

Задача А. Словарь синонимов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вам дан словарь, состоящий из пар слов. Каждое слово является синонимом к парному ему слову. Все слова в словаре различны. Для одного данного слова определите его синоним.

Формат входных данных

Программа получает на вход количество пар синонимов $N \leq 100000$. Далее следует N строк, каждая строка содержит ровно два слова-синонима. После этого следует одно слово.

Формат выходных данных

Программа должна вывести синоним к данному слову.

Пример

стандартный ввод	стандартный вывод
3 Hello Hi Bye Goodbye List Array Goodbye	Bye

Замечание

Используйте для решения средства стандартной библиотеки!

Задача В. Минимум на отрезке

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Рассмотрим последовательность целых чисел длины N . По ней с шагом 1 движется «окно» длины K , то есть сначала в «окне» видно первые K чисел, на следующем шаге в «окне» уже будут находиться K чисел, начиная со второго, и так далее до конца последовательности. Требуется для каждого положения «окна» определить минимум в нём.

Формат входных данных

В первой строке входных данных содержатся два числа N и K ($1 \leq N \leq 150000$, $1 \leq K \leq 100000$, $K \leq N$) – длины последовательности и «окна», соответственно. На следующей строке находятся N чисел – сама последовательность. Числа последовательности не превосходят по модулю 10^5 .

Формат выходных данных

Выходные данные должны содержать $N - K + 1$ строк – минимумы для каждого положения «окна».

Пример

стандартный ввод	стандартный вывод
7 3 1 3 2 4 5 3 1	1 2 2 3 1

Задача С. Симметрическая разность множеств

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Реализуйте бинарную операцию над множествами — симметрическую разность. Ее результатом являются те элементы двух множеств, которые принадлежат ровно одному из множеств (но не принадлежат их пересечению).

Формат входных данных

В первой строке записано натуральное число — количество элементов первого множества.

Во второй строке через пробел перечислены эти элементы (натуральные числа).

В третьей строке указано количество элементов второго множества (натуральное).

В четвертой строке перечислены через пробел элементы второго множества (натуральные числа).

Формат выходных данных

Результат требуется вывести в том же формате: сначала количество элементов, полученных в результате операции, затем (если получено ненулевое количество элементов) вывести сами элементы в порядке возрастания.

Множества во входных данных могут быть неупорядочены и содержать равные элементы. Множество в выводе должно быть упорядочено и все элементы его должны быть различными.

Пример

стандартный ввод	стандартный вывод
3	1
1 2 3	3
2	
1 2	

Задача D. Личные дела

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Однажды, неловкая секретарша перепутала личные дела учащихся. Теперь их снова необходимо упорядочить сначала по классам, а внутри класса по фамилиям

Формат входных данных

В первой строке дано число N ($1 \leq N \leq 1000$) — количество личных дел. Далее для каждого из N учащихся следующие данные, разделенные одиночными пробелами: фамилия и имя, класс, дата рождения. Фамилия и имя — строки не более чем из 20 символов, класс — строка состоящая из числа (от 1 до 11) и латинской буквы (от «A» до «Z»), дата рождения — дата в формате «ДД.ММ.ГГ». Гарантируется, что внутри одного класса нет однофамильцев.

Формат выходных данных

В выходной файл требуется вывести N строк, в каждой из которых записаны данные по одному учащемуся. Строки должны быть упорядочены сначала по классам, а затем по фамилиям.

Пример

стандартный ввод	стандартный вывод
3	9A Ivanov Ivan 10.04.93
Sidorov	9A Sidorov Sidor 20.07.93
Sidor	9B Petrov Petr 23.10.92
9A	
20.07.93	
Petrov	
Petr	
9B	
23.10.92	
Ivanov	
Ivan	
9A	
10.04.93	

Задача E. Англо-латинский словарь

Имя входного файла:	dictionary.in
Имя выходного файла:	dictionary.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Однажды, разбирая старые книги на чердаке, школьник Вася нашёл англо-латинский словарь. Английский он к тому времени знал в совершенстве, и его мечтой было изучить латынь. Поэтому попавшийся словарь был как раз кстати.

