

## Задача 2. Упражнение по арифметике

Имя входного файла: `standand input`  
Имя выходного файла: `standand output`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

В конце урока учитель арифметики задал ученикам на дом задание, но Вова очень спешил на перемену и не записал его. Теперь же дома он пытается вспомнить это задание. Он помнит лишь, что это пример, в котором были записаны несколько первых натуральных чисел, перед каждым из них стоял знак сложения или вычитания, и не было скобок:

$$? 1 ? 2 ? 3 ? \dots ? N =$$

А еще он успел подсмотреть в книгу учителя и знает, что ответом является целое число  $k$ . Теперь Вова хочет попытаться восстановить пример из задания.

### Формат входного файла

Во входном файле записано целое число  $k$  ( $|k| \leq 10^{100000}$ ).

### Формат выходного файла

В выходной файл выведите наименьшее натуральное число  $N$  ( $N \geq 1$ ), для которого существует хотя бы одна такая расстановка знаков в примере, которая дает ответ  $k$ .

### Примеры

standand input	standand output
4	3
-12	7

## Задача 3. Lecture

Имя входного файла: `lecture.in`  
Имя выходного файла: `lecture.out`  
Ограничение по времени: 10 seconds  
Ограничение по памяти: 256 Mebibytes

Один передовой японский режиссёр решил снять фильм «Двенадцать стульев», выдержанный в духе традиционной японской культуры.

В этом фильме великий комбинатор, сопровождаемый верным соратником Кисаво, придя в селение Васуки, распускает слухи о том, что прибыл истинный мастер игры в го, имеющий рекомендации от одного из принцев крови, а также лично от В.А. Кузнецова.

За небольшие пожертвования в пользу несчастного Кисаво мастер соглашается раскрыть секреты своего мастерства, рассказав, каким образом можно в общем случае оценить выигрышность или проигрышность плана игры ещё до первого хода. Вот изложение его рассказа:

«Пусть нам дано поле  $N \times N$ , и на нём выбраны  $N$  прямоугольников. Рассмотрим все способы выбрать из каждого прямоугольника по элементу. Для каждого способа составим матрицу с единицами в выбранных элементах и нулями в остальных местах. Секрет состоит в том, что если сумма определителей полученных таким способом матриц положительна, то это добрый знак и Вы выиграете. Если сумма отрицательна — откажитесь от своего плана, он приведёт Вас к краху. Если же сумма равна нулю, то всё решит дальнейшая игра».

Дальнейшие события мало отличаются от описанных в оригинале. После выхода фильма, имевшего шумный успех, поклонники творчества режиссёра создали фан-сайт этого фильма. На нём, в частности, планируется размещение программы, вычисляющей знак вышеупомянутой суммы определителей. А написание этой программы поручено вам.

### Формат входного файла

В первой строке находится число  $1 \leq N \leq 10^6$  — размер поля, в следующих  $N$  строках находятся четыре числа  $x_1, x_2, y_1, y_2$  ( $1 \leq x_1, x_2, y_1, y_2 \leq N$ ), задающих прямоугольники. Левый верхний элемент прямоугольника имеет координаты  $(x_1, y_1)$ , правый нижний имеет координаты  $(x_2, y_2)$ .

### Формат выходного файла

Если сумма положительна, выведите 1, если отрицательна, то  $-1$ , если равна нулю — выведите 0.

### Примеры

<code>lecture.in</code>	<code>lecture.out</code>
3 1 1 1 1 2 2 3 3 3 3 2 2	-1
3 1 2 1 2 2 3 2 3 1 2 2 3	0

## Задача 5. Newspaper

Имя входного файла:	<code>newspaper.in</code>
Имя выходного файла:	<code>newspaper.out</code>
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 Mebibytes

Редакция одной московской газеты переехала в новый офис. При переезде среди мебели обнаружилось несколько антикварных стульев, купленных ещё в 30-е годы прошлого века.

Один из журналистов неосторожно написал про это, упомянув, что, возможно, найдены прототипы знаменитых полукресел из романа «12 стульев». После выхода этой статьи боевая организация старгородского Общества ценителей творчества Ильфа и Петрова решила, что пора вернуть экспонаты в Старгород.

Руководитель боевой организации сумел раздобыть информацию об устройстве офиса. Оказалось, что офис состоит из небольших фойе, соединённых между собой непересекающимися прямолинейными коридорами, причём в каждом фойе и в середине каждого из коридоров находится ровно один стул. Двигаться по коридорам можно только в одну сторону — в целях антитеррористической защиты в офисе установлена сигнализация, которая срабатывает при движении по коридору в «неправильном» направлении.

