

Задача 5. Светофоры

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

По территории компьютерного лагеря проложен маршрут для поездок на электрокарах. Поскольку на электрокаре можно добраться до ИКТ-центра, то школьник Пахом решил воспользоваться им. Следуя по маршруту, электрокар проехал с постоянной скоростью один за другим два светофора с зеленым светом. Пахому известно, что оба светофора находятся на расстоянии x метров друг от друга и переключаются абсолютно синхронно: зеленый свет горит a минут, потом включается красный свет и горит в течение b минут, после чего светофор переключается опять на зеленый свет и он горит также в течение a минут, и так далее. Переключений на желтый свет у светофоров нет. Скорость движения электрокара по маршруту не превышает 1000 м/мин. Электрокар может проехать на светофоре в тот момент, когда светофор переключается с одного света на другой.

Приехав в ИКТ-центр, Пахом заинтересовался, с какой максимальной постоянной скоростью он мог ехать на электрокаре между двумя светофорами.

Требуется написать программу, которая позволит Пахому выяснить это.

Формат входного файла

Первая строка входного файла содержит три целых числа: a , b и x ($1 \leq a \leq 100$, $1 \leq b \leq 100$, $1 \leq x \leq 100\,000$).

Формат выходного файла

Выходной файл должен содержать одно число – максимальную возможную скорость электрокара между двумя светофорами. Ответ должен отличаться от правильного не более чем на 10^{-9} .

Примеры входных и выходных файлов

lights.in	lights.out
3 5 4000	800
5 10 21010	840.4

Система оценивания

Правильные решения для тестов, в которых ответ является целочисленным, будут оцениваться из 50 баллов.

Несмотря на выделение отдельной группы тестов для целочисленных ответов, на окончательную проверку будут приниматься только решения, правильно работающие для всех тестов из условия задачи.

Задача 6. Кондиционеры

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

При реализации проекта «Умная школа» было решено в каждый учебный класс выбранной для этого школы установить по кондиционеру нового поколения для автоматического охлаждения и вентиляции воздуха. По проекту в каждом классе должен быть установлен только один кондиционер и мощность кондиционера должна быть достаточной для размеров класса. Чем больше класс, тем мощнее должен быть кондиционер.

Все классы школы пронумерованы последовательно от 1 до n . Известно, что для каждого класса с номером i , требуется ровно один кондиционер, мощность которого больше или равна a_i ватт.

Администрации школы предоставили список из m различных моделей кондиционеров, которые можно закупить. Для каждой модели кондиционера известна его мощность и стоимость. Требуется написать программу, которая определит, за какую минимальную суммарную стоимость кондиционеров можно оснастить все классы школы.

Формат входного файла

Первая строка входного файла содержит одно целое число n ($1 \leq n \leq 50\,000$) - количество классов в школе.

Вторая строка содержит n целых чисел a_i ($1 \leq a_i \leq 1000$) - минимальная мощность кондиционера в ваттах, который можно установить в классе с номером i .

Третья строка содержит одно целое число m ($1 \leq m \leq 50\,000$) - количество предложенных моделей кондиционеров.

Далее, в каждой из m строк содержится пара целых чисел b_j и c_j ($1 \leq b_j \leq 1000$, $1 \leq c_j \leq 1000$) - мощность в ваттах j -й модели кондиционера и его цена в рублях соответственно.

Формат выходного файла

Выходной файл должен содержать одно число - минимальную суммарную стоимость кондиционеров в рублях. Гарантируется, что хотя бы один корректный выбор кондиционеров существует, и во всех классах можно установить подходящий кондиционер.

Примеры входных и выходных файлов

<code>cond.in</code>	<code>cond.out</code>
1 800 1 800 1000	1000
3 1 2 3 4 1 10 1 5 10 7 2 3	13

Пояснения к примерам

В первом примере нужно купить один единственно возможный кондиционер за 1000 рублей.

Во втором примере оптимально будет установить в первом и втором классах кондиционеры четвертого типа, а в третьем классе – кондиционер третьего типа. Суммарная стоимость этих кондиционеров будет составлять 13 рублей (3 + 3 + 7).

Система оценивания

Частичные решения для $n, m \leq 1000$ будут оцениваться из 50 баллов.

Задача 7. Конфеты

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

Кондитерская фабрика города П, в котором живет Петя, делает очень вкусные конфеты. Как-то раз, Петя собрался в гости к своему другу Васе, который живет в городе М. От города П до города М Петя решил доехать на поезде и взять с собой в подарок как можно больше коробок вкусных конфет.

Каждая коробка конфет имеет размеры $a \times b \times c$ сантиметров, где a – длина, b – ширина и c – высота коробки. Для перевозки конфет Петя хочет использовать один большой ящик в форме прямоугольного параллелепипеда. В ящик должны быть уложены все коробки конфет. Для того чтобы не повредить их, все коробки в ящике должны сохранять исходную ориентацию и располагаться в одном направлении. Петя может использовать ящик любого размера, но по правилам железнодорожных перевозок размер ящика по сумме трех измерений не может превышать N сантиметров.

Требуется написать программу, которая по заданным числам N , a , b и c определяет такой размер ящика, который должен использовать Петя, чтобы в него поместилось максимальное количество коробок конфет.

Формат входного файла

Первая строка входного файла содержит разделенные пробелами четыре целых числа: N , a , b , c ($1 \leq N, a, b, c \leq 10^9$).

Формат выходного файла

Выходной файл должен содержать три числа – длину, ширину и высоту ящика, который должен выбрать Петя и в который поместится максимальное количество коробок конфет. Если подходящих ответов несколько, необходимо вывести любой.

