

Задача А. Рекламный щит

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

Для рекламы своей новой продукции в Китае одна компания решила разместить на небоскребе рекламный щит. Щит состоит из лампочек, организованных в форме прямоугольной сетки из n строк и m столбцов. В любой момент каждая из лампочек может быть либо включена, либо выключена.

Рекламное сообщение состоит из k иероглифов, которые будут показываться один за другим. Для каждого иероглифа известно, какие лампочки должны быть включены при отображении этого иероглифа. Остальные лампочки должны быть выключены.

Для управления рекламным щитом разрабатывается специальная система. Система может включать и выключать лампочки целыми группами. Все лампочки разбиваются на несколько групп так, что в каждом иероглифе лампочки из одной группы должны быть либо все включены, либо все выключены.

Для оптимизации работы системы управления необходимо разбить лампочки на минимальное возможное число таких групп. Помогите сотрудникам рекламного отдела компании решить эту задачу.

Формат входного файла

В первой строке входного файла заданы числа k , n и m ($1 \leq k, n, m \leq 100$) — количество иероглифов в рекламном сообщении, высота и ширина рекламного щита.

Далее, в kn строках идет описание иероглифов. Каждый из k иероглифов задается n строками по m символов в каждой. Все эти строки состоят только из символов «*» и «.», «*» соответствует включенной лампочке, «.» — выключенной.

Формат выходного файла

Выведите минимальное число групп, на которое можно разбить лампочки.

Пример

<code>billboard.in</code>	<code>billboard.out</code>
<pre>3 2 3 *.. *.. **. *..*.</pre>	<pre>4</pre>

В приведенном примере можно разбить лампочки на группы следующим образом: две лампочки из первого столбца образуют одну группу, две лампочки из последнего столбца — вторую, а каждая из двух оставшихся лампочек образует отдельную группу.

Задача В. Хаотическая перестановка

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

Сегодня Васю заставили убираться в классе. Устав наводить порядок, он решил, что теперь он просто должен в качестве компенсации устроить где-нибудь хаос. И тут ему на глаза попала написанная учителем на доске перестановка чисел от 1 до n . Напомним, что перестановкой чисел от 1 до n называется последовательность из n чисел, в которой каждое из них встречается ровно один раз.

Вася считает, что три подряд идущих элемента находятся в порядке, если они упорядочены либо по возрастанию, либо по убыванию. Он называет перестановку хаотической, если никакая тройка подряд идущих элементов не находится в порядке.

Вася решил изменить перестановку на доске, сделав ее хаотической. Для этого он решил не более n раз поменять местами два соседних элемента в перестановке.

Помогите Васе сделать перестановку хаотической, пока не пришел учитель и не наказал его за то, что он занимается ерундой вместо уборки.

Формат входного файла

Во входном файле задана исходная перестановка, которая написана на доске. Первая строка содержит целое число n — длину перестановки ($3 \leq n \leq 1000$). Вторая строка содержит n различных целых чисел, каждое из которых лежит в диапазоне от 1 до n — саму перестановку.

Формат выходного файла

В первой строке выведите число k — количество операций, которое необходимо сделать Васе. В следующей строке выведите k чисел — саму последовательность операций. Если на очередном шаге надо поменять местами i -й и $i + 1$ -й элементы перестановки, необходимо вывести число i .

Если ответов несколько, вы можете вывести любой. Обратите внимание, что вам не обязательно минимизировать количество операций. Достаточно, чтобы оно было не больше, чем n . Если решения не существует, выведите число -1 .

Примеры

<code>chaotic.in</code>	<code>chaotic.out</code>
5 1 2 3 4 5	3 4 1 2
5 1 2 3 4 5	2 4 2
5 2 3 1 5 4	0

Пояснение

В первом примере, перестановка будет иметь такой вид $\langle 12345 \rangle \rightarrow \langle 12354 \rangle \rightarrow \langle 21354 \rangle \rightarrow \langle 23154 \rangle$. Ответ, предложенный во втором примере, тоже корректен.

В третьем примере перестановка уже хаотическая, и ничего менять не надо.

Задача С. Экзамен

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

Мария Петровна работает преподавателем в университете. В середине семестра она дала своим студентам контрольную работу, в которой было n заданий, упорядоченных по возрастанию сложности.