В одном из фойе расположен вход в офис, в другом — выход. При этом от входа можно дойти до каждого фойе так, что при этом сигнализация не сработает. Аналогично, от каждого фойе можно без срабатывания сигнализации дойти до выхода. Никакие два фойе не соединены напрямую более, чем одним коридором. При этом офис ориентирован так, что для каждого коридора при движении в «правильном» направлении начальное фойе всегда расположено строго севернее конечного.

Был разработан план действий, согласно которому отправляемые в Москву под видом реставраторов мебели агенты заходят в офис, проходят одним из возможных маршрутов до выхода так, чтобы сигнализация не сработала, по пути забирая необходимые стулья из фойе и коридоров. В случае, если не все стулья собраны, ситуация повторяется — вход в офис, проход по другому маршруту, сбор стульев из попутных фойе и коридоров, выход — и так далее, пока все стулья не будут реквизированы.

Какое минимальное количество проходов по зданию потребуется агентам, чтобы выполнить задание и вернуть реликвию в Старгород?

### Формат входного файла

В первой строке входного файла задано целое положительное число  $T \leq 20$  — количество тестовых примеров. Первая строка каждого тестового примера содержит одно число  $n$  ( $2 \leq n \leq 2 \cdot 10^4$ ) — количество фойе в офисе редакции. В фойе с номером 1 находится вход в офис, в фойе с номером  $n$  — выход. В последующих  $n - 1$  строках заданы коридоры.  $i$ -я из этих строк описывает все коридоры, выходящие из фойе с номером  $i$ . Сначала задаётся количество коридоров  $k_i$  ( $1 \leq k_i < n$ ), затем идут  $k_i$  различных целых неотрицательных чисел  $f_{i,j}$  ( $2 \leq f_{i,j} \leq n$ ) — номера фойе, в которые ведут коридоры из комнаты  $i$ . При этом коридоры перечислены с запада на восток (то есть каждый последующий в порядке перечисления коридор, выходящий из некоторого фойе, ведёт восточнее, чем предыдущий).

### Формат выходного файла

Для каждого тестового примера выведите в отдельной строке одно число — минимальное количество проходов по зданию от входа до выхода, которое потребуется агентам для того, чтобы собрать все стулья.

## Пример

newspaper.in	newspaper.out
1	4
7	
3 4 3 6	
1 7	
1 4	
1 7	
1 7	
2 5 2	

## Задача 7. Game on Chessboard

Имя входного файла: `chessgame.in`  
Имя выходного файла: `chessgame.out`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 64 Mebibytes

Есть шахматная доска размера  $M \times N$ . На ней в некоторых клетках расположены  $K$  вымышленных шахматных фигур, называемых  $(p, q)$ -скакунами ( $p < q$ ). Ход скакуна похож на ход обычного шахматного коня. При своем ходе скакун перемещается на  $p$  клеток по горизонтали и на  $q$  по вертикали или на  $q$  клеток по горизонтали и на  $p$  по вертикали. При этом перемещение на  $q$  должно выполняться обязательно либо вверх, либо влево (то есть в сторону уменьшения соответствующей координаты). Недопустим ход, который выводит фигуру за пределы доски, однако в одной клетке могут находиться несколько фигур.

Два игрока играют в игру. Они ходят по очереди. В свой ход игрок обязан выбрать одного из скакунов и выполнить им допустимый ход. Проигрывает тот, кто не может сделать ход.

Определите кто выиграет, предполагая что оба игрока играют оптимально,

### Формат входного файла

В первой строке входного файла задано 5 целых чисел  $M, N, K, p, q$  ( $1 \leq M, N \leq 10^9, 1 \leq K \leq 10^5, 1 \leq p < q \leq 20$ ). В каждой из последующих  $K$  строк задаются координаты соответствующего скакуна  $r_i$  и  $c_i$  ( $1 \leq r_i \leq M, 1 \leq c_i \leq N$ ).

### Формат выходного файла

В выходной файл выведите 'First', если при оптимальной игре выигрывает первый игрок, и 'Second' в противном случае.

### Примеры

<code>chessgame.in</code>	<code>chessgame.out</code>
10 10 2 1 2 3 7 7 3	Second
7 5 3 1 3 2 3 1 5 4 3	First

## Задача 11. Радиовышки

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 8 секунд  
Ограничение по памяти: 256 мегабайта

На одной из военных баз столкнулись с непредвиденной проблемой. В поле были размещены здания, но не было никакой связи между ними. Для устранения этого, было решено поставить на этом поле несколько радиовышек и по одной радиовышке на каждое здание. Тогда сигналы можно будет передавать от одной радиовышки до другой. Но, так как это дело не первой необходимости, то денег на вышки надо потратить как можно меньше.

