник Кругляковский развязал Великую Гражданскую Войну. В ходе ожесточенных сражений не погиб никто.

"Хроники Берляндии", том XIXIVLXLIV.

Главному командующему Берляндии отряд разведчиков передал в руки план с изображением N близлежащих деревень. План был добыт в ставке оппозиции с большим риском для жизни. Известно, что стратегически важный объект противников «Окоп» был изображен на карте в виде прямой. К сожалению, дождь, снег, солнце и контрразведка противника стерли с карты изображение окопа, но разведчики смогли запомнить его отличительную черту: все изображение на карте симметрично относительно окопа.

Фон карты имеет белый цвет, а каждая деревня представлена некоторым черным кругом. Ваша задача найти количество возможных положений окопа. Возможно, карта таким образом «подправлена» контрразведкой, что искомое количество окажется равным 0.

Формат входного файла

В первой строке входного файла содержится натуральное число N ($2 \leq N \leq 200$). Далее в файле содержатся описания деревень в виде X_i, Y_i, R_i , где (X_i, Y_i) — координаты центра i -ой деревни, а R_i ($0 \leq R_i \leq 10$) ее радиус. Все числа целые, по абсолютной величине не превосходящие 10000. Никакие две деревни не пересекаются.

Формат выходного файла

Выведите количество возможных положений окопа.

Пример

<code>map.in</code>	<code>map.out</code>
4 10 10 0 10 -10 0 -10 10 0 -10 -10 0	4
4 10 10 1 10 -10 1 -10 10 0 -10 -10 0	1

Задача F. Ленивый художник

Имя входного файла: lazy.in
Имя выходного файла: lazy.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Говорят, преклонных лет художник Калевич невероятно обленился! Ему лень даже передвигать руку для рисования картин! Да что говорить: ему лень поднять руку от холста! Однако, при всем при этом Калевич не прекращает рисовать картины, и он уже задумал новый шедевр. Он решил нарисовать на холсте N отрезков! Гениально! Восхитительно! Современное искусство еще не знало такого полета мысли!

Художник уже придумал, как конкретно будут расположены на холсте отрезки. Единственное, что его смущает, это тот объем работы, который ему предстоит. Объем работы, по мнению мастера, равен минимальному количеству раз отрыва руки от холста, необходимых, чтобы нарисовать картину, если он хочет, чтобы каждая линия на картине была прорисована ровно K раз. Заметим, что отрезки могут накладываться, и их пересечение рассматривается как одна линия.

Формат входного файла

В первой строке входного файла записана пара натуральных чисел N, K ($1 \leq N \leq 64, 1 \leq K \leq 10^6$). Далее в файле заданы парами координат своих концов N отрезков. Отрезки не могут вырождаться в точки. Все координаты целые, не превосходящие 10^6 по абсолютной величине.

Формат выходного файла

Выведите, какое наименьшее количество раз художник должен отрывать руку от холста, для того, чтобы каждая линия на картине была прорисована ровно K раз

Пример

lazy.in	lazy.out
2 1 0 -10 0 10 -1 -1 1 1	1
2 1 0 -10 0 10 0 -5 0 5	0

Задача G. Укрепление охранных сооружений

Имя входного файла: `guard.in`
Имя выходного файла: `guard.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В соответствии с указом президента от 3.14.15 номер 271828 в связи с вероятным нападением противника следует усилить охрану всех объектов стратегического значения. Один из таких объектов (СВО — стратегически важный объект) находится в вашем ведомстве. На карте этот объект обозначается точкой (X, Y) . В настоящий момент он охраняется N спецвышками, расположенными в точках (x_i, y_i) . Вы решили произвести усиление охраны путем построения прямоугольного забора наибольшего периметра, окружающего СВО. На границе забора СВО находится не может. Прямоугольник на карте должен иметь стороны, параллельные осям координат, и окружать часть плоскости положительной площади. Каждая сторона прямоугольника должна содержать хотя бы одну спецвышку. После непродолжительного размышления вы решили, что один забор менее эффективен, чем два. Таким образом, решено обнести СВО вторым забором наибольшего периметра, удовлетворяющего всем свойствам первого. Второй забор должен находиться строго внутри первого. Тревога не покидала вас, и вы решили продолжить укрепление построением третьего, четвертого и т.д. забора, тем более что процесс когда-нибудь прервется сам собой в тот момент, когда построить очередной забор уже будет нельзя. Сколько же всего заборов было построено?

Формат входного файла

Во входном файле содержится тройка целых чисел N, X, Y ($1 \leq N \leq 10^5$, $-10^5 \leq X, Y \leq 10^5$). Далее содержится N описаний спецвышек, заданных парами своих целочисленных координат x_i, y_i ($-10^5 \leq x_i, y_i \leq 10^5$). Любые точки во входном файле могут совпадать.

