

Задача А. Управляющий совет

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В управляющий совет школы входят родители, учителя и учащиеся школы, причем родителей должно быть не менее одной трети от общего числа членов совета. В настоящий момент в совет входит N человек, из них K родителей. Определите, сколько родителей нужно дополнительно ввести в совет, чтобы их число стало составлять не менее трети от числа членов совета.

Формат входного файла

В первой строке целое число N , а во второй – целое число K ($0 < N \leq 2 \cdot 10^9$, $0 \leq K \leq N$), обозначающие соответственно текущее число членов совета и число родителей в совете.

Формат выходного файла

Выведите единственное число – минимальное число родителей, которое необходимо ввести в совет.

Пример

standard input	standard output
27 7	3

Замечание

В примере совет состоит из 27 человек, из которых родители составляют 7 человек. Если в совет ввести еще 3 родителей, то в совете станет 30 человек, из которых родителей будет 10.

Задача В. Подготовка к олимпиаде

Имя входного файла: standard input
Имя выходного файла: standard output
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Юра решил подготовиться к олимпиаде по информатике. Для этого он должен решить не менее, чем N задач. В первый день Юра решил K задач, а в каждый следующий день решал на одну задачу больше, чем в предыдущий день. Определите, сколько дней уйдет у Юры на подготовку к олимпиаде.

Формат входного файла

В первой строке целое число N , а во второй – целое число K ($1 \leq N, K \leq 2 \cdot 10^9$).

Формат выходного файла

Выведите количество дней, которое потребовалось Юре для решения задач.

Пример

standard input	standard output
10	3
3	

Замечание

В примере в первый день Юра решит 3 задачи, во второй день – 4, в третий день – 5, итого на решение не менее 10 задач у Юры уйдет 3 дня.

Задача С. Следующий палиндром

Имя входного файла: standard input
Имя выходного файла: standard output
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Натуральное число называется палиндромом, если его запись в десятичной системе счисления одинаково читается как слева направо, так и справа налево. По данному натуральному числу N определите следующее за ним натуральное число (то есть наименьшее число, которое превосходит N), являющееся палиндромом.

Формат входного файла

В единственной строке задано натуральное число N ($1 \leq N \leq 10^{100}$).

Формат выходного файла

Выведите наименьшее натуральное число, которое больше N и является палиндромом.

Пример

standard input	standard output
4321	4334

Задача D. Построение

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

На соревнование пришли N участников. У каждого участника имеется некоторый номер на спине, при этом у некоторых участников номера могут совпадать. На открытии соревнования все участники должны построиться в один ряд. Однако сделать это нужно так, чтобы наличие повторов не бросалось в глаза, то есть чтобы никакие два участника с одинаковыми номерами не стояли рядом. Ваша задача – выполнить требуемое построение.

Формат входного файла

В первой строке задано натуральное число N ($1 \leq N \leq 100$). В последующих N строках содержатся номера, которые написаны на спинах у участников. Все номера – натуральные числа от 1 до 3000.

Формат выходного файла

Выведите N строк. Каждая строка должна содержать одно число – номер, написанный на спине у участника, стоящего в ряду на соответствующей позиции. Если существует несколько расстановок с требуемым свойством, можно выводить любую из них. Если таких расстановок не существует, выведите одно число 0.

Примеры

<code>standard input</code>	<code>standard output</code>
4 1005 1005 5 2005	2005 1005 5 1005
4 1005 1005 2005 1005	0

Задача Е. Остров

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

На клетчатой бумаге нарисована карта острова (клетки острова закрашены). При этом остров является клетчато-выпуклой фигурой, то есть каждая горизонтальная или вертикальная линия на карте либо не пересекает остров, либо пересекает его по отрезку (линия пересечения не содержит разрывов). Также остров является связной фигурой, то есть любые две клетки острова соединены путем, каждые две соседние клетки которого имеют общую сторону.

Клетка считается соседней с островом, если она не принадлежит острову, но имеет общую сторону или угол с одной из клеток острова.

Самолет должен облететь вокруг острова по соседним с ним клеткам, не вторгаясь на территорию острова. Программа должна составить маршрут полета самолета. Самолет начинает облет острова в одной из соседних клеток с островом и должен побывать во всех клетках, соседних с островом, ровно один раз. При этом самолет может перемещаться из одной клетки в другую клетку, только если эти клетки имеют общую сторону.

Формат входного файла

В первой строке задано натуральное число N , а во второй – натуральное число M ($1 \leq N, M \leq 100$), определяющие соответственно количество строк и столбцов карты. В последующих N строках содержится по M символов. Каждый символ карты может быть либо символом “.”, что означает клетку, не принадлежащую острову, либо символом “#”, что означает клетку острова. При этом остров не касается края карты.

