

Что такое юбилейный тур?

Чемпионатам Урала уже двадцать лет! Сегодня мы хотим напомнить вам некоторые интересные задачи предыдущих девятнадцати чемпионатов. Так, сегодняшняя задача В была предложена на Чемпионате Урала в 1997 году, задача С — в 1998 году, ..., задача Т — в 2015 году.

Какой формат проведения тура?

20 лет — возраст, в котором советуют избавляться от лишнего. Мы подошли к делу серьёзно и решили избавиться от тестов к задачам. Теперь в каждой задаче ровно один тест, содержимое которого мы выдадим вам. Тест абсолютно одинаковый для всех участников и не меняется в процессе тура (если об этом не будет сказано особо). Ссылку на архив с входными данными тестов вы можете найти в сообщении проверяющей системы.

Для успешной сдачи задачи вам, как и всегда, нужно написать программу. Однако на юбилейном туре ей достаточно вывести правильный ответ на тот самый единственный тест к задаче. **Обратите внимание, что отправлять на проверку нужно именно эту программу, а не результат её работы!** Программе будут подаваться на вход данные из теста. Это означает, что можно отправить на проверку как программу, решающую задачу в общем виде, так и программу вида `print "Answer"`. Ввод и вывод во всех задачах стандартные, программа не должна работать с файлами.

В условиях задач сегодня нет примеров. Читайте внимательней раздел условия «формат выходных данных», чтобы вывести именно то, что нужно! По этой же причине функция тестирования на примерах из условия сегодня не работает — за любое отправленное неверное решение, если оно успешно скомпилировалось, вы получите стандартный штраф в 20 минут.

Во всём остальном сегодня обычный АСМ-тур. **В каждой задаче установлен Time Limit в 1 секунду и Memory Limit в 64 мегабайта.**

Но ведь большинство из этих задач есть на Тимусе?!

Это правда! Вы можете незаметно от нас выйти в Интернет, скачать из архива задач Тимуса своё решение по задаче и отправить его на проверку. Или, если вы прихватили с собой на ноутбуке архив с АС-решениями с Тимуса, то вы можете даже не нарушать правила соревнований. Никакой дисквалификации за это сегодня не будет, но вы, вероятно, получите меньше удовольствия, чем если бы решали задачи честно.

Вы украли идею у организаторов последнего полуфинала?

Да, но наш юбилейный тур круче!

Задача А. Проба пера

Всё, что нужно в этой задаче, — сложить два числа. Ну, и не забыть выдать результат.

Формат входных данных

Целые числа a и b .

Формат выходных данных

Выведите единственное целое число — сумму чисел a и b .

Задача В. Удав

Некоторое время назад на многих компьютерных системах пользовалась популярностью игра «Удав». Эта игра заключается в следующем. На игровом поле 30×30 клеток расположен удав. Удав представляет собой несамопересекающуюся последовательность клеток, каждая из которых имеет общую сторону со следующей. Начальная клетка последовательности является головой удава, конечная — хвостом. Остальные клетки образуют собой туловище.

Движение удава происходит следующим образом. За каждый ход голова удава может двигаться на свободную клетку поля вперёд, вправо или влево (направление вперёд считается противоположным по отношению к клетке удава, следующей за головой). Одновременно клетка, следующая за головой, занимает место головы, следующая за ней клетка встаёт на её место и т.д. Клетка, которую занимал хвост удава, становится свободной.

Смоделируйте движение удава по игровому полю, выполнив заданную последовательность ходов, и выведите конечное состояние игрового поля.

Формат входных данных

Сначала задаётся игровое поле из 30 строк длиной по 30 символов каждая. Каждый символ соответствует клетке поля. Клетки поля задаются следующими символами:

«. » — пустая клетка;

«@» — голова удава;

«#» — клетка туловища или хвост удава.

Состояние исходного поля не допускает неоднозначной интерпретации расположения удава.

