

Задача А. Нестандартный подход

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Избранный, призванный восстановить Пламя и завершить этот цикл возжигания и угасания, должен на своем пути сразиться с множеством могущественных врагов.

Несмотря на известный факт, что все бои с боссами очень сложны, в некоторых случаях можно выбить и более легкую победу, если применить нестандартную тактику. Так, дракон Мидир Пожиратель Тьмы может быть намного проще побежден с помощью заклинания «Ядовитый туман», которое позволит заметно сократить длительность боя, нанося дракону постоянный урон.

Представим поле сражения с драконом в виде прямоугольной сетки размера $n \times m$ (n строк и m столбцов) и обозначим за (i, j) клетку поля на пересечении i -й слева строки и j -го сверху столбца. Также назовем *соседними* клетки поля, имеющие общую сторону.

Определим *расстояние* между двумя клетками A и B как минимальное такое k , что существует последовательность клеток $C_0, C_1, \dots, C_k$, в которой $C_0 = A$, $C_k = B$ и любые два C_i и C_{i+1} — соседние. Так, расстояние между $(1, 2)$ и $(4, 1)$ равно 4, так как из одной в другую можно попасть за четыре перемещения: $(1, 2) \rightarrow (2, 2) \rightarrow (3, 2) \rightarrow (3, 1) \rightarrow (4, 1)$.

Перед началом боя, то есть пока вы еще не оказались на поле, вы можете подойти к любой клетке A на его границе (то есть к клетке, хотя бы одна из сторон которой находится на границе поля) и создать на ней ядовитый туман. Туман распространяется из каждой клетки во все соседние с ней за одну секунду, пока не заполнит все поле. Таким образом, время, за которое туман распространится до произвольной клетки B , равно расстоянию между A и B .

Вы не хотите долго ходить вокруг поля и ждать начала боя, поэтому вам необходимо определить координаты клетки, в которой следует создать ядовитый туман, чтобы он заполнил все поле за минимальное время.

Формат входных данных

В единственной строке ввода даны два целых числа n и m — число строк и столбцов в сетке поля боя ($1 \leq n, m \leq 10^{18}$).

Формат выходных данных

Выведите два целых числа i и j ($1 \leq i \leq n$; $1 \leq j \leq m$) — координаты клетки, которую следует выбрать, чтобы заполнить поле ядовитым туманом за минимальное время. Если таких клеток несколько, выведите любую из них.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Обратите внимание, что прохождение тестов из условия не является необходимым для тестирования на некоторых группах тестов.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
0	—	примеры из условия		полная
1	11	$n = 1$		первая ошибка
2	16	$n = m$		первая ошибка
3	18	$n, m \leq 40$	0	первая ошибка
4	15	$n, m \leq 100$	0, 3	первая ошибка
5	15	$n, m \leq 1000$	0, 3, 4	первая ошибка
6	25	нет	0 – 5	первая ошибка

Примеры

стандартный ввод	стандартный вывод
2 5	1 3
4 31	1 16

Задача В. Спасти Лютика!

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Лютик скрывается от четырех преступных группировок Новиграда через подземелье из n расположенных подряд холлов, разделенных дверьми. Холлы пронумерованы от 1 до n в порядке от выхода ко входу, то есть вход в подземелье ведет в n -й холл, а выйти можно только из первого. При этом в каждом из помещений одновременно может находиться не более t живых бандитов, но изначально подземелье пустое.

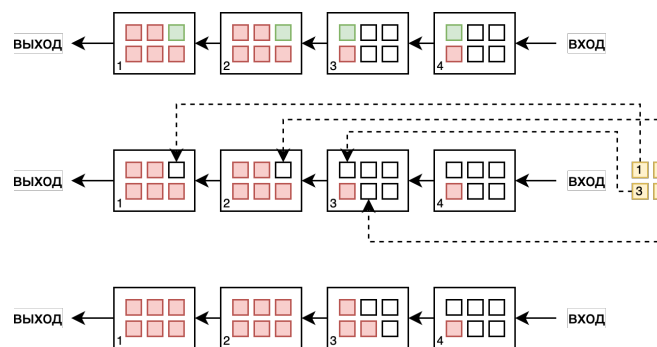
Сейчас Лютик уже успел выйти из подземелья и ровно через s минут он окажется в безопасности на отбывающем из города корабле. А в это время Геральт перекрыл выход из подземелья, и запустился квест, ваша цель в котором — задержать прибывающих в подземелье бандитов.