Мария Петровна знает, что каждый из ее студентов относится к одной из трех категорий:

- *Трудолюбивый* студент решает на контрольной задачи в порядке увеличения сложности, начиная с самой легкой, и не пропускает ни одной задачи, пока не закончится время контрольной.
- *Умный* студент решает на контрольной задачи в порядке уменьшения сложности, начиная с самой сложной, и не пропускает ни одной задачи, также пока не закончится время контрольной.
- *Плохой* студент списывает ровно одну задачу у какого-либо трудолюбивого или умного студента. При этом, два плохих студента не могут списать одну и ту же задачу у одного и того же человека.

Трудолюбивые и умные студенты всегда правильно решают все задачи, которые успевают.

Мария Петровна очень не любит студентов, которые списывают. Поэтому, если она узнает, что кто-то у кого-то списал, то обоим студентам она эту задачу не засчитывает. По итогам контрольной Мария Петровна внесла в ведомость зачтенные задачи для каждого студента и про каждую задачу посчитала, скольким студентам она ее зачла.

Поскольку приближаются экзамены, Мария Петровна решила оценить число плохих студентов в группе, чтобы спланировать время на пересдачи. К сожалению, она потеряла ведомость с зачтенными задачами и сохранила лишь листок, в котором записано для каждой задачи, какому числу студентов эта задача зачтена.

Мария Петровна интересуется, какое минимальное число плохих студентов может быть в группе. Помогите ей это выяснить.

Формат входного файла

Входной файл содержит несколько тестов. В первой строке входного файла содержится целое число T — число тестов. Далее следуют описания тестов.

Описание каждого теста состоит из двух строк. В первой строке описания задано целое число n — количество заданий в контрольной работе. Далее для каждого задания, в порядке возрастания сложности, указано число a_i — количество студентов, которым зачтена соответствующая задача ($0 \leq a_i \leq 10^9$).

Гарантируется, что суммарное число заданий во всех тестах не превосходит 10^5 , а также то, что в контрольной была хотя бы одна задача.

Формат выходного файла

Для каждого теста выведите единственное число — искомое минимальное возможное число плохих студентов.

Пример

exam.in	exam.out
2	0
4	3
3 2 0 1	
3	
1 5 1	

Пояснение

В первом тесте у Марии Петровны могло быть три трудолюбивых студента, два из которых решили по две задачи, а один — только одну, и еще один умный студент, который решил самую сложную задачу (но ничего больше решить не успел).

Во втором тесте у Марии Петровны мог быть один умный студент, который решил две последних задачи, четыре трудолюбивых студента, которые решили по две первые задачи и три плохих студента, которые списали первую задачу, каждый у одного из трудолюбивых студентов. В результате из семи решений первой задачи шесть было аннулировано, зачтенным это задание осталось только у трудолюбивого студента, у которого никто не списал.

Задача D. Лесопосадки

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

Часто при вырубке леса для строительства, например, дорог проводятся компенсационные лесопосадки — равное или большее число деревьев высаживается в другом месте. Однако посаженный таким образом лес отличается от самостоятельно выросшего — при компенсационной посадке деревья обычно сажают в узлах регулярной решетки, что изменяет влияние деревьев друг на друга.

Павел занимается математическим моделированием роста деревьев, высаженных в узлах прямоугольной решетки. Будем считать, что в узлы решетки размером $n \times m$ высажены деревья. Два дерева считаются соседними, если узлы, в которых они растут, являются соседними по вертикали или по горизонтали. У каждого дерева есть своя высота, равная некоторому целому числу метров.

Павел считает, что высота деревьев изменяется с годами по следующему закону:

- если у дерева есть хотя бы один сосед, высота которого ровно на один метр больше высоты самого дерева, то через год высота дерева увеличится ровно на один метр;
- если у дерева нет такого соседа, его высота через год останется прежней.

При моделировании роста деревьев по этим правилам рост всех деревьев остановится в тот момент, когда соседних деревьев с разницей в росте, равной одному метру, не останется. Павел хочет выяснить, сколько лет пройдет до этого момента и какая высота будет в этот момент у каждого дерева.

Помогите ему, напишите программу, которая по заданным начальным высотам деревьев выяснит, через сколько лет рост всех деревьев остановится и какой будет высота каждого дерева в этот момент.

Формат входного файла

В первой строке входного файла находятся два целых числа n и m — размеры участка леса ($1 \leq n, m \leq 100$). Следующие n строк содержат по m натуральных чисел, каждое из которых задает высоту соответствующего дерева. Высота каждого дерева не превышает 100.