Формат выходного файла

Выведите координаты клеток карты в порядке их облета самолетом. Каждая строка вывода должна содержать два числа x и y – координаты самолета ($1 \leq x \leq N, 1 \leq y \leq M$), где x – номер строки (сверху вниз), y – номер столбца (слева направо). Самолет должен побывать в каждой соседней с островом клетке ровно один раз. Каждые две клетки, идущие подряд в выводе, должны иметь общую сторону. Можно вывести любой возможный маршрут облета острова.

Пример

standard input	standard output
6	3 5
7	4 5
.....	5 5
.....	6 5
.....	6 4
.###...	6 3
.###...	6 2
.....	6 1
	5 1
	4 1
	3 1
	3 2
	3 3
	3 4

Задача F. Усреднение

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Имеется пять чисел. Необходимо найти значение того из них, которое ближе к их среднему арифметическому.

Формат входного файла

В единственной строке задаются пять натуральных чисел. Каждое из них не превосходит $2 \cdot 10^9$.

Формат выходного файла

Выведите ближайшее к среднему арифметическому число из заданных.

Пример

<code>standard input</code>	<code>standard output</code>
6 2 8 1 10	6

Задача G. Картография

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Карта задана массивом из N строк и M столбцов. На карте обозначены города. Расстоянием между двумя городами назовем количество перемещений между клетками карты по горизонтали или вертикали, необходимое для достижения одного города из другой (так, расстояние между городами с координатами $(1, 1)$ и $(2, 2)$ равно 2). Найдите кратчайшее из расстояний между двумя различными городами.

Формат входного файла

В первой строке задаются два натуральных числа N и M – размеры карты ($1 \leq N, M \leq 100$). В последующих N строках записано по M чисел, определяющих карту – 0, если в этом месте карты нет города, и 1, если есть. Гарантируется, что на карте есть не менее двух городов.

Формат выходного файла

Выведите расстояние и координаты городов, разделенных этим расстоянием. Если вариантов несколько, вывести любой их них.

Пример

standard input	standard output
3 3	1 3 2 3 3
1 0 1	
0 0 0	
0 1 1	

Задача Н. Тройение

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Число назовем тройственным, если сумма любых трех подряд идущих цифр в его десятичной записи делится на 3 без остатка. Найдите количество тройственных K -значных чисел.

Формат входного файла

В единственной строке задается целое число K ($3 \leq K \leq 30$).

Формат выходного файла

Выведите искомое количество.

Пример

<code>standard input</code>	<code>standard output</code>
3	300

Задача I. Окна 1000

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Аркадий является официальным распространителем программного обеспечения “Окна 1000”. Совсем недавно он получил заказ на большое количество копий этого программного продукта, каждая копия размещена на одном CD-диске. На складе уже подготовили N ящиков к отправке. Ящик с номер i содержит A_i копий продукта, при этом его вместимость P_i копий. Однако поступил заказ ровно на K копий продукта. В связи с этим Аркадию придется перед отправкой поехать на склад. За одну минуту может распаковать только один ящик, положить или достать оттуда один диск и запаковать ящик. Так как он очень паникует, то достает (кладет) всегда ровно по одной копии. У Аркадия есть неограниченный запас копий.

Требуется посчитать минимальное время, за которое возможно подготовить заказ к отправке.

Формат входного файла

Первая строка содержит два целых числа N, K ($1 \leq N \leq 100, 1 \leq K \leq 1000$) – количество ящиков, которые подготовили на складе, и количество копий, которое требуется заказчику соответственно. Каждая из последующих N строк содержат по два целых числа A_i, P_i ($1 \leq P_i \leq 1000, 0 \leq A_i \leq P_i$) – количество копий в i -ом ящике и его вместимость соответственно. Ящики нумеруются последовательно, в порядке их ввода, начиная с единицы.

Гарантируется, что суммарная вместимость всех имеющихся в наличии ящиков не менее K .

Формат выходного файла

Первая строка должна содержать одно целое число M – минимальное количество минут, которое потребуется Аркадию для подготовки заказа.

В следующих M строках должны содержаться целые числа X_i , где $|X_i|$ – номер ящика, с которым предстоит возиться Аркадию на i -й по счету минуте. При этом, если X_i меньше нуля, то из ящика достается одна копия программного продукта, иначе в ящик добавляется одна копия программного продукта.

Надо учесть, что в любой момент времени в каждом ящике должно быть неотрицательное количество копий, и количество копий в ящике не должно превышать вместимость ящика. Если решений несколько, то выведете любое из них.