Обозначим за x последовательность из n чисел от 0 до t — число бандитов в каждом холле в данный момент. Тогда во время квеста каждую минуту происходят два события именно в таком порядке:

1. С вашим текущим уровнем персонажа вы успеваете сразиться по e бандитов в каждом из n холлов. То есть для каждого i от 1 до n в i -м холле число бандитов становится равным $\max(0, x_i - e)$.
2. Затем ко входу в подземелье прибывают ровно k новых бандитов. Они по очереди заходят в подземелье и перемещаются в ближайший к выходу еще не заполненный холл.

Иными словами, каждый из новых k бандитов по очереди находит минимальное i , для которого $x_i < t$, и перемещается в i -й холл, после чего x_i увеличивается на 1. Если для очередного бандита не находится места в подземелье, он разворачивается и уходит.

Обратите внимание, что уже находящиеся в подземелье бандиты не перемещаются. Например, пусть $n = 4$, $t = 6$, $e = 1$ и $k = 4$. Если сейчас число бандитов в холлах равно $x = (6, 6, 2, 2)$, то сначала в каждом холле станет на одного бандита меньше (см. зеленые квадраты на рисунке ниже), а затем появятся $k = 4$ новых бандита (желтые квадраты). Первые два из них пойдут в первый и второй холл, а оставшиеся двое займут третий. После чего x станет равно $(6, 6, 3, 1)$. Бандит, уже находившийся в четвертом холле, останется в нем, несмотря на наличие места в третьем.



По окончании квеста вы хотели бы успеть сыграть пару партий в Гвинт, поэтому скрываться из подземелья надо быстро. Для этого определите, в каком холле по истечении s минут будет меньше всего бандитов, и сколько их будет. В случае, если таких холлов несколько, предпочтительнее будет холл ближе к выходу.

Обратите внимание, что необходимо рассматривать состояние подземелья после того, как на s -й минуте появились еще k бандитов.

Формат входных данных

В первой строке ввода даны три целых числа n , t и s — количество холлов в подземелье, вместимость каждого из них и длительность квеста в минутах ($1 \leq n, t \leq 10^9$; $1 \leq s \leq 10^5$).

Во второй строке даны два целых числа e и k — количество побеждаемых Геральтом за минуту бандитов в каждом холле и количество прибывающих с каждой минутой новых бандитов ($0 \leq k \leq 10^9$, $0 \leq e \leq m$).

Формат выходных данных

Выведите через пробел два целых числа — сначала номер холла, в котором будет наименьшее число бандитов через s минут, а затем само число бандитов в нем. Если под критерий подходит несколько холлов, выведите наименьший номер такого холла.

Система оценки

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

Подзадача	Баллы	Доп. ограничения	Необходимые подзадачи	Информация о проверке
0	—	примеры из условия		полная
1	8	$n, m, s, k, e \leq 10$	0	полная
2	5	$e = 0, k = 1$	0	первая ошибка
3	7	$m = 1$	0	первая ошибка
4	10	$m \leq e \leq k$	0	первая ошибка
5	10	$n, m, s \leq 100$	0, 1	первая ошибка
6	12	$n, s \leq 1000$	0, 1, 5	первая ошибка
7	8	$e \leq k \leq m$	0, 4	первая ошибка
8	20	$n \leq 10^5$	0, 1, 5, 6	первая ошибка
9	20	нет	0 – 8	первая ошибка

Примеры

стандартный ввод	стандартный вывод
5 4 2 1 3	3 0
2 10 7 3 5	2 2
1 4 2 1 2	1 3

Замечание

В первом примере из условия необходимо понять, что будет с подземельем через две минуты.

