

## Задача А. Незнайка учится считать

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	128 мегабайт
Количество тестов:	20
Максимальный балл:	100

Незнайка захотел изучить арифметику. Он определился, что начнёт с целых чисел, а потом займётся более интересными вещами — арифметикой с плавающей запятой. Но, увы, даже натуральные числа вызвали у Незнайки массу трудностей. Знайка решил помочь товарищу и подарил ему набор деревянных палочек для счёта. Знайка надеялся, что Незнайке так будет проще освоить основные операции над числами в унарной системе счисления (с основанием 1).

Палочки для счёта изготавливаются местным деревообрабатывающим комбинатом из отходов мебельного производства, поэтому качество их оставляет желать лучшего. Набор состоит из  $N$  деревянных палочек, но вовсе не обязательно, чтобы все палочки в наборе имели одинаковую длину (как повезёт). Тем не менее, в силу особенностей техпроцесса длина каждой палочки выражается целым числом сантиметров.

Незнайка решил подравнять все палочки в своём наборе, сделать так, чтобы все они стали иметь одинаковую длину. Линейки у Незнайки не было, да и пользоваться ей он не умел, хоть пилой и владел неплохо. Поэтому Незнайка решил действовать следующим образом. Всякий раз он выбирал из набора произвольную пару палочек разной длины, прикладывал более короткую палочку параллельно более длинной, совмещая концы, и отпиливал от более длинной палочки кусок, равный по длине короткой палочке. Строго говоря, если Незнайка извлекал из набора палочки длиной  $X$  и  $Y$  сантиметров, где  $X < Y$ , то после отпиливания у него получались три палочки с длинами  $X$ ,  $X$  и  $Y - X$  сантиметров. Затем три новые палочки Незнайка клал обратно в набор. Потом вновь выбирал две палочки разной длины, повторял описанную процедуру, и так пока не оказалось, что все палочки в наборе имеют равные длины.

Теперь Незнайка задался вопросом, сколько же палочек у него получилось. Считать Незнайка пока не умеет, так что вся надежда на вас. Зная длины всех палочек в исходном наборе, определите, сколько палочек получилось у Незнайки.

### Формат входных данных

В первой строке входного файла записано целое число  $N$  ( $1 \leq N \leq 10\,000$ ) — количество палочек в наборе, произведённом местным деревообрабатывающим комбинатом. Далее следуют  $N$  строк, в каждой задана длина одной палочки  $A_i$  (в сантиметрах,  $1 \leq A_i \leq 10\,000$ ).

### Формат выходных данных

Выведите одно число — сколько палочек в итоге получилось у Незнайки.

### Примеры

input.txt	output.txt
3 2 4 6	6
4 4 4 4 4	4

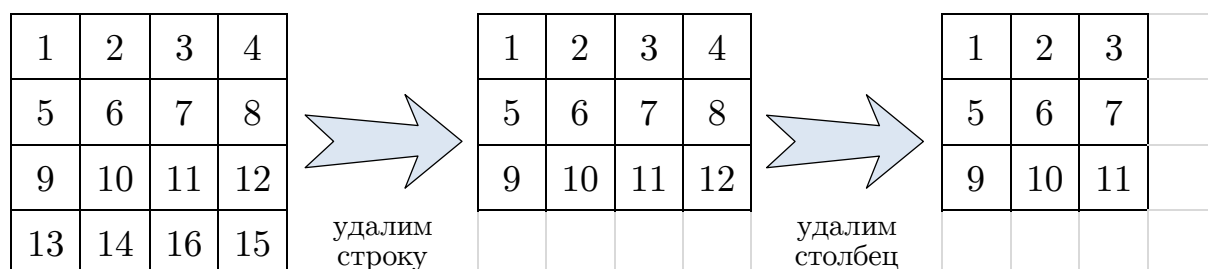
### Частичная оценка

За тесты, где  $N \leq 2$ , вы получите не менее 20 баллов. За все тесты, где  $N$  и  $A_i$ ,  $1 \leq i \leq N$ , не превосходят 100, вы можете получить не менее 40 баллов.

## Задача В. Простая игра

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 128 мегабайт  
Количество тестов: 50  
Максимальный балл: 100

Дед Мазай и заяц играют в очень простую игру. Перед ними — огромная прямоугольная таблица, заполненная целыми числами. Изначально таблица имеет размеры  $N \times N$ . Каждый из игроков во время своего хода может удалить из этой таблицы либо самую нижнюю строку, либо самый правый столбец, но только при условии, что сумма чисел в удаляемых ячейках чётна. Игроки ходят по очереди.



