

Задача 1. Флеш и Зум на пробежке

Ограничение по времени: 1 секунда

Флеш и Зум сильно постарели, из-за чего у них больше нет сил для того, чтобы сражаться друг с другом. Поэтому сейчас они просто друзья, которые каждое утро выходят на пробежку.

Сегодня они решили бегать по стадиону, который представляет собой окружность длины d километров. Герои одновременно начнут бежать в одном направлении из одной точки. При этом Флеш побегит со скоростью v_1 км/с, а Зум — со скоростью v_2 км/с. После t секунд бега они оба остановятся и будут отдыхать. А вам нужно посчитать, сколько километров еще надо пробежать Флешу (**не обязательно в том же направлении, в котором он бежал ранее**), чтобы оказаться с Зумом в одной точке.

Формат входных данных

Первая строка содержит одно целое число d ($1 \leq d \leq 10^9$) — длину стадиона в километрах.

Вторая строка содержит одно целое число v_1 ($1 \leq v_1 \leq 10^9$) — скорость Флеша в км/с.

Третья строка содержит одно целое число v_2 ($1 \leq v_2 \leq 10^9$) — скорость Зума в км/с.

Четвертая строка содержит одно целое число t ($1 \leq t \leq 10^9$) — длительность пробежки в секундах.

Гарантируется, что $v_1 \cdot t \leq 10^9$ и $v_2 \cdot t \leq 10^9$.

Формат выходных данных

Выведите одно целое число — минимальное количество километров, которое необходимо пробежать Флешу, чтобы оказаться с Зумом в одной точке.

Система оценки

Решения, правильно работающие при $t \leq 100\,000$, будут оцениваться в 50 баллов.

Пример

стандартный ввод	стандартный вывод
10	4
3	
5	
3	

Замечание

Рассмотрим первый пример. За три секунды Флеш пробежит 9 километров, а Зум — 15 километров. Тогда расстояние по направлению бега от Флеша до Зума будет равно 6 километрам, а если Флеш развернется, то 4 километрам. Таким образом, ответ — 4 километра.

Задача 2. Миша и сериалы

Ограничение по времени: 1 секунда

Недавно вышел новый сезон любимого сериала Миши. Каждая серия в нем длится ровно n секунд. При этом перед началом каждой серии проигрывается одинаковое интро длиной m секунд, которое одинаковое во всех сериях, и очень надоело Мише.

Миша купил себе новую клавиатуру, на которой есть кнопка для перемотки видео. Одно нажатие на такую кнопку перематывает воспроизведение видео на k секунд вперед. Нажатие на кнопку и перемотка происходят мгновенно, а также Мише не нужно делать перерывов между нажатиями, поэтому он может мгновенно несколько раз нажать на кнопку перемотки.

К сожалению, не всегда удастся перемотать интро так, чтобы не потерять ни секунды событий сериала. Миша решил, что если придется пропустить не более, чем t секунд сериала, то он готов смириться с этим. Теперь Мише интересно, какое максимальное количество секунд сериала он посмотрит, если пропустить как можно больше интро, не пропустив при этом более t секунд сериала.

Проще говоря, если у Миши получится перемотать интро целиком, потеряв при этом не более t секунд сериала, то он поступит именно так. Иначе он пропустит столько секунд интро, сколько возможно.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 10^9$) — длительность одной серии в секундах.

Вторая строка содержит одно целое число m ($0 \leq m \leq 10^9$) — длительность интро в секундах.

Третья строка содержит одно целое число k ($1 \leq k \leq 10^9$) — количество секунд, которые перематываются при нажатии на кнопку.

Четвертая строка содержит одно целое число t ($0 \leq t < n$) — максимальное количество секунд сериала, которые Миша готов пропустить.

Формат выходных данных

Выведите одно целое число — количество секунд сериала, которые посмотрит Миша.

Система оценки

Помимо примеров из условия, задача содержит 25 тестов. Каждый из них оценивается в 4 балла.