Примеры

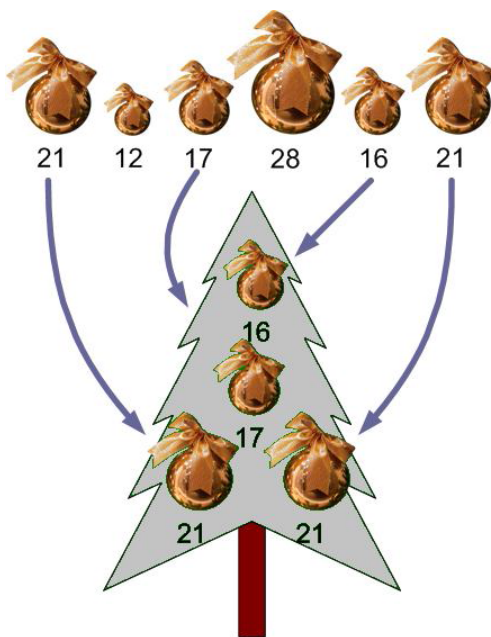
standard input	standard output
4 10 0 1 0 2 0 3 0 4	10 1 2 3 2 3 4 4 3 4 4
4 10 1 1 2 2 3 3 5 5	1 -4
2 1 1 1 1 1	1 -1

Задача J. Новогодняя елка

Имя входного файла: standard input
Имя выходного файла: standard output
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Так повелось, что в Байтландии елки украшаются только новогодними одноцветными шарами. В связи с этим знаменитой компании по производству декоративных украшений “Yellow Toys” было поручено важное и ответственное задание – украшение главной новогодней елки страны.

Сотрудникам компании был составлен строгий перечень имеющихся новогодних игрушек (шаров), согласно которому в наличии имеется N одноцветных шаров, диаметр i -го шара равен D_i миллиметров. Сотрудникам известно, что согласно национальным стандартам главная елка страны должны быть украшена не менее чем K шарами. Также стандарты четко регламентируют понятие показатель некрасивости – равный максимально возможному числу $D'_i - D'_j$, где D' – массив диаметров игрушек, которыми украшена елка. Для приведенного ниже примера показатель некрасивости равен 5.



Ваша задача – помочь сотрудникам компании из имеющихся N игрушек, выбрать M ($M \geq K$) игрушек так, чтобы для набора выбранных M шаров показатель некрасивости был минимальным.

Формат входного файла

Первая строка содержит два целых числа N , K ($2 \leq N \leq 10^5$, $2 \leq K \leq N$). Вторая строка состоит из N целых чисел D_i ($1 \leq D_i \leq 10^9$) – диаметр i -й новогодней игрушки (шара). Числа разделены одиночными пробелами. Игрушки нумеруются последовательно, в порядке их ввода, начиная с единицы.

Формат выходного файла

Первая строка должна содержать одно целое число M ($K \leq M \leq N$) – количество игрушек на елке.

Вторая строка должна описывать набор выбранных для украшения елки игрушек и содержать M целых чисел Z_i ($1 \leq Z_i \leq N$, $Z_i \neq Z_j$, если $i \neq j$) – номера выбранных для украшения игрушек. Если решений несколько, то выведите любое из них.

Примеры

standard input	standard output
3 2 1 3 7	2 1 2
6 4 21 12 17 28 16 21	4 1 5 3 6
8 5 10 10 10 10 20 20 20 20	6 3 5 6 4 7 8

Задача К. Дорожная система

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 Мб

Велика и прекрасна страна Байтландия! Она состоит из N городов, пронумерованных от 1 до N , и M дорог между ними. Каждая дорога связывает между собой два различных города и обеспечивает передвижение между ними. Дорожная система в Байтландии весьма специфична. Все дороги имеют двустороннее движение, и между каждой парой городов проходит не более одной дороги.

С древних времен дорожная система Байтландии удовлетворяет свойству *нечетности*. С самого начала это свойство поддерживалось из религиозных соображений древних байтландцев, а в настоящее время как дань древней традиции, такое же, как нечетное количество цветов в праздничном букете. Сформулируем свойство *emph*нечетности:

Конечную последовательность номеров городов C_1, \dots, C_K ($K \leq 2$) будем называть путем, если для любой соседней пары элементов последовательности C_i, C_{i+1} ($C_i \neq C_{i+1}$ для $1 \leq i < K$) существует дорога между городами с номерами C_i и C_{i+1} . Если $C_1 = C_K$, то такой путь будем называть *замкнутым*. Длину пути C_1, \dots, C_K будем считать равной длине последовательности, то есть равной K . Итак, правило *нечетности* гласит, что *все замкнутые пути в Байтландии имеют нечетную длину, то есть не существует замкнутого пути четной длины*.

