

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 разделимся!

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 разделиться и исследовать страшное и подозрительное место небольшими группами.

В этот раз компания из $n + 2$ человек, исследуя заброшенную хижину в темном лесу, решила разделиться 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. Самая страшная история

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
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 совершить с этой историей m операций, каждая из которых заключается либо в проверке того, насколько история страшная, либо в небольшом изменении этой истории. Формально, Джек может делать с историей четыре вида операций:

- по номеру символа в истории узнать порядковый номер слова и позицию символа в этом слове;
- по номеру слова и позиции символа в нем узнать его номер в истории;
- вставить символ (букву или пробел) в определенную позицию в истории;
- удалить символ на определенной позиции в истории.

Помогите Джеку быстро совершить с историей все операции, чтобы он мог наконец-то рассказать ее друзьям!

Формат входных данных

В первой строке ввода через пробел даны два числа n и m — количество слов в истории и количество операций ($1 \leq n, m \leq 2 \cdot 10^5$).

В следующей строке записана история, написанная Джеком — n слов из строчных латинских букв, разделенные пробелами. Гарантируется, что суммарная длина слов не превышает 10^5 .

В i из следующих m строк дано описание i -й операции:

- «?1 i » — найти номер слова и позицию в слове для символа под номером i ($1 \leq i \leq L$, где L — текущая длина истории);
- «?2 w p » — найти для p -го символа в w -м слове его позицию в истории ($1 \leq w \leq W$; $1 \leq p \leq P$, где W — текущее количество слов, а P — длина w -го слова);
- «+ i c » — вставить символ c на позицию i в историю ($1 \leq i \leq L$; c — строчная латинская буква или «_» для пробела);
- «- i » — удалить i -й символ из истории ($1 \leq i \leq L$).

Гарантируется, что ни в какой момент времени в истории нет двух пробелов подряд и нет пробела в начале, и что символы в запросах первого типа — всегда буквы, а не пробелы.

Формат выходных данных

Для каждого запроса первого типа выведите в отдельной строке пару чисел w и p — номер слова и позицию символа в нем. Для каждого запроса второго типа, аналогично, выведите в отдельной строке число i — позицию запрошенного символа в истории.

Пример

стандартный ввод	стандартный вывод
3 16	1 1
hell spirits fear	14
?1 1	13
?2 3 1	3 1
+ 12 _	3 2
?2 3 1	4 1
+ 13 i	2 2
?1 13	1 7
?1 14	8
?1 16	10
?1 7	14
- 5	
?1 7	
?2 1 8	
- 1	
- 1	
?2 2 1	
?2 3 2	

Задача E. Trick or Treat!

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

Одно из распространенных среди детей развлечений на Хэллоуин — наряжаться в костюмы и ходить ~~выпрашивать~~ собирать конфеты. Однако, в этом году что-то пошло не так, и на улицы вышло слишком много детей, поэтому конфет на всех может не хватить!

Поэтому дети решили собраться в группы, чтобы иметь больше шансов собрать конфеты. К сожалению, они не успели вовремя скоординироваться, поэтому каждый ребенок решил пойти в сторону ближайшего к нему другого ребенка. Разумеется, это не лучшая стратегия, ведь может так оказаться, что ребенок А пошел в сторону ребенка В, а тот, в свою очередь, уже выдвинулся в сторону ребенка С. Но, будем надеяться, какие-то группы они все же смогут сформировать...

Всего на улицы Манхэттена вышло n детей, при чем i -й ребенок находится в точке с координатами (x_i, y_i) . Как известно, манхэттэнское расстояние между точками (x_i, y_i) и (x_j, y_j) равно $|x_i - x_j| + |y_i - y_j|$.

Чтобы предотвратить хаос на дорогах, вам поручено определить для каждого ребенка номер ближайшего к нему другого ребенка, чтобы иметь возможность хотя бы примерно предсказать траектории их перемещения по городу.

Формат входных данных

В первой строке ввода дано целое число n — количество детей в городе ($2 \leq n \leq 10^5$).

В i -й из следующих n строк через пробел даны два целых числа x_i и y_i — координаты i -го ребенка ($0 \leq x_i, y_i \leq 10^9$). **Не гарантируется**, что все дети находятся в разных точках — если два ребенка имеют одинаковые координаты, для них обоих кратчайшее расстояние будет равно 0.

Формат выходных данных

Выведите через пробел n целых чисел от 1 до n , i -е из которых равно номеру ребенка, ближайшего по манхэттэнскому расстоянию к i -му.

Примеры

стандартный ввод	стандартный вывод
3 1 1 3 3 6 6	2 1 2
4 1 6 0 4 3 8 7 3	2 1 1 2

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 клетке на пересечении x -й строки и y -го столбца.

