

Задача А. Укладка асфальта

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

До следующей юбилейной межгалактической олимпиады по программированию осталось совсем мало времени, а подготовка к ней только началась. В результате жеребьёвки было решено, что олимпиада пройдёт на планете Земля.

Объединённое правительство Земли решило построить для проведения олимпиады инновационный город, чтобы показать всю мощь и технологическое развитие планеты. Конечно, в городе нужно будет построить много объектов для проведения соревнований: отели, клубы, магазины и много других зданий. Но в то же время никакой город не может обойтись без дорог!

Вам, как директору фирмы «Дороги Земли», достались планы нового города с указанием разработать дорожную инфраструктуру. Город имеет форму квадрата $N \times N$ с $N + 1$ прямолинейной улицей, идущей с севера на юг, и $N + 1$ прямолинейной улицей, идущей с запада на восток. Расстояние между двумя соседними параллельными улицами равно K метров. На пересечении любых двух дорог будет организован перекрёсток.

Правительство Земли поручило вам организовать укладку асфальта на всех дорогах города, а вы, конечно, хотите сэкономить и потратить как можно меньше асфальта. Но не всё так просто! Вы должны уложить асфальт на некоторые части дорог таким образом, чтобы с любого перекрёстка можно было добраться до любого другого перекрёстка, перемещаясь только по заасфальтированным частям дорог. Если это условие не будет выполнено, вы лишитесь работы.

Посчитайте, какую минимальную суммарную длину дорог вам придётся заасфальтировать.

Формат входных данных

В единственной строке через пробел записаны два числа N и K ($1 \leq N, K \leq 10^6$).

Формат выходных данных

Выведите одно число — минимальную суммарную длину заасфальтированных частей дорог.

Система оценки

Подзадача	Баллы	Дополнительные ограничения	Оценка	Необходимые подзадачи
0	0	Тесты из условия	потестовая	—
1	40	$1 \leq N, K \leq 100$	подзадача	—
2	60	—	подзадача	1

Получение информации о результатах проверки

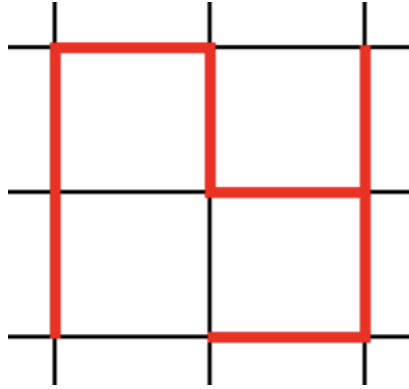
Для каждой подзадачи сообщаются набранные баллы, а также результат тестирования на первом непройденном тесте.

Примеры

стандартный ввод	стандартный вывод
2 4	32
1 1	3

Замечание

Ниже приведён один из способов покрыть минимальную суммарную длину дорог асфальтом. Черными линиями обозначены дороги. Красным обозначены участки, на которых нужно уложить асфальт.



Задача В. Очередь в магазине

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

В одном известном магазине случилась распродажа, однако администрация не учла одну проблему: в магазине всего одна касса! Сразу после начала распродажи возле кассы организовалась длинная очередь. Никто не любит очереди, поэтому у покупателей постепенно возрастает уровень агрессии. От вас требуется рассмотреть процесс продвижения очереди.

Могут происходить события трёх типов:

1. В конец очереди встал человек с уровнем агрессии a ;
2. Первый человек в очереди начал ругаться с кассиром, в результате чего уровень его агрессии увеличился на x , а уровень агрессии каждого из **остальных** людей в очереди (если в очереди стоит не один человек) увеличился на y ;
3. Первый человек в очереди оплатил покупку и ушёл из магазина.

От вас требуется обработать N событий. Будем считать, что изначально очередь пуста. Так как администрация магазина заботится о своей репутации, им важно знать, насколько агрессивными их покупатели уходят из магазина. Поэтому для каждого события третьего типа нужно определить уровень агрессии человека, который ушёл из магазина.

