

Задача А. Тамада

Имя входного файла: `sadgegrdzelo.in`
Имя выходного файла: `sadgegrdzelo.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

По мнению тех, кто знал видного советского математика Илью Несторовича Векуа, он был интересным собеседником и хорошим тамадой.

Одним из показателей, по которым можно оценивать качество руководства застольем, является коэффициент тостирования. Коэффициент тостирования застолья равен минимуму индивидуальных коэффициентов участников данного застолья. Индивидуальный коэффициент - это количество тостов, объектами которых оказался данный участник на протяжении всего периода данного застолья. Например, если застолье посвящено именинам, то участник, являющийся другом именинника, будет одним из объектов тоста за друзей именинника, но если он еще и одноклассник именинника, то будет объектом тоста за одноклассников именинника, разумеется, если таковой будет провозглашён. Говорят, что в свое время существовала Всесоюзная Федерация Проведения Застольий, располагавшаяся в Сигнахи (Грузия, Кахетия). И говорят, что для получения звания Тамады Всесоюзной Категории необходимо было провести 3 квалификационных застолья со значением коэффициента тостирования не меньшим, чем 2. Та же федерация, оказывается, присуждала звание Тамады Республиканской Категории за проведение 5 квалификационных застолий с коэффициентом тостирования, не меньшим, чем 3. О нормативах для получения звания Тамады Кахетинской Категории молва умалчивает. По данным о тостах, провозглашённых во время заданного застолья, Вам необходимо определить, получил ли тамада этого застолья балл для присуждения ему почётного звания Тамады и если да, то какой категории соответствует этот балл.

Формат входного файла

В первой строке входного файла задаются два целых числа $1 \leq n \leq 25$ и $2 \leq m \leq 100$ — соответственно количество тостов, провозглашённых тамадой на данном застолье и общее количество участников застолья. Затем расположены n строк, причём в i -й из этих строк расположены номера тех участников застолья, которые подпадают под i -ый тост. Участники перенумерованы натуральными числами от 1 до m .

Формат выходного файла

В выходной файл выведите текст NO, если тамада не смог заслужить балл для почётного звания, TUK, если тамада заслужил балл для присвоения звания Тамады Всесоюзной Категории и TRK, если тамада заслужил как балл для присвоения звания Тамады Всесоюзной Категории, так и балл для присвоения звания Тамады Республиканской Категории.

Пример

<code>sadgegrdzelo.in</code>	<code>sadgegrdzelo.out</code>
1 3 2 3	NO
2 3 2 3 1 2	NO
2 3 1 2 3 3 2 1	TUK
5 4 2 3 2 4 1 1 3 2 4 4 3 2 1	TRK

Задача В. Остатки

Имя входного файла: `rest.in`
Имя выходного файла: `rest.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

Грузинские оружейники издавна славились своими кинжалами. Ножны хорошего кинжала бывают инкрустированы самоцветами. И в одно грузинское селение пришёл заказ — изготовить для участников n -й международной морской конференции в Батуми максимальную подарочную партию ножен для кортиков с n разными самоцветами, причём так, чтобы каждые ножны были инкрустированы полосой самоцветов, состоящей из n одинаковых слоёв, на одних ножнах никакие два слоя не были бы сделаны из одного и того же самоцвета, и ни для каких двух ножен из этой партии порядок самоцветов на их рукоятках бы не совпадал. Чтобы произведения искусства не повредились при перевозке, упаковать произведённую партию было предписано в ящики $n \times n$. Для составления отчёта заказчику требуется выяснить, должен ли остаться в конце незаполненный до конца ящик, и если останется, то сколько ячеек в нём будет заполнено.

Учитывая, что подобные заказы приходят мастерам из этого селения достаточно часто (например, следующий заказ будет сделан для участников очередного Кубка И.Н. Векуа), Вас попросили написать программу, которая бы вычисляла необходимый остаток, позволяя мастерам не отвлекаться от высокого искусства на бухгалтерские формальности.

Формат входного файла

В первой строке входного файла записано натуральное число M , не превосходящее 10000. В последующих M строках записаны различные n_i — количества самоцветов в i - заказе, не меньшие двух и не большие 10^6 , по одному на каждой строке.

Формат выходного файла

Для каждого заказа выведите соответствующий остаток (или 0, если незаполненного ящика не останется) по одному в строке, так, что заказу из $i + 1$ -й строки входного файла соответствует остаток в i -я строка выходного файла.