К сожалению, для полноценного изучения языка недостаточно только одного словаря: кроме англо-латинского необходим латинско-английский. За неимением лучшего он решил сделать второй словарь из первого.

Как известно, словарь состоит из переводимых слов, к каждому из которых приводится несколько слов-переводов. Для каждого латинского слова, встречающегося где-либо в словаре, Вася предлагает найти все его переводы (то есть все английские слова, для которых наше латинское встречалось в его списке переводов), и считать их и только их переводами этого латинского слова.

Помогите Васе выполнить работу по созданию латинско-английского словаря из англо-латинского.

Формат входных данных

Во входном файле содержатся несколько описаний английских слов. Каждое описание содержится в отдельной строке, в которой записано сначала английское слово, затем отведёнными пробелами дефис (символ номер 45), затем разделённые запятыми с пробелами переводы этого английского слова на латинский. Переводы отсортированы в лексикографическом порядке. Порядок следования английских слов в словаре также лексикографический.

Все слова состоят только из маленьких латинских букв, длина каждого слова не превосходит 15 символов. Общее количество слов на входе не превышает 100 000.

Формат выходных данных

Программа должна вывести количество латинских слов в словаре k . В следующих k строках программа должна вывести латинско-английский словарь, соответствующий входному словарю, в точности соблюдая формат входных данных. В частности, первым должен идти перевод лексикографически минимального латинского слова, далее — второго в этом порядке и т. д. Внутри перевода английские слова должны быть также отсортированы лексикографически.

Пример

dictionary.in	dictionary.out
apple - malum, pomum, popula	7
fruit - baca, bacca, popum	baca - fruit
punishment - malum, multa	bacca - fruit
	malum - apple, punishment
	multa - punishment
	popum - apple
	popula - apple
	popum - fruit

Задача F. Счастливое число

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.9 секунд
Ограничение по памяти: 512 мегабайт

Ванечка подарил Анечке массив из n чисел. У Анечки есть любимое счастливое число — оно всегда помогает ей сдавать даже неправильные решения задач.

Анечке стало очень интересно, сколько пар чисел из подаренного массива в сумме дают число k . К сожалению, сейчас она занята подготовкой к контрольной и не может это узнать. Помогите ей!

Формат входных данных

В первой строке находится число t ($1 \leq t \leq 100$) — количество тестов.

Для каждого теста в первой строке вводятся два числа n, k ($2 \leq n \leq 10^5, |k| < 10^9$), где n — это размер подаренного Ванечкой массива и k — счастливое число. В следующей строке дано n чисел, по модулю не превосходящих 10^9 — числа в Анечкином массиве.

Формат выходных данных

Для каждого теста на отдельной строке выведите ответ на задачу.

Пример

стандартный ввод	стандартный вывод
3	2
3 1	1
0 1 0	1
2 -1	
-2 1	
4 3	
1 2 3 -1	

Задача G. Машины

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Петя, которому три года, очень любит играть с машинками. Всего у Пети N различных машинок, которые хранятся на полке шкафа так высоко, что он сам не может до них дотянуться. Одновременно на полу комнаты может находиться не более K машинок. Петя играет с одной из машинок на полу и если он хочет поиграть с другой машинкой, которая также находится на полу, то дотягивается до нее сам. Если же машинка находится на полке, то он обращается за помощью к маме. Мама может достать для Пети машинку с полки и одновременно с этим поставить на полку любую машинку с пола. Мама очень хорошо знает своего ребенка и может предугадать последовательность, в которой Петя захочет играть с машинками. При этом, чтобы не мешать Петиней игре, она хочет совершить как можно меньше операций по подъему машинки с пола, каждый раз правильно выбирая машинку, которую следует убрать на полку. Ваша задача состоит в том, чтобы определить минимальное количество операций. Перед тем, как Петя начал играть, все машинки стоят на полке.

