

Задача А. Ключ к разгадке

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Джонни в очередной раз решил проверить свой интеллект, приняв участие в невероятном квесте. Как опытный квестоман, Джонни вмиг разгадал все загадки, кроме последней. В этой последней задачке он нашёл листочек с написанной на нём строкой. Примечательно, что строка состоит только из строчных букв латинского алфавита и цифр.

На листе также была дана подсказка – нарисован символ кавычки, и дано пояснение. Джонни понял, что ему требуется расставлять кавычки в заданной строке. Когда кавычки расставлены, строка преобразуется следующим образом: если перед подстрокой t , вокруг которой стоят кавычки, следует некоторое число a , то подстрока t повторяется a раз. В качестве числа a может быть выбрана любая непрерывная последовательность цифр, стоящая непосредственно перед подстрокой с кавычками. Такая операция производится для всех подстрок t , заключённых в кавычки, а остальные символы остаются в строке без изменения. Вложенные кавычки запрещены. Ключом к сейфу загадки будет число, обозначающее длину максимальной строки, которую можно получить из исходной при таком подходе. Решение этой задачи оказалось не по зубам Джонни, поэтому он позвонил вам и попросил помочь.

Помогите Джонни! Посчитайте максимальную длину строки, которую можно получить описанным образом из исходной. Известно, что ключ к сейфу состоит не более чем из 13 цифр.

Формат входных данных

Во входном файле содержится непустая строка, длина которой не превосходит 10^5 символов.

Формат выходных данных

В выходной файл необходимо вывести одно целое число n — длину максимальной строки, которую можно получить расстановкой кавычек в заданной строке и последующим её преобразованием.

Гарантируется, что количество цифр в десятичной записи числа n без лидирующих нулей не превосходит 13.

Система оценки

Решения, корректно работающие для строк, длина которых не превышает 20, будут оцениваться в 40 баллов.

Решения, корректно работающие для строк, длина которых не превышает 1000 символов, будут оцениваться в 70 баллов.

Примеры

input.txt	output.txt
10a29b99c	139
011b	12

Замечание

В первом примере необходимо расставить кавычки следующим образом: 10«a29b»99«c», тогда подстрока «a29b» повторяется 10 раз, а буквы «с» — 99 раз, в итоге получается строка длины 139. Заметьте, что в таком случае нельзя взять букву «b» в дополнительные кавычки, так как вложенные кавычки запрещены.

В случае расстановки 10«a»29«b»99«c» количество символов в итоговой строке было бы равно 138, поэтому такой вариант не является оптимальным.

Во втором примере можно расставить кавычки следующим образом: 011«b». В этом случае, в качестве числа a перед подстрокой «b» могут выступать числа 1, 11 или 011. Выгоднее всего взять число 11, тогда подстрока «b» повторится 11 раз, а 0 останется без изменения в начале строки, итого после трансформации получается строка 0bbbbbbbbbbb, её длина равна 12.

Задача В. Города в пустыне

Имя входного файла: `input.txt`
 Имя выходного файла: `output.txt`
 Ограничение по времени: 4 секунды
 Ограничение по памяти: 256 мегабайт

Совсем недавно Джонни читал книгу про загадочный Восток, и его заинтересовал следующий факт. Торговые караваны при переходе по пустыне могут пополнять свои запасы воды и пищи только в оазисах. Следовательно, они должны рассчитывать, какое количество припасов необходимо нести с собой при переходах.

Теперь Джонни раздумывает, какое же количество воды необходимо караванам иметь с собой на самый долгий переход между парой оазисов, если на единицу пути тратится одна единица воды. Для простоты Джонни представляет оазисы точками на плоскости, а караван обязательно начинает свой переход из оазиса и движется к другому оазису по кратчайшему пути.

Помогите Джонни найти минимальное количество воды, которое потребуется для самого долгого перехода между парой оазисов, если он не знает, между какими двумя оазисами будет лежать путь каравана.

Формат входных данных

В первой строке входного файла записано целое число N – количество оазисов в пустыне ($2 \leq N \leq 2000$).

В следующих N строках заданы координаты оазисов. Каждая строка содержит пару целых чисел, по модулю не превосходящих 10^9 . Гарантируется, что координаты всех оазисов различны.

Формат выходных данных

В выходной файл требуется вывести одно вещественное число – количество воды, требуемое для самого долгого перехода. Ответ будет считаться правильным, если его абсолютная или относительная погрешность не превышает 10^{-9} .

Система оценки

Решения, корректно работающие для $N \leq 100$, будут оцениваться в 40 баллов.