Пример

<code>rest.in</code>	<code>rest.out</code>
2	2
2	6
3	

Задача С. Авиалинии

Имя входного файла: `avia.in`
Имя выходного файла: `avia.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

Руководители команд, прибывших на очередной розыгрыш Кубка Векуа, решили предложить своим правительствам Проект, в результате осуществления которого все города, представленные на Кубке Векуа, будут соединены друг с другом прямыми авиарейсами. Поскольку известно, что в проекте могут быть задействованы 3 авиакомпании, а Оргкомитет Кубка Векуа не хочет создавать ситуацию, в которой какая-то из компаний, используя своё монопольное положение, взвинтит цены или понизит качество услуг, необходимо составить Проект таким образом, чтобы минимизировать максимальное количество городов, между которыми можно было бы путешествовать, пользуясь авиарейсами только одной компании.

Формат входного файла

Во входном файле задано единственное число n — количество городов, упоминаемых в проекте.

Формат выходного файла

Выведите целое число - искомый минимум максимального количества городов, между которыми существует маршрут (возможно, с пересадками), обслуживаемый рейсами только одной авиакомпании.

Пример

<code>avia.in</code>	<code>avia.out</code>
3	2

Задача D. Объяснительные записки

Имя входного файла: `sko.in`
Имя выходного файла: `sko.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

Про первого ректора НГУ академика Векуа ходят легенды. Вот одна из них: В бытность ректором НГУ, академику Векуа часто приходилось иметь дело с объяснительными записками студентов. В те времена ещё не было возможности получить универсальную «отмазку» у вице-преьера, так что объяснительных было много. Через какое-то время ректор заметил, что при перечислении уважительных причин в текстах объяснительных часто встречаются скобки. Будучи математиком, он заинтересовался этим вопросом и провёл исследование. И действительно, оказалось, что тип объяснительной записки полностью определяется взаимным расположением скобок, а общее количество скобок в каждой записке одинаково. После чего было принято решение упорядочить все возможные типы записок студентов следующим образом: из двух записок разного типа раньше идёт та, у которой первая несовпадающая скобка является открывающей и пронумеровать полученный список, начиная с единицы.

При этом естественно, что все записки написаны корректно с точки зрения расстановки скобок, то есть что общее количество открывающих скобок в тексте равно общему количеству закрывающих и что в произвольной подстроке, начало которой совпадает с началом текста, количество закрывающих скобок не превосходит количество открывающих.

Суеверный студент хочет выбрать для своей объяснительной записки тип, который соответствует «счастливному» для него числу k и обратился к вам за помощью — по заданному числу он хочет получить скобочную последовательность, имеющую номер, совпадающий с этим числом.

Формат входного файла

Во входном файле записаны два натуральных числа $n \leq 4000$ — количество открывающих скобок в объяснительной записке и $k \leq 10^{18}$ — «счастливый» для студента номер.

Формат выходного файла

В выходной файл выведите скобочную последовательность, которая среди последовательностей длины $2n$ имеет номер k .

Пример

<code>sko.in</code>	<code>sko.out</code>
3 2	((()())

Задача Е. Дороги

Имя входного файла:	road.in
Имя выходного файла:	road.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайт

Новосибирский Академгородок строился в лесу, причём строители старались оставить как можно больше живых участков леса на территории Академгородка. Так что НГУ, как и многие другие здания в Академгородке, находится посередине небольшого лесного массива. Университет активно развивался, появлялись новые здания, в лесу требовалось прокладывать новые дороги. В бытность ректором НГУ Ильё Несторовичу Векуа пришлось уделять много внимания в том числе и вопросам расширения территории университета.

Однажды ему от имени руководства университета пришлось подписывать смету на оплату проложенных по территории лесного массива дорог. Проложенные дороги состояли из прямолинейных участков и разбивались на несколько маршрутов: если два участка имеют хотя бы одну общую точку, они принадлежат к одному маршруту, и если участок имеет хотя бы одну общую точку с каким-то маршрутом, этот участок считается принадлежащим к данному маршруту. Для вычисления суммы, требующейся для оплаты работ, подрядчикам было необходимо знать как количество маршрутов, так и количество прямолинейных участков, принадлежащих каждому маршруту.

Так как Академгородок активно развивался, то подобного рода договоры были достаточно стандартными, в связи с чем подрядчики предложили Университету в порядке хоздоговорных работ составить программу, по координатам начальных и конечных точек прямолинейных участков генерирующую указанную информацию. Программисты НГУ быстро справились с задачей. Попробуйте справиться с ней и вы.