Формат выходного файла

В первой строке выходного файла выведите t — число лет, которое пройдет до того момента, когда все деревья перестанут расти. После этого выведите n строк по m чисел — каждое число должно быть равно высоте соответствующего дерева через t лет.

Пример

<code>forest.in</code>	<code>forest.out</code>
3 4	9
1 1 1 2	3 3 3 3
1 5 5 1	3 5 5 3
3 1 1 1	3 3 3 3

Задача Е. Академия Джедаев

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

Чтобы стать джедаем, необходимо овладеть многими теоретическими и практическим навыками. В Академии Джедаев можно обучиться всему необходимому, если, конечно, у тебя есть способности.

Новый студент академии Фил очень способный и обучается индивидуально. Он вправе самостоятельно составлять свое расписание. Фил очень любознателен, поэтому он все свое время посвящает учебе. Фил не учится только в те моменты, когда переходит из одного корпуса академии в другой или идет из общежития в один из корпусов.

Академия состоит из двух корпусов, в одном из которых обучают теоретическим навыкам, а в другом — практическим. Путь из одного корпуса в другой занимает ровно a минут. Изучение любого навыка занимает ровно b минут. В начальный момент Фил находится в общежитии, путь от которого до каждого из корпусов также занимает a минут.

Разумеется, навыки нельзя изучать в произвольном порядке. Например, прежде чем овладеть световым мечом, нужно сначала изучить основы оптики и искусство рукопашного боя. Филу важно учесть это при составлении своего расписания.

Фил хочет как можно скорее стать джедаем, но для этого ему нужно обучиться всем необходимым навыкам. Прежде чем приступить к учебе, он хочет узнать через какое минимальное число минут он сможет изучить все навыки. Помогите ему это выяснить. После окончания учебы Фил сразу же отправится сражаться со злом, возвращаться в общежитие ему не требуется.

Формат входного файла

В первой строке входного файла задано целое число n — количество навыков, которым нужно обучиться в академии ($1 \leq n \leq 10^5$). Все навыки пронумерованы от 1 до n .

В следующих n строках описаны требования для изучения каждого навыка. В начале i -й из этих строк записано число 1 или 2, оно указывает в каком корпусе можно изучить i -й навык. Затем следует число k — количество навыков, необходимых для изучения навыка i . Затем в этой же строке следует k целых чисел — навыки, необходимые для изучения навыка i . Гарантируется, что сумма чисел k по всем навыкам не превосходит 10^5 .

В следующей строке задано два целых числа a — время в минутах на путь из одного корпуса в другой или из общежития до корпуса, и b — время на изучение одного навыка ($1 \leq a, b \leq 10^4$).

Гарантируется, что существует такой порядок изучения всех навыков, при котором каждый раз при изучении очередного навыка все необходимые для него уже изучены.

Формат выходного файла

В единственной строке выходного файла выведите минимальное время в минутах, которое потребуется Филу, чтобы стать джедаем.

Пример

jediacademy.in	jediacademy.out
6	285
1 3 3 4 5	
2 1 4	
2 2 5 6	
1 1 6	
1 0	
1 0	
15 40	

Задача F. Спасти котенка

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

Артур принимает участие в телешоу, в котором участникам необходимо выполнять различные интеллектуальные и физические задания, чтобы зарабатывать очки. В одном из таких заданий Артуру необходимо спасти маленького котенка.

Поле для выполнения задания представляет собой прямоугольник размером $n \times m$ метров, разбитый на квадраты единичной площади. В одном из квадратов исходно находится Артур, в некотором другом квадрате находится котенок. Кроме того, один из квадратов содержит лифт, встав на который вместе с котенком, Артур успешно выполняет задание.

За один шаг Артур может перемещаться на любой квадрат, имеющий общую сторону с тем, на котором он стоит. После этого квадрат, на котором перед этим шагом стоял Артур, исчезает и больше на него вставать нельзя. Таким образом исчезают в том числе и квадрат, на котором исходно стоял Артур, и квадрат с котенком. Цель Артура — дойти до котенка, взять его и затем дойти до лифта. При этом очки за выполнение задания, зависят от числа шагов, которое сделает Артур, поэтому ему необходимо сделать минимальное число шагов.