Для примера рассмотрим рисунок. Из исходной таблицы  $4 \times 4$  можно удалить последнюю строку (сумма чисел в ней 58), но нельзя удалить последний столбец, поскольку сумма его элементов равна нечётному числу 39. Допустим, первый игрок удалил строку и таблица теперь имеет размеры  $3 \times 4$ . Второй игрок может удалить или последнюю строку с суммой 42, или последний столбец с суммой 24. На рисунке изображён вариант, когда игрок решил удалить столбец.

Начинает игру либо дед Мазай, либо заяц. Тот, кто не может сделать ход, проигрывает. Если в таблице больше нет чисел, делать ход нельзя. Нетрудно видеть, что с каждым ходом размеры таблицы уменьшаются, поэтому игра заканчивается за конечное число ходов победой одного из игроков.

Требуется написать программу, которая при заданных исходных данных определяет победителя в этой игре. При этом следует учитывать, что игроки играют оптимально.

### Формат входных данных

В первой строке входного файла записано целое число  $N$  ( $1 \leq N \leq 1000$ ). Далее следуют  $N$  строк — это строки таблицы  $N \times N$  в порядке перечисления сверху вниз. В каждой строке записано ровно  $N$  разделённых пробелами целых чисел, каждое число не менее 1 и не более  $10^9$ . Эти числа — элементы соответствующей строки таблицы  $N \times N$ , выписанные слева направо.

### Формат выходных данных

Выведите «First» (без кавычек), если при оптимальной игре выиграет первый, т. е. у игрока, который ходит первым, есть выигрышная стратегия вне зависимости от действий второго игрока. Выведите «Second» (без кавычек) в противном случае.

## Примеры

input.txt	output.txt
4 1 2 3 4 5 6 7 8 9 10 11 12 13 14 16 15	Second
2 2 1 2 2	First
1 928408027	Second

## Частичная оценка

Тесты к задаче объединены в группы по 5 тестов. Если пройдены все тесты группы, то за группу тестов вам будет выставлено 10 баллов, иначе будет выставлено 0 баллов.

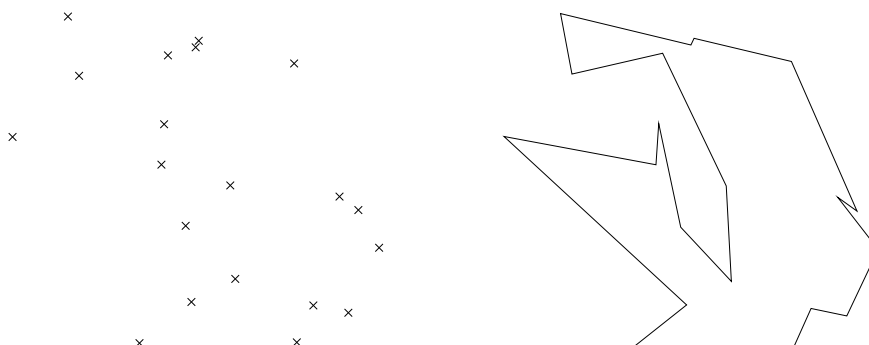
Если ваша программа будет правильно работать на тестах, где  $N \leq 4$ , вы получите не менее 20 баллов. За тесты, где  $N \leq 100$ , можно получить не менее 70 баллов.

## Задача С. Впуклая оболочка

Имя входного файла:       input.txt  
Имя выходного файла:       output.txt  
Ограничение по времени:   2 секунды  
Ограничение по памяти:     128 мегабайт  
Количество тестов:         100  
Максимальный балл:         100

Вам дано множество из  $N$  точек на плоскости. Постройте многоугольник без самопересечений и самокасаний с вершинами в этих точках. Более формально, каждая точка множества должна являться вершиной этого многоугольника, и многоугольник не должен иметь вершин, отличных от точек заданного множества. Никакие два отрезка, являющиеся сторонами многоугольника, не могут иметь общих точек, за исключением пар смежных сторон (они должны иметь ровно одну общую точку — вершину).

Например, для заданного на рисунке слева множества точек (они для наглядности изображены крестиками) справа приведено одно из возможных решений.



Гарантируется, что такой многоугольник существует. Если вариантов построения многоугольника несколько, можно вывести любой. Более того, все  $N$  точек исходного множества попарно различны и не все точки лежат на одной прямой.

