

Задача 1. Имена собственные

Входной файл	proper.in
Выходной файл	proper.out
Ограничение по времени	1 секунда
Ограничение по памяти	256 мегабайт

Вася очень любит читать. При этом он старается каждое произведение читать на языке оригинала. Но у него часто возникает проблема: во многих произведениях встречается очень много имен собственных — имен героев, названий мест и т.д. — и их все очень сложно удержать в голове. Поэтому Вася очень хочет иметь программу, которая из текста книги выделит все имена собственные.

Очередная книга, которую собирается читать Вася, написана на английском языке. Вася знает, что в английском языке (в отличие от русского) в именах собственных *все* слова пишутся с большой буквы. Поэтому, для простоты, Вася решил считать именем собственным любую последовательность подряд идущих слов одного предложения, каждое из которых начинается с большой буквы; при этом такая последовательность не должна включать первое слово соответствующего предложения (потому что оно и так всегда пишется с большой буквы). Словом считается любая последовательность подряд идущих английских букв. Предложением считается последовательность слов, заканчивающаяся на точку, восклицательный или вопросительный знак.

Более формально: вам дан текст, содержащий только английские маленькие и большие буквы, пробелы, переводы строк и знаки препинания. Символы «точка», «восклицательный знак» и «вопросительный знак» считаются *символами конца предложения*, они разбивают текст на предложения. А именно, первым предложением считается последовательность символов от начала текста до первого символа конца предложения, вторым предложением — последовательность символов между первым и вторым символами конца предложения, и т.д.

Все остальные символы, кроме букв, считаются *пробельными символами*, они разбивают предложения на слова. А именно, первым словом предложения считается последовательность букв начиная с первой буквы, присутствующей в предложении, до первого пробельного символа, идущего после этой буквы. Вторым словом считается последовательность букв начиная со следующей буквы и до ближайшего после нее пробельного символа, и т.д.

Слово считается *частью имени собственного*, если оно начинается с большой буквы и при этом не является первым словом в предложении. *Именем собственным* считается каждая последовательность слов-частей имени собственного, принадлежащих одному предложению и ограниченных с обеих сторон или концами предложения, или словами, не являющимися частями имени собственного.

Напишите программу, которая выведет все имена собственные, встречающиеся в данном тексте.

Формат входных данных

Выходной файл содержит заданный текст. В тексте встречаются только маленькие и большие латинские буквы, пробелы, переводы строки и следующие знаки препинания: `!?.;:,"()-`. Длина текста не превосходит 100 000 символов. Гарантируется, что текст заканчивается на символ конца предложения, после которого могут идти несколько пробелов и/или переводов строки.

Формат выходных данных

Выведите в выходной файл все имена собственные, встречающиеся в заданном тексте, по одному имени собственному на строке, в том порядке, в котором они встречаются в тексте. Если одно и то же имя собственное встречается несколько раз в разных местах текста, его надо выводить каждый раз.

Слова в имени собственном разделяйте пробелами; знаки препинания должны отсутствовать в выходном файле.

Пример на следующей странице

Пример

<i>Входной файл</i>	<i>Выходной файл</i>
<p>This is the living-room of the house occupied by the eminent Professor Michael Verres-Evans, and his wife, Mrs Petunia Evans-Verres, and their adopted son, Harry James Potter-Evans-Verres. There is a letter lying on the living-room table, and an unstamped envelope of yellowish parchment, addressed to Mr H Potter in emerald-green ink. The Professor and his wife are speaking sharply at each other, but they are not shouting. The Professor considers shouting to be uncivilised.</p>	<p>Professor Michael Verres Evans Mrs Petunia Evans Verres Harry James Potter Evans Verres Mr H Potter Professor Professor</p>
<p>The week after Taffimai Metallumai (we will still call her Taffy, Best Beloved) made that little mistake about her Daddy spear and the Stranger-man and the picture-letter and all, she went carp-fishing again with her Daddy. Her Mummy wanted her to stay at home and help hang up hides to dry on the big drying-poles outside their Neolithic Cave, but Taffy slipped away down to her Daddy quite early, and they fished.</p>	<p>Taffimai Metallumai Taffy Best Beloved Daddy Stranger Daddy Mummy Neolithic Cave Taffy Daddy</p>

Задача 2. Железная дорога

Входной файл	bzd.in
Выходной файл	bzd.out
Ограничение по времени	1 секунда
Ограничение по памяти	256 мегабайт

На сайте Берляндских Железных Дорог (БЖД) любой может купить билет. Для этого ему лишь нужно заполнить несложную форму, а затем пройти еще несколько простых шагов. Руководство БЖД недавно задумалось о том, насколько легко людям использовать эту форму, и поручило вам провести измерения, чтобы оптимизировать количество действий, нужных для заполнения формы.