Выяснив, сколько шагов ему придется сделать, Артур заинтересовался, сколько существует различных способов дойти до котенка, а затем с ним до лифта, сделав в сумме минимальное число шагов. Помогите ему это выяснить. Это число может быть довольно большим, поэтому Артур просит найти его по модулю $10^9 + 7$.

Формат входного файла

Первая строка входного файла содержит два натуральных числа n и m — размеры поля для выполнения задания ($2 \leq n, m \leq 100$).

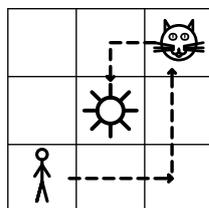
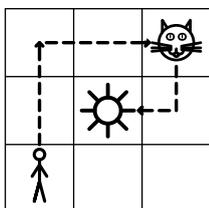
Вторая строка содержит два целых числа x_A и y_A — координаты квадрата, на котором исходно находится Артур ($1 \leq x_A \leq n, 1 \leq y_A \leq m$). Третья строка содержит два целых числа x_K и y_K — координаты квадрата, на котором сидит котенок ($1 \leq x_K \leq n, 1 \leq y_K \leq m$). Четвертая строка содержит два целых числа x_E и y_E — координаты квадрата, на котором находится лифт ($1 \leq x_E \leq n, 1 \leq y_E \leq m$). Эти три квадрата попарно различны.

Формат выходного файла

В единственной строке выходного файла выведите одно число — число способов дойти до котенка и затем до лифта, не наступая на один квадрат два раза, совершив при этом минимальное количество шагов. Число необходимо вывести по модулю $10^9 + 7$.

Пример

<code>kitten.in</code>	<code>kitten.out</code>
3 3 1 1 3 3 2 2	2



Два способа для поля, приведенного в примере.

Задача G. Топот котов

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

В одном городе люди постоянно жаловались на то, что им мешают спать. Каждый день у соответствующих чиновников собиралась большая куча заявлений о слишком громком поведении некоторых людей ночью. С этим необходимо было что-то делать. Тогда на очередном собрании было решено принять закон, который запрещает издавать громкие звуки после одиннадцати часов вечера.

В соответствии с бюрократическими традициями, закон должен содержать расшифровку понятия «громкий звук». В результате обсуждения, ночью решили запретить, например, играть на музыкальных инструментах, передвигать мебель, забивать гвозди.

Когда закон уже собирались принимать, один депутат заметил, что холодильник не является мебелью, и его перемещение не попадает под действие закона. Другие депутаты также начали придумывать дополнительные запреты, которые исходно не попали в закон. В результате были запрещены ночные стоны, скрипы, лай собак и *топот котов*.

За нарушение закона был введен штраф в размере a рублей.

Узнав о законе, Петя решил выяснить, какой штраф может быть наложен на жильцов его дома. Дом, в котором живет Петя, имеет n этажей, на каждом этаже находится по m квартир. Квартиры в доме пронумерованы от 1 до nm . Если на некотором не последнем этаже находится квартира номер x , то непосредственно над ней расположена квартира номер $x + m$.

Известно, что в i -й квартире живет b_i котов. Петя предположил, что жильцы некоторой квартиры будут жаловаться на соседей сверху только в том случае, если коты сверху топают *существенно громче*, чем их собственные. Проведя эксперименты, Петя решил, что p котов топают *существенно громче*, чем q котов, если $p > 2q$.

Выясните, какой суммарный штраф придется заплатить жильцам этого дома, если все, у кого коты в квартире непосредственно сверху топают *существенно громче*, чем их собственные коты, пожалуются на своих соседей сверху и на тех будет наложен штраф.

Формат входного файла

Первая строка входного файла содержит три целых числа n , m , a — количество этажей, количество квартир на каждом этаже и размер штрафа ($1 \leq n \leq 20$, $1 \leq m \leq 10$, $1 \leq a \leq 1000$). В следующей строке содержится nm целых чисел b_1, b_2, \dots, b_{nm} , где b_i — количество котов в i -й квартире ($1 \leq b_i \leq 30$).

Формат выходного файла

Выведите в выходной файл искомый суммарный штраф.

Пример

<code>loudcats.in</code>	<code>loudcats.out</code>
2 3 10	10
3 5 2 4 10 5	

Пояснение

В примере штраф придется заплатить только жильцам 6-й квартиры.

Задача Н. Странный город

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

В одной стране есть странный город. В этом городе есть n перекрестков и m улиц. Каждая улица соединяет ровно два различных перекрестка и по каждой улице можно двигаться в обоих направлениях.