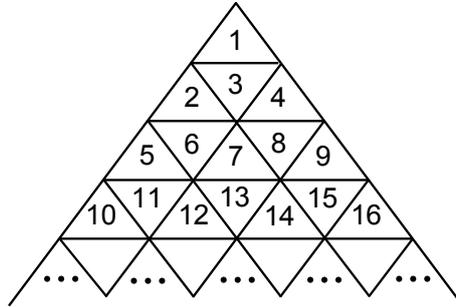
В последней строке дана последовательность направлений движения головы удава. Каждое направление задано одним символом («F» — вперёд, «L» — влево, «R» — вправо).

Формат выходных данных

Выведите состояние игрового поля после завершения движения удава. Гарантируется, что заданное движение удава не приводит к его столкновению с собственным туловищем или хвостом или к выходу за пределы поля. Конечное состояние игрового поля должно иметь тот же формат, что поле во входных данных.

Задача С. Дельта-волна

На треугольном поле, устроенном так, как показано на рисунке, клетки пронумерованы последовательными целыми числами от единицы до бесконечности.



Путешественнику требуется пройти из клетки с номером m в клетку с номером n . Путешественник может попадать в соседние клетки только через ребра треугольников (не через вершины). Количество ребер, которое ему нужно будет пересечь в пути, называется длиной маршрута.

Напишите программу, которая вычисляет длину кратчайшего маршрута для заданных клеток m и n .

Формат входных данных

На входе содержатся целые числа m и n , записанные через пробел.

Формат выходных данных

Выведите длину кратчайшего маршрута.

Задача D. Кратная сумма

Дано множество из n положительных целых чисел. Ваша задача — выбрать из этого множества несколько чисел так, чтобы сумма выбранных вами чисел была кратна числу n (то есть была представима в виде $n \cdot k$ для некоторого целого k).

Формат входных данных

В первой строке записано целое число n . В каждой из следующих n строк записано по одному числу из заданного множества.

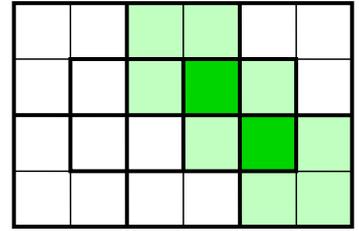
Формат выходных данных

Если задача имеет решение, выведите в первой строке размер искомого подмножества — целое число m в пределах от 1 до n — и далее в m строках числа, составляющие подмножество (сами числа, а не их номера в исходном множестве). Числа можно выводить в произвольном порядке. Если существует несколько подходящих подмножеств, можно вывести любое из них.

Если задача не имеет решения, в единственной строке выведите число 0.

Задача E. Пирамида

На прямоугольной клетчатой площадке размером $m \times n$ (m и n чётные) была построена пирамида из стеклянных плиток размером 2×2 . Нижний слой имеет размер $m \times n$ и состоит из плиток, полностью заполняющих площадку. Второй слой имеет размер $(m - 2) \times (n - 2)$ и расположен так, что вершины плиток, образующих его, лежат в центрах плиток предыдущего слоя. Аналогично строится третий слой и т.д., пока не получится слой из одного ряда плиток (в частности, при $m = n$ он состоит из одной плитки).



Известно, что часть плиток, использованных при строительстве, была бесцветной, а часть имела зелёный оттенок. Взглянув на пирамиду сверху, можно определить сколько зелёных плиток лежит над каждой клеткой площадки.

Требуется по заданному виду сверху определить расположение бесцветных и зелёных плиток в каждом слое, если известно, что ни одна зелёная плитка не лежит точно над бесцветной.

Формат входных данных

Первая строка содержит целые числа m и n — размеры площадки. Далее следуют m строк по n неотрицательных чисел в каждой. Каждое число показывает количество зелёных плиток, лежащих над соответствующей клеткой. Гарантируется, что входные данные задают некоторое возможное расположение плиток в пирамиде.