Форма состоит из двух полей: станция отправления и станция назначения. Когда человек заходит на сайт, в этих полях уже введены станции — те, которые сайт считает наиболее вероятными. Также под каждым полем есть список названий некоторых, самых популярных, станций — клик по элементу списка подставляет название станции в соответствующее поле. Наконец, есть кнопка «Поменять местами станцию отправления и станцию назначения». Кроме того, конечно, пользователь также волен ввести названия нужных ему станций вручную. К сожалению, из-за ошибки в коде сайта, редактировать введенное или выбранное название нельзя, можно только ввести его заново.

Для целей эксперимента клик по кнопке смены станций отправления и назначения или по станции в списке популярных станций оценивается в одно очко действия. Ввод названия станции вручную стоит столько очков действия, сколько символов (включая пробелы и дефисы) содержит это название.

Вам нужно написать программу, которая по желаемым станциям отправления и назначения и начальному виду страницы (введенные станции, список популярных станций) определит минимальное количество очков действия, необходимое для того, чтобы ввести желаемые станции в соответствующие поля ввода.

Формат входных данных

В первой строке входного файла записано количество тестов в этом файле — единственное целое число K ($1 \leq K \leq 10$). В последующих строках записаны K тестов.

Каждый отдельный тест состоит из нескольких строк. В первой строке теста записано название желаемой станции отправления. Во второй строке — название желаемой станции назначения. Гарантируется, что желаемая станция отправления не совпадает с желаемой станцией назначения.

В третьей строке — текущее введенное название станции отправления. В четвертой строке — текущее введенное название станции назначения.

В пятой строке — одно целое число N ($0 \leq N \leq 100$) — количество популярных вариантов станции отправления. В следующих N строках записаны эти варианты, по одному на строке. Все варианты различны.

После этого записано количество популярных вариантов станции назначения — одно целое число M ($0 \leq M \leq 100$). В следующих M строках записаны эти варианты, по одному на строке. Все варианты различны.

Названия станций могут состоять из больших и маленьких латинских букв, пробелов и дефисов. Длина каждого названия не превышает 100 символов. Названия станций всегда начинаются и заканчиваются буквой. Любой пробел или дефис обязательно окружен буквами.

Формат выходных данных

Для каждого теста выведите единственное целое число — количество очков действия, необходимое, чтобы ввести желаемые станции отправления и назначения в соответствующие поля. Выводите по одному ответу на строку. Порядок ответов должен соответствовать порядку тестов во входном файле.

Пример на следующей странице

Пример

<i>Входной файл</i>	<i>Выходной файл</i>
2 Nizhny Novgorod Moskva Samara Petrozavodsk 2 Arkhangelsk Nizhny Novgorod 2 Moskva Samara Moskva Nizhny Novgorod Moskv Novgorod 2 Arkhangelsk Ulyanovsk 2 Volgograd Samara	2 21
1 Moskva Nizhny Novgorod Arkhangelsk Moskva 3 Nizhny Novgorod Ulyanovsk Kazan 3 Volgograd Kazan Samara	2

Примечание: Первый пример состоит из двух тестов. В первом тесте пользователю достаточно кликнуть по Nizhny Novgorod в списке популярных станций отправления, чтобы заменить Samara на желаемую станцию отправления, а затем кликнуть по Moskva в списке популярных станций назначения, чтобы заменить Petrozavodsk на желаемую станцию. Два клика требуют двух очков действия.

Во втором тесте у пользователя нет выбора, кроме как ввести названия станций вручную. Ввод Moskva потребует 6 очков действий, а ввод Nizhny Novgorod — 15 очков. В сумме требуется 21 очко.

Второй пример стоит из единственного теста. В этом тесте пользователю достаточно сначала кликнуть по Nizhny Novgorod в списке популярных станций отправления, а затем нажать на кнопку «Поменять местами станцию отправления и станцию назначения». На это потребуется два очка.