Примеры

input.txt	output.txt
3 -1 1 -1 -2 2 1	4.2426406871
6 1 1 -1 0 -3 -1 -2 -2 2 3 4 -2	7.0710678119

Задача С. Джонни и записная книжка

Имя входного файла: `input.txt`
 Имя выходного файла: `output.txt`
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 256 мегабайт

Джонни — толковый математик. Сегодня он решил поработать с целыми числами, которые случайно оказались в его записной книжке. Джонни будет изменять числа по шагам. За один шаг он может взять одно из представленных чисел, возвести его в степень t и заменить старое число на результат. Чтобы избежать беспредела в своей записной книжке, Джонни решил, что в качестве t на каждом шаге он будет использовать одно из чисел: 0.6, 0.7, либо 0.8. Конечно, при возведении в такую степень может получиться нецелое число, но и тут Джонни всё предусмотрел — он решил, что после каждого шага будет округлять полученное число вниз до ближайшего целого.

Джонни хочет, чтобы после его махинаций с числами, в записной книжке как можно больше значений делилось на число M . Конечно, может существовать множество вариантов преобразования исходных чисел требуемым образом. В таком случае, его будет интересовать вариант, в котором количество шагов минимально.

Помогите Джонни — напишите программу, которая поможет ему посчитать требуемую информацию.

Формат входных данных

В первой строке входного файла записаны два целых числа N и M — количество чисел в записной книжке и число, на которое должны делиться числа после изменений ($1 \leq N \leq 10^5$, $1 \leq M \leq 10^{18}$).

В следующей строке задано N целых положительных чисел, не превосходящих 10^{13} .

Формат выходных данных

В выходной файл требуется вывести через пробел два целых числа — максимальное количество чисел, которое может делиться на M после преобразований Джонни и минимальное суммарное количество шагов, которое придётся при этом выполнить.

Система оценки

Решения, корректно работающие для $N = 1$, будут оцениваться в 30 баллов.

Решения, корректно работающие для чисел, не превосходящих 10^7 , будут оцениваться в 30 баллов.

Пример

<code>input.txt</code>	<code>output.txt</code>
1 7333 478473239	1 2

Замечание

В заданном примере число 478473239 сначала может быть возведено в степень 0.8, тогда получится значение 8787920.555, которое Джонни округляет до 8787920. Полученное число может быть возведено в степень 0.6, тогда после округления получится 14666, оно без остатка делится на 7333.

Заметим, что за один шаг невозможно получить число, которое бы без остатка делилось на 7333.

Значит, ответ в данном примере — 1 2, то есть, максимум мы можем получить одно число, которое будет делиться на M , для этого потребуется два шага.

Задача D. Дорожная реформа

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Всем известно, что в Берляндии N городов, пронумерованных числами от 1 до N . Столицей является город с номером 1. Дорожная система страны состоит из $N - 1$ дороги, используя которые можно добраться из любого города до любого другого. По каждой дороге можно ездить в обе стороны, однако за проезд в любую сторону нужно заплатить определённую сумму, зависящую от дороги.

Текущие цены за проезд не нравятся Королю Берляндии, поэтому он решил провести реформу, согласно которой можно выбрать некоторую дорогу и заменить цену проезда по ней на M , однако сделать это можно не более K раз. Естественно, он хочет добиться того, чтобы суммарная стоимость проезда от столицы до остальных городов была максимальной. Помогите ему заменить цены проезда так, чтобы суммарная стоимость проезда от столицы до остальных городов была наибольшей.

Формат входных данных

Первая строка входного файла содержит три числа N , K и M – число городов в Берляндии, количество дорог, на которых можно поменять цену проезда, и величина изменения цены ($1 \leq N \leq 2 \cdot 10^5$, $0 \leq K \leq N - 1$, $1 \leq M \leq 10^7$). Далее в $N - 1$ строке содержится описание дорог. В i -ой строке записаны три целых числа u_i , v_i и c_i – номера городов между которыми проложена дорога, и цена проезда по ней до реформы ($1 \leq u_i, v_i \leq N$, $u_i \neq v_i$, $1 \leq c_i \leq 10^7$). Дороги пронумерованы числами от 1 до $N - 1$ в порядке описания.

Формат выходных данных

В первую строку выходного файла необходимо вывести два числа – наибольшую суммарную стоимость проезда от столицы до других городов после реформы и требуемое количество замен Q стоимости проезда. Во второй строке должны быть записаны в произвольном порядке Q чисел – номера дорог, на которых нужно поменять цену.

Система оценки

Решения, верно работающие для $1 \leq n \leq 20$, будут оцениваться в 20 баллов. Решения, верно работающие для $1 \leq n \leq 2000$, будут оцениваться в 60 баллов.