Формат выходного файла

Выведите искомое число заборов.

Пример

<code>guard.in</code>	<code>guard.out</code>
5 0 0 -100 0 100 0 0 100 0 -100 10 10	1

Задача Н. Постройка Бремля

Имя входного файла: `breml.in`
Имя выходного файла: `breml.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Берляндское правительство решило воздвигнуть новое архитектурное чудо, исполняющее роль правительственного дворца. Новостройку было решено назвать Бремль. Площадь для постройки Бремля представляет собой треугольник с вершинами в точках (x_1, y_1) , (x_2, y_2) , (x_3, y_3) . В этих точках расположены древние башни. По замыслу архитекторов Бремль должен состоять из N зданий, некоторые из которых соединены стенами. Здания должны быть расположены строго внутри треугольной площади. Разумеется, никакие две стены не должны пересекаться. Архитекторы уже решили, что три древние башни следует включить в архитектурный ансамбль и попарно соединить стенами, они будут образовывать внешний периметр. Позиции остальных N зданий пока неизвестны. Архитекторы решили, что все внутренние площади Бремля (участки, окруженные стенами) должны представлять собой треугольники, а в любом строении (как внутреннем здании, так и в любой древней башне) должно сходиться одинаковое число стен. Так как каждая внутренняя площадь – треугольник, то вдоль ее границы должно располагаться точно 3 башни (т.е. развернутые углы не допускаются).

Помогите архитекторам расположить оставшиеся N строений и соединить некоторые пары из них стенами (здания можно соединять стенами и с башнями), таким образом, чтобы удовлетворить всем указанным выше условиям.

Формат входного файла

В первой строке входного файла содержится натуральное число N ($1 \leq N \leq 1000$). Далее содержится три пары целых чисел (x_1, y_1) , (x_2, y_2) , (x_3, y_3) . Все координаты по абсолютной величине не превосходят 50. Заданный треугольник не вырожден.

Формат выходного файла

Выведите в первую строку -1 если решить задачу невозможно, или количество стен Q , которые следует добавить к проекту в противном случае. Далее выведите Q строк, содержащих описания стен. Каждая стена должна задаваться двумя парами координат точек, которые она соединяет. Координаты зданий, добавленных к проекту, могут быть вещественными. Разумеется, все эти здания должны лежать в треугольнике (x_1, y_1) , (x_2, y_2) , (x_3, y_3) . Выводите координаты не более, чем с тремя знаками после запятой.

Пример

breml.in	breml.out
1	3
-10 0	0 5 10 0
10 0	-10 0 0 5
0 10	0 10 0 5

Задача I. Великая Берляндская Стена

Имя входного файла: wall.in
Имя выходного файла: wall.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Полковник Кругляковский в одной минуте от капитуляции в гражданской войне. Он решает на отчаянный шаг — постройку Великой Берляндской Стены. Стена будет иметь форму окружности и заключать в себе часть городов Берляндии. Каждый город задан своими координатами (x_i, y_i) на плоскости. Полковник определился, что стена будет проходить через два важных города — через историческую родину полковника город Круглесберг и через город Тарасов, потому что этот город просто нравится полковнику. Для устойчивости конструкции необходимо, чтобы стена проходила не менее чем через три города. Полковник приказал подполковнику Берляеву «разузнать обстановку по поводу возможных местоположений стены». Подполковник Берляев приказал майору Понину выполнить пришедший сверху приказ. Майор Понин передал приказ старшему лейтенанту Тукалису, а тот в свою очередь лейтенанту Ложкину. Лейтенант Ложкин приказал прапорщику Звереву. Тот, со свойственным ему чувством долга — вам. Оказалось, что «разузнать обстановку по поводу возможных местоположений стены» обозначает для каждого натурального C найти количество таких возможных положений будущей постройки, что внутри и на границе постройки ровно C городов.

Формат входного файла

В первой строке входного файла содержится натуральное число N ($2 \leq N \leq 15000$), количество городов в стране. Далее в файле заданы описания N местоположений парами целочисленных координат x_i, y_i ($-250 \leq x_i, y_i \leq 250$). Первый и второй город в описании — это столь дорогие сердцу полковника города. Никакой город не лежит на прямой, соединяющей первый и второй город. Местоположения первого и второго города не совпадают.

Формат выходного файла

В первую строку выведите количество различных позиций будущей постройки. Далее для каждого такого C , что соответствующее искомое значение Q_C положительно, выведите пару C Q_C . Пары выводите в порядке возрастания значений C .