### Формат входных данных

В первой строке входного файла записано целое число  $N$  ( $3 \leq N \leq 2000$ ) — количество точек. Во второй и последующих строках даны координаты точек (целые числа  $X_i$  и  $Y_i$  из промежутка от  $-10\,000$  до  $10\,000$  включительно, разделённые пробелом). В  $(i + 1)$ -й строке входного файла даются координаты  $i$ -й точки,  $1 \leq i \leq N$ .

### Формат выходных данных

Выведите в единственной строке перестановку чисел от 1 до  $N$ . Другими словами, выведите  $N$  целых чисел от 1 до  $N$ , каждое число ровно один раз. Эти числа соответствуют номерам точек в порядке их описания во входном файле. Если в соответствии с этой перестановкой соединить пары последовательно идущих точек отрезками прямых, а последнюю соединить с первой, должен получиться многоугольник, который удовлетворяет указанным выше требованиям.

### Примеры

input.txt	output.txt
3 -1 2 7 4 3 -9	3 2 1
4 0 0 1 1 1 0 0 1	1 3 2 4

### Примечание

В первом примере любая из шести перестановок чисел 1, 2, 3 будет засчитана как правильный ответ.

Во втором примере можно вывести любую из следующих восьми перестановок:

1 3 2 4	4 2 3 1
3 2 4 1	2 3 1 4
2 4 1 3	3 1 4 2
4 1 3 2	1 4 2 3

### Частичная оценка

Если ваша программа корректно работает на тестах, в которых  $N \leq 10$ , вы получите не менее 40 баллов.

## Задача D. Подгонка тестов

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 128 мегабайт  
Количество тестов: 1  
Максимальный балл: 100

Пётр Васильевич Колошин в 2011 г. возглавил белорусский филиал одной из крупнейших IT-компаний России «Gold & Silver Software». Тем не менее, в свободное от работы время Пётр Васильевич по-прежнему занимается своим любимым делом — составляет задачи для школьных республиканских олимпиад по информатике. В данный момент наш герой трудится над такой новой и нигде ранее не встречавшейся задачей.

В некоторой стране есть  $N$  городов и  $M$  платных дорог с односторонним движением. Города пронумерованы числами от 1 до  $N$ . Для любой пары городов  $A$  и  $B$  существует не более одной дороги, ведущей из города  $A$  в город  $B$ , и не более одной, ведущей из города  $B$  в город  $A$ . Никакая дорога не ведёт из города в него же. Для каждой дороги задана стоимость проезда по ней — неотрицательное целое число. Требуется вычислить стоимость самого дешёвого пути из города 1 в город  $N$ . Гарантируется, что такой путь существует. Под стоимостью пути понимается сумма стоимостей проезда по всем дорогам, в него входящим.

Пётр Васильевич уже оформил условие задачи, написал авторское решение, проверяющую программу, сгенерировал случайные тесты. Но когда дело дошло до выбора тестов, которые следует поместить в условие в качестве примеров, Пётр Васильевич задумался: «Некрасиво, если в примерах ответы будут случайными числами. Подгоню-ка я стоимости проезда в своих тестах так, чтобы ответ на первый был равен, например, 42 (это же ответ на главный вопрос жизни, вселенной и всего такого), на второй — 2012 (какой у нас год на дворе?), на третий тест ответ сделаю...».

Сейчас Пётр Васильевич занят этой подгонкой. Он взял один из уже сгенерированных тестов, запустил авторское решение на этом тесте и получил ответ  $D$  (можете считать, что авторское решение у программиста такого уровня, как Пётр Васильевич, наверняка правильное). Однако согласно новому замыслу творца ответ на этот тест должен быть  $C$ , причём так оказалось, что  $C < D$ .

Теперь Петру Васильевичу интересно, какое минимальное число правок придётся внести в тест, чтобы ответ на задачу стал равен  $C$ , а не  $D$ . Под одной правкой следует понимать изменение стоимости проезда по одной дороге. Разрешается менять стоимость проезда на любое неотрицательное целое число.