Примеры

input.txt	output.txt
7 1 2 1 2 3 1 3 2 1 4 4 1 5 1 6 5 15 7 5 10	40 1 4
2 1 1 1 2 10	10 0
1 0 10	0 0

Замечание

В первом тесте выгодно поменять цену проезда из первого города в пятый. Во втором тесте выгодно оставить цену прежней. В третьем тесте нет дорог, а значит суммарная стоимость проезда равна 0.

Задача Е. Новогодние чудеса

Имя входного файла: `input.txt`
 Имя выходного файла: `output.txt`
 Ограничение по времени: 5 секунд
 Ограничение по памяти: 256 мегабайт

Для кого-то Новый Год – праздник, а для кого-то – работа. В конце дня каждому хочется добраться до дома как можно быстрее, и Дед Мороз – не исключение. Однако перед новогодней ночью случилось чудо, и сани Деда Мороза утратили способность летать, ему придётся добираться до дома по земле.

Город, в котором работает Дед Мороз, состоит из домов, соединённых дорогами. Изначально дороги в городе были двусторонними, и по ним можно было добраться из любого дома до любого другого. Но в новогоднюю ночь свершилось другое чудо – дороги в городе стали односторонними. Поэтому, возможно, между некоторыми домами исчезли пути.

К счастью, в новогоднюю ночь магия Деда Мороза никуда не пропадает, и он может менять направления любых дорог на противоположные. Только делать это часто он не хочет – уж слишком много энергии требуется. Поэтому Дед Мороз хочет развернуть минимальное количество дорог так, чтобы можно было добраться от дома, где он закончил работу, до своего дома.

Деду Морозу некогда решать задачи, поэтому он попросил вас помочь ему.

Формат входных данных

В первой строке входного файла заданы два числа N и M – количество домов и количество соединяющих их дорог ($1 \leq N, M \leq 10^6$).

В каждой из следующих M строк заданы пары чисел a и b ($1 \leq a, b \leq N$). Каждая пара означает, что есть направленная дорога из дома a в дом b .

В последней строке заданы два числа s и t – номер дома, где Дед Мороз закончил работу, и номер дома, где он живет ($1 \leq s, t \leq N$).

Гарантируется, что если бы дороги были двунаправленными, то из любого дома можно было бы добраться до любого другого.

Формат выходных данных

В выходной файл требуется вывести одно целое число – количество дорог, которые требуется развернуть, чтобы добраться из дома s в дом t .

Система оценки

Решения, корректно работающие на всех тестах для $n \leq 10000$, будут оцениваться в 30 баллов.

Примеры

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

Задача F. Новогодний подарок

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

На Новый Год принято дарить подарки. Чем больше подарки по объёму, тем они круче, не так ли? Но какой же подарок без сюрприза? — Правильно, никакой. Именно поэтому наибольший по объёму подарок нужно спрятать в шкафу, тогда до праздника его никто не найдёт.

Шкаф представляет собой прямоугольный параллелепипед со сторонами a , b и c . Подарок тоже должен представлять собой прямоугольный параллелепипед, ведь он будет упакован в коробку. Нижняя грань этой коробки должна лежать в шкафу параллельно нижней грани шкафа, иначе подарок может испортиться, пока будет ждать своего часа.

Проблема заключается в том, что шкаф не пустой, в нём уже лежит некоторое количество вещей. Необходимо найти в этом шкафу максимальное по объёму пустое место в форме прямоугольного параллелепипеда для коробки с подарком.

Шкаф разбит на равные по объёму области. Каждая область представляет собой кубик размера $1 \times 1 \times 1$. Для каждой из областей известно свободна ли она или занята вещами.

Формат входных данных

В первой строке входного файла заданы три целых числа a , b и c — размеры сторон шкафа ($1 \leq a, b, c \leq 100$).

Далее шкаф будет описан слоями. Всего будет задано a слоёв, размера $b \cdot c$ каждый. Каждый слой представлен прямоугольным полем размера $b \cdot c$ — это b строк из нулей и единиц по c символов в каждой строке. Для каждого слоя с номером i , строки с номером j в этом слое и k -го символа в этой строке цифра 1 означает, что область шкафа (i, j, k) занята вещами, а 0 — свободна.

Формат выходных данных

В выходной файл нужно вывести одно целое число — максимальный размер пустой области в форме прямоугольного параллелепипеда внутри шкафа.

Система оценки

Решения, корректно работающие для $1 \leq a, b, c \leq 5$, будут оцениваться из 20 баллов.