Формат входных данных

В первой строке записано одно число N — количество событий ($2 \leq N \leq 300000$).

В каждой из следующих N строк содержится описание очередного события:

- 1 a , если произошло событие первого типа;
- 2 x y , если произошло событие второго типа;
- 3, если произошло событие третьего типа.

Для всех событий верно, что $1 \leq a, x, y \leq 10^9$. Гарантируется, что события второго и третьего типов происходят только в том случае, если в очереди есть хотя бы один человек. Также гарантируется, что после N событий в очереди не останется ни одного человека. Возможны случаи, когда первый человек в очереди несколько раз подряд ссорится с кассиром.

Формат выходных данных

Для каждого запроса третьего типа выведите одно число — уровень агрессии человека, который ушёл из магазина. Каждое число следует выводить на отдельной строке.

Система оценки

Подзадача	Баллы	Дополнительные ограничения	Оценка	Необходимые подзадачи
0	0	Тесты из условия	потестовая	—
1	40	$2 \leq N \leq 1000$ Для всех событий $1 \leq a, x, y \leq 1000$	подзадача	—
2	60	—	подзадача	1

Получение информации о результатах проверки

Для каждой подзадачи сообщаются набранные баллы, а также результат тестирования на первом непройденном тесте.

Пример

стандартный ввод	стандартный вывод
8	10
1 4	13
1 2	1
2 6 1	
3	
2 10 20	
1 1	
3	
3	

Замечание

Сначала в очередь встали два человека с уровнями агрессии 4 и 2 соответственно. Затем первый человек поссорился с кассиром, после чего уровни агрессии людей стали равны 10 и 3. После этого первый человек ушёл из очереди, а второй поссорился с кассиром. Теперь уровень его агрессии равен 13. Затем в очередь встал человек с уровнем агрессии 1, после чего оба человека ушли из магазина.

Задача С. Драгоценный камень

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

Давным-давно, на просторах Эллады (или, как мы сказали бы сегодня, Древней Греции) жил торговец Николаус. Однажды ему удалось приобрести один весьма редкий драгоценный камень турмалин. Однако в ту пору познания людей в геологии были невелики, а потому определить ценность этого камня было проблематично. Николаус, как любой предприимчивый торговец, решил на этом заработать.

Ценность турмалина может существенно меняться изо дня в день. Для воплощения своего плана, Николаус отправился на гору Парнас, к оракулу, способному предсказывать будущее. Оракул сообщил Николаусу, какая будет цена у турмалина в следующие N дней: в день i его стоимость составит a_i драхм. Тогда Николаус решил, что он выберет несколько дней i_1, i_2, \dots, i_k ($1 \leq i_1 < i_2 < \dots < i_k \leq N$), и в первый выбранный день продаст турмалин за a_{i_1} драхм, затем в другой день купит его обратно за a_{i_2} драхм, и так далее. Что произойдет с турмалином после N дней Николаусу не известно, а потому в последний выбранный день он обязательно хочет продать свой драгоценный камень за a_{i_k} драхм, то есть выбранное k должно быть нечетным.

Николаус хочет максимизировать полученную им прибыль $a_{i_1} - a_{i_2} + a_{i_3} - a_{i_4} + \dots + a_{i_k}$. Однако познания древних греков в математике также были невелики, и вам предстоит ответить на вопрос: какую максимальную прибыль может получить Николаус?

Формат входных данных

В первой строке задано количество дней N ($1 \leq N \leq 200000$).

Во второй строке содержатся N целых чисел a_i ($1 \leq a_i \leq 10^9$) — стоимости турмалина, предсказанные оракулом.

Формат выходных данных

Выведите одно число — наибольшую прибыль, которую может получить Николаус.

Система оценки

Подзадача	Баллы	Дополнительные ограничения	Оценка	Необходимые подзадачи
0	0	Тесты из условия	потестовая	—
1	20	$1 \leq N \leq 10, 1 \leq a_i \leq 1000$	подзадача	—
2	30	$1 \leq N \leq 1000, 1 \leq a_i \leq 1000$	подзадача	1
3	50	—	подзадача	1-2

Получение информации о результатах проверки

Для каждой подзадачи сообщаются набранные баллы, а также результат тестирования на первом непройденном тесте.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3	3
4 4 1 2 5	8

Замечание

В первом примере наиболее выгодным является продать турмалин в третий день и получить 3 драхмы.

