

Задача А. Космическое поселение

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайт

Для освоения Марса требуется построить исследовательскую базу. База должна состоять из n одинаковых модулей. Каждый модуль представляет собой жилой отсек, который в основании имеет форму прямоугольника размером $a \times b$ метров.

Для повышения надежности модулей инженеры могут добавить вокруг каждого модуля дополнительный защитный слой. Толщина этого слоя должна составлять целое число метров, и все модули должны иметь одинаковую толщину защитного слоя. Модуль с защитным слоем, толщина которой равна d метрам, будет иметь в основании форму прямоугольника размером $(a + 2d) \times (b + 2d)$ метров.

Все модули должны быть расположены на заранее подготовленном прямоугольном поле размером $w \times h$ метров. При этом они должны быть организованы в виде регулярной сетки: их стороны должны быть параллельны сторонам поля, и модули должны быть ориентированы одинаково.

Требуется написать программу, которая по заданному количеству и размеру модулей, а также размеру поля для их размещения, определяет максимальную толщину дополнительного защитного слоя, который можно добавить к каждому модулю.

Формат входного файла

Входной файл содержит пять разделенных пробелами целых чисел: n , a , b , w и h ($1 \leq n, a, b, w, h \leq 10^{18}$). Гарантируется, что без дополнительного защитного слоя все модули можно разместить в поселении описанным образом.

Формат выходного файла

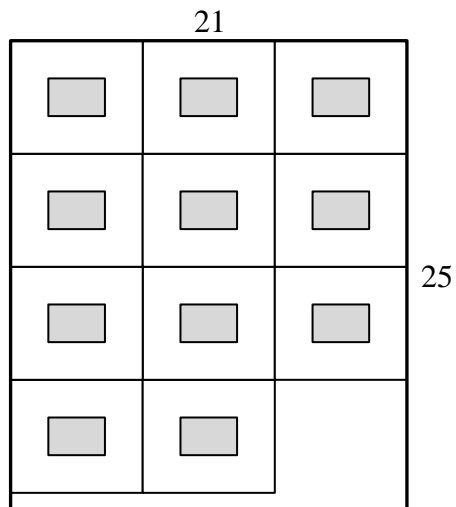
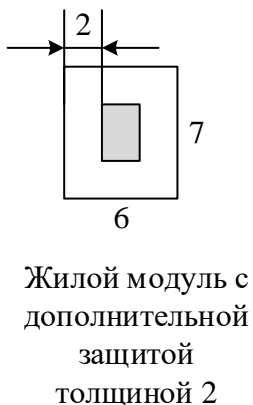
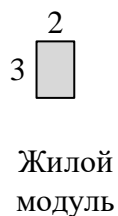
Выходной файл должен содержать одно целое число: максимальную возможную толщину дополнительного защитного слоя. Если дополнительный защитный слой установить не удастся, требуется вывести число 0.

Примеры входных и выходных файлов

<code>space.in</code>	<code>space.out</code>
11 2 3 21 25	2
1 5 5 6 6	0

Пояснение к примерам

В первом примере можно установить дополнительный защитный слой толщиной 2 метра и разместить модули на поле, как показано на рисунке.



Размещение модулей на поле

Во втором примере жилой отсек имеет в основании размер 5×5 метров, а поле – размер 6×6 метров. Добавить дополнительный защитный слой к модулю нельзя.

Задача В. Выбор зала

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайт

Для проведения церемонии открытия олимпиады по информатике организаторы осуществляют поиск подходящего зала. Зал должен иметь форму прямоугольника, длина каждой из сторон которого является целым положительным числом.

Чтобы все участники церемонии поместились в зале, и при этом он не выглядел слишком пустым, площадь зала должна находиться в пределах от A до B квадратных метров, включительно.

Чтобы разместить на стенах зала плакаты, рассказывающие об успехах школьников на олимпиадах, но при этом не создать ощущения, что успехов слишком мало, периметр зала должен находиться в пределах от C до D метров, включительно.

Прежде чем сделать окончательный выбор, организаторы олимпиады решили просмотреть по одному залу каждого подходящего размера. Залы с размерами $Y \times Z$ и $Z \times Y$ считаются одинаковыми. Чтобы понять необходимый объем работ по просмотру залов организаторы задались вопросом, сколько различных залов удовлетворяют приведенным выше ограничениям.

Требуется написать программу, которая по заданным A , B , C и D определяет количество различных залов, площадь которых находится в пределах от A до B , а периметр — от C до D , включительно.

