

Задача А. Банк

Имя входного файла: `bank.in`
Имя выходного файла: `bank.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В одном далеком мире, в славном городе Эрбовле открыли новый банк. В банке есть m сотрудников, работающих с клиентами, и один главный бухгалтер.

Для решения своих проблем в банк приходят гномы. Известно, что i -й гном приходит в банк через t_i минут после открытия банка. Сначала ему нужно провести a_i минут у одного из m сотрудников, а потом еще b_i минут в офисе главного бухгалтера.

Разумеется несколько гномов не могут одновременно находиться у одного сотрудника или в офисе главного бухгалтера, поэтому к сотрудникам и к главному бухгалтеру формируются очереди.

Очередь к сотрудникам общая, при этом гном из очереди идет к первому освободившемуся сотруднику. Если в банк одновременно приходят два гнома, то первым в очередь к сотрудникам встает тот, чей номер меньше. Если гном начал обслуживаться у сотрудника в момент x , то он освобождается в момент $x + a_i$, в этот момент другой гном может начать обслуживаться у этого же сотрудника. Гном, пришедший в банк в момент t , может начать обслуживаться у сотрудника в любой момент, начиная с t .

Решив свои проблемы у сотрудника, гном идет в очередь к главному бухгалтеру. Аналогично, если два гнома приходят в эту очередь одновременно, первым встает гном с меньшим номером, в момент, когда заканчивается обслуживание одного из гномов, может сразу начаться обслуживание следующего, гном может попасть к главному бухгалтеру, начиная с того момента, когда закончил обслуживаться у сотрудника.

Сегодня в банк собирается прийти n гномов, про каждого известно: во сколько он заходит в банк, сколько времени он хочет провести у окошка и сколько времени он хочет провести у бухгалтера. Нужно сообщить время выхода из банка для каждого гнома.

Формат входного файла

В первой строке заданы два целых числа n и m ($1 \leq n \leq 100\,000$, $1 \leq m \leq 10$) — число гномов и сотрудников, соответственно. Далее, в n строках задано по три целых числа t_i , a_i и b_i ($1 \leq t_i, a_i, b_i \leq 10^9$) — время прихода i -го гнома, сколько минут i -й гном должен провести у сотрудника банка и сколько минут он должен провести в офисе главного бухгалтера. Известно, что гномы заданы в порядке прихода в банк, то есть для любой пары $i < j$ выполняется $t_i \leq t_j$.

Формат выходного файла

Выведите n целых чисел, i -е число должно быть равно числу минут после открытия, когда i -й гном покинет банк.

Примеры

bank.in	bank.out
4 2	8
1 3 3	5
1 2 2	9
2 2 1	13
2 1 4	

Задача В. Ставка

Имя входного файла: `bet.in`
Имя выходного файла: `bet.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Андрей очень любит играть в *Космический покер*.

В космическом покере вместо карт используются фишки трех цветов. Казино определяет два числа A и C — коэффициенты для вычисления ставок. Затем игрок по определенным правилам ставит фишки трех цветов: красного, зеленого и синего. Выигрыш игрока вычисляется по формуле: $A \cdot (r^2 + g^2 + b^2) + C \cdot \min\{r, g, b\}$, где r, g, b — количество фишек красного, зеленого и синего соответственно.

Правила, по которым делаются ставки, очень сложны, но сейчас перед Андреем стоит следующая задача. На поле уже есть r красных, g зеленых и b синих фишек. Прежде чем будет определен его выигрыш, он может добавить на поле ровно одну фишку любого цвета. Помогите ему выбрать цвет фишки, которую следует добавить на поле, чтобы максимизировать выигрыш.

Формат входного файла

Во входном файле содержится несколько игровых ситуаций, которые требуется проанализировать.

В первой строке задано одно целое число t ($1 \leq t \leq 10\,000$) — количество игровых ситуаций.

Каждая игровая ситуация описывается двумя строками. В первой строке задано два целых числа A и C ($1 \leq A, C \leq 10$) — коэффициенты для вычисления выигрыша. Во второй строке задано три целых числа r, g и b ($0 \leq r, g, b \leq 15$) — количество фишек красного, зеленого и синего цвета, соответственно.