Формат выходных данных

Выведите описание слоёв — от самого нижнего до самого верхнего. Для вывода информации об i -м слое используется k_i строк по l_i символов в каждой ($k_1 = m/2$; $l_1 = n/2$; $k_i = k_{i-1} - 1$; $l_i = l_{i-1} - 1$). Символ «G» обозначает плитку из зелёного стекла, «W» — плитку из бесцветного стекла. Каждый последующий слой отделяется от предыдущего пустой строкой. Если существует несколько решений, можно вывести любое из них.

Задача F. Радиомаяки

На плоскости расположено n радиомаяков. Точное положение маяков неизвестно, но известно, что их координаты — целые числа в пределах от 1 до 200. Каждый радиомаяк излучает свой уникальный сигнал, по которому его можно легко отличить от прочих.

В нескольких различных контрольных точках, координаты которых известны, были проведены измерения. В результате этих измерений стали известны расстояния от каждой из контрольных точек до некоторых источников сигнала. Расстояние между точками A и B следует считать равным $\max(|A_x - B_x|, |A_y - B_y|)$.

Необходимо по координатам контрольных точек и результатам измерений определить положение маяков на поле, если это возможно.

Формат входных данных

В первой строке дано целое число m — количество контрольных точек. Далее следуют m строк, каждая из которых содержит информацию, полученную в одной из контрольных точек, в формате: $X_i, Y_i: n_1-R_1 [, n_2-R_2] [, \dots]$ где X_i, Y_i — координаты контрольных точек, R_k — расстояние от i -й контрольной точки до маяка с номером n_k .

Гарантируется, что измерения непротиворечивы, т.е. положение любого маяка существует, но может быть неоднозначным.

Формат выходных данных

Выведите n строк, описывающих положение маяков. Если существует ровно одно положение (x_k, y_k) k -го маяка, удовлетворяющее результатам измерений и такое, что $1 \leq x_k, y_k \leq 200$, выведите в k -й строке « $k:x_k, y_k$ ». В противном случае выведите в k -й строке « $k:UNKNOWN$ ». Строки следует выводить в порядке возрастания k .

Задача G. Pentium vs ENIAC

Вычисление 2000 знаков числа π на машине ENIAC в 1949 году потребовало 70 часов (не считая программирования!). Современные компьютеры (и программисты) могут найти 2000 знаков числа π гораздо быстрее.

Для вычисления можно использовать ряд

$$\pi/4 = 1 - 1/3 + 1/5 - 1/7 + 1/9 - \dots$$

Но этот ряд ужасно медленно сходится. Гораздо лучше ряд для арктангенса x при $|x| < 1$

$$\operatorname{arctg} x = x - x^3/3 + x^5/5 - x^7/7 + \dots$$

Объединяя его с формулой для сложения тангенса

$$\operatorname{tg}(a + b) = (\operatorname{tg} a + \operatorname{tg} b)/(1 - \operatorname{tg} a \cdot \operatorname{tg} b)$$

и выбирая a и b так, чтобы $\operatorname{tg}(a + b) = 1 = \operatorname{tg} \pi/4$.

На практике обычно используют следующие формулы:

$$\pi = 16 \operatorname{arctg}(1/5) - 4 \operatorname{arctg}(1/239);$$

$$\pi = 32 \operatorname{arctg}(1/10) - 16 \operatorname{arctg}(1/515) - 4 \operatorname{arctg}(1/239);$$

$$\pi = 12 \operatorname{arctg}(1/4) + 4 \operatorname{arctg}(1/20) + 4 \operatorname{arctg}(1/1985).$$

Во всех формулах необходимо вычислять $\operatorname{arctg}(1/k)$, где $k \geq 2$. Напишите программу, которая выполняет это вычисление.

Формат входных данных

В единственной строке записано целое число k .

Формат выходных данных

Выведите значение $\operatorname{arctg}(1/k)$. Ваш ответ должен отличаться от верного не более чем на 10^{-2000} .

Задача Н. Кодовый замок

Однажды конструктор Трурль заглянул в гости к Клапауцию, и тот похвастался новым кодовым замком, стоящим на двери секретной лаборатории.