Формат входных данных

В первой строке содержатся три числа N , K и P ($1 \leq K, N \leq 100000$, $1 \leq P \leq 500000$). В следующих P строках записаны номера машинок в том порядке, в котором Петя захочет играть с ними.

Формат выходных данных

Выведите единственное число: минимальное количество операций, которое надо совершить Петиней маме.

Пример

стандартный ввод	стандартный вывод
3 2 7 1 2 3 1 3 1 2	4

Замечание

В этой задаче можно использовать STL.

Пояснения к примеру:

Операция 1: снять машинку 1

Операция 2: снять машинку 2

Операция 3: поднять машинку 2 и снять машинку 3

Операция 4: поднять машинку 3 или 1 и снять машинку 2

Задача N. Трамвай

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

С окраины в центр города каждое утро по одному маршруту едут в трамвае N человек. За долгое время поездок они достаточно хорошо узнали друг друга. Чтобы никому не было обидно, они захотели решить, кто из них и между какими остановками маршрута должен сидеть, а кто должен стоять. Все остановки пронумерованы от 1 до P .

Один из пассажиров оказался знатоком теории математического моделирования. Он предложил рассмотреть значение суммарного удовлетворения пассажиров. Для каждого i -го пассажира он оценил две величины — a_i и b_i . Если в течение одного переезда между остановками пассажир сидит, то к суммарному удовлетворению прибавляется a_i , если же он стоит, то прибавляется b_i .

Всего в трамвае M сидячих мест. Вставать и садиться пассажиры могут мгновенно на любой остановке. Кроме того, некоторые пассажиры предпочитают ехать стоя, даже если в трамвае есть свободные места (для них $a_i < b_i$).

Требуется написать программу, которая вычисляет значение максимально достижимого суммарного удовлетворения, если для каждого i -го пассажира известны величины a_i и b_i , а также номера остановок, на которых он садится и выходит из трамвая.

Формат входных данных

Первая строка входного файла содержит разделенные пробелом три целых числа N , M и P — число пассажиров, число сидячих мест и число остановок на маршруте соответственно ($1 \leq N, M, P \leq 100000$; $2 \leq P$).

Каждая из следующих N строк содержит информацию об очередном пассажире в виде четырёх целых чисел a_i, b_i, c_i, d_i , где первые два числа определяют вклад в параметр счастья, третье — номер остановки, на которой пассажир садится в трамвай, и последнее — номер остановки, на которой он выходит из трамвая ($-10^6 \leq a_i, b_i \leq 10^6$; $1 \leq c_i < d_i \leq P$).

Формат выходных данных

В выходной файл необходимо вывести одно целое число — максимальное суммарное удовлетворение, которого могут добиться пассажиры.

Пример

стандартный ввод	стандартный вывод
4 2 4	28
10 -10 2 3	
-1 -3 1 4	
6 -6 1 3	
7 4 2 4	

Замечание

Максимальное суммарное довольство достигается следующим образом:

На первой остановке входят и садятся второй и третий пассажиры;

На второй остановке входят первый и четвёртый пассажиры, второй уступает место первому;

На третьей остановке встают и выходят первый и третий пассажиры, второй и четвёртый садятся на их места;

На четвёртой остановке выходят второй и четвёртый пассажиры.

Задача I. Бесконечный этаж

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Знаете ли вы, почему четвертый этаж заперт и там не останавливается лифт? Потому что на самом деле четвертый, запертый, этаж, где не останавливается лифт, содержит бесконечное количество комнат, пронумерованных натуральными числами. На этот этаж регулярно приезжают дети, каждый из которых заранее выбрал, в какую комнату он хочет заселиться. Если выбранная комната оказывается свободна, то ребенок занимает ее, в противном случае он занимает первую свободную комнату с номером, большим исходного.

Кроме того, иногда дети уезжают. Сразу после отъезда ребенка его комната становится доступна для заселения следующего.