Формат выходного файла

Выведите t строк. В k -ой строке выведите «RED», если оптимально добавить красную фишку, «GREEN», если оптимально добавить зеленую фишку или «BLUE», если оптимально добавить синюю фишку. Если есть несколько оптимальных вариантов, можно вывести любой из них

Примеры

<code>bet.in</code>	<code>bet.out</code>
3	RED
2 10	BLUE
2 4 4	GREEN
1 2	
3 4 5	
4 2	
7 7 7	

Задача С. Взлом шифра

Имя входного файла: `cipher.in`
Имя выходного файла: `cipher.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайта

Алан любит вскрывать шифры и кодовые замки. На этот раз ему попался необычайно сложный замок, найти ключ к которому Алану не удалось, поэтому он решил перебрать все возможные комбинации, чтобы узнать ключ.

Замок представляет собой n кнопок, пронумерованных целыми числами от 1 до n . Замок открывается только тогда, когда какие-то последовательные n нажатий на кнопки образуют некоторую секретную перестановку. Кнопки замка следует нажимать по очереди, нажать одновременно две или более кнопки нельзя.

Более формально: предположим, что Алан нажал на кнопки k раз. Пусть a_i ($1 \leq i \leq k$) — номер кнопки, которую Алан нажал i по счету, а b_1, b_2, \dots, b_n — секретная перестановка. Тогда замок открывается, если существует такое число x ($1 \leq x \leq k - n + 1$), что $b_1 = a_x, b_2 = a_{x+1}, \dots, b_n = a_{x+n-1}$.

Алан хочет придумать такую универсальную последовательность нажатий, что при нажатии кнопок в такой последовательности замок откроется для любой секретной перестановки. Также Алан хочет, чтобы эта последовательность не была слишком длинной, а именно, ее длина не превышала $2n!$, где $n! = 1 \cdot 2 \cdot \dots \cdot n$. Например, для $n = 3$ длина последовательности не должна превышать 12.

Помогите Алану найти такую последовательность.

Формат входного файла

В единственной строке входного файла находится целое число n ($1 \leq n \leq 9$) — количество кнопок на кодовом замке.

Формат выходного файла

В первой строке выходного файла выведите число k ($0 \leq k \leq 2n!$) — длину универсальной последовательности. Во второй строке выведите k целых чисел a_i , разделенных пробелами ($1 \leq a_i \leq n$) — порядок, в котором следует нажимать кнопки. Обратите внимание, что достаточно вывести любую последовательность длины не более $2n!$, минимизировать длину не нужно. Гарантируется, что такая последовательность существует для любого n .

Примеры

<code>cipher.in</code>	<code>cipher.out</code>
2	3 1 2 1
3	10 1 2 3 1 3 2 1 3 1 2

Задача D. Болезнь

Имя входного файла: `disease.in`
Имя выходного файла: `disease.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В Байтландии вспыхнула эпидемия опасной болезни. Известно, что возбудителями болезни являются n различных болезнетворных бактерий.

Для правильного лечения пациента врачам необходимо знать, чем именно была вызвана его болезнь. Для этого пациент сдает m анализов: каждый анализ проверяет наличие или отсутствие некоторых видов бактерий. Анализ дает положительный результат, если в крови у человека есть хотя бы один из проверяемых этим анализом возбудителей болезни.

Помогите врачам по результатам анализов выяснить про каждую бактерию, заражен ли ею пациент.

Формат входного файла

В первой строке входного файла заданы два числа n ($1 \leq n \leq 100$) — число различных возбудителей болезни и m — число анализов. Следующие m ($1 \leq m \leq 10\,000$) строк содержат по $n + 1$ числу. Первые n чисел описывают, какие возбудители обнаруживаются этим анализом, i -е число равно 1, если анализ проверяет наличие i -го возбудителя и 0 — в противном случае.

Последнее число в строке равно 1, если анализ дал положительный результат, и 0 — в противном случае.

