

Комета Бармалея

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Как известно, комета Бармалея видна с Земли каждые C лет. Любопытно, что это происходит в годы, кратные C , т.е. $C, 2 \times C, 3 \times C$ и т.д. Не каждому суждено увидеть эту комету хотя бы однажды в жизни. Впрочем, находятся счастливые долгожители, заставшие её прилёт даже несколько раз. Считается, что впервые эту комету увидел и документировал знаменитый средневековый астроном Бармалео Бармалей. В честь него она и получила своё имя. Говорят, за свою долгую жизнь он успел сделать много великих открытий в самых разных областях науки. Однако недавно историки засомневались, правда ли все открытия, которые ему приписываются, Бармалео Бармалей сделал сам. В частности, они заинтересовались, сколько раз за свою жизнь учёный мог видеть комету, названную в его честь.

Бармалео Бармалей родился 1 января в год A и умер 31 декабря в год B . Сколько раз за его жизнь комета была видна с Земли? Мы считаем, что он мог видеть комету, даже будучи младенцем или глубоким стариком, т.е. если она прилетала в год A или B .

Программа получает на вход три целых числа A, B и C ($1 \leq A \leq B \leq 2 \times 10^9$,

$1 \leq C \leq 2 \times 10^9$) и должна вывести одно целое число – количество раз, которое комета была видна между годами A и B включительно.

Система оценивания.

Решение, правильно работающее только для случаев, когда все числа не превосходят 10 000, будет оцениваться в 60 баллов.

Пример

Ввод	Вывод
1850 1900 50	2

Переключение окон

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Дима – программист, поэтому на его компьютере всегда открыто много окон. Так как у Димы не очень большой монитор, на нём может отображаться только одно окно. В каждый момент времени оконный менеджер хранит список открытых окон, первое окно списка отображается на мониторе. Для переключения окон Дима использует сочетание клавиш Alt + Tab. Если удерживать эту кнопку нажатой в течение T секунд, то $T + 1$ -е по счёту окно в текущей нумерации переместится на первую позицию, а относительный порядок остальных окон не изменится. Например, на рисунке ниже показано, что произойдёт с порядком окон, если нажимать на Alt + Tab в течение 3 секунд.



Если держать Alt + Tab $N - 1$ секунду, то первым станет последнее окно из списка. Список открытых окон «зациклен», за последним окном следует первое окно из списка, т. е. если удерживать Alt + Tab нажатым N секунд, то окно, которое было первым в списке, останется на первом месте. Если удерживать Alt + Tab $N + 1$ секунду, на первое место переместится второе по счёту окно и т.д.

В начале рабочего дня любимая среда разработки Димы имела номер M в списке открытых окон. В течение дня Дима K раз использовал сочетание клавиш Alt + Tab. Определите, на какой позиции находится его любимая среда разработки в конце дня.

Первая строка входных данных содержит целое число N , $1 \leq N \leq 10^5$ – количество окон на экране. Вторая строка содержит целое число M , $1 \leq M \leq N$ – номер, который имела любимая среда разработки Димы в начале дня. Третья строка содержит целое число K , $1 \leq K \leq 10^5$ – количество раз, которое Дима нажимал Alt + Tab. В последующих K строках содержатся целые положительные числа, не превосходящие 10^5 – длительность каждого нажатия в секундах.

Программа должна вывести одно целое число – позицию любимой среды Димы в конце рабочего дня.

Система оценивания

Решение, правильно работающее только для случаев, когда $1 \leq N \leq 3$, $1 \leq K \leq 3$ и все продолжительности нажатий не превосходят $N - 1$, будет оцениваться в 30 баллов.

Решение, правильно работающее только для случаев, когда $1 \leq N \leq 100$ и $1 \leq K \leq 100$, будет оцениваться в 60 баллов.

Пример

Ввод	Вывод
3	3
2	
3	
1	
5	
2	

SNTP

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Для того чтобы компьютеры поддерживали актуальное время, они могут обращаться к серверам точного времени SNTP (Simple Network Time Protocol). К сожалению, компьютер не может просто получить время у сервера, потому что информация по сети передаётся не мгновенно: пока сообщение с текущим временем дойдёт до компьютера, оно потеряет свою актуальность. Протокол взаимодействия клиента (компьютера, запрашивающего точное время) и сервера (компьютера, выдающего точное время) выглядит следующим образом:

- 1) Клиент отправляет запрос на сервер и запоминает время отправления A (по клиентскому времени).
- 2) Сервер получает запрос в момент времени B (по точному серверному времени) и отправляет клиенту сообщение, содержащее время B .
- 3) Клиент получает ответ на свой запрос в момент времени C (по клиентскому времени) и запоминает его. Теперь клиент, из предположения, что сетевые задержки при передаче сообщений от клиента серверу и от сервера клиенту одинаковы, может определить и установить себе точное время, используя известные значения A , B , C .

