

Задача А. Страшные числа

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Все мы, конечно же, очень любим числа, а также задачи на математику! Но, говорят, в Хэллоуин все задачи на математику становятся гораздо более страшными... Сегодня вашей задачей будет посчитать количество *страшных* чисел!

Число x называется *k-страшным*, если количество множителей в его разложении на простые числа равно k . Пока что, возможно, вам не кажется страшным такое определение, но вы еще не видели, в чем заключается вопрос задачи!

Требуется ответить на q запросов, каждый из которых описывается тремя целыми числами l , r и k . Ответом на запрос является количество *k-страшных* чисел, лежащих на отрезке от l до r . Если вас все еще не пугает эта задача, ответьте на все q запросов.

Формат входных данных

В первой строке ввода дано единственное целое число q — количество запросов ($1 \leq q \leq 10^5$).

В i -й из следующих q строк через пробел записаны три натуральных числа l_i , r_i и k_i — параметры запроса (границы отрезка и k из определения *k-страшного* числа) ($2 \leq l_i \leq r_i \leq 10^5$; $1 \leq k \leq 16$).

Формат выходных данных

Для i -го запроса выведите в отдельной строке количество *k_i-страшных* чисел на отрезке $[l_i, r_i]$.

Примеры

стандартный ввод	стандартный вывод
3	4
2 10 1	1
12 15 3	3
10 20 2	
5	4
21 40 1	0
46 65 9	17
50 100 2	12
100 150 3	7
150 200 4	

Задача В. Кошмар наяву

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

Многие люди на Хэллоуин балуются проведением разных магических ритуалов. Чаще всего, все эти ритуалы проводятся просто в шутку для развлечения, но вам сегодня попался настоящий магический артефакт, представляющий из себя коробочку с довольно интересного вида устройством на крышке.

Вы решили во что бы то ни стало открыть эту коробочку (хотя в фильмах это обычно плохо заканчивается). Устройство на крышке можно представить в виде плоскости, на которой расположены

- круг C с центром в точке (x_0, y_0) и радиусом r ;
- прямая L , заданная уравнением $ax + by + c = 0$;
- две отмеченные точки с координатами (x_1, y_1) и (x_2, y_2) .

Круг на крышке можно вращать относительно центра, в таком случае те из двух отмеченных точек, которые расположены на нем, вращаются вместе с ним.

Взяв коробочку в руки, вы моментально поняли, что чтобы ее открыть, требуется добиться параллельности прямой L и прямой, проходящей через две отмеченные точки. Можно ли повернуть круг так, чтобы эти прямые были параллельны, или же коробочка останется навсегда закрытой?

Формат входных данных

В первой строке через пробел перечислены три целых числа x_0 , y_0 и r — координаты центра круга и его радиус ($|x_0|, |y_0| \leq 10^4$; $0 \leq r \leq 10^4$).

Во второй строке через пробел перечислены три целых числа a , b , и c — коэффициенты уравнения прямой L ($|a|, |b|, |c| \leq 10^4$).

В третьей строке через пробел даны координаты первой отмеченной точки x_1 и y_1 ($|x_1|, |y_1| \leq 10^4$). В четвертой строке в том же формате даны координаты второй точки x_2 и y_2 .

Формат выходных данных

Выведите в единственной строке «YES» (без кавычек), если коробочку получится открыть, и «NO» иначе.

Примеры

стандартный ввод	стандартный вывод
0 0 6 1 1 1 0 0 1 1	YES
0 0 6 1 -1 1 0 0 1 1	YES
0 0 6 1 1 0 6 0 10 0	NO

Задача С. Давайте поделимся!

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

Как известно, во всех фильмах ужасов кому-то рано или поздно приходит в голову гениальная мысль разделить и исследовать страшное и подозрительное место небольшими группами.

В этот раз компания из $n + 2$ человек, исследуя заброшенную хижину в темном лесу, решила разделить на две группы. В компании есть два лидера, имеющих *степень безрассудства* a_1 и a_2 , соответственно. Также для исследования доступны два помещения, с *подозрительностью*, равной b_1 и b_2 , соответственно.

