

Задача А. Канатные дороги

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Недавно Вова купил новый широкоугольный объектив для своего фотоаппарата. Теперь он мечтает заснять запоминающуюся панораму. Для этого он решил подняться на вершину ближайшей горы.

До вершины можно добраться по узкой извилистой тропинке, которая идёт вверх по склону. Вова мысленно ввёл систему координат, направив ось OY вдоль склона по направлению от подножия к вершине, а ось OX — перпендикулярно ей. Тропинка представляет собой ломаную из n звеньев, вершины которой следуют в порядке строгого возрастания y -координаты. Для каждого звена ломаной Вова известно время, которое понадобится, чтобы изменить x -координату на единицу при прохождении этого звена.

Кроме того, Вова может воспользоваться канатными дорогами. Эти дороги начинаются в некоторой точке тропинки, идут строго вверх по склону параллельно оси OY и заканчиваются в точке следующего пересечения с тропинкой. Для каждой канатной дороги известно время путешествия по ней.

Помогите Вова подняться на вершину горы, пока не стемнело! Учтите, что Вова столь целеустремлён, что откажется идти вниз по склону, даже если это поможет ему добраться до вершины быстрее.

Формат входного файла

В первой строке записано целое число n ($1 \leq n \leq 10^5$) — количество отрезков, из которых состоит тропинка. В следующей строке через пробел записано $n + 1$ целое число — x -координаты концов этих отрезков в порядке от подножия склона к вершине. x -координаты любых двух последовательных вершин ломаной различны. В i -й из следующих n строк записаны целые числа v_i и t_i — время, необходимое на изменение x -координаты на единицу при прохождении по i -му отрезку, и количество канатных дорог, начинающихся на i -м отрезке, соответственно. Далее в строке идёт t_i пар чисел, описывающих канатные дороги. Каждая дорога задаётся x -координатой её начала и временем, необходимым на её прохождение. Канатные дороги описаны в порядке от подножия к вершине. Никакие две канатные дороги не начинаются в одной точке. Если канатная дорога начинается в общей точке двух отрезков тропинки, то она присутствует в описании только одного отрезка. Гарантируется, что для все канатные дороги заканчиваются на тропинке. Числа в строках разделены пробелами. Все координаты целые и не превосходят по модулю 10^6 . Все времена целые, положительные и не превосходят 10^6 . Общее количество канатных дорог не превосходит 10^5 .

Формат выходного файла

Выведите единственное целое число — минимальное время, за которое можно добраться до вершины.

Примеры

input.txt	output.txt
4 0 5 15 10 0 1 0 1 1 7 1 1 0 1 0	15
3 0 10 0 10 10 0 10 1 10 1 10 0	101

Задача В. Локальные корни

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Рассмотрим слово w , состоящее из n символов. Мы можем *разбить* его в точке i ($1 \leq i \leq n - 1$) на префикс p длины i и суффикс s длины $n - i$. *Локальным корнем* слова w в точке i называется такое непустое слово r , что:

- p является суффиксом r , или r является суффиксом p , или r равно p ;
- s является префиксом r , или r является префиксом s , или r равно s ;
- r имеет минимально возможную длину.

Ваша задача — найти точку разбиения слова w , в которой длина локального корня максимальна.

Формат входного файла

В единственной строке записано слово w , состоящее из строчных латинских букв. Его длина не меньше двух и не больше 300 000 символов.

Формат выходного файла

Выведите через пробел два числа — точку критического разбиения слова w и длину его локального корня в этой точке. Если возможных ответов несколько, выведите любой.

Пример

<code>input.txt</code>	<code>output.txt</code>
aababaaa	5 6

Локальные корни слова aababaaa в различных точках:

Точка	Локальный корень
1	a
2	babaa
3	ab
4	ba
5	aaabab
6	a
7	a

Задача С. Разрезание параллелепипеда

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Младший брат Эрнеста потерял свой кубик Рубика. На следующий день он притащил домой ножовку. Его целью был прямоугольный параллелепипед Эрнеста, все шесть граней которого были покрашены в разные цвета...