Формат входного файла

Входной файл содержит четыре разделенных пробелами целых числа: A , B , C и D ($1 \leq A \leq B \leq 10^9$, $4 \leq C \leq D \leq 10^9$).

Формат выходного файла

Выходной файл должен содержать одно число — искомое количество залов.

Пример входных и выходных файлов

<code>hall.in</code>	<code>hall.out</code>
2 10 4 8	3

Пояснения к примеру

В примере ограничениям удовлетворяют залы следующих размеров: 1×2 , 1×3 , 2×2 .

Задача С. Вырубка леса

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Фермер Николай нанял двух лесорубов: Дмитрия и Федора, чтобы вырубить лес, на месте которого должно быть кукурузное поле. В лесу растут X деревьев.

Дмитрий срубает по A деревьев в день, но каждый K -й день он отдыхает и не срубает ни одного дерева. Таким образом, Дмитрий отдыхает в K -й, $2K$ -й, $3K$ -й день, и т.д.

Федор срубает по B деревьев в день, но каждый M -й день он отдыхает и не срубает ни одного дерева. Таким образом, Федор отдыхает в M -й, $2M$ -й, $3M$ -й день, и т.д.

Лесорубы работают параллельно и, таким образом, в дни, когда никто из них не отдыхает, они срубает $A + B$ деревьев, в дни, когда отдыхает только Федор — A деревьев, а в дни, когда отдыхает только Дмитрий — B деревьев. В дни, когда оба лесоруба отдыхают, ни одно дерево не срубается.

Фермер Николай хочет понять, за сколько дней лесорубы срубят все деревья, и он сможет засеять кукурузное поле.

Требуется написать программу, которая по заданным целым числам A , K , B , M и X определяет, за сколько дней все деревья в лесу будут вырублены.

Формат входного файла

Входной файл содержит пять целых чисел, разделенных пробелами: A , K , B , M и X ($1 \leq A, B \leq 10^9$, $2 \leq K, M \leq 10^{18}$, $1 \leq X \leq 10^{18}$).

Формат выходного файла

Выходной файл должен содержать одно целое число — искомое количество дней.

Пример входных и выходных файлов

<code>forest.in</code>	<code>forest.out</code>
2 4 3 3 25	7

Пояснение к примеру

В приведенном примере лесорубы вырубят 25 деревьев за 7 дней следующим образом:

- 1-й день: Дмитрий срубает 2 дерева, Федор срубает 3 дерева, итого 5 деревьев;
- 2-й день: Дмитрий срубает 2 дерева, Федор срубает 3 дерева, итого 10 деревьев;
- 3-й день: Дмитрий срубает 2 дерева, Федор отдыхает, итого 12 деревьев;
- 4-й день: Дмитрий отдыхает, Федор срубает 3 дерева, итого 15 деревьев;
- 5-й день: Дмитрий срубает 2 дерева, Федор срубает 3 дерева, итого 20 деревьев;
- 6-й день: Дмитрий срубает 2 дерева, Федор отдыхает, итого 22 дерева;

- 7-й день: Дмитрий срубает 2 дерева, Федор срубает оставшееся 1 дерево, итого все 25 деревьев срублены.

Задача D. Кампус

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайт

Новое здание кампуса Университета Байтбурга имеет n этажей, пронумерованных снизу вверх от 1 до n . Комнаты студентов расположены в нескольких подъездах.

В каждом подъезде на этажах, номер которых кратен числу k , расположено по x комнат, а на остальных этажах расположено по y комнат.

Комнаты внутри каждого подъезда пронумерованы последовательными натуральными числами. Номера комнат на первом этаже имеют наименьшие значения в этом подъезде, затем следуют номера комнат на втором этаже, и так далее. Комнаты в первом подъезде пронумерованы, начиная с 1, в каждом следующем подъезде нумерация комнат начинается с числа, следующего после максимального номера комнаты в предыдущем подъезде.

На рис. 1 показаны номера комнат в здании с $n = 7$ этажами, 3 подъездами, и параметрами $k = 3$, $x = 2$, $y = 3$.

	Подъезд 1	Подъезд 2	Подъезд 3
7 этаж	17, 18, 19	36, 37, 38	55, 56, 57
6 этаж	15, 16	34, 35	53, 54
5 этаж	12, 13, 14	31, 32, 33	50, 51, 52
4 этаж	9, 10, 11	28, 29, 30	47, 48, 49
3 этаж	7, 8	26, 27	45, 46
2 этаж	4, 5, 6	23, 24, 25	42, 43, 44
1 этаж	1, 2, 3	20, 21, 22	39, 40, 41

Рис. 1. Пример нумерации комнат в здании

Для организации расселения студентов администрация кампуса должна по номеру комнаты оперативно определять этаж, на котором она находится.