Промоделируйте работу вожатых, ответственных за четвертый этаж и научитесь быстро сообщать приезжающим детям, какую комнату им следует занимать.

Формат входных данных

Первая строка входного файла содержит натуральное число n — количество прибытий и отъездов ($1 \leq n \leq 100000$).

Следующие n строк содержат информацию о школьниках. Число $a > 0$ обозначает, что приехал школьник, желающий занять комнату номер a ($1 \leq a \leq 100000$). Число $a < 0$ обозначает, что из комнаты номер $|a|$ уехал школьник (гарантируется, что эта комната не была пуста).

Формат выходных данных

Для каждого приезжающего школьника выведите одно натуральное число — номер комнаты, в которую он поселится.

Пример

стандартный ввод	стандартный вывод
6	5
5	6
5	7
5	6
-6	8
5	
5	

Задача J. Вычислительная ихтиология

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Игорь работает младшим лаборантом в НИИ ихтиологии. Ему вверены n аквариумов, стоящих в ряд, в каждом из которых живет колония рыбок гуппи. Про каждую колонию заранее известна ее численность.

В лабораторных условиях НИИ ихтиологии колония рыбок гуппи растет по следующему правилу: достигнув популяции в f рыбок, колония живет в течение $\max(1000 - f, 1)$ секунд, после чего на свет появляется новая рыбка. От начального момента времени до рождения первой рыбки колония размера f также ждет $\max(1000 - f, 1)$ секунд.

Например, колония с начальным размером 996 будет размножаться следующим образом:

время	число рыб	время до очередной рыбки
0	996	4
4	997	3
7	998	2
9	999	1
10	1000	1
11	1001	1
...

Появление на свет каждой новой рыбки Игорь должен фиксировать в специальном журнале. Будем считать, что запись он делает мгновенно, но при этом он должен в момент рождения новой рыбки находиться рядом с аквариумом, в котором это произошло.

На перемещение от одного аквариума к соседнему у Игоря уходит одна секунда. В начальный момент времени Игорь стоит около первого аквариума.

Вычислите, в течение какого наибольшего периода времени Игорь сможет добросовестно выполнять свою работу.

Формат входных данных

В первой строке входного файла содержится целое число n ($2 \leq n \leq 50$) — количество аквариумов с рыбками гуппи в НИИ ихтиологии. Каждая из следующих n строк содержит одно целое число a_i ($1 \leq a_i \leq 2007$) — численность i -й колонии.

Формат выходных данных

В выходной файл выведите момент времени, когда родится первая рыбка гуппи, запись о рождении которой Игорь сделать не сможет.

Пример

стандартный ввод	стандартный вывод
3 996 1 994	7

Замечание

В приведенном примере Игорь сначала ждет у первого аквариума появления рыбки на 4-й секунде. После этого он бежит к третьему аквариуму (на это у него уходит 2 секунды) и как раз успевает к рождению рыбки на 6-й секунде. Однако вернуться к первому аквариуму, где следующая рыбка родится на 7-й секунде, он уже не успевает.

Задача К. T2005

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Клавиатура сотового телефона выглядит так:

1 пробел	2 abc	3 def
4 ghi	5 jkl	6 mno
7 pqrs	8 tuv	9 wxyz

Режим ввода T2005 устроен следующим образом. В телефоне есть словарь. Пользователь, чтобы ввести слово, последовательно нажимает клавиши, на которых написаны буквы этого слова. Например, чтобы ввести слово `begin` пользователь должен нажимать клавиши `23446`. Но как только в словаре оказывается только одно слово с таким началом, это слово автоматически подставляется и, кроме того, после этого слова автоматически добавляется пробел. Например, пусть пользователь нажал клавиши `234`, и оказалось, что слов, ввод которых начинается с нажатия именно этих клавиш, — ровно одно. Тогда автоматически подставится это слово и пробел после него, а все последующие нажатия клавиш уже будут относиться к вводу следующего слова.