Однажды с целью борьбы с пробками мэр города решил провести дорожную реформу. Он решил запретить автомобильное движение на некоторых улицах и сделать их пешеходными. При этом исследования департамента транспорта показали, что результаты реформы будут оптимальными, если после ее осуществления из каждого перекрестка будет выходить нечетное число улиц, на которых разрешено автомобильное движение.

Подчиненные мэра озадачены. Они никак не могут исполнить приказ мэра. Вам предлагается сделать это за них. Вам задана карта города, требуется решить, какие улицы необходимо сделать пешеходными, а какие — оставить для автомобилей, чтобы в результате реформы из каждого перекрестка выходило нечетное число улиц, по которым разрешено движение автомобилей.

Формат входного файла

В первой строке заданы целые числа n и m ($1 \leq n \leq 2 \cdot 10^5$, $0 \leq m \leq 2 \cdot 10^5$) — количество перекрестков и количество дорог в городе соответственно.

В следующих m строках заданы дороги, каждая дорога описана в отдельной строке. В $(i + 1)$ -й строке описывается дорога с номером i . Каждая дорога задается двумя целыми числами v и u — номерами двух перекрестков, которые эта дорога соединяет ($1 \leq v, u \leq n$).

Гарантируется, что дорога не соединяет перекресток с самим собой.

Гарантируется, что любая пара перекрестков соединена не более чем одной дорогой.

Не гарантируется, что от исходно любого перекрестка можно добраться до любого другого по улицам.

Формат выходного файла

Если удовлетворить условие в приказе мэра невозможно, выведите единственное число -1 в первой строке выходного файла.

Иначе в первой строке выведите число k — количество дорог, которые нужно оставить для автомобильного движения. В следующей строке выведите k различных чисел через пробел — номера этих дорог. Дороги пронумерованы от 1 до m в том порядке, в котором они заданы во входном файле.

Примеры

odd.in	odd.out
2 1	1
1 2	1
4 4	2
1 2	1 3
2 3	
3 4	
4 1	
3 1	-1
1 2	

Задача I. Длинный питон

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

Раз, два, левой, правой,
дважды два — очень просто
измеряются удавы
пятью пять — любого роста

Попугай

После того, как Мартышка и Попугай досконально исследовали длину Удава, им стало очень скучно. Тут Слоненок вспомнил, что в лесу живет еще Питон, которого тоже можно измерять! Друзья сразу же отправились на его поиски.

Питон, как и Удав, тоже целый, поэтому его нельзя мерить половинками. Измерив Питона, Мартышка и Попугай узнали, что в Питоне помещается n целых Попугаев или m целых Мартышек. Обрадованная Мартышка убежала сообщить полученный результат Слоненку. Когда она ушла, Попугая заинтересовал следующий вопрос: а сколько раз он помещается в одной Мартышке?

Так как Мартышка убежала, и измерить ее он не может, Попугай решил попытаться выяснить, сколько целых Попугаев может поместиться в одной Мартышке, используя результаты измерения Питона. По заданным n и m выясните, какое минимальное и максимальное число целых Попугаев может помещаться в одной Мартышке.

Формат входного файла

Во входном файле заданы два целых числа n и m , каждое на своей строке — количество Попугаев и Мартышек в Питоне, соответственно ($1 \leq n, m \leq 10^9$).

Формат выходного файла

В выходной файл выведите два числа — минимальное и максимальное количество целых Попугаев в одной Мартышке.

Пример

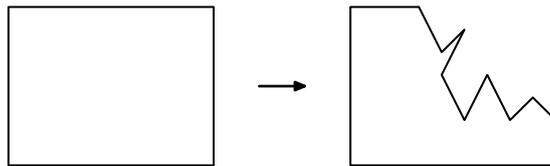
<code>python.in</code>	<code>python.out</code>
38	6
5	7

Задача J. Швабра

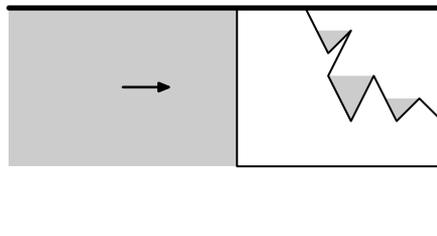
Имя входного файла: rag.in
Имя выходного файла: rag.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Первокурсник Рома приехал в общежитие и, удивившись беспорядку в комнате и толстому слою пыли на полу, начал наводить порядок. Первым делом он решил вымыть пол. Для этого Рома в магазине приобрел инновационную швабру.