Когда Эрнест увидел, что происходит, брат уже сделал несколько распилов, параллельных граням параллелепипеда. Распилы проходили через весь параллелепипед. После того, как все распилы были сделаны, параллелепипед распался на куски. Помогите Эрнесту определить для каждого куска его исходное место в параллелепипеде.

Формат входного файла

В первой строке описан исходный параллелепипед. Сперва через пробел записаны целые числа w , d и h ($1 \leq w, d, h \leq 1000$) — расстояние между передней и задней гранью, между нижней и верхней гранью и между левой и правой гранью, соответственно. Затем идёт строка из шести символов — цвета граней. Цвет задаётся одним из символов «R», «O», «Y», «G», «B», «V». Передняя нижняя левая вершина имеет координаты $(0, 0, 0)$, а задняя верхняя правая — (w, d, h) . Цвета граней описываются в порядке: передняя, задняя, нижняя, верхняя, левая, правая. Во второй строке записано число n ($1 \leq n \leq 1000$) — количество кусков. В следующих n строках эти куски описываются в том же формате, что и исходный параллелепипед. Если какая-либо грань куска не была внешней, то её цвет задаётся символом «.».

Формат выходного файла

Для каждого куска опишите его расположение в исходном параллелепипеде. Сперва выведите символ, обозначающий грань, которая должна стать передней, затем через пробел символ, обозначающий грань, которая должна стать нижней, а затем выведите координаты передней нижней левой вершины этого куска в исходном параллелепипеде. Для обозначения передней грани используется символ «F», задней — «B», нижней — «D», верхней — «U», левой — «L», правой — «R». Положение кусков следует описывать в том же порядке, в котором они заданы на входе. Если решений несколько, выведите любое. Гарантируется, что решение существует.

Пример

input.txt	output.txt
2 2 2 ROYGBV	F D 0 0 0
2	R F 1 0 0
1 2 2 R.YGBV	
2 2 1 YGVBO.	

Задача D. Лидер армии магов

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	3 секунды
Ограничение по памяти:	64 мегабайта

После того как правительство королевства Магов собрало новую армию, на повестку дня встал вопрос о выборе её единоличного лидера. Два мудрейших Наставника — глава королевства Загамиус и лич Сандро — решили провести собеседование с достойными кандидатами. Они разошлись по разным кабинетам и стали независимо друг от друга вызывать к себе нужного мага или же отпускать того, с кем они уже пообщались. При этом как Сандро, так и Загамиус могли вызывать к себе одного и того же мага несколько раз, а в их кабинетах одновременно могли находиться несколько магов одновременно.

Оба Наставника заранее определили порядок, в котором они будут вызывать и отпускать магов. Если в некоторый момент времени оба Наставника могут совершить очередное действие, то один из них совершает его, а другой ждёт. В некоторых случаях, только один Наставник может совершить очередное действие. Например, если Сандро вызовет к себе мага a , а он в это время будет находиться в кабинете Загамиуса, то Сандро придется ждать до тех пор, пока маг a не освободится.

К сожалению, может возникнуть ситуация, в которой оба Наставника будут ждать друг друга. В этом случае они не смогут выбрать лидера своей армии и проиграют ещё не начавшуюся войну. Вы должны определить, возможно ли такое глупое поражение.

Формат входного файла

В первой строке записано целое число t ($1 \leq t \leq 10$) — количество тестов. Каждый тест начинается строкой, в которой через пробел записаны целые числа n , m и k ($1 \leq n, m, k \leq 1000$) — количество записей в плане собеседования Сандро, количество записей в плане собеседования Загамиуса и количество магов в новой армии. Маги занумерованы целыми числами от 1 до k . Далее в n строках содержится план Сандро: в каждой строке через пробел записаны знак «+» или «-» и целое число i ($1 \leq i \leq k$). Знак «+» означает, что Сандро собирается пригласить i -го мага, а знак «-» — отпустить его. Далее в m строках содержится план Загамиуса в таком же формате. В обоих планах каждый маг покидает кабинет только если он там находится. Гарантируется, что ни один Наставник не вызывает мага, если он уже находится в его кабинете. Также известно, что в случае выполнения всех запланированных действий в кабинетах Наставников не останется ни одного мага.