— Вот, смотри — на замке несколько кнопок с буквами. Можно нажимать кнопки сколько угодно — дверь не откроется, пока k последних нажатых букв не совпадут с правильной комбинацией. Буквы в этой комбинации не должны повторяться, так как в замок встроены специальные защитные механизмы. Если среди любых k последних введенных букв есть хотя бы две одинаковые, то на мой пейджер поступает сообщение, что кто-то пытается открыть дверь, подбирая код.

Через несколько дней, когда Клапауций отправился на собрание Артели Свободных Механиков, Трурль проник в его дом, чтобы ознакомиться с новыми изобретениями Клапауция в его секретной лаборатории. Чтобы открыть кодовый замок, нужно было всего лишь перебрать все комбинации, не вызвав тревоги. И следовало бы поторопиться, так как Клапауций мог явиться с собрания в любой момент.

Напишите программу, которая по количеству букв в комбинации для открытия кодового замка и количеству кнопок на замке определит минимальную последовательность нажатий кнопок замка, которая содержит все возможные комбинации и не вызывает тревоги.

Формат входных данных

В единственной строке записаны целые числа k и n — количество букв в комбинации для открытия кодового замка и количество кнопок на замке. Кнопки обозначаются последовательными прописными латинскими буквами, начиная с «A».

Формат выходных данных

Выведите минимальную последовательность нажатий кнопок замка, содержащую все возможные комбинации. Если существует несколько вариантов, выведите любой из них. Если перебрать все комбинации без тревоги не удастся, выведите сообщение «IMPOSSIBLE».

Задача I. Архиватор

Как правило, жюри олимпиад по программированию норовит дать к каждой задаче «сказку», цели которой — наметить мнимую связь задачи с реальностью, а также замутить воду вокруг сути задачи, особенно если условие задачи кажется слишком простым для понимания. Однако в данной задаче сказки не будет — во-первых, потому что условие является достаточно непростым и необычным, а во-вторых, потому что сама задача — о краткости.

Пусть дан некоторый текст. Архивом этого текста называется текст, удовлетворяющий следующим требованиям:

1. **Архив является программой на Pascal или C++.**
2. Первой строкой архива является строка «{PAS}» или строка «//CPP».
3. Если архив скомпилировать и выполнить, то программа выведет текст, в точности совпадающий с исходным. **Для компиляции архива жюри будет использовать компиляторы FreePascal 2.0 и Visual C++ 2010!**
4. Размер архива строго меньше размера исходного текста.

Требуется написать программу, которая по заданному тексту строит его архив. При проверке решения архив компилируется и исполняется с использованием тех же ограничений, параметров и условий, что и решения участников. Решение задачи не обязано быть написано на том же языке, что и архив. При проверке жюри определяет язык архива по его первой строке («{PAS}» — Pascal, «//CPP» — C++).

Формат входных данных

Текст, содержащий большие и маленькие латинские буквы, знаки препинания, пробелы, переводы строк.

Формат выходных данных

Архив исходного текста.

Задача J. Полутона

Иван Пяточников является большим любителем электронной техники. Среди друзей он пользуется большим авторитетом. Ещё бы, ведь он чинит практически любую железку, которую к нему принесут. Он разберётся с любой неправильно настроенной программой. Исправит все ошибки в любом коде.

Но вот сегодня у него день рождения, и ребята собрались сделать для него обалденный подарок — бубен. Да, конечно же бубен: ни один настоящий компьютерщик не может обойтись без столь необходимой в жизни вещи, как бубен.

Но есть одна проблема. Бубен не будет работать, если на нём не запечатлеть изображение владельца. Теперь Ванины друзья в унынии — у них есть только матричный принтер и Ванина чёрно-белая фотография. Вот только принтер умеет печатать картинки, все пиксели которых либо чёрные, либо белые, а фотография содержит все градации серого цвета. И ребята боятся, что картинка не выйдет достаточно похожей.