Более формально. Дана таблица размером  $n \times m$ . Каждая ячейка либо пуста, либо содержит здание. В любую пустую ячейку можно поставить радиовышку, и, обязательно, на каждое здание требуется установить по радиовышке. Каждая радиовышка имеет мощность от 1 до 9. Это число означает, на каком расстоянии она будет работать (отправлять сигналы до других радиовышек). Расстояние считать Евклидовым. Стоимость установки одной вышки равна  $a$ . Если вышка имеет мощность  $p$ , то надо дополнительно заплатить  $p^2$  единиц.

Определите, где и какие вышки необходимо разместить, чтобы можно было передать сигнал от любого здания к любому другому (возможно не напрямую), и чтобы стоимость размещения вышек была минимальна.

### Формат входного файла

Первая строка входного файла содержит 2 целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 20$ ,  $2 \leq n \times m \leq 25$ ) — размер поля. Далее следуют  $n$  строк по  $m$  символов — описание поля: «.» — пустая ячейка, «\*» — ячейка, занятая зданием. Гарантируется, что на поле всегда будут присутствовать по крайней мере 2 здания.

Последняя строка входного файла содержит одно целое число  $a$  ( $1 \leq a \leq 500$ ).

### Формат выходного файла

Выходной файл должен содержать  $n$  строк по  $m$  символов — поле, после установки вышек: «.» — пустая ячейка, «x» — ячейка, занятая вышкой мощностью  $x$ , где  $x$  — целое число от 1 до 9.

### Примеры

input.txt	output.txt
3 3 ..* ... *.. 4	..2 .2. 2.. ...
5 4 *... ...* .... *... ...* 9	1... 3..3 .... 2... ...3 ....

## Задача 13. Supercomputer

Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 Mebibytes

International Brotherhood of Magicians (IBM) разработали новейший суперкомпьютер. Отличительной особенностью этого суперкомпьютера является то, что он содержит бесконечное количество процессоров и неограниченный объем памяти! Более того, благодаря применению архитектуры гиперкуб и тахионной шины, время доступа процессоров к памяти равно нулю. Тем не менее, вычисления с плавающей запятой всё же требуют, как правило, некоторых временных затрат. В частности, время сложения и вычитания равно  $t_a$ , время умножения —  $t_m$ , деления —  $t_d$ , вычисления квадратного корня из числа —  $t_s$ . Время операции смены знака равно нулю. Однако операции сравнения чисел с плавающей запятой пока работают неэффективно и поэтому заблокированы. Кроме того, за счет аппаратной реализации библиотеки *libastral*, операции в целочисленной арифметике, включая сравнения, а также все операции перехода выполняются мгновенно. К сожалению, реализация ее все же неполна и не позволяет сразу узнавать ответ на любую задачу.

Одним из первых покупателей этого суперкомпьютера оказалась некая организация MJ12. Задача, которую требуется решить им, формулируется следующим образом. Пусть имеется симметричная матрица  $A$  размера  $N \times N$  с действительными коэффициентами, для которой известно, на каких позициях находятся ее ненулевые коэффициенты. Разложением Холецкого матрицы  $A$  называется такая нижне-треугольная матрица  $L$ , что  $A = LL^T$ , где  $L^T$  — транспонированная матрица  $L$ . Для того чтобы  $L$  существовала, необходимо, чтобы коэффициенты матрицы  $A$  удовлетворяли определенным условиям. Мы предполагаем, что эти условия выполняются. Тогда коэффициенты  $L$  могут быть вычислены по следующим рекуррентным формулам:

$$L_{jj} = \sqrt{A_{jj} - \sum_{k < j} L_{jk}^2}$$

$$L_{ij} = \frac{A_{ij} - \sum_{k < j} L_{ik} L_{jk}}{L_{jj}}, \quad i > j$$

MJ12 требуется вычислять по заданной матрице  $A$  соответствующую ей матрицу  $L$ . Ясно, что поскольку некоторые коэффициенты матрицы  $A$  являются нулевыми, то некоторые коэффициенты матрицы  $L$  также обязательно будут нулевыми, и их вычислять не требуется. Все остальные коэффициенты матрицы  $L$  вычислять необходимо. Перед покупкой суперкомпьютера MJ12 хотят узнать, насколько быстро он сможет решать эту задачу на их сверхсекретных матрицах. Чтобы не предоставлять их IBM, они попросили предоставить им программу, которая по заданным позициям ненулевых элементов матрицы  $A$  вычисляет минимальное время, за которое суперкомпьютер сможет вычислить коэффициенты матрицы  $L$ . Так как инженеры IBM заняты «допиливанием» *libastral*, они поручили написать такую программу вам.