Требуется написать программу, которая по заданным числам n , k , x и y , а также по номерам комнат, определяет для каждой комнаты, на каком этаже она находится.

Формат входного файла

Первая строка входного файла содержит натуральные числа n , k , x и y ($1 \leq n \leq 10^9$, $1 \leq k \leq n$, $1 \leq x, y \leq 10^9$). Соседние числа разделены ровно одним пробелом.

Вторая строка входного файла содержит натуральное число q — количество номеров комнат, для которых требуется определить этаж ($1 \leq q \leq 1000$).

Третья строка содержит q целых чисел a_1, a_2, \dots, a_q — номера комнат ($1 \leq a_i \leq 10^{18}$). Можно считать, что в здании так много подъездов, что все комнаты с заданными номерами существуют.

Формат выходного файла

Требуется вывести q чисел, по одному на строке. Для каждого номера комнаты во входном файле требуется вывести номер этажа, на котором она находится.

Пример входных и выходных файлов

<code>building.in</code>	<code>building.out</code>
7 3 2 3	1
4	7
1 19 20 50	1
	5

Описание подзадач и системы оценивания

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения		Необходимые подзадачи
		n, x, y	q, a_i	
1	31	$1 \leq n \leq 10,$ $1 \leq x, y \leq 10$	$q = 1$ $1 \leq a_i \leq 100$	
2	19	$1 \leq n \leq 10^7,$ $1 \leq x, y \leq 10^9$	$q = 1$ $1 \leq a_i \leq 10^7$	1
3	16	$1 \leq n \leq 10^9,$ $1 \leq x, y \leq 10^9,$ $x = y$	$1 \leq q \leq 1000$ $1 \leq a_i \leq 10^{18}$	
4	34	$1 \leq n \leq 10^9$ $1 \leq x, y \leq 10^9$	$1 \leq q \leq 1000$ $1 \leq a_i \leq 10^{18}$	1, 2, 3

Получение информации о результатах окончательной проверки

По запросу сообщается результат окончательной проверки на каждом тесте.

Задача Е. Улучшение успеваемости

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайт

В лицее на уроках информатики ответы учеников оцениваются целым числом баллов от 2 до 5. Итоговая оценка по информатике выставляется как среднее арифметическое оценок на всех уроках, округленное до ближайшего целого числа. Если среднее значение находится ровно посередине между двумя целыми числами, то оценка округляется вверх.

Примеры округления оценок приведены в таблице.

Оценки на уроках	Среднее арифметическое	Итоговая оценка
2, 3, 5	$\frac{2 + 3 + 5}{3} = 3\frac{1}{3}$	3
3, 3, 4, 4	$\frac{3 + 3 + 4 + 4}{4} = 3\frac{1}{2}$	4
5, 5, 5, 3, 5	$\frac{5 + 5 + 5 + 3 + 5}{5} = 4\frac{3}{5}$	5

Все ученики лицея стремятся получить итоговую оценку по информатике не ниже 4 баллов. К сожалению, один из учеников получил на уроках a двоек, b троек и c четверок. Теперь он планирует получить несколько пятерок, причем хочет, чтобы итоговая оценка была не меньше 4 баллов. Ему надо понять, какое минимальное количество пятерок ему необходимо получить, чтобы добиться своей цели.

Требуется написать программу, которая по заданным целым неотрицательным числам a , b и c определяет минимальное количество пятерок, которое необходимо получить ученику, чтобы его итоговая оценка по информатике была не меньше 4 баллов.

Формат входных данных

Входные данные содержат три строки. Первая строка содержит целое неотрицательное число a , вторая строка содержит целое неотрицательное число b , третья строка содержит целое неотрицательное число c ($0 \leq a, b, c \leq 10^{15}$, $a + b + c \geq 1$).

Формат выходных данных

Выходные данные должны содержать одно число: минимальное число пятерок, которое необходимо получить ученику, чтобы итоговая оценка была не меньше 4 баллов.

Примечание

Следует обратить внимание, что входные данные в этой и других задачах не помещаются в стандартный 32-битный тип данных. Необходимо использовать 64-битный тип данных (`long long` в C++, `int64` в Паскале, `long` в Java).

