

Задача А. Оценки

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

Вася учится в шестом классе. Недавно он решил стать отличником и получить за четверть пятерки по всем предметам. Для этого он решил получить по каждому предмету такое количество пятерок, чтобы пятеркой стала и оценка за четверть. Оценка за четверть выставляется следующим образом: вычисляется среднее арифметическое всех оценок в четверти и округляется до ближайшего целого. В случае неоднозначности округления (например 4.5), округление ведется в пользу ученика, то есть вверх.

Вам даны оценки Васи по одному из предметов. Требуется определить, какое минимальное количество пятерок ему требуется получить, чтобы пятеркой стала оценка за четверть.

Формат входного файла

В первой строке входного файла записано число N — число оценок, уже полученных Васей по этому предмету ($1 \leq N \leq 100$). Во второй строке записано N целых чисел, через пробел — сами оценки. Оценки лежат в пределах от 2 до 5.

Формат выходного файла

Выведите единственное число — требуемое Васе количество пятерок.

Примеры

<code>marks.in</code>	<code>marks.out</code>
1 2	5
5 2 3 4 5 5	7

Задача В. Воздушные шары

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

Несколько недель назад у Васи был день рождения. На этот праздник он получил большое количество воздушных шаров. Вася заметил, что каждый из шаров сдувается. Он заинтересовался в каком порядке шарики сдуются. Для этого он привлек своего папу с его оборудованием. Про каждый шарик известен его исходный объем c_i (единиц объема) и скорость, с которой шарик сдувается, v_i единиц объема в единицу времени.

Формат входного файла

В первой строке входного файла записано число N ($1 \leq N \leq 1000$) — число шариков, полученных Васей на день рождения. Следующие N строк содержат по два числа, $i + 1$ строка входного файла содержит изначальный объем i -ого шарика c_i и скорость сдувания шарика v_i . ($1 \leq c_i, v_i \leq 100000$)

Формат выходного файла

Выходной файл в единственной строке должен содержать порядок, в котором шарики сдуются. Если какие-то шарики сдуются в одно и тоже время, то выведите их в любом порядке.

Примеры

balloons.in	balloons.out
2 3 1 2 1	2 1
4 2 2 3 1 5 3 4 2	1 3 4 2

Задача С. Турнир

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

В Летней Карельской Компьютерной Школе кроме решения задачек бывают и внеучебные мероприятия. Одно из них представляет собой турнир по настольному теннису. Турнир проходит по следующим правилам:

1. Два произвольных игрока играют между собой.
2. Проигравший выбывает из турнира.
3. Если есть человек которому нет пары, он ждет одного из победителей другой пары.
4. Игра заканчивается победой одного из игроков.

На турнир записалось N ЛККШат. Ваша задача — помочь рассчитать количество партий в настольный теннис, необходимых для выявления чемпиона первенства ЛККШ.

Формат входного файла

Входной файл содержит одно число N ($1 \leq N \leq 999999999$).

Формат выходного файла

В выходной файл выведете ответ на задачу.

Примеры

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

Задача D. Lines

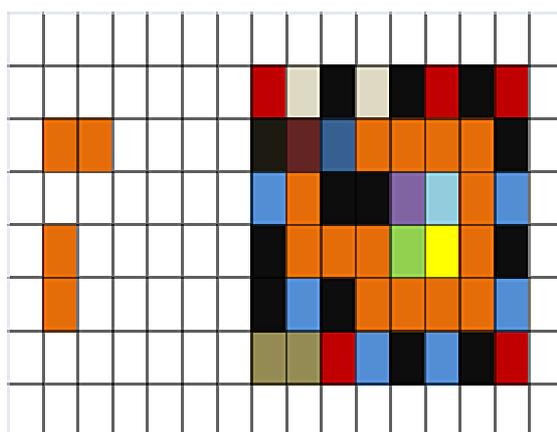
Имя входного файла: `lines.in`
Имя выходного файла: `lines.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Ученики ЛККШа каждый день никак не могут заснуть без их любимой игры *Lines*. Но играют они в неё очень долго и ложатся далеко за полночь. Учителям это сильно не нравится. Поэтому вам поручено написать программу, автоматически играющую за ЛККШат в *Lines*, пока они готовятся ко сну.

Правила ЛККШатского *Lines*:

1. Поле игры представляет собой квадрат $N \times N$. Низом называется нижняя строка. Верхом — верхняя.
2. Изначально на поле расставляются N^2 квадратиков произвольных цветов из диапазона от 1 до K ;
3. Назовем 2 квадратика одного цвета связными, если они стоят в клетках, соседних по горизонтали или вертикали. Назовем набор квадратиков одного цвета связным, если от каждого квадратика из набора до каждого можно прийти, передвигаясь только между связными квадратиками.
4. Ход игры заключается в одновременном уничтожении всех связных одноцветных наборов шариков с количеством шариков в них больше 2. Если под квадратиком нет другого квадратика и он не находится в нижней строке, то квадратик сдвигается на клетку вниз (гравитацию никто не отменял).
5. Игра заканчивается, когда нельзя сделать ход.