В каждой группе должен быть ровно один из двух лидеров, при чем если группа i -го лидера из k_i человек (**не считая лидера**) идет исследовать j -е помещение, *опасность* такого исследования равна $D = a_i \cdot k_i \cdot b_j$.

Разумеется, вы хотите им помочь минимизировать *опасность* такого сюжета, поэтому перед вами стоит задача разделить n человек, не являющихся лидерами, на две группы (в том числе одна группа может быть пустой), и назначить каждой группе своего лидера и свое помещение так, чтобы максимальная из двух *опасностей* была как можно меньше.

Формат входных данных

В первой строке ввода дано единственное целое число n — количество человек, не считая двух лидеров ($1 \leq n \leq 10^9$).

Во второй строке через пробел перечислены два целых числа a_1 и a_2 — степени безрассудства двух лидеров ($1 \leq a_1, a_2 \leq 10^4$).

В третьей строке так же даны целые числа b_1 и b_2 — подозрительности помещений ($1 \leq b_1, b_2 \leq 10^4$).

Формат выходных данных

В первой строке выведите единственное целое число — наименьшее возможное значение максимальной из опасностей для двух групп.

В следующей строке выведите через пробел номер помещения, в которое следует отправиться группе первого лидера, и количество людей (не включая лидера) в его группе. В последней строке в том же формате выведите описание группы второго лидера.

Если возможных ответов несколько, выведите любой из них.

Пример

стандартный ввод	стандартный вывод
20	1900
10 30	2 19
40 10	1 1

Задача D. Расколбас с Франкенштейном

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

Доктор Франкенштейн решил заняться самым интересным занятием на Земле — смешиванием колб с реагентами.

Каждый реагент описывается одним действительным числом — коэффициентом. От этого числа зависит какому классу принадлежит этот реагент. Существуют четыре класса:

1. \mathbb{N} : если коэффициент является натуральным числом, т.е. число, возникающее естественным образом при счёте (например, 1, 2, 3, 4 и так далее).
2. \mathbb{Z} : если коэффициент — целое число. Целые числа включают в себя натуральные числа, числа, противоположные натуральным, и 0. (например, 5, -1 , 0)
3. \mathbb{Q} : если коэффициент — рациональное число, или же число, которые можно представить в виде дроби $\frac{m}{n}$, где m и n — целые числа (например, 1, 2.7, 4.(3))
4. \mathbb{R} : если коэффициент — действительное число. Действительные числа включают в себя и те, которые возможно представить в виде дроби, и те, которые невозможно. (например, π , $\sqrt{2}$)

Как видим, каждый следующий класс содержит в себе предыдущий, поэтому для любого реагента выбирается класс, минимальный по включению.

Можно взять два реагента и смешать их, в результате смешивания может произойти три реакции — реакция сложения, вычитания и умножения. После этого возникает новый реагент, коэффициент которого равен сумме коэффициентов, разности коэффициентов и произведению коэффициентов соответственно.

Теперь задача — Франкенштейн взял два реагента и смешал их. Дана реакция и классы, которым принадлежат реагенты. Определить класс, которому принадлежит результат реакции. Считайте, что надо определить класс в общем случае (так, например, существуют конкретные примеры вида $\sqrt{2} \cdot \sqrt{2} = 2 \in \mathbb{N}$, но в общем случае произведение двух вещественных чисел всегда будет вещественным, и не всегда хотя бы рациональным).

Формат входных данных

В единственной строке вводятся три символа. Второй символ может быть равен «+», «-» или «*». Остальные могут быть равны «N», «Z», «Q» или «R».

Первый символ — класс первого реагента, второй обозначает осуществлённую реакцию («+» — сложение, «-» — вычитание, «*» — умножение), третий — класс второго реагента.

Формат выходных данных

Выведите единственный символ, равный «N», «Z», «Q» или «R» — класс, которому принадлежит реагент-результат.

Пример

стандартный ввод	стандартный вывод
Q + Z	Q

Задача Е. Самая страшная история

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

Маленький Джек решил написать самую страшную историю, чтобы напугать своих друзей на Хэллоуин.

Назовем историей непустую последовательность из слов, разделенных пробелами. Слово в истории — непустая последовательность строчных букв латинского алфавита.