Иллюстрация ко второму примеру:

Станции отправления и прибытия

Arkhangelsk

Moskva

Nizhny Novgorod, Ulyanovsk, Kazan

Volgograd, Kazan, Samara

Задача 3. Последнее число

Входной файл	last.in
Выходной файл	last.out
Ограничение по времени	1 секунда
Ограничение по памяти	256 мегабайт

На доске записано n целых чисел a_1, a_2, \dots, a_n . Миша может стереть любые два числа a_i и a_j и вместо них записать их сумму $a_i + a_j$. При этом Миша выплачивает своему другу Жене количество монет, равное минимуму из этих двух стертых чисел.

Миша хочет оставить на доске ровно одно число, отдав Жене как можно меньше монет. Помогите ему в этом.

Формат входных данных

В первой строке входных данных находится число n ($1 \leq n \leq 10^3$). Во второй строке находится последовательность чисел a_1, a_2, \dots, a_n , записанная через пробелы ($1 \leq a_i \leq 10^3$).

Формат выходных данных

Выведите единственное число — минимальное количество монет, которое необходимо Мише заплатить Жене, чтобы на доске осталось ровно одно число.

Пример

Входной файл	Выходной файл
2 7 3	3
3 10 20 30	30

Примечание: Во втором примере Миша может сначала стереть числа 20 и 30 и записать на доске число 50. Затем стереть числа 10 и 50 и записать 60. Итого Миша должен отдать Жене $20 + 10 = 30$ монет.

Задача 4. Взять все остатки

Входной файл	rem.in
Выходной файл	rem.out
Ограничение по времени	1 секунда
Ограничение по памяти	256 мегабайт

Недавно Петин учитель рассказал ему об операции деления с остатком. Число p называется *частным*, а число q — *остатком* от деления x на y , если $x = y \cdot p + q$, а также $0 \leq q < y$, и p и q целые. С вычислением *частного* у Пети проблем не возникает, а вот получение *остатка* дается ему с трудом. Но так как Петя — отличник, он решил поупражняться в вычислении остатка.

Дома Петя нашел его любимую последовательность чисел a_1, a_2, \dots, a_n . Теперь Петя будет выбирать числа b_1, b_2, \dots, b_m и с каждым b_i из них выполнять следующие операции:

1. Делить с остатком b_i на a_1 ,
2. Получившийся остаток делить с остатком на a_2 ,
3. Получившийся после второй операции остаток делить на a_3 , и так далее,
- ...
- n . Остаток от $(n - 1)$ -ого деления делить с остатком на a_n .

Для того, чтобы проверять себя, Петя попросил вас написать программу, которая по заданным последовательностям a_1, a_2, \dots, a_n и b_1, b_2, \dots, b_m найдет для каждого b_i остаток от последнего (n -ого) деления. Помогите Пете!

Формат входных данных

В первой строке входных данных находится число n ($1 \leq n \leq 10^5$). Во второй строке находится последовательность a_1, a_2, \dots, a_n , записанная через пробелы ($1 \leq a_i \leq 10^9$). Далее следует число m ($1 \leq m \leq 10^5$), а после него на четвертой строке — числа b_1, b_2, \dots, b_m , разделенные пробелами ($0 \leq b_i \leq 10^9$). Все числа во входных данных целые.

Формат выходных данных

Выведите m чисел — остатки от последней операции для каждого b_i .

Пример

Входной файл	Выходной файл
4 10 9 5 7 5 14 8 27 11 25	4 3 2 1 0

Примечание: В примере $b_3 = 27$. Последовательность действий, выполняемая Петей, будет следующая. Сначала разделим 27 на $a_1 = 10$, получим в остатке 7. Далее разделим 7 на $a_2 = 9$, получим в остатке 7. Разделим 7 на $a_3 = 5$, получим в остатке 2. Разделим 2 на $a_4 = 7$, получим в остатке 2, это и будет число, которое необходимо вывести.

Задача 5. Передача опыта

Входной файл	travel-cards.in
Выходной файл	travel-cards.out
Ограничение по времени	3 секунды
Ограничение по памяти	256 мегабайт

Как известно, Берляндия — хорошо развитая страна, а поэтому в каждом городе есть школа. На недавнем съезде учителей стало ясно, что методики преподавания различных предметов во всех школах разные. Чтобы исправить это недоразумение, а также для улучшения качества образования было решено, что школам необходимо обмениваться опытом. Так как школы действительно разные, и у каждой имеется уникальный опыт, поступило предложение из каждой школы в каждую другую отправить делегата для перенимания опыта.