Вам дано поле и изначальная расстановка квадратиков. Определите, каким будет поле в конце игры.



*Слева на рисунке изображены примеры связных квадратиков по горизонтали и вертикали.
Справа на рисунке изображен пример одного единственного связного набора квадратиков одного цвета количеством больше двух штук.*

Формат входного файла

В первой строке входного файла записаны через пробел два числа N и K ($1 \leq N, K \leq 100$). Далее следуют N строк по N чисел в каждой — цвета квадратиков. Числа разделены пробелом и не превышают K .

Формат выходного файла

В выходной файл выведите N строк по N чисел, разделенных пробелом — состояние поля в конце игры. Пустые клетки обозначьте 0.

Примеры

lines.in	lines.out
4 7	0 0 0 0
1 1 1 3	0 0 0 0
1 2 2 3	0 2 2 0
4 4 5 3	0 7 5 0
4 7 3 3	

Задача Е. Комплексные числа

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

Комплексные числа — расширение множества вещественных чисел, обычно обозначается \mathbb{C} . Любое комплексное число может быть представлено как формальная сумма $x + iy$, где x и y — вещественные числа, i — мнимая единица, то есть число, удовлетворяющее уравнению $i^2 = -1$. Число x в такой записи комплексного числа называется действительной частью, число y — мнимой частью. Два комплексных числа считаются равными, если у них равны действительные и мнимые части. Сложение и вычитание комплексных чисел осуществляется покомпонентно (то есть складываются/вычитаются отдельно действительные и мнимые части).

Даны два комплексных числа. Посчитайте их частное. Частное двух комплексных чисел тоже является комплексным числом.

Частным чисел $A+iB$ и $C+iD$ называется такое комплексное число $P+iQ$, что верно следующее равенство:

$A+iB = (C+iD)(P+iQ)$. При этом произведение чисел $A+iB$ и $C+iD$ по определению равно $(A+iB)(C+iD) = (AC - BD) + i(AD + BC)$.

Формат входного файла

В первой строке входного файла записаны через пробел два целых числа A и B ($-1000 \leq A, B \leq 1000$). Во второй строке входного файла записаны через пробел два целых числа C и D ($-1000 \leq C, D \leq 1000$). При этом C и D одновременно не равняются 0.

Формат выходного файла

В выходной файл выведите через пробел два вещественных числа P и Q , с шестью знаками после запятой.

Примеры

<code>complex.in</code>	<code>complex.out</code>
5 7 1 1	6.0 1.0
3 4 1 2	2.2 -0.4

Задача F. Строки

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

Васин младший брат Вова играет в игру со строками. Изначально Вова пишет на листе бумаги некоторый набор строк, состоящих из строчных латинских букв и, возможно, знаков доллара `$`. При этом все строки, записанные Вовой на листе, различны. Далее, Вова хочет произвести следующее действие: взять какую-либо из изначальных строк, содержащую внутри один или несколько знаков доллара и заменить какие-то знаки доллара на те из исходных строк, в которых долларов нет. При это какие-то знаки доллара могут остаться неизменными, а некоторые знаки доллара могут быть заменены на одну и ту же исходную строку. Вову интересует, сколько различных таких операций он может произвести. При этом операции считаются различными, даже если в результате получаются одинаковые строки (но при этом, конечно же, производимые Вовой действия должны хоть где-то различаться).

Формат входного файла

В первой строке входного файла записано число $1 \leq N \leq 100$ — количество исходных строк, записанных на листе бумаги. Далее следует N строк, содержащих сами исходные строки, по одной в строке.

Формат выходного файла

Выведите единственное число — сколько операций можно произвести. Так как ответ может получиться очень большим, выведите остаток от его деления на 1000000007.

Примеры

<code>strings.in</code>	<code>strings.out</code>
2 a b	0
3 a\$ \$ b	2
4 \$abacaba\$ preved medved krossavcheg	15

Задача G. Циклы

Имя входного файла: `cycles.in`
Имя выходного файла: `cycles.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Вася нашёл карту Тридевятого царства. На этой карте отмечено N городов, некоторые пары городов соединены дорогами с односторонним движением. На карте все города пронумерованы числами от 1 до N . Вася хочет совершить туристическую поездку. Число x назовём хорошим, если Вася может выйти из какого-либо города, пройти ровно x дорог и вернуться в этот же город. При этом один и тот же город можно посетить несколько раз. Перед началом своего путешествия Вася хочет знать все хорошие числа на промежутке от 1 до K .