Примеры

стандартный ввод	стандартный вывод
10 5 2 1	9
10 5 2 0	10
30 10 5 2	30

Замечание

В первом примере серия начинается с интро длиной 5 секунд, после чего начинается основная часть серии, которая длится 10 секунд. За одно нажатие на кнопку Миша проматывает 2 секунды, при этом он готов пропустить не более 1 секунды сериала. Поэтому мальчик может три раза нажать на кнопку перемотки, после чего он посмотрит 9 секунд сериала.

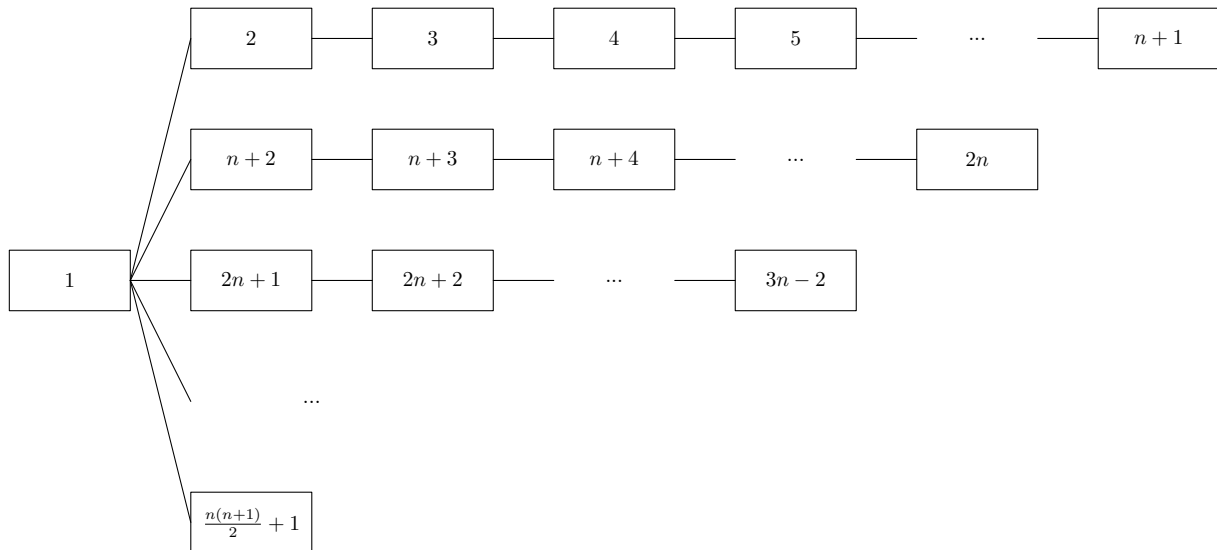
Во втором примере Миша не готов жертвовать просмотром серии, поэтому он посмотрит всю серию целиком, перемотав 4 секунды интро из 5.

В третьем примере Миша сможет полностью пропустить интро за два нажатия, после чего он посмотрит всю серию.

Задача 3. Серверы

Ограничение по времени: 1 секунда

Тестирующая система располагается на $\frac{n(n+1)}{2} + 1$ серверах, пронумерованных целыми числами от 1 до $\frac{n(n+1)}{2} + 1$. Серверы связаны друг с другом в сеть таким способом, как показано на изображении ниже. Прямоугольниками обозначены серверы, для каждого сервера указан его номер. Линиями указаны кабели, соединяющие серверы.



Главный сервер системы имеет номер 1. Остальные серверы соединены в n цепочек при помощи кабелей. Первая цепочка состоит из n серверов с номерами от 2 до $n+1$. Вторая цепочка состоит из $n-1$ серверов с номерами от $n+2$ до $2n$. Третья цепочка состоит из $n-2$ серверов с номерами от $2n+1$ до $3n-2$. И так далее. Последняя цепочка состоит из одного сервера с номером $\frac{n(n+1)}{2} + 1$. Серверы в каждой цепочке соединены друг с другом как показано на иллюстрации. Главный сервер соединен с первым сервером каждой цепочки.