Помогите Ваниным друзьям сделать настоящий подарок.

Формат входных данных

В первой строке находятся целые числа n и m — размеры изображения. Далее следует n строк по m чисел от 0 до 255 в каждой — само изображение.

Формат выходных данных

Выведите n строк по m чисел в каждой — нужное чёрно-белое изображение. Все числа должны быть равны 0 или 255. При этом для того, чтобы чёрно-белое изображение было похоже на оригинал, для каждого пикселя оригинала (i, j) :

$$\left| \frac{\sum_{\substack{|k-i| \leq 10, 1 \leq k \leq n \\ |l-j| \leq 10, 1 \leq l \leq m}} (B[k, l] - A[k, l])}{\sum_{\substack{|k-i| \leq 10, 1 \leq k \leq n \\ |l-j| \leq 10, 1 \leq l \leq m}} 1} \right| \leq 20,$$

если яркость пикселя на оригинальном изображении обозначить через $A[i, j]$, а на чёрно-белом $B[i, j]$.

Задача К. Дымолётный порошок

Колдуны и ведьмы могут перемещаться с помощью дымолётного порошка. Щепотки этого порошка достаточно, чтобы перенести колдуна от одного камина к любому другому камину, подключенному к дымолётной сети.

В секретной комнате в министерстве магии есть заколдованный сундук, в котором хранится неприкосновенный запас дымолётного порошка на случай экстренной эвакуации. Комната надежно защищена от грабителей, и открыть этот сундук может только министр магии. Чтобы сберечь порошок от мышей, сундук висит на волшебных цепях.

Новый министр магии заметил, что прежний министр, хоть и тщательно охранял сундук заклинаниями, не слишком заботился о правильных условиях хранения, и сундук элементарно разохся. Доски немного разошлись, и в дне сундука образовалась тонкая прямая щель, через которую часть порошка могла высыпаться и быть использована мышами для каких-то своих, загадочных целей.

Помогите министру подсчитать, хватит ли оставшегося порошка для эвакуации всех сотрудников.

Длина сундука — 1 метр, ширина и высота сундука — 50 см. Порошок насыпали в сундук через специальную маленькую дырочку в центре крышки. Из насыпанного порошка в сундуке образовался правильный конус с естественным углом осыпания. Если основание конуса пересекается с щелью, то порошок высыпается так, что вдоль щели образуются плоские склоны с тем же углом осыпания.

Формат входных данных

В первой строке записано целое число r — радиус основания исходного конуса. Во второй строке записано целое число d — расстояние от центра этой окружности до щели. В третьей строке записано целое число h — начальная высота конуса. Все величины указаны в сантиметрах.

Формат выходных данных

Выведите объём оставшегося в сундуке дымолётного порошка в кубических сантиметрах. Ответ следует выводить с абсолютной или относительной точностью не менее 10^{-9} .

Задача L. Турнир по сумо

В Токио проходит турнир по сумо, в котором участвуют 2^n спортсменов. В каждом поединке обязательно есть победитель, а проигравший выбывает из турнира. Таким образом, чтобы определить победителя турнира, необходимо провести n раундов.

Организаторы хотят, чтобы в как можно большем числе раундов все поединки прошли между сумоистами из разных префектур Японии. Для этого они могут подтасовать результаты жеребьёвки произвольным образом.

Формат входных данных

В первой строке записано целое число n . В каждой из следующих 2^n строк записаны имя сумоиста и префектура, которую он представляет.

Формат выходных данных

Выведите максимальное количество раундов, в которых сумоисты из одной префектуры гарантированно не встретятся (то есть найдите наибольшее возможное число k такое, что при какой-то начальной расстановке соперников, независимо от исхода поединков, по крайней мере в k раундах все поединки будет проведены между сумоистами из разных префектур).

Задача М. Трамвайная плитка

В Екатеринбурге активно развивается общественный транспорт. В ближайшее время городские власти планируют отремонтировать около 20 км трамвайных путей по новой технологии с применением специально разработанной плитки. Внедрение нового способа ремонта межрельсового пространства началось минувшим летом, и первые результаты видны уже сейчас.