Формат выходного файла

Для каждого теста в отдельной строке выведите «:-)», если оба Наставника обязательно выполнят свой план и выберут лидера армии. В противном случае выведите «:-(».

Пример

input.txt	output.txt
2	: -)
4 4 2	: -(
+ 1	
+ 2	
- 2	
- 1	
+ 1	
+ 2	
- 2	
- 1	
4 4 2	
+ 1	
+ 2	
- 1	
- 2	
+ 2	
+ 1	
- 2	
- 1	

Задача Е. Целочисленный комплексный определитель

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Пусть $\mathbb{Z}[i]$ — множество целых Гауссовых чисел, то есть, комплексных чисел с целочисленными составляющими. Если $a, b, q, r \in \mathbb{Z}[i]$, $a = bq + r$, $|r| < |b|$, то r называется *остатком* от деления a на b .

Дано число $p \in \mathbb{Z}[i]$ и матрица $X = (x_{ij})$ размером $n \times n$, элементы которой принадлежат множеству $\mathbb{Z}[i]$. Посчитайте остаток от деления определителя матрицы X на p . Напомним, что определитель матрицы X равен

$$\sum_{\pi} \text{sign}(\pi) \cdot x_{1\pi(1)} \cdot x_{2\pi(2)} \cdot \dots \cdot x_{n\pi(n)},$$

где сумма берется по всему множеству перестановок n элементов. Сложение и умножение являются обычными сложением и умножением комплексных чисел.

Формат входного файла

В первой строке записано целое число n ($1 \leq n \leq 50$). В каждой из следующих n строк через пробел записаны n комплексных чисел — элементы матрицы X . В последней строке записано комплексное число p , не равное нулю. Комплексное число задаётся действительной и мнимой частью, записанными через пробел. Все составляющие комплексных чисел не превосходят по модулю 10^4 .

Формат выходного файла

Выведите через пробел действительную и мнимую часть остатка от деления определителя X на p . Если возможных ответов несколько, выведите любой. Гарантируется, что ответ существует.

Пример

input.txt	output.txt
2 2 0 -7 0 1 0 0 -1 3 1	3 0

Задача F. НОД 2010

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Вам предложили работу в секретном проекте Агентства Федеральной Безопасности под кодовым названием «НОД 2010». Основным объектом исследования является набор целых положительных чисел. Вы должны понять, как будет изменяться наибольший общий делитель всех чисел этого набора при добавлении в него новых чисел или удалении лежащих там чисел. В начале эксперимента набор чисел пуст.

Формат входного файла

В первой строке записано целое число q ($1 \leq q \leq 10^5$) — количество операций с набором. Каждая из следующих q строк имеет вид «+ x » или «- x ». В первом случае число x добавляется в набор, а во втором случае — удаляется из него. Число x целое, положительное и не превосходит 10^9 . Гарантируется, что из набора будут удаляться только числа, которые в нём лежат.

Формат выходного файла

Выведите наибольший общий делитель всех чисел набора после каждой описанной операции. Согласно распоряжению 190р, наибольшим общим делителем пустого набора является единица.

Пример

<code>input.txt</code>	<code>output.txt</code>
5	8
+ 8	2
+ 6	2
+ 8	2
- 8	6
- 8	

Задача G. Замкадьё

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В королевстве Замкадьё, расположенном на Великой Равнине, n городов. Давным-давно основатель королевства построил $n - 1$ прямую двустороннюю дорогу. От любого города по этим дорогам можно было добраться до любого другого города. Основатель был очень бережлив, поэтому суммарная длина этих дорог была минимально возможной.

Король Замкадьё Карл I на старости лет решил отправиться в путешествие и посетить все города своего королевства. Он начал своё путешествие в городе Понаеховске, где располагался королевский дворец, и действовал следующим образом:

- Если Карл I приезжал в город, где до этого не был, он детально описывал свои впечатления в дневнике и запоминал город, откуда он попал в этот.
- Если из города, где находился Карл I, какая-нибудь дорога вела в город, где он ещё не был, то он ехал по ней. В противном случае Карл I возвращался в город, откуда он впервые попал в данный.

Вскоре после путешествия Карл I умер, завещав свой дневник сыну, Карлу II.