- После первой минуты стадия сражения будет пропущена (бандитов еще нет). Затем в первом холле появятся три бандита.
- После второй минуты в каждом холле Геральт победит по одному бандиту, после чего x_1 станет равно 2. Далее в подземелье зайдут три бандита, и им не хватит места в первом холле, поэтому один из них расположится во втором: $x_1 = 4$, $x_2 = 1$.

При $x = (4, 1, 0, 0, 0)$ меньше всего бандитов в холлах 3, 4 и 5 (по 0), поэтому в качестве ответа выбирается третий холл.

Задача С. Полное прохождение

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Изначально в игре было шесть возможных концовок: Эпоха Разлома, Эпоха Порядка, Эпоха Сумрака, Эпоха Отчаяния, Владыка Яростного Пламени и Эра Звезд. А с выходом множества модов и DLC число возможных концовок вообще стало равным n . Многие говорят, что Эра Звезд — лучшая из них, но зачем останавливаться на чем-то одном, если можно пройти все?

На получение i -й концовки потребуется ровно d_i времени для всех i от 1 до n . Причем $d_1 = 0$ — чтобы познакомиться с игрой, первую и стандартную концовку вы уже получили, победив Радана и восстановив Кольцо Элдена.

Оставшиеся концовки вы намерены получить строго по порядку, то есть сначала у вас уйдет d_2 времени на получение второй концовки, затем d_3 времени на третью концовку, и так далее до d_n времени на последнюю концовку.

Помимо прохождения вы хотите после каждой концовки посмотреть стрим с разбором ее сюжета на своем любимом канале. Пусть сейчас момент времени 0, тогда разбор i -й концовки случится строго в момент времени a_i (то есть через a_i времени от текущего момента). Если вы его пропустите, то уже не сможете его посмотреть потом. Также после получения i -й концовки вы уже не станете смотреть разборы предыдущих концовок.

Отметим, что время на просмотр видео-разбора сюжета бесконечно мало по сравнению с затратами времени на прохождение, поэтому будем считать время просмотра видео нулевым и не будем его учитывать. Таким образом, для каждого i от 1 до n после получения i -й концовки у вас есть следующие варианты действий:

- Вы можете моментально перейти к $(i + 1)$ -му прохождению игры. В этом случае вы точно не посмотрите разбор i -й концовки, но завершите $(i + 1)$ -е прохождение ровно через d_{i+1} времени;
- Если с момента начала прошло не больше a_i времени, вы можете сделать паузу в прохождении, дождаться момента времени a_i и посмотреть стрим. Время просмотра стрима бесконечно мало, поэтому вы закончите просмотр также в момент времени a_i . После чего вы сразу начнете следующее прохождение, которое завершится ровно в момент времени $a_i + d_{i+1}$.

Определите, какое максимальное количество стримов по сюжетам концовок вы сможете посмотреть при таком прохождении.

Формат входных данных

Первая строка теста содержит одно целое число n — количество концовок игры ($2 \leq n \leq 2 \cdot 10^5$).

Во второй строке перечислены n целых чисел a_i — в какой момент времени случится каждый стрим с разбором концовки ($1 \leq a_i \leq 10^9$).

В третьей строке перечислены $n - 1$ целых чисел $d_2, d_3, \dots, d_n$ — время на получение каждой концовки со 2-й по n -ю ($1 \leq d_i \leq 10^4$).

Формат выходных данных

Выведите одно целое число — максимальное количество стримов с разборами сюжета, на которые вы сможете попасть.

Система оценки

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

Подзадача	Баллы	Доп. ограничения	Необходимые подзадачи	Информация о проверке
0	–	примеры из условия		полная
1	12	$n \leq 10$		полная
2	17	$d_i = 1, 0 \leq a_{i+1} - a_i \leq 1$	–	первая ошибка
3	29	$n \leq 1000$	1	первая ошибка
4	42	нет	1 – 3	первая ошибка

Примеры

стандартный ввод	стандартный вывод
3 3 4 6 2 2	2
5 1 1 2 3 4 1 1 1 1	4
6 1 3 6 8 14 15 3 2 1 7 1	4

Замечание

В первом примере имеет смысл пропустить разбор первой концовки, подождать два часа после прохождения второй, и затем посмотреть разбор второй концовки, получить третью и тут же пойти смотреть стрим по ней.