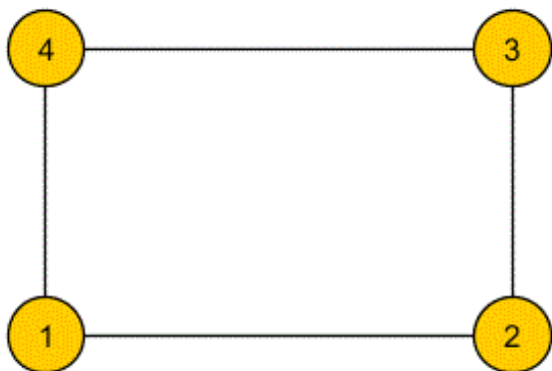


Рисунок 1

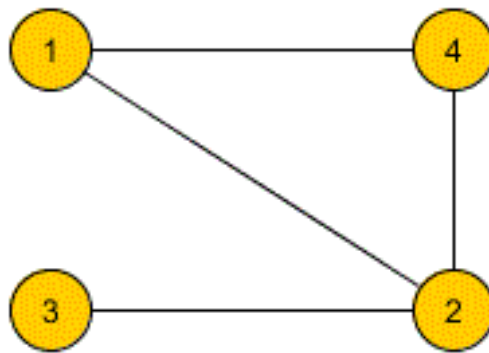


Рисунок 2

Дорожная сеть, изображенная на рисунке 1, обладает свойством нечетности, а дорожная система на рисунке 2 не обладает этим свойством из-за наличия в ней замкнутого пути четной длины 4: 1 2 4 1.

Недавно Министр транспорта Байтландии решил, что текущая дорожная система неэффективна, и необходимо построить несколько новых дорог. Причем новая дорожная система, полученная из старой добавлением некоторого числа дорог, должна обладать свойством нечетности. Все новые дороги должны связывать между собой различные города. Кроме этого, в новой дорожной сети между каждой парой городов должно проходить не более одной дороги.

Для улучшения эффективности дорожной системы было выделено много денег, поэтому Министр Байтландии решил построить как можно больше новых дорог таким образом, чтобы полученная дорожная система удовлетворяла описанным выше условиям. Но эта задача оказалась довольно сложной, и министерство транспорта Байтландии решило обратиться к Вам за консультацией.

Итак, вам дано описание исходной дорожной сети. Вам необходимо найти максимальное число дорог, которое можно добавить к исходной сети, не нарушая вышеописанных свойств.

Формат входного файла

Первая строка входного файла содержит целые числа N ($1 \leq N \leq 10^4$) и M ($0 \leq M \leq 10^5$), разделенные пробелом. Следующие M строк описывают дороги Байтландии. В каждой строке находится описание ровно одной дороги. Каждая дорога описывается двумя целыми числами X и Y ($1 \leq X, Y \leq N$, $X \neq Y$), разделенными пробелом. Эти числа соответствуют номерам городов, связанных дорогой. Города нумеруются последовательно целыми числами от 1 до N . Гарантируется, что заданная дорожная система удовлетворяет свойству нечетности, а также любые два города связаны не более чем одной дорогой.

Формат выходного файла

Выходной файл должен содержать одно целое число – максимальное число дорог, которое можно добавить в исходную дорожную сеть.

Примеры

standard input	standard output
4 4 1 2 2 3 3 4 1 4	0
6 4 1 2 6 5 3 2 4 5	5

Замечание

Дорожная сеть из первого примера приведена на рисунке 1. Исходная дорожная сеть из второго примера представлена на рисунке 3, а ее состояние после добавления в нее новых дорог изображено на рисунке 4.

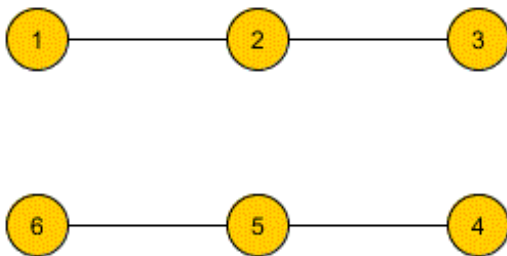


Рисунок 3

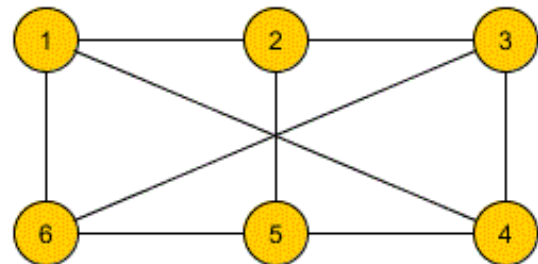


Рисунок 4