Прошло много лет. Карл II состарился и решил посетить все города королевства в том порядке, в котором они были описаны в дневнике его отца. Техника за прошедшие годы шагнула далеко вперёд, и Карл II собрался путешествовать на дирижабле — на нём можно от любого города до любого другого лететь напрямик, не пользуясь дорогами. Карл II был более ленив, чем его отец, поэтому решил посетить все города ровно по одному разу, после чего вернуться в Понаеховск.

К вам за помощью обратилась жена Карла II. Она хочет знать, когда вернётся её муж, но ей неизвестен порядок городов, в котором они описаны в дневнике. Вычислите минимально возможное расстояние, которое придётся преодолеть Карлу II.

Формат входного файла

В первой строке записано целое число n ($2 \leq n \leq 1000$) — количество городов в Замкадьё. Города занумерованы числами от 1 до n , Понаеховск имеет номер 1. В i -й из следующих n строк через пробел записаны два целых числа — координаты i -го города. Координаты не превосходят по модулю 10^4 . Далее следует пустая строка, после которой в $n - 1$ строке описываются дороги Замкадьё. Каждая дорога задаётся парой номеров городов, которые она соединяет, записанных через пробел.

Формат выходного файла

Выведите минимальное расстояние, которое придётся преодолеть Карлу II на дирижабле, с относительной погрешностью не более 10^{-9} .

Пример

input.txt	output.txt
4 0 0 1 0 1 1 0 1 1 2 1 4 4 3	4.000

Задача Н. Охота на мух

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 6 секунд
Ограничение по памяти: 64 мегабайта

Пахом и его друзья решили поохотиться на мух. Каждый из них взял свою любимую мухобойку, после чего они подкрались к стене, на которой сидели n ничего не подозревающих мух. По команде друзья одновременно ударили своими мухобойками по стене так сильно, что на стене остались следы от ударов. След каждой мухобойки представляет собой простой многоугольник с внутренностью. Следы никакой пары мухобоек не имеют общих точек. Муха была убита, если она находилась внутри или на границе следа. Помогите друзьям вычислить, сколько мух убил каждый из них.

Формат входного файла

В первой строке записано целое число n ($1 \leq n \leq 10^5$) — количество мух. В каждой из следующих n строк через пробел записаны координаты очередной мухи. Никакие две мухи не расположены в одной точке. В следующей строке записано целое число m ($1 \leq m \leq 30\,000$) — количество мухобоек. Каждая из следующих m строк описывает след очередной мухобойки. След задаётся количеством вершин в соответствующем ему многоугольнике и координатами этих вершин в порядке обхода против часовой стрелки. Суммарное количество вершин во всех многоугольниках не превосходит 10^5 . Все координаты целые и не превосходят по модулю 10^7 .

Формат выходного файла

Для каждой мухобойки выведите количество убитых ей мух. Описывайте мухобойки в том же порядке, в котором они заданы на входе.

Пример

input.txt	output.txt
3	0
0 0	2
1 1	1
4 4	
3	
3 -2 -2 -3 -3 -2 -3	
3 0 0 2 0 1 1	
3 3 0 5 0 4 6	

Задача I. Охота на зайцев 2

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 8 секунд
Ограничение по памяти: 64 мегабайта

Хороший охотник убивает одним выстрелом двух зайцев. Но чтобы стать лучшим охотником в мире, вам нужно научиться убивать максимально возможное количество зайцев одним выстрелом. В этом вам поможет программа *Rabbit Hunt 2*. Программа считает всех зайцев точками на плоскости. Ей на вход подаются координаты n зайцев, после чего она генерирует q возможных вариантов выстрела и выбирает из них тот, который убьёт наибольшее количество зайцев. Выстрел задаётся точкой, где должен стоять охотник, и направляющим вектором луча, по которому летит пуля. Считается, что выстрел убивает всех зайцев, лежащих на этом луче, в том числе зайца, расположенного в той же точке, что и охотник.