Во втором примере снова надо пропустить разбор первой концовки, после чего без перерывов чередовать прохождения и стримы.

В третьем примере подойдет способ, в котором вы смотрите разборы первой, третьей, пятой и шестой концовок.

Задача D. Не доверяйте свиткам

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Это задача с двойным запуском. Ваше решение будет запущено дважды.

За игровым столом четыре игрока — мастер игры, вы (паладин) и два ваших друга — волшебник и плут. Ваш отряд в данный момент проводит инвентаризацию; оказалось, что десятки ненужных вам подобранных по пути предметов снаряжения, которыми забиты ваши рюкзаки, мешает эффективно перемещаться по лесу.

В своем рюкзаке среди добытого лута волшебник нашел свиток с отчетностью о тратах отряда, которую он вел последние n дней (вы подозревали, что плут втайне кладет пару лишних монет в свой карман). Известно, что

- ваш отряд расходует золото очень ответственно, поэтому в каждый из n дней было потрачено одинаковое число монет;
- все траты разделены на три категории: еда, снаряжение и вечеринки в таверне.

Иными словами, на свитке были записаны три массива неотрицательных целых чисел a (траты на еду), b (траты на снаряжение) и c (траты на вечеринки), каждый длины n , и известно, что $a_i + b_i + c_i = a_j + b_j + c_j$ для всех i и j .

Внимательно посмотрев на свиток, волшебник заметил, что с ним что-то не так: массивы были разных размеров, и условие на сохранение суммы соответствующих элементов трех массивов перестало выполняться. Произнеся заклинание «Обнаружение магии» и бросив кубик, волшебник установил, что это «Проклятый свиток сжатия», который удаляет из массивов подряд идущие одинаковые элементы. Например, если на свитке были записаны $a = [2, 2, 2, 1, 5]$, $b = [3, 3, 1, 2, 3]$ и $c = [4, 4, 6, 6, 1]$, то спустя некоторое время они превратятся в $a' = [2, 1, 5]$, $b' = [3, 1, 2, 3]$ и $c' = [4, 6, 1]$.

Ваш персонаж волшебнику тоже не до конца доверяет. Поэтому у вас в кармане случайно нашлась записка с сокращенной информацией о тратах S , состоящей из не более k целых чисел от 0 до $2^{30} - 1$. На удивление, этой информации хватило, чтобы восстановить как могли выглядеть исходные a , b и c .

Какую информацию S следует запомнить, чтобы по n , a' , b' и c' можно было восстановить какие-то a , b и c , удовлетворяющие условию?

Формат входных данных

В первой строке ввода дано единственное целое число 1 или 2 — номер запуска.

Первый запуск

Во второй строке ввода даны два целых числа n и k — число дней в отчетности и ограничение на длину S ($1 \leq n \leq 30\,000$; $1 \leq k \leq 60\,000$).

В третьей строке перечислены n целых чисел a_i — траты отряда на еду в каждый из дней ($0 \leq a_i < 2^{20}$). В четвертой и пятой строках в том же формате перечислены b_i и c_i — траты на снаряжение и на вечеринки ($0 \leq b_i, c_i < 2^{20}$). Гарантируется, что $a_i + b_i + c_i$ — постоянная величина.

Второй запуск

Во второй строке ввода даны пять целых чисел n , m_s , $m_{a'}$, $m_{b'}$ и $m_{c'}$ — изначальное n из первого запуска, длина выведенной вашим решением при первом запуске последовательности S , а также длины a' , b' и c' ($1 \leq m_{a'}, m_{b'}, m_{c'} \leq n \leq 30\,000$; $1 \leq m_s \leq k$).

В следующих четырех строках перечислены элементы S , a' , b' и c' соответственно. Гарантируется, что S получено при первом запуске вашего решения, а a' , b' и c' получены из a , b и c для первого запуска удалением повторений подряд идущих элементов.