Когда участник отправляет решение задачи на проверку, оно поступает на главный сервер. После этого происходит следующий процесс: сначала главный сервер тратит a секунд на подготовку данных, после чего мгновенно отправляет их первому серверу первой цепочки (то есть серверу с номером 2). Затем главный сервер тратит еще a секунд на подготовку данных, после чего мгновенно отправляет их первому серверу второй цепочки (то есть серверу с номером $n+2$). И так далее до тех пор, пока главный сервер не отправит последнее сообщение, которое адресовано первому серверу n -й цепочки.

Каждый из серверов, кроме главного, работает следующим образом: после получения сообщения сервер тратит b секунд на обработку данных, после чего мгновенно передает их следующему серверу в цепочке. Когда данные доходят до последнего сервера цепочки, он тратит b секунд на их обработку, после чего мгновенно передает их предпоследнему серверу в цепочке. После этого данные начинают передаваться между серверами по направлению к началу цепочки таким же образом, каким они передавались ранее (каждый сервер тратит b секунд на обработку данных, после чего передает их предыдущему серверу по цепочке).

Наконец, когда данные проходят по всей цепочке серверов и попадают на первый сервер цепочки, он тратит a секунд на обработку данных и передает их на главный сервер. После этого работа данной цепочки серверов прекращается.

Как видите, на Марсе процесс проверки решения участника является достаточно сложной задачей. Вам предстоит вычислить, через сколько времени главный сервер получит обработанные данные от первых серверов всех цепочек. Считайте, что главный сервер получил решение на проверку в момент времени 0.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 10^9$) — количество цепочек серверов.

Вторая строка содержит одно целое число a ($1 \leq a \leq 10^9$) — количество секунд, которое тратит на обработку данных главный сервер.

Третья строка содержит одно целое число b ($1 \leq b \leq 10^9$) — количество секунд, которое тратят на обработку данных остальные серверы.

Формат выходных данных

Выведите одно целое число — количество секунд после приема решения на проверку, после которых главный сервер получит обработанные данные от первых серверов всех цепочек.

Обратите внимание, что ответ в этой задаче может превышать возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битные целочисленные типы данных (тип `int64` в языке Pascal, тип `long long` в C++, тип `long` в Java и C#).

Система оценки

В этой задаче, помимо тестов из условия, есть 20 тестов. Каждый из них оценивается в 5 баллов.

Решения, правильно работающие при $n \leq 10^5$, будут набирать не менее 50 баллов.

Примеры

стандартный ввод	стандартный вывод
3 3 2	14
5 2 3	28

Замечание

Рассмотрим первый пример.

Перемещение данных по первой цепочке выглядит следующим образом:

- В момент времени 3 сервер 1 передаст данные серверу 2.
- В момент времени 5 сервер 2 передаст данные серверу 3.
- В момент времени 7 сервер 3 передаст данные серверу 4.
- В момент времени 9 сервер 4 передаст данные серверу 3.
- В момент времени 11 сервер 3 передаст данные серверу 2.
- В момент времени 14 сервер 2 передаст данные серверу 1.

Перемещение данных по второй цепочке выглядит следующим образом:

- В момент времени 6 сервер 1 передаст данные серверу 3.
- В момент времени 8 сервер 3 передаст данные серверу 4.
- В момент времени 10 сервер 4 передаст данные серверу 3.
- В момент времени 13 сервер 3 передаст данные серверу 1.

Перемещение данных по третьей цепочке выглядит следующим образом:

- В момент времени 9 сервер 1 передаст данные серверу 4.
- В момент времени 12 сервер 4 передаст данные серверу 1.

Таким образом, главный сервер получит обработанные данные в моменты 14, 13 и 12, поэтому ответ равен 14.

Задача 4. Скучные квартиры

Ограничение по времени: 1 секунда

Как известно, одно из самых веселых занятий для маленького ребенка — звонить в случайные квартиры, используя домофон.

Мальчик Вася решил обзвонить квартиры в своем доме. Всего в доме n квартир, которые пронумерованы целыми числами от 1 до n . Мальчик решил звонить только в квартиры со *скучными* номерами. Номер называется скучным, если и только если он состоит из одинаковых цифр. Например, квартиры с номерами 222, 1 и 999 являются скучными, а квартиры с номерами 42 и 20 — не являются.