Во втором примере наиболее выгодным является продать турмалин в первый день за 4 драхмы, затем купить его во второй день за 1 драхму, а затем снова продать в четвертый день за 5 драхм. Полученная прибыль равна: $4 - 1 + 5 = 8$.

Задача D. Приближение прогрессией

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

Представьте, что вы стали призером очередной очень известной олимпиады по программированию, и теперь едете домой. Вдруг вам стало скучно, а в голову пришла отличная мысль — заняться решением задач по программированию. Вы открыли один очень известный сайт, на котором размещены 1136 задач различного уровня сложности. Так как вы являетесь призёром многих олимпиад по программированию, вы не побоялись трудностей, выбрали случайную задачу и начали думать над ее решением. Написав решение, вы получили вердикт «Неправильный ответ на тесте 2» и начали искать ошибку.

Для того, чтобы проверить корректность решения, вы перебрали всевозможные ответы на поставленную задачу. Теперь для каждого ответа вам нужно понять, является ли он правильным. Для этого вам предстоит решить не менее интересную задачу...

Вам дана последовательность чисел a_1, a_2, \dots, a_n , а также арифметическая прогрессия b_1, b_2, \dots, b_n , где $b_k = b + (k - 1) \cdot d$ для всех k от 1 до n . Таким образом арифметическая прогрессия задана при помощи двух чисел: b и d . Перед вами стоит довольно простая задача — найти **отклонение** последовательности от прогрессии. Отклонение определяется следующим образом: $(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2$.

Ваша задача — посчитать отклонения для нескольких арифметических прогрессий.

Формат входных данных

В первой строке записано число n ($1 \leq n \leq 100000$).

Во второй строке через пробел записаны n целых чисел a_i ($-10^5 \leq a_i \leq 10^5$).

В третьей строке записано число m — количество арифметических прогрессий, для которых нужно вычислить отклонение от последовательности a_i ($1 \leq m \leq 300000$).

В следующих m строках через пробел записаны пары целых чисел b, d , которые задают очередную арифметическую прогрессию ($-10^5 \leq b, d \leq 10^5$). Гарантируется, что $-10^5 \leq b_i \leq 10^5$.

Формат выходных данных

Выведите m целых чисел — отклонение каждой арифметической прогрессии от последовательности a_1, a_2, \dots, a_n . Каждое число нужно вывести на отдельной строке.

Система оценки

Подзадача	Баллы	Дополнительные ограничения	Оценка	Необходимые подзадачи
0	0	Тесты из условия	потестовая	—
1	30	$1 \leq n, m \leq 1000$ $-100 \leq a_i, b, d \leq 100$ Гарантируется, что $-500 \leq b_i \leq 500$	подзадача	—
2	10	У всех прогрессий $d = 0$, также $a_1 = a_2 = \dots = a_n$	подзадача	—
3	10	У всех прогрессий $d = 0$	подзадача	2
4	10	$a_1 = a_2 = \dots = a_n$	подзадача	2
5	40	—	подзадача	1, 2, 3, 4

Получение информации о результатах проверки

Для каждой подзадачи сообщаются набранные баллы, а также результат тестирования на первом непройденном тесте.

Пример

стандартный ввод	стандартный вывод
5	33
1 4 2 6 8	12
3	146
4 0	
1 2	
-3 1	

Замечание

Первая арифметическая прогрессия: $b_1 = 4, b_2 = 4, b_3 = 4, b_4 = 4, b_5 = 4$.

Отклонение равно: $(1 - 4)^2 + (4 - 4)^2 + (2 - 4)^2 + (6 - 4)^2 + (8 - 4)^2 = 9 + 0 + 4 + 4 + 16 = 33$.