Примеры входных и выходных файлов

sweets.in	sweets.out
10 1 2 3	3 4 3
14 8 3 2	9 3 2

Пояснения к примерам

В первом примере выгоднее всего взять ящик размером $3 \times 4 \times 3$ сантиметров, в который поместится три коробки конфет в длину, две коробки конфет в ширину и одна коробка конфет в высоту.

Во втором примере для того, чтобы разместить хотя бы две коробки конфет, нужен ящик размером хотя бы $8 \times 3 \times 4$, у которого сумма измерений равна 15. В подходящий ящик поместится максимум одна коробка конфет. Подходящим также является ящик размером $9 \times 3 \times 2$, хотя он и не является минимальным.

Система оценивания

Частичные правильные решения для тестов, в которых $N \leq 1000$, будут оцениваться из 30 баллов.

Частичные правильные решения для тестов, в которых $N \leq 100\,000$, будут оцениваться из 60 баллов.

Задача 8. Волонтеры

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

В проведении межпланетной олимпиады по информатике принимает участие большое количество андроидов. Андроиды, придумывающие задачи, пишущие условия и разрабатывающие наборы тестов, являются членами научного комитета. Те андроиды, которые обеспечивают работоспособность компьютеров и установленного на них программного обеспечения, являются членами технического комитета. Есть третья большая группа андроидов, участвующих в организации и проведении многих других мероприятий, – это волонтеры.

У каждого волонтера есть свой непосредственный начальник в научном комитете и свой непосредственный начальник в техническом комитете. У каждого члена научного комитета, кроме его председателя, есть свой непосредственный начальник – другой член научного комитета. У каждого члена технического комитета, кроме его председателя, также есть свой непосредственный начальник, являющийся членом технического комитета.

В самом начале олимпиады происходит процесс, определяющий начальников для всех волонтеров, членов научного комитета и членов технического комитета. Происходит он следующим образом.

Все n волонтеров получают уникальные номера от одного до n , после чего выстраиваются друг за другом в порядке возрастания своих номеров. После этого m членов научного комитета, также имеющие уникальные номера от одного до m , по очереди, в порядке возрастания своих номеров, выбирают себе подчиненных следующим образом. Вначале, первый член научного комитета выбирает себе в подчинение некоторое количество волонтеров, стоящих в данный момент в строю подряд друг за другом. После этого те, кого он выбрал, выходят из строя, а он сам встает в строй на то место, где стояли его новоиспеченные подчиненные, заменяя собой сразу несколько андроидов в строю. Затем второй член научного комитета выбирает себе подчиненных из тех, кто теперь стоит в строю и заменяет их собой аналогично первому. Так этот процесс продолжается и каждый последующий член научного комитета может выбрать себе в подчинение находящихся в строю волонтеров и членов научного комитета. Последним выбирает себе подчиненных председатель научного комитета, и он должен выбрать всех андроидов, кто на тот момент будет находиться в строю.

После того, как в строю останется только председатель научного комитета, аналогичный процесс повторяется с самого начала, но уже с членами технического комитета. Важно то, что номера волонтеров при этом сохраняются и волонтеры выстраиваются в том же порядке, что и при выборе подчиненных членами научного комитета.

Считается, что член научного комитета может конфликтовать с членом технического комитета, если один и тот же волонтер подчиняется, возможно, не напрямую, и члену научного комитета, и члену технического комитета. Из этого определения, в частности, следует, что председатель научного комитета всегда может конфликтовать с председателем технического комитета. Несложно понять, что от общего количества таких возможных конфликтов между андроидами зависит то, насколько качественно будет проведена олимпиада.

Требуется написать программу, которая для каждого члена научного комитета подсчитывает количество членов технического комитета, с которыми он может конфликтовать, и определяет суммарное количество таких конфликтов.

Формат входного файла

Первая строка входного файла содержит числа n , m и k ($1 \leq n, m, k \leq 100\,000$) – количество волонтеров, членов научного комитета и членов технического комитета соответственно.

Последующие n строк содержат по два числа a_i и b_i ($1 \leq a_i \leq m, 1 \leq b_i \leq k$) – номер начальника соответствующего волонтера среди членов научного комитета и номер начальника соответствующего волонтера среди членов технического комитета.

Следующая строка содержит $(m-1)$ чисел c_i ($i = 0 \dots (m-1), i < c_i \leq m$) – номер начальника i -го члена научного комитета. Председатель научного комитета имеет номер m .

Следующая строка содержит $(k-1)$ чисел d_i ($i = 0 \dots (k-1), i < d_i \leq k$) – номер начальника i -го члена технического комитета. Председатель технического комитета имеет номер k .

Гарантируется, что все входные данные корректны, то есть они получены в результате процесса назначения подчиненных, описанного выше.

Формат выходного файла

Первая строка выходного файла должна содержать одно число – суммарное количество искоемых конфликтов.

Примеры входных и выходных файлов

<code>volunteers.in</code>	<code>volunteers.out</code>
4 3 4 1 2 1 1 2 1 2 4 3 3 3 3 4	11

Пояснения к примерам

В представленном примере только второй член научного комитета и второй член технического комитета не могут конфликтовать друг с другом.

Система оценивания

Частичные правильные решения для тестов, в которых количества волонтеров, членов научного комитета и членов технического комитета не превышают 100, будут оцениваться из 30 баллов.

Частичные правильные решения для тестов, в которых количества волонтеров, членов научного комитета и членов технического комитета не превышают 5000, будут оцениваться из 60 баллов.