Пример входных и выходных данных

входные данные	выходные данные
2 0 0	2

Задача F. Превышение скорости

Превышение скорости является опасным нарушением, значительно увеличивающим вероятность трагических последствий транспортных происшествий. К сожалению контроль скорости с использованием радаров и камер не решает проблему полностью. Притормаживая перед камерами, водители едут со значительным превышением на участках дорог, где контроль не ведётся. С целью предотвращения такого поведения используется назначение штрафа за гарантированное превышение скорости, основанное на времени проезда дороги.

Рассмотрим дорогу, состоящую из n участков, пронумерованных от 1 до n . Длина i -го участка составляет l_i метров. На i -м из участков установлено ограничение по скорости в v_i м/с.

За превышение скорости предусмотрены штрафы. В зависимости от превышения, установлены различные штрафы, величина штрафа вычисляется следующим образом.

Пусть e — максимальное превышение разрешённой скорости в процессе пребывания автомобиля на всей дороге, то есть максимальная разница между скоростью автомобиля и максимальной разрешённой скоростью на участке, где он в этот момент находится. Если превышения скорости не было, то штраф не взимается. В противном случае штраф вычисляется так:

- если $0 < e \leq a_1$, то штраф составляет f_1 денежных единиц;
- если $a_1 < e \leq a_2$, то штраф составляет f_2 денежных единиц;
- ...
- если $a_{m-2} < e \leq a_{m-1}$, то штраф составляет f_{m-1} денежных единиц;
- если $a_{m-1} < e$, то штраф составляет f_m денежных единиц.

Таким образом, есть m диапазонов превышения скорости и соответствующие им штрафы.

Автоматическая система назначения штрафов получила данные о q автомобилях. Для удобства пронумеруем их от 1 до q . Известно, что i -й автомобиль заехал на дорогу в момент времени s_i , проехал все n участков, после чего выехал с нее в момент времени t_i . Отсчёт времени будем вести в секундах с открытия дороги.

Для каждого из автомобилей система должна определить, какой максимальный штраф можно гарантированно выписать этому автомобилю, основываясь только на времени заезда на дорогу и выезда с нее.

Требуется написать программу, которая по описанию границ диапазонов превышения скорости, соответствующих штрафов и временам въезда/выезда автомобилей определяет для каждого автомобиля максимальный штраф, который можно выписать этому автомобилю.

Формат входных данных

Первая строка входных данных содержит единственное целое число n — количество участков на дороге ($1 \leq n \leq 10$).

Вторая строка содержит n целых чисел v_i — ограничения скорости на участках ($1 \leq v_i \leq 10^9$).

Третья строка содержит n целых чисел l_i — длины участков ($1 \leq l_i \leq 10^9$).

Четвертая строка содержит единственное целое число m — количество границ диапазонов превышения скорости ($1 \leq m \leq 10^5$).

Пятая строка содержит $m - 1$ целых чисел a_i — границы диапазонов превышения скорости ($1 \leq a_i \leq 10^9$). Гарантируется, что значения a_i строго возрастают. Обратите внимание, что если $m = 1$, то пятая строка ввода пустая.

Шестая строка содержит m целых чисел f_i — штрафы за диапазоны превышения скоростей ($1 \leq f_i \leq 10^9$). Гарантируется, что значения f_i возрастают.

Седьмая строка содержит единственное целое число q — количество автомобилей, которые надо обработать ($1 \leq q \leq 10^5$).

Каждая из следующих q строк содержит два целых числа s_i и t_i — время заезда на трассу и выезда с неё i -го из рассматриваемых автомобилей ($1 \leq s_i < t_i \leq 10^9$).

Формат выходных данных

Для каждого из q автомобилей выведите в отдельной строке максимальный штраф, который гарантированно можно выписать этому автомобилю, основываясь только на временах его заезда на дорогу и выезда с нее. Если возможна ситуация, что автомобиль ни разу не превысил разрешённую скорость, следует вывести 0.

Гарантируется, что если время въезда или выезда автомобиля изменить не более чем на 10^{-5} , штраф, который можно ему выписать, не изменится.

Система оценивания

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	5	$n = 1, m = 1$		первая ошибка
2	10	$m = 1$	1	первая ошибка
3	9	$n = 1, m \leq 10$	1	первая ошибка
4	12	$n = 1$	1, 3	первая ошибка
5	13	$m \leq 10, a_i \leq 10$		первая ошибка
6	14	$m \leq 10$	1, 2, 3, 5	первая ошибка
7	37		1 – 6	первая ошибка

Пример

стандартный ввод	стандартный вывод
3	0
10 20 30	800
400 500 600	600
6	
1 5 10 12 16	
100 300 600 800 1000 1500	
3	
10 100	
20 70	
45 100	