К сожалению, города в Берляндии находятся далеко друг от друга, поэтому их связывает минимальная возможная сеть дорог: из каждого города можно добраться по дорогам в любой другой не посещая ни один город дважды, но при этом лишь единственным способом. Также, еще большую проблему для учителей создает сложная система оплаты проезда по дорогам Берляндии.

Каждой дорогой владеет одна из K компаний, и для проезда по дорогам любой компании человек должен один раз купить проездной у этой компании. После этого он может проезжать по любым дорогам этой компании без дополнительной платы. Стоимость проездного i -ой компании ($1 \leq i \leq K$) равна c_i бурлей.

Каждому из делегатов нужно купить необходимые ему проездные. Помогите учителям посчитать, сколько же всего бурлей будут стоить все проездные для всех делегатов. Так как это число может быть большим, найдите остаток от деления этого числа на $1\,000\,000\,007$ ($10^9 + 7$).

Формат входных данных

В первой строке входных данных находятся два натуральных числа N и K ($1 \leq N \leq 1\,500\,000$, $1 \leq K \leq 1\,500\,000$) — число городов в Берляндии и число компаний-владельцев дорог соответственно. На следующих $(N-1)$ строках дано описание дорог Берляндии, на i -ой из этих строк (эти строки нумеруются с единицы, города тоже нумеруются с единицы) находятся два числа b_i и t_i ($1 \leq b_i \leq N$, $1 \leq t_i \leq K$), означающие, что существует дорога из города с номером $i+1$ в город с номером b_i , и владеет ей компания с номером t_i . По всем дорогам можно передвигаться в обе стороны; также, гарантируется, что из любого города можно добраться в любой другой по существующим дорогам.

На последней строке входных данных находятся 4 числа A , B , C , c_0 , задающие стоимости проездных ($1 \leq C \leq 1\,000\,000\,001$, $0 \leq A, B, c_0 < C$). А именно, стоимости проездных вычисляются по формуле $c_i = (A \cdot c_{i-1} + B) \bmod C$, где c_i — стоимость проездного i -ой компании, а « $X \bmod Y$ » — остаток от деления X на Y .

Формат выходных данных

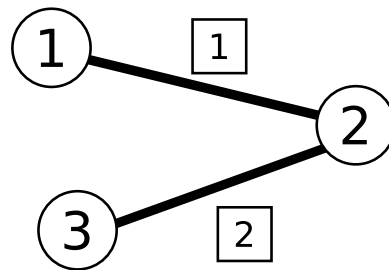
Выведите единственное число — остаток от деления суммарной стоимости необходимых проездных на $1\,000\,000\,007$.

Пример

Входной файл	Выходной файл
3 2 1 1 2 2 1 2 100 1	32
3 1 1 1 2 1 1 2 100 1	18

Примечание к примеру на следующей странице

Примечание: В первом тесте из примера карта дорог выглядит так, как показано на рисунке. Кругами обозначены города, линиями — дороги, а числа в квадратах около дорог обозначают номера компаний, обслуживающих эти дороги. Цены проездных в этом примере равны $c_1 = 3$ бурля, $c_2 = 5$ бурлей. Таким образом, делегаты, едущие из первого города в третий, или обратно, должны купить оба проездных, остальным делегатам нужно проехать только одну дорогу, поэтому им нужно купить только один проездной. Итоговая стоимость проездных равна $2 \cdot (3 + 5) + 2 \cdot 3 + 2 \cdot 5 = 32$ бурлей.



Во втором тесте схема дорог выглядит так же, а компания-владелец только одна. Стоимость проездного равна 3 бурля. Поэтому все 6 делегатов должны купить единственный проездной, и суммарная стоимость $6 \cdot 3 = 18$ бурлей.

Тесты, в которых выполняются дополнительные ограничения $1 \leq N, K \leq 100$, имеют суммарную стоимость не менее 30 баллов.

Тесты, в которых выполняются дополнительные ограничения $1 \leq N, K \leq 1000$, имеют суммарную стоимость не менее 50 баллов.

Тесты, в которых выполняются дополнительные ограничения $1 \leq N, K \leq 100\,000$, имеют суммарную стоимость не менее 75 баллов.