Как известно, на качество истории влияют не только слова, содержащиеся в ней, но и символы, содержащиеся в этих словах.

Джек уже составил историю из n слов. Теперь он хочет проверить m гипотез относительно получившейся истории, чтобы убедиться, что она действительно страшная. Для проверки каждой гипотезы ему необходимо по номеру символа в истории узнать порядковый номер слова и позицию символа в этом слове.

Формат входных данных

В первой строке ввода через пробел даны два числа n и m — количество слов в истории и количество гипотез ($1 \leq n \leq 10^5$; $1 \leq m \leq 5 \cdot 10^5$).

В следующей строке записана история, написанная Джеком — n слов из строчных латинских букв, разделенные пробелами. Гарантируется, что суммарная длина слов не превышает 10^6 .

В последней строке ввода через пробел перечислены m целых чисел — номера символов в гипотезах Джека ($1 \leq x_i \leq \sum_{i=1}^n |s_i|$).

Формат выходных данных

Выведите m пар чисел, каждую в отдельной строке. Пара чисел в i -й строке — порядковый номер слова, в котором содержится i -й символ, и номер этого символа в слове (слова и символы нумеруются с единицы).

Пример

стандартный ввод	стандартный вывод
3 15	1 1
hell spirits fear	1 2
1	1 3
2	1 4
3	2 1
4	2 2
5	2 3
6	2 4
7	2 5
8	2 6
9	2 7
10	3 1
11	3 2
12	3 3
13	3 4
14	
15	

Задача F. Стрелочник

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Загулявшись поздно Хэллоуинской ночью, вы и сами не заметили, как попали в ловушку к демону-стрелочнику. Было бы здорово выбраться из нее до рассвета, а иначе у вас будут все шансы остаться в ней навсегда (ну или как минимум до следующего Хэллоуина).

Ловушка представляет из себя матрицу размера $n \times m$. Некоторые клетки матрицы пусты, а в некоторых нарисованы стрелочки в соседние по стороне или углу клетки. Каждую секунду все стрелочки поворачиваются на 45° градусов по часовой стрелке.

Обозначим направление вверх как 0, вправо-вверх как 1 и так далее, пустую клетку обозначим точкой. Вы находитесь в клетке с координатами (a_x, a_y) , и,

- находясь в пустой клетке, можете либо секунду подождать в ней, либо за секунду переместиться в соседнюю по стороне клетку;
- попадая на клетку со стрелочкой, вы моментально (за 0 секунд) переноситесь туда, куда она указывает.

Когда вы переходите на клетку со стрелкой, она уже успевает повернуться за ту секунду, что вы шагали. Ваша задача — выбраться из ловушки как можно скорее. Попадите из стартовой точки (a_x, a_y) в конечную (b_x, b_y) за минимальное количество секунд, либо определите, что это невозможно, и смиритесь с тем, что вам не выбраться.

Формат входных данных

В первой строке через пробел даны два целых числа n и m — размеры ловушки ($1 \leq n, m \leq 1000$).

Во второй строке даны два целых числа a_x и a_y — координаты стартовой клетки ($1 \leq a_x \leq n$; $1 \leq a_y \leq m$).

В третьей строке так же даны два целых числа b_x и b_y — координаты конечной клетки ($1 \leq b_x \leq n$, $1 \leq b_y \leq m$).

Далее следуют n строк по m символов — описание матрицы. Гарантируется, что ни в стартовой, ни в конечной точке нет стрелочек.

Формат выходных данных

В качестве ответа выведите минимальное время, необходимое, чтобы добраться из (a_x, a_y) в (b_x, b_y) , либо -1 , если это невозможно.

Примеры

стандартный ввод	стандартный вывод
2 2 1 1 2 2 .4 2.	8
1 5 1 1 1 5 .007.	-1
3 3 1 1 3 3 .05 655 01.	-1
3 3 1 1 3 3 .12 345 67.	7

Задача G. Уиджа

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

Наверное, многие видели в фильмах как люди используют для спиритических сеансов доску уиджа. Стоит только задать вопрос, и кто-нибудь с той стороны обязательно ответит, двигая указатель по буквам на доске вашими пальцами. Это достаточно опасная затея, но, к сожалению, не все подходят к ней достаточно серьезно.