Для того чтобы Васе было весело, он решил обзванивать квартиры в следующем порядке. Сначала мальчик обзвонит все квартиры с номерами, состоящими только из единиц, в порядке возрастания номеров квартир. Затем он обзвонит все квартиры с номерами, состоящими только из двоек, в порядке возрастания номеров, и так далее. Таким образом, последовательность номеров квартир, в которые будет звонить мальчик, окажется следующей:

1, 11, 111, ..., 2, 22, 222, ..., 3, 33, 333, ..., 4, 44, 444, ..., 5, 55, 555, ..., 6, 66, 666, ..., 7, 77, 777, ..., 8, 88, 888, ..., 9, 99, 999, ...

Шалость Васи могла бы продолжаться долго, но люди, живущие в квартире с номером k ответили мальчику, из-за чего он испугался и сразу же убежал. Теперь Васе интересно, как много квартир он успел обзвонить. Но он еще очень маленький, поэтому посчитать количество квартир придется вам.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 10^9$) — количество квартир в доме.

Вторая строка содержит одно целое число k ($1 \leq k \leq n$) — номер квартиры, жители которой ответили Васе. Гарантируется, что k является скучным номером.

Формат выходных данных

Выведите одно целое число — количество квартир, которое успел обзвонить Вася. Обратите внимание, что квартира с номером k также должна быть учтена.

Система оценки

Решения, правильно работающие, если n является скучным номером, будут оцениваться не менее чем в 40 баллов.

Решения, правильно работающие, если $n \leq 1000$, будут оцениваться в 60 баллов.

Примеры

стандартный ввод	стандартный вывод
100 22	4
10 1	1
100 99	18

Замечание

В доме из первого примера 100 квартир, а Васе ответили жители 22-й квартиры. То есть Вася успел позвонить в квартиры 1, 11, 2 и 22 (именно в таком порядке), после чего убежал.

В доме из второго примера 10 квартир, и Васе ответили жители квартиры 1.

В доме из третьего примера 100 квартир, а Васе ответили жители 99-й квартиры. Поэтому он позвонил во все квартиры с двузначными и однозначными скучными номерами.

Задача 5. Университетская команда

Ограничение по времени: 1 секунда

Ваш университет всюю готовится к новому сезону командных олимпиад. Как несложно догадаться, для этого необходимо собрать команду, которая будет представлять учебное заведение.

Всего в университете учатся n студентов, i -й из них имеет силу a_i . Для участия в олимпиадах необходимо выбрать **ровно** k людей и расположить их в каком-то порядке. Пусть были выбраны студенты с номерами i_1, i_2, \dots, i_k (именно в таком порядке). Тогда *слабость* команды равна $|a_{i_2} - a_{i_1}| + |a_{i_3} - a_{i_2}| + \dots + |a_{i_k} - a_{i_{k-1}}|$. Иначе говоря, слабость команды равна сумме модулей разностей сил соседних участников команды, если их расположить в выбранном порядке. **Обратите внимание, что порядок студентов вы выбираете сами.**

Администрация университета просит вас определить минимально возможную слабость команды.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 100\,000$) — количество студентов, обучающихся в университете.

Вторая строка содержит одно целое число k ($1 \leq k \leq n$) — количество людей в составе команды.

Каждая из следующих n строк содержит целое число a_i ($1 \leq a_i \leq 10^9$) — силу i -го студента.

Формат выходных данных

Выведите одно целое число — минимально возможную слабость команды.

Система оценки

Решения, правильно работающие при $n \leq 10$, будут оцениваться в 25 баллов.

Решения, правильно работающие при $n \leq 1\,000$, будут оцениваться в 50 баллов.

Решения, правильно работающие при $a_i \leq 1\,000\,000$, будут оцениваться в 50 баллов.

Пример

стандартный ввод	стандартный вывод
5 3 1 5 3 2 1	1

Замечание

Рассмотрим пример. Если взять студентов с номерами 1, 5 и 4 (именно в таком порядке), то слабость команды будет равна $|1 - 1| + |2 - 1| = 0 + 1 = 1$. Можно доказать, что меньшую слабость получить не получится.