Решения, корректно работающие для $1 \leq a, b, c \leq 20$, будут оцениваться из 40 баллов.

Решения, корректно работающие для $1 \leq a, b, c \leq 100$, будут оцениваться из 100 баллов.

Примеры

input.txt	output.txt
2 2 2 00 00 11 11	4
2 2 2 00 00 00 00	8
3 3 3 000 010 111 010 010 111 010 010 111	6

Задача G. Квадратные уравнения

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Тем школьникам, кто вёл себя хорошо в 2018 году, Дед Мороз придумал особые подарки – квадратные трёхчлены, т.е. выражения вида $x^2 + bx + c$, где b, c – некоторые коэффициенты, а x – переменная.

Но уравнения он хочет выбрать особые, такие, которые имеют целые корни, при этом каждый корень такого уравнения должен быть равен некоторому уникальному элементу из волшебного списка Деда Мороза. Этот список состоит из N элементов, каждый из которых – целое число. Известно, что в списке могут встречаться одинаковые числа. Помогите Деду Морозу сосчитать количество различных уравнений, которые он сможет подарить школьникам на Новый год.

Формат входных данных

В первой строке входного файла записано одно целое число N ($1 \leq N \leq 2 \cdot 10^5$).

В следующей строке через пробел записаны N целых чисел из списка Деда Мороза, по модулю не превосходящих 10^9 .

Формат выходных данных

В выходной файл необходимо вывести ответ на задачу – количество различных уравнений, каждое из которых имеет два корня – два элемента из списка Деда Мороза. Элементы в списке, соответствующие корням одного уравнения, должны быть разными, но могут совпадать по значению. Два уравнения считаются различными, если у них отличается хотя бы один коэффициент.

Примеры

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

Замечание

Решения, корректно работающие для списка размера не более 500, содержащего числа, по модулю не превышающие 30000, будут оцениваться из 25 баллов.

Решения, корректно работающие для списка размера не более 10000, будут оцениваться из 60 баллов.

Решения, корректно работающие на всех тестах, набирают 100 баллов.

Задача Н. Картофель на Марсе

Имя входного файла: `input.txt`
 Имя выходного файла: `output.txt`
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 256 мегабайт

Один астронавт, участвовавший в исследовательской миссии на Марсе, попал в затруднительное положение на красной планете, оставшись там совсем один.

Из еды у астронавта был лишь небольшой запас картошки – S штук. Узнав о его бедственном положении, с Международной космической станции за ним немедленно отправилась спасательная экспедиция, но до Марса она долетит только через M дней. Чтобы продержаться это время, питаясь одной картошкой, астронавту придется съесть в день P картофелин. Однако он имеет ученую степень по биологии и может выращивать картошку даже на Марсе. Из каждой посаженной им картофелины через X дней вырастает Y новых картофелин. Теперь астронавта очень интересует вопрос – сможет ли он продержаться таким образом или пора начинать из подручных материалов строить ракету?

Помогите астронавту решить эту задачу как можно скорее.

Формат входных данных

В первой строке входного файла задано число T – количество тестовых случаев ($1 \leq T \leq 1000$). Далее следует T строк, каждая из которых содержит по пять целых чисел:

- S – начальное количество картофелин у астронавта ($1 \leq S \leq 10^{12}$),
- M – количество дней, которое ему необходимо продержаться ($1 \leq M \leq 10^{18}$),
- P – количество картофелин, которое необходимо съесть астронавту каждый день ($1 \leq P \leq 10^{12}$),
- X – срок, через который созревает картофель на Марсе ($1 \leq X \leq 100$),
- Y – количество картофелин, которое даёт каждый куст ($1 \leq Y \leq 100$).

Формат выходных данных

В выходной файл необходимо вывести T строк, каждая из которых содержит либо **Yes**, если астронавту удастся дождаться спасательной экспедиции, либо **No** – его запасов не хватит, чтобы продержаться нужное количество дней.

Система оценки

Решения, корректно работающие для тестов, в которых каждый из входных параметров не превышает 10, будут оцениваться в 20 баллов. Решения, корректно работающие для $M \leq 10^5$, будут оцениваться в 50 баллов.

Пример

input.txt	output.txt
3	Yes
4 2 2 1 1	Yes
2 5 1 1 6	No
3 10 2 1 2	

Замечание

В первом примере астронавту требуется прожить 2 дня, съедая по 2 картофелины. Изначально у него есть ровно 4 картофелины, и ему достаточно этого количества, чтобы дождаться спасения.

Во втором примере для того, чтобы прожить необходимые 5 дней, астронавт может съесть одну картофелину, а вторую посадить. Тогда на следующий день у него будет уже 6 картофелин, которых хватит на остальные 4 дня.