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 перечислены 4 целых числа n , m , x и y — размеры поля и координаты указателя ($1 \leq x \leq n \leq 10^5$; $1 \leq y \leq m \leq 10^5$).

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 2 2 R 1 R 1 YOU WIN	2 C 1 C 1
4 5 3 2 C 4 C 2 C 1 YOU WIN	1 R 3 C 1 R 1 R 1

Задача Н. Расстановка тыкв

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Для Хэллоуина m жителей решили расставить множество тыкв вдоль забора своего дома. Предварительно были выбраны n мест, в которых можно расположить тыквы, при чем i -е место характеризуется двумя параметрами: x_i — расстоянием от начала забора до этого места, и c_i — уровнем *недовольства* жителей, если в данном месте будет расположена тыква (у каждого свое понимание об идеальной расстановке).

Чтобы вычислить *удовлетворенность* каждого жителя расстановкой, всех попросили назвать их любимое число. Житель номер i назвал число d_i , которое означает, что

- уровень его удовлетворенности двумя соседними тыквами на расстоянии d друг от друга равен $|d - d_i|$;
- суммарная *удовлетворенность* жителя равна сумме удовлетворенности для каждой двух соседних тыкв.

Поскольку жители дома хотят, чтобы отмечание праздника всех порадовало, было решено максимизировать суммарную удовлетворенность расстановкой тыкв. Однако также было учтено суммарное *недовольство* жителей всеми выбранными местами. Таким образом, если тыквы расположить в местах j_1, j_2, \dots, j_k (в порядке отдаления от начала забора), суммарная удовлетворенность будет вычисляться как

$$\left(\sum_{i=1}^m \sum_{t=2}^k |x_{j_t} - x_{j_{t-1}} - d_i| \right) - \left(\sum_{t=1}^k c_{j_t} \right).$$

Выведите максимальную суммарную удовлетворенность, которую можно достичь, оптимально выбрав места для тыкв.

Формат входных данных

В первой строке ввода через пробел даны два числа n и m — количество мест для тыкв и количество жителей дома, соответственно ($2 \leq n \leq 10^5$; $1 \leq m \leq 10^5$).

Во второй строке через пробел перечислены m чисел d_i — любимые числа каждого жителя ($0 \leq d_i \leq 10^7$).

В каждой из следующих n строк записаны числа x_i и c_i — параметры выделенных мест для расположения тыкв ($0 \leq x_i \leq 10^7$; $0 \leq |c_i| \leq 10^{12}$). Гарантируется, что $x_1 < x_2 < \dots < x_n$.

Формат выходных данных

Выведите одно число — максимальную суммарную удовлетворенность жителей дома.

Примеры

стандартный ввод	стандартный вывод
2 1 10 0 5 20 3	2
3 3 3 7 10 2 20 5 4 10 -3	-1
9 5 30 64 2 93 67 0 81 1 256 6 251 13 256 23 180 52 256 72 94 77 256 97 12	137

Задача I. Интересные празднования

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

Когда ваша семья празднует Хэллоуин каждый год, начинает хотеться как-то разнообразить празднование, чтобы Хэллоуин не надоедал.

Достоверно известно, что есть всего 26 способов отпраздновать Хэллоуин, i -й из которых может быть обозначен i -й буквой латинского алфавита (от 'a' до 'z'). Семья Майерсов отмечает Хэллоуин уже очень давно, и, разумеется, они ведут записи о том, каким способом они его отмечали каждый год.

Теперь им стало интересно, сколько подпоследовательностей лет (не обязательно идущих подряд) были *интересными*. Всего есть ровно 26 возможных *интересных* последовательностей способа отмечания, которые определяются следующим образом:

- $\text{seq}_1 = \text{«a»}$;
- $\text{seq}_{i+1} = \text{seq}_i + \text{«}c_{i+1}\text{»} + \text{seq}_i$, где c_{i+1} — $(i+1)$ -й символ латинского алфавита.

Так, первые три *интересные* последовательности равны «a», «aba» и «abacaba».

Вам дана строка s , i -й символ которой равен способу отмечания Хэллоуина в i -й год. Помогите Майерсам определить количество ее подпоследовательностей, которые являются *интересными*. Поскольку это число может оказаться слишком большим, достаточно вычислить его по модулю 998244353. Напомним, что подпоследовательностью называется строка, полученная из данной вычеркиванием некоторого, возможно нулевого, количества символов.

Формат входных данных

В единственной строке задана строка s ($1 \leq |s| \leq 5000$).

Формат выходных данных

Выведите число подпоследовательностей этой строки вида seq_i по модулю 998244353.

Примеры

стандартный ввод	стандартный вывод
abacaba	11
b	0

Замечание

Из строки «abacaba» можно выбрать 4 подпоследовательности «a», 6 подпоследовательностей «aba» и одну подпоследовательность «abacaba».

Задача J. 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 который 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