Формат входного файла

В первой строке входного файла задано количество тестовых примеров. Далее идут тестовые примеры в следующем формате: В первой строке примера задано натуральное число $n \leq 250$ — количество прямолинейных участков В последующих n строках идут целые числа x_1, y_1, x_2, y_2 - координаты начала и конца каждого прямолинейного участка, по модулю не превосходящие 100.

Формат выходного файла

Для каждого тестового примера в отдельной строке выведите информацию по полученным маршрутам в формате $n_1 \times a_1 \ n_2 \times a_2 \dots \ n_r \times a_r$, где n_i - количество маршрутов, состоящих ровно из a_i участков. Список должен быть отсортирован по убыванию a_i .

Пример

road.in	road.out
2	1x2 1x1
3	1x2 2x1
0 0 2 0	
0 1 2 3	
2 1 0 3	
4	
0 1 0 2	
1 0 2 0	
1 1 3 3	
2 2 4 4	

Задача F. Перфокарты

Имя входного файла: `perfo.in`
Имя выходного файла: `perfo.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

Как и все крупные учёные советского времени, Илья Несторович Векуа участвовал в решении задач для оборонной промышленности. Конкретные детали этих задач до сих пор остаются засекреченными. Для примера приведём соответствующим образом изменённую часть одной из задач такого рода.

При разработке узлов одного из шифровальных устройств, работающего на перфокартах (дело было в начале 60-х годов XX века), использовался следующий алгоритм. Сначала в устройство была загружена колода из n перфокарт, при этом i -я карта содержала число i . Затем на каждом шаге отбрасывались k первых карт в колоде, сумма чисел на этих картах фиксировалась на новой перфокарте, которая добавлялась в конец колоды. В случае, когда в колоде оставалось меньше k карт, производилось суммирование чисел, заданных на всех оставшимся картам (при этом на следующем шаге в колоде оставалась одна карта).

При модернизации системы было принято решение заменить данный узел. Вам же предлагается написать программу, которая по заданному n и k вычисляет, какое именно число записывалось на перфокарту на r -м шаге.

Формат входного файла

В первой строке входного файла задано натуральное число m , не большее 100 — количество запросов к программе. В следующих m строках находятся запросы, по одному на каждой строке. Каждый запрос задаётся тремя числами n , k и r ($1 \leq n, k, r \leq 10^9$) — изначальным количеством перфокарт в колоде, количеством суммируемых на каждом шаге перфокарт и номер шага.

Формат выходного файла

Для каждого запроса выведите с новой строки одно число, которое после указанного в запросе шага будет записано на добавляемой перфокарте. Запросу из $i + 1$ -й строки входного файла соответствует число в i -й строке.

Пример

<code>perfo.in</code>	<code>perfo.out</code>
6	3
6 2 1	7
6 2 2	11
6 2 3	10
6 2 4	21
6 2 5	21
6 2 6	

Задача G. Рассадка

Имя входного файла: `seat.in`
Имя выходного файла: `seat.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 64 мегабайт

В зале, в котором участники пишут личный турнир Кубка Векуа, в линию установлено $m + 1$ компьютеров, занумерованных от 0 до m . За 5 минут до начала соревнований стало известно, что делегация из k гуманоидов с Тау Кита, подававшая заявку, всё-таки добралась до Батуми и таким образом Кубок Векуа стал первым межзвёздным соревнованием по программированию.

Тау-китянам надо было предоставить рабочие места. К тому моменту остались свободными $n \geq k$ компьютеров. Для предотвращения несанкционированного обмена информацией между тау-китайцами (тем более возможной, что их язык никто не понимал) было принято решение рассадить инопланетных участников так, чтобы максимизировать наименьшее расстояние (разницу в номерах компьютеров) между двумя участниками с Тау Кита. Так как времени до старта оставалось мало, Вам предложили написать программу, которая бы решала данную задачу.

Формат входного файла

В первой строке входного файла задано количество тестовых случаев $P < 10$. Далее идут P тестовых примеров. Каждый тестовый пример состоит из двух строк. В первой из них заданы два натуральных числа n и k , $1 \leq k \leq n \leq 10^5$ - количество прибывших участников с Тау Кита и количество свободных компьютеров на момент их прибытия, во второй — n различных неотрицательных чисел $c_i \leq 10^9$: номера свободных компьютеров.

Формат выходного файла

Для i -го тестового примера в i -й строке выведите одно целое число — максимальное значение минимального расстояния между двумя участниками с Тау Кита при условии рассадки всех k участников по одному за свободные компьютеры.

Пример

<code>seat.in</code>	<code>seat.out</code>
2	1
4 4	4
1 2 3 4	
5 3	
1 3 6 9 10	