Формат выходного файла

Если входные данные противоречивы, выведите в выходной файл единственную строку «Incorrect». В противном случае выведите в выходной файл три строки. Каждая строка задается в формате: число бактерий, далее их номера.

В первой строке необходимо вывести номера бактерий, которые не могут являться причиной болезни, во второй — номера бактерий, про которые можно точно утверждать, что они являются причиной болезни, в третьей — номера бактерий, про которые по результатам анализов ничего утверждать нельзя.

Примеры

disease.in	disease.out
3 3 1 0 0 0 1 1 1 1 0 1 0 0	2 1 2 1 3 0
2 2 1 1 0 1 0 1	Incorrect
2 1 1 1 1	0 0 2 1 2

Задача Е. Разделение королевства

Имя входного файла: `division.in`
Имя выходного файла: `division.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайта

Королевство Флатландия имеет вид бесконечной двумерной плоскости. В королевстве находится n замков. Для более удобного составления карт в Флатландии была введена Декартова система координат. Известно, что i -й замок находится в точке с координатами $(x_i + 0.5, y_i + 0.5)$, где x_i, y_i — целые числа. Местоположения всех замков попарно различны.

На старости лет король решил разделить на карте королевство между своими сыновьями прямыми, параллельными осям координат. Если прямая параллельна оси Ox , то y координата у всех точек на прямой должна быть целым числом, иначе x координата у всех точек должна быть целым числом. В обоих случаях соответствующие целые координаты по модулю не должны превышать $2 \cdot 10^9$. При этом Его величество хочет, что бы после разделения королевства любые два замка оказались бы в различных частях.

Помогите королю разделить королевство, используя не более чем $n - 1$ прямую. У любой пары прямых должно быть не более одной общей точки.

Формат входного файла

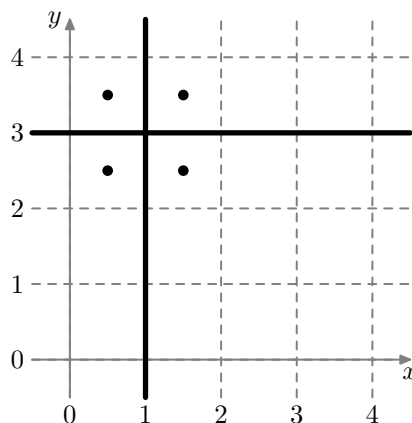
В первой строке задано целое число n ($1 \leq n \leq 100\,000$) — количество замков в королевстве. В следующих n строках записано по два числа x_i и y_i ($-10^9 \leq x_i \leq 10^9, -10^9 \leq y_i \leq 10^9$) — целые части координат замков.

Формат выходного файла

В первой строке выходного файла выведите количество используемых прямых. В следующих строчках выведите сами прямые, по одной в каждой строке. Если прямая параллельна оси Ox , то выведите символ « y », а затем через пробел y координату всех точек на этой прямой, иначе выведите символ « x », а затем через пробел x координату всех точек на этой прямой.

Примеры

<code>division.in</code>	<code>division.out</code>
4	2
0 2	x 1
0 3	y 3
1 2	
1 3	



Задача F. Загадочное уравнение

Имя входного файла: `equation.in`
Имя выходного файла: `equation.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Маленький Вася очень любит уравнения. Однажды ему на глаза попало уравнение $x + y + xy = n$. Вася захотел узнать количество пар целых неотрицательных чисел x и y , которые являются решениями этого уравнения.

Так как Вася еще маленький, то он попросил вас посчитать это количество.

Формат входного файла

В единственной строке входного файла дано число n ($0 \leq n \leq 10^9$).

Формат выходного файла

В единственную строку выходного файла выведите ответ на задачу.