Генератор позиций использует числа $a_x, b_x, a_y, b_y, a_{vx}, b_{vx}, a_{vy}, b_{vy}$. Координаты охотника в первой позиции — (x_1, y_1) , а направление первого выстрела — (vx_1, vy_1) . Координаты охотника и направляющий вектор для i -го выстрела ($i > 1$) вычисляются по формулам:

$$\begin{aligned}x_i &= (a_x * (x_{i-1} + 10\,000) + b_x) \bmod 20\,001 - 10\,000, \\y_i &= (a_y * (y_{i-1} + 10\,000) + b_y) \bmod 20\,001 - 10\,000, \\vx_i &= (a_{vx} * (vx_{i-1} + 10) + b_{vx}) \bmod 21 - 10, \\vy_i &= (a_{vy} * (vy_{i-1} + 10) + b_{vy}) \bmod 21 - 10.\end{aligned}$$

Проблема в том, что программы *Rabbit Hunt 2* ещё не существует. Вам предстоит её написать.

Формат входного файла

В первой строке записано целое число n ($1 \leq n \leq 10^4$) — количество зайцев. В каждой из следующих n строк через пробел записаны координаты очередного зайца. Никакие два зайца не расположены в одной точке. В следующей строке записано целое число q ($1 \leq q \leq 10^6$) — количество генерируемых позиций. В следующей строке через пробел записаны целые числа x_1, y_1, vx_1, vy_1 ($-10^4 \leq x_1, y_1 \leq 10^4$; $-10 \leq vx_1, vy_1 \leq 10$). В последней строке через пробел записаны неотрицательные целые числа $a_x, b_x, a_y, b_y, a_{vx}, b_{vx}, a_{vy}, b_{vy}$. Все эти числа не превосходят 10^5 . Гарантируется, что ни в одной из первых q позиций направляющий вектор выстрела не будет нулевым.

Формат выходного файла

Выведите через пробел два числа — номер позиции, для которой количество убитых зайцев будет максимально, и это количество. Если таких позиций несколько, выведите позицию с наибольшим номером.

Пример

input.txt	output.txt
4 0 0 2 0 3 0 2 2 3 -1 -1 1 0 1 0 1 1 1 0 1 0	2 3

Задача J. Космический туризм

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

В звёздной системе Бетельгейзе n планет. Каждая планета населена некоторой разумной расой, на разных планетах могут обитать одинаковые расы. Все планеты вращаются вокруг Бетельгейзе по круговым орбитам с одинаковой угловой скоростью и лежат на луче, начинающемся из звезды. В министерство космического туризма часто обращаются туристы-одиночки из других галактик, которые хотят, чтобы их вместе с их личным кораблём доставили на какую-нибудь планету, откуда они смогли бы отправиться в путешествие по планетам системы.

Корабли всех туристов снабжены небольшим топливным баком, который позволяет покрыть без дозаправки расстояние не более d астрономических единиц. На любой планете системы можно заправить топливный бак до максимума. Кроме того, для каждого корабля известен диапазон расстояний до звезды, в которых он может нормально функционировать, используя энергию излучения звезды. Туриста можно доставить на планету только в том случае, если расстояние от этой планеты до звезды попадает в данный диапазон.

Министерство поручило вам вычислить для каждого туриста максимальное количество рас, с которыми он сможет установить контакт.

Формат входного файла

В первой строке через пробел записаны целые числа n , s и d ($1 \leq n, s, d \leq 10^5$) — количество планет в системе Бетельгейзе, количество известных науке рас и максимальное расстояние, которое могут пролететь корабли туристов без дозаправки. В каждой из следующих n строк через пробел записаны два целых положительных числа — расстояние от очередной планеты системы до звезды и номер расы, населяющей эту планету. Расы занумерованы целыми числами от 1 до s . Расстояния от Бетельгейзе до всех планет различны и не превосходят 10^6 астрономических единиц. В следующей строке записано целое число m ($1 \leq m \leq 10^5$) — количество туристов. В каждой из следующих m строк через пробел записаны два целых положительных числа — минимальное и максимальное расстояние до звезды, на котором может находиться корабль очередного туриста. Эти расстояния не превосходят 10^6 .

Формат выходного файла

Для каждого туриста выведите в отдельной строке максимальное количество рас, с которыми он сможет познакомиться

Пример

input.txt	output.txt
5 5 11	2
10 1	1
50 2	
30 1	
20 1	
60 3	
2	
5 65	
15 45	