Если для ввода какого-то слова нужно нажать последовательность клавиш, которая может являться началом какого-то другого слова, то после ввода этого слова нужно нажать клавишу `1`, что соответствует вводу пробела. При вводе пробела считается, что вы ввели все слово целиком (а не только какое-либо его начало). Если после ввода пробела оказалось, что в словаре такой последовательности клавиш удовлетворяет несколько слов, подставляется первое из них в алфавитном порядке. Если (опять же после ввода пробела) оказалось, что в словаре нет слова, которое может быть введено такой последовательностью клавиш, то все, что было введено после предыдущего пробела (введенного или автоматически подставленного, или, если в тексте ранее не встречалось ни одного пробела — от начала текста) удаляется. Если после ввода пробела (как нажатием «1», так и автоподстановкой) или в начале текста нажимается клавиша «1», то ее нажатие игнорируется.

Вам дан словарь и последовательность нажатий клавиш. Выведите текст, который был введен пользователем.

Примечание: в тексте используются только маленькие латинские буквы и символ пробел.

Формат входных данных

Сначала на вход программы поступает число N — количество слов в словаре ($2 \leq N \leq 100000$). В следующих N строках задается словарь. Каждое слово записано в отдельной строке. Слова расположены в алфавитном порядке. Никакое слово в словаре не встречается дважды. Длина каждого слова не превосходит 10 символов.

Далее вводится число M — количество нажатий клавиш ($1 \leq M \leq 20000$). Затем задается M разделяющихся пробелами чисел, описывающих нажатые клавиши. Последней нажатой клавишей всегда является клавиша «1».

Формат выходных данных

Выведите одну строку — текст, который оказался введен пользователем. Пробел после последнего введенного слова также должен быть выведен.

Пример

стандартный ввод	стандартный вывод
5 po pod sasha shla shosse 12 7 4 5 7 2 7 6 1 7 4 6 1	shla sasha po shosse

Задача L. Поступление

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Напишите программу, которая будет по списку поступающих и университетов определять, кто именно поступит в каждый университет.

В каждом университете есть лимит мест, у каждого абитуриента есть определенный набранный балл и список предпочтений.

Распределение происходит следующим образом: абитуриенты сортируются по убыванию баллов, при равенстве - по возрастанию даты рождения и, затем, по фамилии и имени.

Далее каждый абитуриент по очереди зачисляется в первый университет из его списка, в котором еще остались места (если таких нет - не зачисляется никуда).

Формат входных данных

Первая строка содержит одно целое число N от 1 до 10^4 - количество университетов.

Следующие N строк содержат строку их латинских символов длиной от 5 до 15 символов и число от 0 до 10^9 - название и максимальное количество студентов для очередного университета.

Следующая строка содержит целое число M от 0 до 10^4 - количество абитуриентов.

Далее идут M строк, каждая из которых содержит две строки длиной от 5 до 15 символов - имя и фамилию очередного абитуриента, затем 3 целых числа от 0 до 10^9 - число, месяц и год его рождения, целое число от 0 до 10^9 - балл студента, целое число k от 0 до 200 - количество университетов, в которые абитуриент готов поступать, и k названий университетов из списка в первой части входа.

Формат выходных данных

Для каждого университета в алфавитном порядке выведите его название, затем через табуляцию - имена и фамилии поступивших в него студентов, отсортированных по фамилии, имени и дате рождения.

Пример

стандартный ввод	стандартный вывод
3	HSE Ivan Petrov Petr Petrov
MSU 1	MIPT Alexander Sidorov
HSE 2	MSU Ivan Ivanov
MIPT 100	
5	
Ivan Ivanov 1 1 1900 100 2 MSU HSE	
Petr Petrov 2 1 1900 90 2 MSU HSE	
Alexander Sidorov 3 1 1900 110 2 MIPT HSE	
Ivan Petrov 3 1 1900 100 3 HSE MSU MIPT	
Petr Ivanov 4 1 1900 80 1 HSE	