Вам предстоит реализовать алгоритм, с точностью до секунды определяющий точное время для установки на клиенте по известным A , B и C . При необходимости округлите результат до целого числа секунд по правилам арифметики (в меньшую сторону, если дробная часть числа меньше $\frac{1}{2}$, иначе в большую сторону).

Возможно, что, пока клиент ожидал ответа, по клиентскому времени успели наступить новые сутки, однако известно, что между отправкой клиентом запроса и получением ответа от сервера прошло менее 24 часов.

Программа получает на вход три временные метки A , B , C , по одной в каждой строке. Все временные метки представлены в формате «hh:mm:ss», где «hh» – это часы, «mm» – минуты, «ss» – секунды. Часы, минуты и секунды записываются ровно двумя цифрами каждое (возможно, с дополнительными нулями в начале числа).

Программа должна вывести одну временную метку в формате, описанном во входных данных, – вычисленное точное время для установки на клиенте. В выводе не должно быть пробелов, пустых строк в начале вывода.

Система оценивания

Решение, правильно работающее только для случаев, когда все три входных времени и ответ на задачу принадлежат одним суткам, будет оцениваться в 60 баллов.

Пример

Ввод	Вывод
15:01:00	18:10:05
18:09:45	
15:01:40	

Лифт в бизнес-центре

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Рабочий день закончился, и сотрудники бизнес-центра собрались по домам. Бизнес-центр представляет собой N -этажное здание, этажи пронумерованы от 1 до N снизу вверх. Все сотрудники хотят спуститься на парковку, которая расположена в подвальном помещении на один этаж ниже первого. Бизнес-центр оборудован лифтом, который может перевозить не более K человек одновременно. Лифт перемещается вверх или вниз на один этаж за одну секунду, посадка и высадка пассажиров происходят мгновенно. Изначально лифт расположен на уровне парковки. Известно, сколько людей хотят спуститься на парковку с каждого из N этажей. Определите, какое минимальное время потребуется, чтобы перевезти на парковку всех сотрудников бизнес-центра. Первая строка входных данных содержит наибольшее возможное число людей в лифте K , $1 \leq K \leq 10^9$. Вторая строка содержит число этажей в бизнес-центре N , $1 \leq N \leq 10^5$. Следующие N строк содержат целые неотрицательные числа – число людей, ожидающих лифт на 1, 2, ..., N -м этаже соответственно, эти числа не превосходят 10^9 каждое. В здании находится хотя бы один человек.

Программа должна вывести ровно одно целое число – наименьшее время в секундах, за которое всех людей можно перевезти на парковку.

Система оценивания

Решение, правильно работающее только для случаев, когда $N \leq 100$ и количество людей на каждом этаже не превосходит 100, будет оцениваться в 40 баллов.

Решение, правильно работающее только для случаев, когда $N \leq 100$, а дополнительного ограничения на число людей нет, будет оцениваться в 70 баллов.

Пример

Ввод	Вывод
2	8
3	
3	
0	
1	

Счастливые билеты

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

На автобусных билетах указываются их номера. Номера всех билетов всегда записываются при помощи одного и того же количества цифр, при этом число используемых цифр чётно. При необходимости числа дополняются ведущими нулями. К примеру, если для записи используют 4 цифры, то 514 будет записано как 0514. Билеты отпечатаны на лентах, билеты на каждой ленте нумеруются подряд числами от 00...01 до 99...99.

Счастливым считается тот билет, у которого сумма цифр первой половины равна сумме цифр второй половины, например, билеты 1001 и 123051 счастливые, а 7778 и 39 – нет.

Сегодня Дима зашел в автобус, и кондуктор выдал ему билет с номером N. Поскольку Диме ехать достаточно долго, а заняться чем-нибудь надо, он стал думать, какой номер будет иметь следующий счастливый билет, выданный из той же ленты, что и Димин билет. Если в текущей ленте не осталось счастливых билетов, Диму интересует номер минимального счастливого билета из новой ленты.

В первой и единственной строке входного файла содержится номер Диминого билета N, записанный с ведущими нулями. Количество цифр в записи числа N не превосходит 100 000 и чётно.

Программа должна вывести номер следующего счастливого билета из текущей ленты в таком же формате. Если такого билета не существует, надо вывести номер минимального счастливого билета из новой ленты. В выводе не должно быть пробелов, пустых строк в начале вывода.

Система оценивания

Решение, правильно работающее только для случаев, когда номер билета содержит ровно 4 цифры, будет оцениваться в 20 баллов.

Решение, правильно работающее только для случаев, когда номер билета содержит ровно 8 цифр, будет оцениваться в 20 баллов (вместе с предыдущей группой – 40 баллов).

Решение, правильно работающее только для случаев, когда номер билета содержит не более 16 цифр, будет оцениваться в 60 баллов.

Пример

Ввод	Вывод
0514	0523