Примеры

<code>equation.in</code>	<code>equation.out</code>
5	4
8	3

Пояснение

Ниже перечислены все решения уравнения $x + y + xy = 5$

1. $x = 0, y = 5$
2. $x = 1, y = 2$
3. $x = 2, y = 1$
4. $x = 5, y = 0$

Задача G. Осенний парк

Имя входного файла: `park.in`
Имя выходного файла: `park.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Воскресенье, утро. Пора на олимпиаду. Вениамин взял пачку чистой бумаги, ручку, пару бутербродов — что еще понадобится? — и открыл картографический сайт чтобы посмотреть, куда и как ему добираться. Какая удача! На пути есть прекрасный парк, а Вениамин как раз любит гулять по паркам. Парк представляет собой прямоугольное поле, разбитое на квадратные клеточки, в каждой из которых либо газон с дорожками, либо какое-то препятствие (заросли кустарника, деревья, а то и вовсе какой-нибудь огороженный памятник).

На олимпиаду нужно прийти вовремя, так что Вениамин не может позволить себе долго разгуливать по парку. Перемещение между двумя соседними по стороне клеточками парка занимает одну секунду. Вениамин не может устоять на месте, поэтому каждую секунду он перемещается в какую-либо соседнюю клеточку. Вениамин решил, что может позволить себе гулять лишние две секунды. Ваша задача: посчитать количество способов пройти через парк так, чтобы время прогулки было ровно на две секунды больше минимального.

Вход в парк и выход из него — это некоторые выделенные различные клетки в парке, выходить за пределы парка запрещается, передвигаться можно только между соседними по ребру клетками. Вениамин должен гулять на две секунды дольше оптимального времени прохода от входа к выходу, поэтому он может в качестве промежуточной точки пути оказаться на входе или выходе. Поскольку ответ на задачу может быть довольно большим, от вас требуется остаток от деления количества путей на $10^9 + 9$.

Формат входного файла

В первой строке входного файла заданы два числа h и w : размеры парка. Следующие h строк содержат по w символов в каждой. Символ « $.$ » означает, что в соответствующей клетке дорожка или газон, по которому можно ходить. Символ « $\#$ » означает препятствие. Символы «E» и «X» означают вход в парк и выход из него соответственно.

Ограничения: $1 \leq h \leq 500$, $1 \leq w \leq 500$, символы «E» и «X» встречаются во входном файле ровно по одному разу. Обратите внимание, что вход и выход не обязательно находятся на границе парка: например, клеткой входа может быть расположенный в парке вестибюль метро, из которого Вениамин собирается выходить на своем пути.

Формат выходного файла

В выходной файл выведите одно число: количество путей, которые длиннее кратчайшего ровно на две секунды. Если парк устроен так, что невозможно добраться от входа до выхода, то выведите ноль.

Пример

<code>park.in</code>	<code>park.out</code>
6 9#####X ..#..... .E..####. ..##...#.#...	15

Задача Н. Палиндромные числа

Имя входного файла: `pnumbers.in`
Имя выходного файла: `pnumbers.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася очень любит изучать разные интересные классы чисел. Сегодня он изучает палиндромные числа.

Вася называет число палиндромным, если оно записывается одинаково слева направо и справа налево. При этом, Вася разрешает приписывать к числу несколько (возможно ни одного) лидирующих нулей. Например, числа 22, 4554, 12321, 5050 являются палиндромными. В частности, к числу 5050 необходимо приписать один ноль, чтобы получить 05050, которое читается одинаково слева направо и справа налево.

В числе прочих, Васю интересуют палиндромные числа, отличающиеся на 2. Для их исследования Вася рассматривает такие x , что $x - 1$ и $x + 1$ являются палиндромными. Такие числа Вася называет *междупалиндромными*. Вася хочет найти количество междупалиндромных чисел x от L_k до R_k включительно для нескольких отрезков $[L_k, R_k]$.

Помогите Васе в этом нелегком деле!

Формат входного файла

Входной файл содержит несколько отрезков, которые интересуют Васю. В первой строке задано одно число T ($1 \leq T \leq 2000$) — количество отрезков. В каждой из следующих T строк заданы два числа L_k и R_k ($1 \leq L_k \leq R_k \leq 10^{18}$) — границы отрезка.

Формат выходного файла

Выведите T строк. В k -ой строке выведите одно число — количество междупалиндромных чисел в отрезке от L_k до R_k включительно.