Задача 6. Светофоры

Входной файл	traffic.in
Выходной файл	traffic.out
Ограничение по времени	1 секунда
Ограничение по памяти	256 мегабайт

Вы — министр транспорта в Тридевятом королевстве. Дороги, как известно, являются одной из основных проблем королевства, тем не менее, недавно была закончена постройка Великого Королевского Тракта — самой длинной дороги в Тридевятом царстве. Тракт представляет из себя прямую дорогу, поэтому далее будем называть координатой точки на тракте расстояние от начала тракта до нее.

Вам стало известно, что король скоро собирается проехать по этому тракту от начала до конца, причем поедет он со скоростью 1. Когда именно он собирается начать, вам неизвестно, но зато вы уверены, что это произойдет в какой-то случайный целочисленный момент времени между t_1 и t_2 включительно. В рамках модернизационной программы вы отдали поручение построить на Тракте n светофоров, которые должны будут впечатлить проезжающего короля. Светофоры в некотором смысле одноразовые, а именно они поначалу горят зеленым, затем красным, а потом снова зеленым, не переключаясь больше, пока не сломаются (но можно считать, что сломаются они не скоро). Про каждый светофор вам известно три величины: его положение на Тракте, время, когда он загорится красным, и время, когда он вновь загорится зеленым. Так получилось, что последний светофор стоит прямо на конце Тракта. Понятно, что король не станет подавать плохой пример подданным: никакой светофор он не будет проезжать на красный свет (в том числе это относится и к последнему светофору). В момент, когда красный свет только загорается, ехать уже становится нельзя, а когда загорается зеленый — сразу же можно.

Ваша премия как министра напрямую зависит от того, насколько долго король будет ехать по тракту, поэтому вас интересует вопрос, сколько в среднем времени ему потребуется на то, чтоб проехать Тракт от начала до конца (включая время ожидания на последнем светофоре, если это потребуется). Напишите программу, которая посчитает эту величину.

Формат входных данных

В первой строке входного файла вводится число n — количество светофоров ($1 \leq n \leq 100\,000$). Затем в каждой из следующих n строк — описания светофоров. В каждой из них содержатся 3 целых числа x_i, s_i, f_i ($0 \leq x_i, s_i, f_i \leq 10^9$), описывающих очередной светофор: x_i — это положение светофора на Тракте, s_i — момент времени, когда светофор загорится красным, f_i — момент времени, когда он вновь загорится зеленым ($0 \leq s_i < f_i$). Светофоры упорядочены по величине x , т.е. для любых номеров i и j таких, что $1 \leq i < j \leq n$, верно, что $x_i < x_j$. Кроме того, гарантируется, что длина Тракта больше нуля.

В последней строке задаются целые числа t_1 и t_2 — начальный и конечный момент времени, когда король может начать свой путь ($0 \leq t_1 \leq t_2 \leq 10^9$).

Формат выходных данных

Выведите одно число — время в пути, усредненное по всем возможным временам начала (т.е. среднее арифметическое времен в пути, вычисленных для всех возможных целочисленных времен старта от t_1 до t_2 включительно). Ваш ответ будет считаться верным, если его относительная погрешность будет не более 10^{-9} , т.е., если ваш ответ равен a , а правильный — b , то должно выполняться $|a - b| \leq 10^{-9}b$.

Пример

Входной файл	Выходной файл
2 4 11 12 8 14 15 6 9	8.5000000000

Примечание: В примере король может выехать в любой момент между 6 и 9 включительно.

Если король выезжает в момент времени 6, то к первому светофору он подъезжает в момент времени 10, этот светофор еще горит зеленым, поэтому король едет дальше без остановки. Ко второму светофору король подъезжает в момент времени 14, этот светофор в этот момент загорается красным, поэтому король ждет до момента времени 15, когда светофор загорается опять зеленым. Итого получается время в пути равно 9.

Если король выезжает в момент времени 7, то к первому светофору он подъезжает в момент времени 11, когда этот светофор только загорелся красным — король ждет до момента времени 12. Ко второму светофору король подъезжает в момент времени 16, когда этот светофор уже горит зеленым. Итого получается время в пути равно 9.

Если король выезжает в момент времени 8 или 9, то он не ждет ни на одном светофоре, и время в пути получается 8.

Итого среднее время в пути есть $(9 + 9 + 8 + 8)/4 = 8.5$.