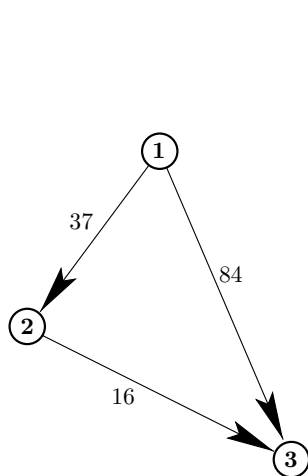


Рис. 1. Иллюстрация к первому примеру

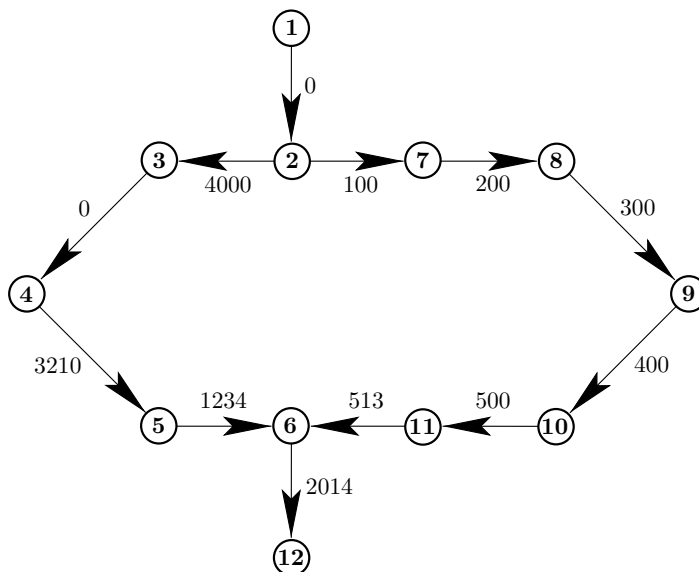


Рис. 2. Иллюстрация ко второму примеру

Помогите Петру Васильевичу. Вам дан тест к задаче и дано число  $C$ . Определите минимальное число правок, необходимое, чтобы подогнать ответ к  $C$ , или сообщите, что это невозможно.

### Формат входных данных

В первой строке входного файла записаны три целых числа  $N$ ,  $M$  и  $C$ , причём  $2 \leq N \leq 100$ ,  $1 \leq M \leq 1000$  и  $0 \leq C \leq 100\,000$ . Это соответственно число городов, число дорог в имеющемся тесте к задаче Петра Васильевича и желаемая стоимость самого дешёвого пути. Числа разделены одиночными пробелами.

Далее следуют  $M$  строк, в каждой строке записаны целые числа  $A_i$ ,  $B_i$  и  $F_i$  — описание  $i$ -й дороги: она идёт из города  $A_i$  в город  $B_i$ ,  $1 \leq A_i, B_i \leq N$ , проезд по ней стоит  $F_i$  единиц стоимости,  $0 \leq F_i \leq 10\,000$ . Числа в этих тройках тоже разделяются одиночными пробелами. Гарантируется, что  $A_i \neq A_j$  или  $B_i \neq B_j$  для всех  $i \neq j$ . Также можно быть уверенным в том, что  $A_i \neq B_i$  для всех  $i$ .

Из города 1 в город  $N$  обязательно есть хотя бы один путь. Более того, путь минимальной стоимости из 1 в  $N$  имеет стоимость  $D$ , строго меньшую, чем та, которой хочет добиться Пётр Васильевич, т. е.  $C < D$ .

### Формат выходных данных

Если решение существует, выведите в единственной строке ровно одно число — минимальное число дорог, цену проезда по которым следует изменить, чтобы самый дешёвый путь из 1 в  $N$  стал стоить  $C$  единиц. Помните, что устанавливать отрицательные стоимости проезда нельзя, равно как нельзя менять саму дорожную сеть (строить новые дороги, закрывать существующие).

Если же возможности сделать ответ равным  $C$  нет, выведите число  $-1$ .

### Примеры

input.txt	output.txt
3 3 42 1 2 37 2 3 16 1 3 84	1
12 12 2012 1 2 0 2 3 4000 3 4 0 4 5 3210 5 6 1234 6 12 2014 2 7 100 7 8 200 8 9 300 9 10 400 10 11 500 11 6 513	2

### Примечание

В первом примере минимальная стоимость пути из города 1 в город 3 равна 53. Чтобы сделать её равной 42, мы можем поменять стоимость проезда по дороге из города 1 в город 3 с 84 на 42. После изменения этот прямой путь станет путём минимальной стоимости.

Рассмотрим второй пример. Здесь минимальная стоимость пути из города 1 в город 12 равна 4027. Чтобы подогнать её к 2012, можно изменить стоимость проезда по дороге из 6 в 12, а также по одной из дорог в правой половине (см. рисунок) дорожной сети. Это можно сделать многими способами, но в любом случае придётся внести не менее двух правок в тест.

### Частичная оценка

Если ваша программа будет корректно работать с тестами, в которых  $M \leq 16$ , вы получите не менее 45 баллов.