В один не очень прекрасный вечер два друга нашли уиджа на чердаке, и решили с ней поиграть. Для этого они расчертили на ней клетчатое поле $n \times m$ и приклеили указатель к левой верхней клетке доски.

Затем друзья по очереди делали ходы. Каждый ход заключается в разрезании доски по горизонтали или по вертикали по границам клеток, после чего та часть, на которой приклеен указатель, остается в игре, а вторая половина выкидывается. Тот, кто не может сделать ход, то есть держит перед своим ходом доску размера 1×1 , проигрывает.

Разумеется, духи с той стороны очень недовольны таким обращением с уиджа, поэтому проигравший отправится сразу на тот свет (хотя друзья пока что об этом не знают). Ваша задача — определить, у кого из игроков, первого или второго, есть выигрышная стратегия, и продемонстрировать ее.

Формат входных данных

В единственной строке ввода через пробел даны два целых числа n и m — размеры поля ($1 \leq n, m \leq 10^5$).

Протокол взаимодействия

Это интерактивная задача.

В рамках взаимодействия с интерактором первым действием ваша программа должна вывести число «1» (без кавычек), если вы выбираете ходить первым, и «2», если вторым.

После чего в соответствии с выбранной очередностью ваша программа и интерактор совершают ходы. Каждый ход описывается парой из символа «R» или «C», обозначающего разрезание по горизонтали или по вертикали, соответственно, и числа t , обозначающего номер строки/столбца, после которой/которого делается разрез. Например, если размеры оставшейся доски равны 13×17 , ход «R 5» означает, что доска разрезается по горизонтали на части размеров 5×17 и 8×17 . Строки и столбцы нумеруются с единицы слева-направо и сверху-вниз, начиная от края оставшейся в игре части доски.

Если ваша программа выводит некорректный ход, интерактор выводит в ответ «FAIL» и завершается с вердиктом Wrong Answer. Считав слово «FAIL», ваша программа должна также завершиться во избежание вердиктов Time Limit Exceeded и Idleness Limit Exceeded.

Если интерактор не может сделать ход, он выводит «YOU WIN» и завершается с вердиктом OK, после чего ваша программа также должна завершиться с нулевым кодом возврата. Обратите внимание, что интерактор не завершается, если после его хода у вашей программы не остается возможности походить — чтобы в таком случае получить вердикт Wrong Answer, следует вывести произвольный некорректный ход и завершиться.

После каждой выведенной строки необходимо сбрасывать буфер потока вывода (`sys.stdout.flush()` в Python, `System.out.flush()` в Java, `cout.flush()` в C++).

Примеры

стандартный ввод	стандартный вывод
3 3 R 1 YOU WIN	2 C 1
4 5 C 1 YOU WIN	1 R 2 R 1

Задача Н. Монстры и люди

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

Иногда чтобы проникнуться таинственной атмосферой друзья играют в игру «Монстры и люди». Монстры и люди — это загадочная игра, в которой некоторые игроки являются обычными людьми, а некоторые монстрами.

Монстры, конечно, знают друг друга, а вот обычные люди не знают, кто есть кто.

На очередном ходе игры каждый из n игроков выбирает ровно одного другого игрока (но не себе) и выдвигает обвинения против него. Монстры сотрудничают, поэтому всегда выдвигают обвинения против обычных людей. Обвинения обычных людей при этом основаны только на догадках по ходу игры.

Вы не знаете, кто монстр, а кто обычный житель, но вам известно, какой игрок выдвинул обвинения против какого игрока. Определите, **какое максимальное число монстров** может быть среди игроков!

Формат входных данных

В первой строке содержится целое число n — число игроков в игре ($2 \leq n \leq 5 \cdot 10^5$).

Следующие n строк содержат информацию о том, кто кого обвинил на текущем ходе игры. В i -й строке одно число m_i , которое означает, что игрок с номером i обвинил игрока m_i .

Формат выходных данных

Выведите единственное целое число: максимальное число монстров на текущем ходе игры.

Примеры

stdin	stdout
3 3 1 1	2
3 2 3 1	1
7 3 5 4 6 4 3 4	4