Так, бригада Равшана и Джамшуда оснастила плиткой почти 2 перекрёстка в центре города. Такая поразительная скорость работы объясняется новой хитростью, придуманной их начальником. Перед началом ремонтных работ он сам размещает между рельсами несколько плиток так, чтобы можно было продолжить выкладывать их только одним способом. В результате экономится время на проектирование плиточных узоров, которое занимает большую часть рабочего дня у других бригад.

Каждая плитка имеет размер 1×2 . Требуется замостить плиткой прямоугольную область размером $n \times m$. Определите минимальное число плиток, которое должен выложить начальник, чтобы итоговый узор (то есть полное замощение прямоугольника плитками) определялся однозначно. Два узора считаются одинаковыми, если расположение всех плиток в них совпадает.

Формат входных данных

Единственная строка содержит целые числа n и m .

Формат выходных данных

Выведите в первой строке минимальное число плиток, которое должен выложить начальник. Следующие n строк должны содержать по m символов, описывающих одно из возможных начальных расположений. Символ «1» обозначает клетку, занятую плиткой, символ «0» — пустое место.

Задача N. Борис, ты не прав!

Недавно Борис придумал четвёртый признак равенства треугольников.

Теорема. Треугольники $\triangle A_1B_1C_1$ и $\triangle A_2B_2C_2$ равны, если две стороны и угол, лежащий напротив одной из них, в одном треугольнике равны соответствующим сторонам и углу в другом треугольнике:

- $A_1B_1 = A_2B_2$,
- $B_1C_1 = B_2C_2$,
- $\angle B_1A_1C_1 = \angle B_2A_2C_2$.

Покажите Борису, что он не прав. Пусть дан треугольник $\triangle A_1B_1C_1$, постройте такой треугольник $\triangle A_2B_2C_2$, что по теореме Бориса он равен данному, но на самом деле это не так.

Формат входных данных

В трёх строках записаны координаты точек A_1 , B_1 и C_1 .

Формат выходных данных

В первой строке выведите «YES», если теорема Бориса работает для данного треугольника. В противном случае, когда существует треугольник $\triangle A_2B_2C_2$, равный данному согласно теореме, но не равный ему в действительности, выведите «NO», а в следующих трёх строках выведите координаты точек A_2 , B_2 и C_2 с максимальной точностью. Координаты не должны превышать по модулю 1000. Треугольник должен быть невырожденным.

Задача О. Старая уральская легенда

По легенде, в священном индийском городе Варанаси, в храме, отмечающем центр мира, Брахма установил три алмазных стержня и возложил на них 64 золотых диска. Жрецы храма без устали перекладывают диски со стержня на стержень. Считается, что мир погибнет, как только они закончат свою работу.

На Урале известна другая легенда. Говорят, что на границе Европы и Азии, там, где некогда рос мистический каменный цветок, в лесах затерян монастырь. Живущие там монахи обречены выписывать на пергаменте целые положительные числа до тех пор, пока не начнётся Страшный суд. Никто не знает, когда и как они начали свою работу. Среди монахов живёт предание о том, что после того, как монастырь был построен, к его первому настоятелю во сне явился архангел Гавриил. Он указал камень на склонах Уральских гор, на котором была высечена строка из цифр. Гавриил повелел выписывать все числа, начиная с наименьшего числа, не входящего в эту строку в качестве подстроки. Если это предание истинно, то какое число монахи выписали первым?

Формат входных данных

Дана непустая строка из десятичных цифр, высеченная на камне.

Формат выходных данных

Выведите целое положительное число, которое монахи выписали первым.

Задача Р. Число Исенбаева

Владислав Исенбаев — двукратный чемпион Урала по программированию, вице-чемпион TopCoder Open 2009, абсолютный чемпион ACM ICPC 2009. За то время, которое вы потратите на чтение этого условия, Владислав уже решил бы одну задачу. А может, и две. . .