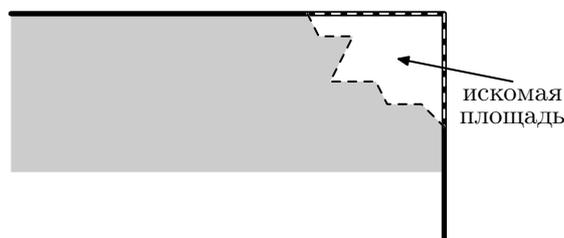
Изначально моющая часть швабры имела идеальную прямоугольную форму, но в процессе перевозки из магазина в общежитие у нее отломился один из углов. Таким образом, теперь она представляет собой многоугольник, граница которого состоит из двух соседних сторон прямоугольника, фрагментов двух оставшихся сторон и ломаной, соединяющей концы этих фрагментов.



Рома живет в большой прямоугольной комнате. Рома провел сломанной шваброй от одной стороны комнаты до другой, не отрывая ее от стены, так что в результате отломанный угол швабры оказался в углу комнаты. При этом часть соответствующей прямоугольной полосы пола в углу осталась невымытой.



Рома считает, что все точки, в которых в какой-то момент находилась моющая часть швабры оказались вымыты. Теперь он решил выяснить, какая часть этой полосы осталась грязной.



Помогите ему вычислить площадь этой части. Можете считать, что размер комнаты, в которой живет Рома, существенно больше размеров моющей части швабры.

Формат входного файла

Первая строка входного файла содержит два целых числа w и h — размеры моющей части швабры до повреждения ($2 \leq w, h \leq 10^5$).

Вторая строка содержит целое число n — число вершин ломаной, соединяющей соседние стороны швабры ($2 \leq n \leq 10^5$). В i -й из следующих n строк заданы два целых числа x_i и y_i ($1 \leq x_i < w$,

$1 \leq y_i < h$, за исключением $y_1 = h, x_n = w$) — координаты i -й вершины ломаной. Ломаная не имеет самопересечений и самокасаний.

Координаты введены таким образом, что стена, вдоль которой Рома провел шваброй, соответствует прямой $y = h$.

Формат выходного файла

В выходной файл выведите площадь невымытой части пола с абсолютной или относительной погрешностью не более 10^{-6} . Это значит, что если правильный ответ a , а вы вывели p , то ваш ответ будет засчитан как правильный, если $\frac{|a-p|}{\max(|a|,1)} \leq 10^{-6}$.

Примеры

rag.in	rag.out
9 7	18.0
9	
3 7	
4 5	
5 6	
4 4	
5 2	
6 4	
7 2	
8 3	
9 2	

Задача К. Вампирские числа

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

Илья увлекается математикой. Недавно он прочитал про *вампи́рские числа*. Они настолько восхитили Илью, что теперь он постоянно придумывает задачи, связанные с этими числами, и пытается их решить.

Число a , десятичная запись которого состоит из n цифр (n чётно), называется вампирским, если его можно представить в виде произведения двух $n/2$ -значных чисел b и c , причем используя все цифры b и c можно записать число a . Каждую цифру при этом разрешается использовать столько раз, сколько раз она суммарно встречается в b и в c . Числа b и c называются *клыками* числа a .

Например, число 6880 — вампирское, так как $6880 = 80 \times 86$, а число 1023 — нет.

Для его новой задачи Илья попросил вас найти k различных вампирских чисел, состоящих из n цифр.

Формат входного файла

В единственной строке входного даны два числа k и n — требуемое количество вампирских чисел и количество цифр в каждом из них соответственно ($1 \leq k \leq 100$, $4 \leq n \leq 100$, n — чётно).

Формат выходного файла

В выходной файл выведите k различных n -значных вампирских чисел в формате $A_i = B_i \times C_i$, где A_i — i -е из найденных вампирских чисел, B_i и C_i — его клыки (между B_i и C_i следует вывести маленькую латинскую букву «x»).

Если ответов несколько, то разрешается вывести любой из них. Гарантируется, что для приведенных во входном файле n и k существует k различных n -значных вампирских чисел.

Примеры

vampire.in	vampire.out
1 6	125460=246x510
3 4	1260=60x21 1395=15x93 1530=30x51