Формат выходных данных

Первый запуск

В первой строке выведите целое число m_s от 1 до k — длину последовательности S , которую вы запишете и уберете в карман.

Во второй строке через пробел выведите m_s целых чисел от 0 до $2^{30} - 1$ — элементы S .

Второй запуск

Выведите три строки, в каждой по n чисел от 0 до $2^{20} - 1$ — элементы массивов a , b и c соответственно.

Если существует несколько возможных подходящих ответов, выведите любой из них — не обязательно восстанавливать в точности те a , b и c , которые были даны во вводе при первом запуске.

Протокол взаимодействия

Во избежание получения некорректных вердиктов вроде `Idleness Limit Exceeded` или `Security Violation` заканчивайте вывод каждой строки символом перевода строки (`'\n'`) и сбросом буфера вывода (`cout.flush()`, `sys.stdout.flush()`, `System.out.flush()`).

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Обратите внимание, что прохождение тестов из условия не является необходимым для тестирования на некоторых группах тестов.

В последней подзадаче 10 тестов, каждый из которых независимо оценивается от 0 до 2 баллов. Пусть решение жюри на тесте выводит S длины j . Тогда ваше решение получит

- 2 балла, если выведет S не длиннее, чем j ;
- 1 балл, если выведет S длины от $j + 1$ до $j + 30$;
- 0 баллов, если выведет S длины хотя бы $j + 31$.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
0	—	примеры из условия		полная
1	4	$n \leq 3, k = 1$		первая ошибка
2	5	$n \leq 10, k = 1$	1	первая ошибка
3	8	$n \leq 100, k = 4$	1, 2	первая ошибка
4	9	$n \leq 1000, k = 1000$	0	первая ошибка
5	4	$k = 60\,000$	0	первая ошибка
6	5	$k = 40\,000$	0, 5	первая ошибка
7	12	$k = 30\,000$	0, 5, 6	первая ошибка
8	13	$k = 15\,000$	0, 5 – 7	первая ошибка
9	7	$a_i = 0$ для всех $i, k = 1$		первая ошибка
10	13	$k = 3\,000$	0, 5 – 8	первая ошибка
11	10×2	$k = 1\,900$	0 – 8, 10	полная

стандартный ввод	стандартный вывод
1 5 15 2 2 2 1 5 3 3 1 2 3 4 4 6 6 1 2 5 3 3 4 3 3 2 4 2 1 5 3 1 2 3 4 6 1	3 3 2 4 2 2 1 5 5 3 1 2 3 3 4 6 6 1 1
1 8 24 4 5 7 5 7 9 7 7 3 3 2 4 4 3 5 6 7 6 5 5 3 2 2 1 2 8 1 7 6 6 11347 4 5 7 5 7 9 7 3 2 4 3 5 6 7 6 5 3 2 1	1 11347 4 5 7 5 7 9 7 7 3 3 2 4 4 3 5 6 7 6 5 5 3 2 2 1

Задача Е. Подъем на Высокий Хротгар

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Как и любой уважающий себя Драконорожденный, вы не можете себе позволить отправиться выполнять основную линию квестов. Поэтому к тому моменту, как вы решаете все-таки подняться на монастырь Высокий Хротгар на вершине горы высотой n метров, вы уже успели изучить несколько драконьих криков, собрать впечатляющий арсенал и набрать себе не менее впечатляющий набор побочных квестов.

Среди этих квестов также есть $m + q$ таких, которые вы можете выполнить по пути на Хротгар и обратно: для i -го из них вам необходимо подняться на высоту a_i метров от подножья горы, забрать у одного NPC какой-то предмет и доставить его другому NPC на высоте b_i от подножья горы. При этом

- на подъем на один метр вы тратите t_u времени;
- на спуск на один метр вы тратите t_d времени;
- на то, чтобы ~~преслужать~~ пропустить диалог, когда вы забираете или отдаете квестовый предмет, время не тратится.

Из этих $m + q$ квестов первые m вы точно намерены выполнить, а оставшиеся q пока что помечены неактивными. Перед началом подъема вы можете выбрать произвольное i от 0 до q и активировать первые i из оставшихся квестов в дополнение к m уже активным. Затем вы можете начать свой подъем.