Поскольку Владислав Исенбаев — выпускник СУНЦ УрГУ, неудивительно, что многие из бывших и действующих олимпиадников УрГУ/УрФУ знакомы с ним уже много лет. Некоторые из них с гордостью заявляют, что играли с Владиславом в одной команде. Или играли в команде с бывшими одноклассниками Владислава. . .

Определим *число Исенбаева* следующим образом. У самого Владислава это число равняется нулю. У тех, кто играл с ним в одной команде, оно равняется единице. У тех, кто играл вместе с одноклассниками Владислава, но не играл с ним самим, это число равняется двум, и так далее. Помогите автоматизировать процесс вычисления чисел Исенбаева, чтобы каждый олимпиадник мог знать, насколько близок он к чемпиону ACM ICPC.

Формат входных данных

В первой строке записано целое число n — количество команд. В каждой из следующих n строк записаны составы этих команд в виде фамилий трёх участников. Фамилия Владислава — «Isenbaev».

Формат выходных данных

Для каждого участника, представленного во входных данных, выведите в отдельной строке через пробел его фамилию и число Исенбаева. Если это число не определено, выведите вместо него «undefined». Участники должны быть упорядочены по фамилии в лексикографическом порядке.

Задача Q. Уроки прошлого

В середине 23-го века инопланетные твари, выйдя из гиперпространственного прыжка, нанесли внезапный удар по Двадцать пятой колонизационной экспедиции в тот момент, когда все её участники собрались на концерте одной из самых популярных певиц во всей Галактике. Тогда в общей панике и давке погибла почти треть всех участников экспедиции.

С тех пор контроль над гиперпространством стал одной из самых приоритетных задач сил обороны. Автоматические дроны с датчиками уровня колебаний гиперпространственного поля были отправлены в самые далёкие уголки обитаемой части Галактики. Однако сбор данных с них стал достаточно сложной технической задачей. Загвоздка в том, что для передачи данных без искажений на подобные расстояния требуется огромное количество энергии. И даже использование промежуточных ретрансмиттеров не сильно улучшает ситуацию. Но инженерам удалось найти достаточно изящное решение. Они предложили сделать так, чтобы ретрансмиттер передавал не само полученное им значение, а модуль его разности с некоторым числом, прошитым в памяти ретрансмиттера. Эти числа было решено подобрать так, чтобы в случае нормального уровня колебаний, зафиксированного датчиком, последний в цепочке ретрансмиттер передавал число, близкое к нулю. Соответственно, аномальным всплеском должна была считаться любая ситуация, когда от последнего ретрансмиттера пришло число, строго большее единицы. Специалистов по безопасности тут же заинтересовал вопрос, насколько надёжна система из k ретрансмиттеров и насколько разные показания датчика будут восприняты ею как спокойное состояние гиперпространственного поля. Исследовать этот вопрос поручили вам.

Формат входных данных

В первой строке записано целое число k — общее количество ретрансмиттеров. Во второй строке следуют целые числа a_1, \dots, a_k — значения, записанные в памяти ретрансмиттеров, в порядке следования сигнала от датчика поля до приёмника на разведывательном корабле.

Формат выходных данных

В первой строке выведите целое число n — количество отрезков в ответе. В следующих n строках выведите по два целых числа $l_i \leq r_i$ — концы очередного отрезка. Отрезки нужно выводить в порядке возрастания координат и без взаимных пересечений.

Задача R. Тестирование игры

В игре Higgs Pong каждый игрок может настроить графику под производительность своего компьютера, включив или выключив некоторые графические опции. Всего в игре n графических опций; при одновременном включении всех их игра начинает тормозить даже на самых мощных компьютерах.

Когда создание игры подходило к концу, выяснилось, что пиарщик Вася уже разместил на нескольких популярных игровых ресурсах информацию о том, что игра Higgs Pong будет отлично работать на персональном компьютере Эверест. Поразмывлив, разработчики купили один такой компьютер и поручили тестировщику Ивану определить, при каких конфигурациях графики игра будет приемлемо работать на нём.