Примеры

<code>pnumbers.in</code>	<code>pnumbers.out</code>
3	1
18 23	1
21 21	0
50 55	

Пояснение

От 17 до 24 палиндромными являются числа 20 и 22. Поэтому единственное междупалиндромное число на отрезке $[18, 23]$ — это 21. Во втором примере, число 21 опять подходит. От 49 до 56 палиндромными являются числа 50, 55. Междупалиндромных чисел на отрезке $[50, 55]$ нет.

Задача I. Необычный экспонат

Имя входного файла: `scanner.in`
Имя выходного файла: `scanner.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Недавно на выставку странных устройств привезли новый экспонат. Его суть заключается в том, что он может генерировать случайную перестановку из чисел от 1 до n , после чего сканировать ее и выводить на экран $n - k + 1$ чисел, i -е из этих чисел обозначает количество инверсий на отрезке с i по $i + k - 1$ сгенерированной перестановки.

Напомним, что инверсией в перестановке p называется всякая пара индексов i, j такая, что $1 \leq i < j \leq n$ и $p[i] > p[j]$.

У этого экспоната кроме экрана есть две ручки — первая определяет n , то есть длину перестановки, а вторая определяет k . Посетитель Вася повернул ручки и увидел на экране числа. Теперь он хочет понять, какую перестановку сгенерировало это странное устройство. Помогите ему в этом.

Формат входного файла

В первой строке содержатся два натуральных числа n и k ($2 \leq n \leq 10^5$, $2 \leq k \leq 5$, $n \geq k$). Во второй строке даны $n - k + 1$ чисел, выведенные на экран устройством. Гарантируется, что устройство исправно, и существует хотя бы одна перестановка, по которой можно сгенерировать эти числа.

Формат выходного файла

Выведите n чисел, разделенных пробелами — сгенерированную устройством перестановку. Если возможных перестановок несколько, то выведите любую.

Пример

<code>scanner.in</code>	<code>scanner.out</code>
3 2 0 1	1 3 2

Задача J. Супрематизм

Имя входного файла: `supreme.in`
Имя выходного файла: `supreme.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Недавно на уроках ИЗО Казимиру рассказали о различных направлениях искусства. Больше всего его впечатлил *супрематизм*, и он решил нарисовать свою первую картину в этом стиле.

Казимир помнил, что в супрематизме картина состоит из простых фигур, поэтому, сначала он нарисовал прямоугольник $n \times m$, составленный из разноцветных квадратов 1×1 . После критического переосмысления своего творения, Казимир пришел к выводу, что получившаяся картина слишком сложна, и не все смогут понять его задумку. Второго холста у него не было, и он решил исправлять эту картину. На *достаточно простой* картине, по мнению Казимира, должен присутствовать всего один цвет.

Казимир решил исправить картину следующим образом. Он может взять строку своей картины, более половины единичных квадратов в которой покрашено в один и тот же цвет, и перекрасить всю строку в этот цвет. Аналогично он может перекрасить столбец, более половины единичных квадратов в котором покрашено в один цвет.

Помогите Казимиру определить, сможет ли он с помощью этих операций исправить свою картину и сделать ее достаточно простой.

Формат входного файла

В первой строке заданы два числа n и m ($1 \leq n, m \leq 300$) — размеры картины. Далее, в n строках задано по m чисел $c_{i,j}$ ($1 \leq c_{i,j} \leq 1\,000\,000$) — цвета квадратов, из которых составлена картина. Гарантируется, что на картине представлено хотя бы два цвета.

Формат выходного файла

Если Казимиру не удастся сделать картину достаточно простой, выведите «Poor Kazimir».

Иначе, выведите в первой строке k — количество действий, которое нужно сделать Казимиру. Действия могут быть двух видов:

- R r — перекрасить строку r ($1 \leq r \leq n$).
- C c — перекрасить столбец c ($1 \leq c \leq m$).

Разрешается сделать не более 1000 действий.

Примеры

supreme.in	supreme.out
3 3 1 1 2 2 1 1 2 2 2	5 R 1 R 2 C 1 C 2 C 3
3 3 1 1 2 2 1 1 2 2 2	4 C 1 C 3 R 1 R 2
2 2 1 2 3 4	Poor Kazimir