В процессе подъема вы можете сколько угодно раз менять направление перемещения и держать в инвентаре произвольное число квестовых предметов одновременно. Как только все $m + i$ квестов будут выполнены, вы можете тут же остановиться, крикнуть *ドラゴン* в пролетающего мимо дракона и улететь на нем исследовать руины.

Обратите внимание, что **не требуется** в процессе выполнения квестов в итоге хотя бы раз побывать в самом Хротгаре на высоте n — мы понимаем, что при выполнении такого числа побочных квестов довольно легко снова забыть про основной сюжет. Также не требуется после выполнения квестов спуститься обратно на высоту 0.

Для каждого i от 0 до q выведите, какое минимальное время вам потребуется на выполнение m активных квестов и первых i из дополнительных квестов, если изначально вы находитесь у подножья горы на высоте 0.

Формат входных данных

В первой строке ввода дано целое число n — высота, на которой расположена вершина горы ($1 \leq n \leq 10^9$).

Во второй строке даны два целых числа t_u и t_d — время на подъем на один метр и время на спуск на один метр ($1 \leq t_u, t_d \leq 10^9$).

В третьей строке дано целое число m — количество активных квестов ($1 \leq m \leq 2 \cdot 10^5$). В i -й из следующих m строк даны два целых числа a_i и b_i — высоты стартовой и конечной точек i -го квеста ($0 \leq a_i, b_i \leq n$).

Затем дано целое число q — количество дополнительных квестов ($0 \leq q \leq 2 \cdot 10^5$), после которого следуют q строк с описаниями этих квестов в том же формате.

Формат выходных данных

В первой строке выведите минимальное время, необходимое для выполнения m активных квестов. Затем выведите еще q строк, в i -й из которых выведите ответ на задачу в случае, если перед подъемом активировать i первых квестов из q оставшихся.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Обратите внимание, что прохождение тестов из условия не является необходимым для тестирования на некоторых группах тестов.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
0	–	примеры из условия		полная
1	8	$q = 0, n, m \leq 100$		полная
2	10	$q = 0, n, m \leq 1000$	1	первая ошибка
3	12	$n, m, q \leq 1000$	1, 2	первая ошибка
4	5	$b_i > a_i$ для всех i		первая ошибка
5	11	$q = 0, n \leq 2 \cdot 10^5$	1, 2	первая ошибка
6	11	$q = 0$	1, 2, 5	первая ошибка
7	20	$n \leq 2 \cdot 10^5$	0 – 3, 5	первая ошибка
8	23	нет	0 – 7	первая ошибка

Пример

стандартный ввод	стандартный вывод
20 1 1 4 1 10 11 20 20 16 15 11 2 5 1 10 6	29 37 39

Задача F. Простая загадка

Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Это интерактивная задача.

Прагматичный и оптимистичный Гейл — хороший выбор игрового персонажа, если вы не против, чтобы половина вашего приключения превратилась в погоню за магическими артефактами. Недавно вы услышали в таверне легенду о заброшенном святилище старого культа, в котором может быть спрятан Разумный Амулет — артефакт, дарующий владельцу возможность использовать два уникальных заклинания. Разумеется, поиск Амулета сразу же стал следующим приоритетным квестом.

По заросшим тропам вы добрались до руин святилища. За тайным проходом, открывшимся после обезвреживания хитроумной ловушки, находился алтарь, где загадочный голос сообщил вам, что теперь вы должны доказать остроту ума, решив задачу, чтобы получить Амулет. А именно, вы должны отгадать два загаданных этим голосом числа l и r (известно, что $l \leq r$).

Поскольку угадывание случайных чисел без какой-либо дополнительной информации — это скорее проверка удачи, чем мудрости, голос разрешил вам сделать m запросов вида «заменить l и r на $l + d_l$ и $r + d_r$ », где d_l и d_r — произвольные выбранные вами числа. После каждого запроса голос будет сообщать вам количество простых чисел на отрезке от $\min(|l|, |r|)$ до $\max(|l|, |r|)$ включительно.