Вторая арифметическая прогрессия: $b_1 = 1, b_2 = 3, b_3 = 5, b_4 = 7, b_5 = 9$.

Отклонение равно: $(1 - 1)^2 + (4 - 3)^2 + (2 - 5)^2 + (6 - 7)^2 + (8 - 9)^2 = 0 + 1 + 9 + 1 + 1 = 12$.

Задача Е. Последний рубеж

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

Это интерактивная задача.

Однажды общительный программист Павел позвал своих друзей на квест. Ребята с лёгкостью решали головоломки и продвигались вперёд. И вот им осталось решить последнюю загадку перед тем, как получить долгожданный приз.

Загадка состоит в том, что перед ребятами находится дверь с N замками, которую нужно открыть. Некоторые замки открыты, а некоторые закрыты. Ребята не знают, какие замки уже открыты, однако, потратив некоторое время на изучение одного конкретного замка, они могут определить, открыт он или нет. Рядом с дверью висит табличка, на которой написано, что самый левый замок открыт, а самый правый закрыт.

В процессе выполнения предыдущих заданий ребята выяснили, как открыть дверь. Пронумеруем замки слева направо числами от 1 до N . Тогда для того, чтобы открыть дверь, ребятам нужно найти замок с номером $i < N$, такой что замок i открыт, а замок $i + 1$ закрыт.

Как уже было сказано, для того, чтобы определить, является ли i -й замок открытым, им нужно подробно его осмотреть, потратив на это некоторое время. Так как у ребят осталось немного времени для выполнения последнего задания, они могут подробно осмотреть не более Q замков.

Помогите ребятам открыть дверь.

Протокол взаимодействия

В начале на вход программе подаётся одно целое число N ($2 \leq N \leq 10^{18}$).

После этого вы можете делать запросы вида $? i$, означающие, что ребята подробно осматривают замок с номером i . В ответ на подобный запрос вы получите число 0, означающее, что замок закрыт, или число 1 в противном случае.

Если вы нашли ответ, вы должны сделать запрос вида $! i$, означающий, что вы считаете, что замок i открыт, а замок $i + 1$ закрыт. После этого запроса вы должны завершить работу программы.

Считается, что в начале вы знаете состояние замков 1 и N .

Вы должны сделать не более Q запросов первого типа. В случае превышения этого ограничения ваше решение получит вердикт «Неправильный ответ».

В случае нарушения каких-либо правил взаимодействия с программой-интерактором, ваше решение может получить любой вердикт.

После каждого запроса, в том числе после запроса второго типа, вы должны выполнить операцию `flush`.

Для сброса буфера вывода (то есть для операции `'flush'`) сразу после вывода запроса и перевода строки нужно сделать:

- `fflush(stdout)` или `cout.flush()` в языке C++;
- `System.out.flush()` в Java;
- `stdout.flush()` в Python;
- `flush(output)` в Pascal;
- смотрите документацию для других языков.

Если вы не сделаете операцию `flush` после какого-либо запроса, ваше решение может получить любой вердикт.

Система оценки

Подзадача	Баллы	Дополнительные ограничения	Оценка	Необходимые подзадачи
0	0	Тесты из условия	потестовая	—
1	20	$Q = 100$ $2 \leq N \leq 100$	подзадача	—
2	10	$Q = 100$ $2 \leq N \leq 10^5$ В начале закрыты ровно два замка	подзадача	—
3	30	$Q = 90$ $2 \leq N \leq 10^9$	подзадача	1, 2
4	20	$Q = 65$ $2 \leq N \leq 10^{18}$	подзадача	1, 2, 3
5	20	$Q = 60$ $2 \leq N \leq 10^{18}$	подзадача	1, 2, 3, 4

Получение информации о результатах проверки

Для каждой подзадачи сообщаются набранные баллы, а также результат тестирования на первом непройденном тесте.

Примеры

стандартный ввод	стандартный вывод
3 0	? 2 ! 1
3 1	? 2 ! 2
5 0 0 1	? 2 ? ? 3 ? ? 4 ! ! 4