Пример

wall.in	wall.out
4 0 0 1 0 0 1 1 1	1 4 1
6 0 0 10 0 10 10 10 10 0 10 6 7	2 3 1 6 1

Задача J. Кинотеатры в Берляндии

Имя входного файла: `cinema.in`
Имя выходного файла: `cinema.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Для нумерации мест в квадратном кинозале $N \times N$ всегда используют целые числа от 1 до N^2 . В Берляндии уже давно решили, что нумеровать места справа налево, сверху вниз (рассматривается вид зала сверху) скучно и неинтересно. По этой причине места в кинотеатрах зачастую нумеруют весьма своеобразными способами. На самом деле, допустим любой способ, лишь бы в каждом ряду номера стояли в возрастающем порядке. Заметим, что ряды в кинозале соответствуют строкам на плане, изображающем кинозал сверху.

Директор кинотеатра « π -онер» узнал, что кинотеатр собирается посетить лидер партии. К сожалению, он не помнит, какая точно это партия: то ли партия максималистов, то ли минималистов. Известно, что гость желает сесть на место, расположенное на плане кинотеатра в K -ом столбце. Директор хочет заготовить два специальных способа нумерации. Первый вариант на тот случай, если лидер из партии максималистов. В этом случае сумма номеров мест в K -ом столбце плана должна быть наибольшей возможной. Второй вариант необходим на случай, если гость из партии минималистов — сумма номеров мест в K -ом столбце плана должна быть наименьшей возможной.

После непродолжительного размышления, директор решил учредить специальную должность составителя нумераций. На эту должность приглашены вы. Составьте искомые нумерации.

Формат входного файла

Во входном файле записано два натуральных числа N, K ($1 \leq N \leq 10^5$, $1 \leq K \leq N$).

Формат выходного файла

Выведите искомые нумерации в виде двух матриц $N \times N$. Числа в строках разделяйте пробелами. Сначала выведите первый вариант нумерации, затем второй. Перед выводом каждой матрицы выведите соответствующую сумму. В случае, если $N > 100$, матрицы получаются слишком громоздкими, и их выводить не следует — выведите только две соответствующие суммы.

Пример

<code>cinema.in</code>	<code>cinema.out</code>
2 2	7 1 3 2 4 6 1 2 3 4

Задача К. Берляндский калькулятор

Имя входного файла: `fract.in`
Имя выходного файла: `fract.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Профессор Кнутов, работающий на кафедре НННП (никому не нужных профессий), изобрел программируемый калькулятор. Особенностью его изобретения является тот факт, что все программы для калькулятора состоят из положительного числа строк вида: $LF = RG_1 \ RG_2 \ \dots \ RG_K$, где $K > 1$. Такая строка при выполнении занесет в переменную LF среднее арифметическое величин $RG_1, RG_2, \dots, RG_N$. В силу целого ряда технических и технологических причин, внутренняя память калькулятора не может хранить два или более равных значения, то есть в любой момент выполнения программы значения всех переменных различны. По аналогичным причинам все значения $RG_1, RG_2, \dots, RG_N$ должны быть различны. Конечно, нельзя использовать неинициализированные переменные. Программа может оперировать с двумя предопределенными константами 0 и 1 (они, разумеется, могут встречаться лишь только правее знака “=”).

После выполнения программы калькулятор выдает на дисплей значение LF (в виде правильной дроби), где LF — соответствующая переменная из последней строки программы.

Профессор Кнутов решил дать вам курсовую работу. Цель данной курсовой работы — написать программу для калькулятора, которая выводит дробь $1/5$. Вы подозреваете, что на следующем курсе он даст вам похожую курсовую, только значение дроби будет другим. Вы решили написать программу, которая создает программу для калькулятора, завершающуюся выводом дроби P/Q .

Помните, что для обозначения переменных следует использовать заглавные латинские буквы, цифры и знак “_” (underscore). Длина любой переменной не должна превосходить 64 символа. Все элементы строки должны быть разделены пробелами.

Формат входного файла

Во входном файле содержится пара натуральных чисел P, Q . Так как профессор Кнутов не любит большие числа, числа удовлетворяют условию: $1 \leq P, Q \leq 60$.

Формат выходного файла

В первую строку выходного выведите “-1”, если такой программы не существует и количество строк в программе в противном случае. Далее, в случае положительного ответа, выведите саму программу. Так как калькулятор работает нестабильно для длинных программ, длина вашей программы не должна превышать 2048 строк. Числитель и знаменатель любой дроби в процессе выполнения программы не должен превышать 100 знаков после сокращения дроби. Суммарное количество величин правее знака “=” во всей программе не должно быть больше 2048. Утверждается, что если программа для данной дроби существует, то найдется программа, удовлетворяющая всем заданным ограничениям.

Пример

<code>fract.in</code>	<code>fract.out</code>
6 8	2 A = 1 0 A_EQ_3_4 = A 1