MJ12 требуют, чтобы вычисление  $L$  проводилось в точном соответствии с указанными формулами. Однако это может быть неэффективно, и вам удалось убедить их разрешить использовать следующие тождества для оптимизации вычислений:

- $a + b = b + a$ ,  $a - b = -(b - a)$
- $(a + b) + c = a + (b + c)$ ,  $(a - b) + c = a - (b - c)$ ,  $(a - b) - c = a - (b + c)$
- $0 + a = a + 0 = a$ ,  $a - 0 = a$ ,  $0 - a = -a$
- $0 \cdot a = a \cdot 0 = 0$

Тождества (1) — (2), фактически, разрешают складывать слагаемые под знаком корня и в числителе дроби в выражениях в произвольном порядке, а тождества (3) — (4) разрешают учитывать структуру матриц  $L$  и  $A$  и не складывать с тождественным нулем и не умножать на тождественный ноль.

Не забудьте, что суперкомпьютер имеет бесконечно много процессоров. Модель параллельных вычислений на бесконечном числе процессоров подразумевает, что если есть арифметическая операция, которую требуется вычислить, то она начнет выполняться сразу, как только будут вычислены все ее аргументы.

## Формат входного файла

В первой строке входного файла записано два целых числа  $N$  и  $M$  ( $1 \leq N \leq 5 \cdot 10^4, 0 \leq M \leq 2 \cdot 10^5$ ) — размер матрицы  $A$  и количество внедиагональных элементов в нижне-треугольной ее половине. В следующих  $M$  строках содержатся пары чисел  $r_j$  и  $c_j$  — строка и столбец, в котором находится внедиагональный элемент  $j$  ( $1 \leq c_j < r_j \leq N$ ). Гарантируется, что каждый элемент будет указан ровно один раз. Ненулевые элементы на диагонали матрицы во входном файле не задаются, так как предполагается, что вся диагональ отлична от нуля. Последняя строка входного файла содержит четверку целых неотрицательных чисел  $t_a, t_m, t_d$  и  $t_s$  — времена вычисления соответствующих операций. Эти числа не превосходят  $10^9$ .

## Формат выходного файла

В единственную строку выходного файла необходимо вывести одно целое число в Б” минимальное время, требуемое для вычисления разложения Холецкого матрицы, заданной во входном файле. Гарантируется, что количество ненулевых элементов матрицы  $L$ , включая элементы на диагонали, не превзойдет  $3 \times 10^5$ .

## Примеры

input.txt	output.txt
2 0 1 1 1 1	1
3 1 3 2 1 2 3 4	14



## Задача 17. Godzilla

Имя входного файла:	<code>standard input</code>
Имя выходного файла:	<code>standard output</code>
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 mebibytes

Сеть кабельного телевидения в Байтландии состоит из  $n$  узлов и  $m$  **односторонних** соединений. Сигнал передаётся на некоторые узлы (называемые *передающими* узлами). Остальные узлы могут принимать сигнал, если существует (не обязательно прямое) соединение, ведущее из передающего узла в данный узел.

Передающие узлы выбираются так, чтобы, во-первых, обеспечить целостность сети (то есть чтобы все узлы сети могли принимать сигнал), и, во-вторых, количество передающих узлов было бы минимально.

Однажды в стране появился злобный монстр Годзилла, который каждый день приводит в негодность по одному соединению. Поэтому для сохранения целостности сети каждый день набор передающих узлов выбирается заново. Требуется написать программу, которая вычисляла бы наименьшее возможное количество передающих узлов для каждого из дней после появления монстра.

### Формат входного файла

В первой строке входного файла заданы два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 100\,000$ ) — соответственно количество узлов в кабельной сети и количество соединений. В следующих  $m$  строках описываются соединения: каждая из этих строк содержит два целых числа  $a$  и  $b$  ( $1 \leq a, b \leq n, a \neq b$ ) и описывает однонаправленное соединение из узла  $a$  в узел  $b$ . Между любыми двумя узлами в одном направлении существует не более одного соединения. Следующая строка содержит целое число  $k$  ( $1 \leq k \leq m$ ) — количество соединений, которые оказались повреждены. Далее следуют  $k$  строк.  $i$ -я из них содержит номер (в порядке перечисления во входном файле, начиная с 1) соединения, повреждённого в  $i$ -й день. При этом все повреждённые узлы попарно различны.

### Формат выходного файла

Выведите  $k$  строк,  $i$ -я из которых содержит минимальное количество передающих узлов, необходимых для обеспечения целостности сети на  $i$ -й день после появления Годзиллы.

### Примеры

standard input	standard output
3 2	2
1 2	3
2 3	
2	
2	
1	