Иван установил некоторую конфигурацию настроек графики, запустил игру, поиграл в неё некоторое время, после чего записал в блокнот результат тестирования. Далее Иван решил перед каждым следующим тестированием изменять в предыдущей конфигурации ровно одну настройку (включать одну из выключенных опций или выключать одну из включенных). Иван записывал все конфигурации, при которых проводилось тестирование, и ни одна конфигурация не была протестирована им дважды. Через некоторое время Ивану показалось, что он проверил уже достаточное количество конфигураций, и он пошёл рассказать разработчикам о результатах тестирования.

Зная начальную и конечную конфигурации настроек графики, определите, какое максимальное количество конфигураций могло быть протестировано Иваном.

Формат входных данных

В первой строке дано единственное целое число n — количество графических опций. Во второй строке приведена начальная конфигурация графики в виде списка целых чисел $k a_1 a_2 \dots a_k$, где k — количество включенных опций, а a_i — номера этих опций ($0 \leq k \leq n$; $1 \leq a_1 < a_2 < \dots < a_k \leq n$). В третьей строке приведена конечная конфигурация графики в том же формате. Начальная и конечная конфигурации различаются.

Формат выходных данных

В первой строке выведите целое число m — максимальное количество протестированных конфигураций, включая начальную и конечную. В следующих m строках перечислите эти конфигурации в том порядке, в котором Иван устанавливал их. Конфигурации следует выводить в том же формате, в котором во входных данных были даны начальная и конечная конфигурации. Если задача имеет несколько оптимальных решений, можно вывести любое из них. Гарантируется, что хотя бы одно решение существует.

Задача S. Jeerman

Правила немецкого языка позволяют составлять такие интересные слова, как Rindeis etikettierungsüberwachungsaufgabenübertragungsgesetz. Оно содержит 65 букв и означает закон о передаче обязанностей контроля маркировки говядины.

В языке Jeerman пошли ещё дальше. Имеется набор базовых слов. Словом является любая последовательность непробельных символов. Ценность слова выражается количеством различных слов из базового набора, которые входят в это слово как подстрока. Требуется сгенерировать слово заданной длины так, чтобы его ценность была как можно больше.

Формат входных данных

В первой строке записаны целые числа n и k . Далее в k строках записаны различные базовые слова языка, состоящие только из строчных латинских букв.

Формат выходных данных

Выведите слово длины n из строчных латинских букв, обладающее максимальной ценностью. Если существует несколько вариантов с максимальной ценностью, выведите любой из них.

Задача Т. Физкультура

Раз в неделю андроид Вася посещает занятия по физической подготовке. Преподаватель, который проводит эти занятия, считает, что тренировать нужно не только физические навыки, но и умение думать. Поэтому он часто даёт своей группе задания, которые не так просто выполнить.

Вот каким было сегодняшнее задание. Изначально n андроидов стояли в одну шеренгу. Преподаватель выдал им различные номера, записанные десятичными цифрами. Все номера лежали в пределах от 1 до n и соответствовали порядку, в котором стояли андроиды, слева направо. По команде преподавателя ученики должны были перестроиться в шеренгу в другом порядке так, чтобы для любых двух рядом стоящих андроидов выполнялось одно из двух условий:

- сумма цифр в номере левого анроида меньше суммы цифр в номере правого;
- суммы цифр их номеров равны, и номер левого анроида меньше номера правого.

Группа долго выполняла это задание, а Васе оно показалось очень скучным. Ведь ему даже не пришлось сдвинуться с места — он стоял первым в шеренге.

Пока все остальные перестраивались, Вася задался вопросом, сколько ещё андроидов остались на своих местах. Помогите Васе посчитать это.

Формат входных данных

В единственной строке записано целое число n — количество андроидов в группе.

Формат выходных данных

Выведите количество андроидов, оставшихся на своих местах.