В любой момент времени вы можете назвать текущие значения l и r . Если вы угадали, Гейл получит Разумный Амулет. Иначе — путь к нему закроется для вас навсегда. Вашим спутникам, Астариону и Карлах, эта задача вряд ли по силам, поэтому придется вам самостоятельно помочь своему персонажу получить этот могущественный артефакт.

Формат входных данных

Каждый тест содержит несколько наборов входных данных. В единственной строке входных данных дано одно целое число t — количество наборов входных данных ($1 \leq t \leq 500$). Для каждого набора входных данных запускается процесс взаимодействия с интерактором.

Протокол взаимодействия

Взаимодействие с интерактором проходит в виде запросов со стороны вашей программы и ответов со стороны интерактора. Вы можете выполнить действие, описанное в условии, не более m раз, где m зависит от подзадачи.

Чтобы сделать запрос, выведите строку «? d_l d_r » ($-2 \cdot 10^6 \leq d_l, d_r \leq 2 \cdot 10^6$), после чего l и r изменятся на d_l и d_r соответственно. Интерактор в ответ выведет на отдельной строке количество простых чисел на отрезке $[\min(|l|, |r|), \max(|l|, |r|)]$ для новых значений l и r .

Чтобы вывести ответ на задачу, выведите строку «! l r », где в качестве l и r должны быть **текущие** загаданные границы. Этот вывод не учитывается в количестве запросов. В ответ интерактор выведет вердикт — 1, если ваше предположение верно, и 0 в противном случае. Если ваш ответ был неверен, ваше решение получит вердикт WA (Wrong Answer), а интерактор завершится. Во избежание получения некорректных вердиктов, считав информацию о том, что выведенный ответ неверен, ваше решение тоже должно завершиться.

Гарантируется, что исходные l и r удовлетворяют условию $1 \leq l \leq r \leq 10^6$.

Если в какой-то момент ваша программа превышает лимит в m запросов, или значения l или r после очередного запроса превысят $2 \cdot 10^6$ по модулю, ваша программа завершится с вердиктом WA.

Важно: не забывайте после каждой выведенной строки сбрасывать буфер вывода, чтобы интерактор получил ваш запрос. Это можно сделать с помощью `std::cout.flush()` в C++, `System.out.flush()` в Java и `sys.stdout.flush()` в Python, а также аналогичными командами в других языках. Если ваша программа не сбрасывает буфер вывода, она получит вердикт TL (Time Limit Exceeded) или IL (Idleness Limit Exceeded).

Система оценки

Баллы за подзадачи 1 – 11 начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

В последней подзадаче 8 тестов, каждый из которых оценивается независимо в 3 балла. Баллы за тест этой подзадачи начисляются только если на каждом из t наборов данных был дан верный ответ без превышения лимита запросов.

В таблице ниже за l и r обозначены изначальные неизвестные вам значения l и r , а m означает максимальное число разрешенных запросов.

Подзадача	Баллы	Доп. ограничения	m	Необходимые подзадачи	Информация о проверке
0	—	примеры из условия	500		полная
1	5	$l, r \leq 10$	150	0	первая ошибка
2	6	$l, r \leq 100$	150	0, 1	первая ошибка
3	6	$l, r \leq 1000$	150	0 – 2	первая ошибка
4	9	$l, r \leq 1000$	50	0 – 3	первая ошибка
5	7	$t \leq 3$	170	0	первая ошибка
6	7	нет	170	0, 5	первая ошибка
7	4	$t \leq 3$	90	0, 5	первая ошибка
8	5	нет	90	0, 5 – 7	первая ошибка
9	9	нет	60	0, 5 – 8	первая ошибка
10	9	нет	40	0, 5 – 9	первая ошибка
11	9	нет	30	0, 5 – 10	первая ошибка
12	8×3	нет	26	0 – 11	каждый тест

Пример

стандартный ввод	стандартный вывод
1	?
2	0 0
1	?
1	0 -1
2	?
1	1 1
2	?
1	2 5
2	!
1	5 8