В третьем примере астронавту нужно выживать длительный срок — 10 дней. Однако после того, как он съест две из трех картофелин в первый день, у него останется одна картофелина, которую можно посадить. На следующий (второй) день, он будет иметь всего лишь 2 картофелины, которые он будет вынужден съесть, чтобы прожить до третьего дня, а уже на третий день картошки не останется.

Задача 9. Определение опытного игрока Dota2

В игре [Dota 2](#) участвуют две стороны: команды сил Света (The Radiant) и сил Тьмы (The Dire). Каждая команда состоит из пяти игроков — персонажей со своими уникальными способностями. Перед началом матча каждый игрок выбирает себе одного героя из 116 возможных. Цель игры — разрушить трон, находящийся на базе противника.

Повышая игровые навыки, игроки повышают свой рейтинг и получают медали опыта. Эксперты по игре Dota 2 утверждают, что легко могут отличить матч новичка от игры опытного игрока. Ваша задача — построить алгоритм, который сможет определять опытность игрока по статистике из одного матча.



Медали новичка

Медали опытного игрока

Для решения задачи участникам предоставляется набор из почти 50'000 примеров матчей, уровень игрока в которых известен. Для каждого из $\approx 21'000$ "тестовых" примеров, участникам необходимо проанализировать данные о матче при помощи своих алгоритмов и дать ответ — опытный ли был игрок. Допускается использование любых языков программирования, библиотек, методов машинного обучения, разрешено использование внешних данных из сети интернет.





Доступные данные

Участникам предоставляются наборы данных с примерами матчей, которые запакованы в файл [dota2_skill_data.zip](#). Каждый пример описывает характеристики матча и статистику одного из игроков на момент окончания матча. Все примеры имеют уникальный идентификатор “id”.

Всего два набора:

- 1) обучающий набор (train), в котором для каждого примера известен тип медали опыта игрока
- 2) тестовый набор (test), для которого участникам неизвестна опытность игрока — ее необходимо вычислить.

Наборы данных предоставляются в двух форматах:

- [CSV](#)-таблицы: skill_train.csv, skill_test.csv;
- [JSON](#)-документы: skill_train.jsonlines, skill_test.jsonlines (расширенный, содержит дополнительную информацию, которой нет в CSV-таблицах).

Описание полей с данными можно найти в руководстве по решению задачи. Про данные достаточно знать, что это некоторый набор чисел, идентификаторов и временных рядов, который потенциально несет в себе полезную информацию. Опыт игры в Dota 2 для использования данных не требуется.

Выходные данные

Для каждого примера из тестового набора необходимо предсказать опытность игрока. В систему необходимо предоставить для проверки CSV-таблицу с предсказаниями, она должна содержать две колонки: id — идентификатор примера, skilled — тип медали (0 — новичок, 1 — опытный).

Пример выходных данных (файл sample_submission.csv):

```
id,skilled
0,0
7,1
9,0
10,0
11,1
15,0
...
```

Оценка качества решения

Доля верно угаданных типов медали опыта по всем тестовым примерам:

$$Accuracy = \frac{1}{N} \sum_{i=1}^N [y_i = p_i],$$

где

- y_i — настоящая опытность игрока,
- p_i — ответ, выданный участником,
- $[x] = 1$ если выражение x верно, 0 — иначе.

Оценка за задачу вычисляется по формуле: $Accuracy * 100$.

Таким образом, максимальная оценка за задачу равна 100 баллам.



Рекомендации по решению задачи

Данная задача не типична для олимпиад по программированию. В отличие от обычных олимпиадных задач, точного решения у организаторов нет, более того — его, скорее всего, не существует, так как данные взяты из реального мира: в компьютерных играх всегда присутствует случайность и человеческий фактор, который не позволяет во всех случаях безошибочно определять целевую переменную. Это открывает огромный простор для творчества и экспериментов. Решая задачу, участникам предлагается ознакомиться с техниками анализа данных и машинного обучения. В частности:

- посмотреть материалы [Академии Искусственного Интеллекта Сбербанка](#);
- обратить внимание на разбор похожих задач на [тренировках по машинному обучению](#).

Для решения удобнее всего использовать язык программирования Python, так как для него есть большое число библиотек для анализа данных: [NumPy](#), [Pandas](#), [SciKit-Learn](#) и другие. В качестве инструмента разработки — интерактивную среду [Jupyter](#).

Базовый пример решения от организаторов доступен в виде Jupyter-тетради:
[Dota2SkillPrediction_Tutorial.ipynb](#)