Формат входного файла

В первой строке входного файла записаны два числа: количество городов $1 \leq N \leq 100$ и количество дорог $1 \leq M \leq 10000$ тридевятого царства. Каждая из следующих M строк содержит по два числа — A и B . Это означает, что в царстве существует дорога из города A в город B . Последняя строка исходного файла содержит число $1 \leq K \leq 100$.

Формат выходного файла

В выходном файле должна содержаться одна строка, состоящая ровно из K чисел. i -тое число должно равняться 1, если i — хорошее число, в противном случае i -ое число должно равняться 0.

Примеры

<code>cycles.in</code>	<code>cycles.out</code>
3 3 1 2 2 3 3 1 6	0 0 1 0 0 1
2 1 1 1 5	1 1 1 1 1
3 4 1 2 2 1 2 3 3 1 10	0 1 1 1 1 1 1 1 1 1

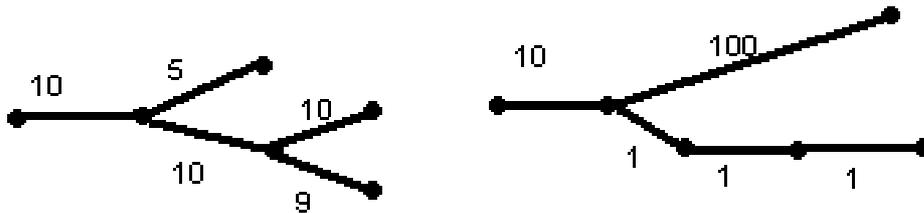
Задача Н. Дерево

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

Вова срубил у себя на даче дерево. Теперь ему нужно убрать это дерево в сарай. Однако дерево целиком в сарай не помещается, посему Вова решил распилить его еще в нескольких местах.

Дерево представляет собой набор сучков и веток. Каждая ветка соединяет ровно два сучка. При этом каждая ветка характеризуется длиной l_i и толщиной s_i . Вова срубил именно дерево, а не марсианский кактус, поэтому от каждого сучка до каждого другого сучка можно добраться единственным образом, двигаясь по веткам (ну или, что то же самое, в дереве не существует циклов из веток).

Вова может распилить любую ветку ровно посередине ее длины, так как пилить дерево рядом с сучком очень трудно. После распила дерева в нескольких местах, у Вовы образуется несколько не связанных друг с другом кусков дерева, которые необходимо убрать в сарай. Кусок дерева можно убрать в сарай, если его длина не превосходит высоты сарая h . Длина же куска дерева определяется следующим образом: у любого куска дерева есть распил, ближайший к исходному корню дерева. Длиной куска назовем максимальную сумму длин веток от вышеупомянутого распила до любого другого распила или сучка в этом куске. Например, длина дерева на рисунке слева равна 30, а на рисунке справа равна 110 (ближайший к исходному корню распил представлен сучком слева).



При распиле какой-либо ветки дерева Вова оставляет ровно половину ее длины с одной стороны и ровно половину с другой. Например, если дерево состоит из одной ветки длины 10, а высота сарая равна 8, то распилив эту единственную ветку, Вова получит два куска дерева длины по 5 каждый, и сможет убрать дерево в сарай. При этом, каждую ветку можно пилить только один раз, то есть, если при том же дереве высота сарая равна 4, убрать дерево в сарай никак не получится.

Ваша задача — так определить места распилов дерева, чтобы после распиливания куски дерева поместились бы в сарай, при этом суммарную толщину распиливаемых веток нужно минимизировать (чтобы Вова справился с деревом побыстрее).

Формат входного файла

В первой строке входного файла находится два числа: N — количество сучков в дереве ($2 \leq N \leq 100$) и h — высота сарая ($1 \leq h \leq 10^6$). Далее следует $N - 1$ строка с описаниями веток дерева. Каждая ветка задается четырьмя целыми числами: a_i, b_i, l_i, s_i — номерами сучков, которые соединяются этой веткой, ее длиной и толщиной ($1 \leq a_i, b_i \leq N$), ($1 \leq l_i, s_i \leq 10^6$). Исходный корень дерева обозначается сучком с номером 1.

Формат выходного файла

В выходной файл выведите единственное число — минимальную суммарную толщину веток, которые придется перепилить Вове. Если же дерево убрать в сарай невозможно, выведите слово "IMPOSSIBLE".

Примеры

tree.in	tree.out
2 8 1 2 10 4	4
2 4 1 2 10 4	IMPOSSIBLE

Задача I. Выражение

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

Однажды Вася очень долго готовился к контрольной по математике, он долго решал разные арифметические выражения. Когда они закончились, он решил сам придумать свои выражения. В процессе подготовки к контрольной Васю заинтересовала такая задача: дано арифметическое выражение, состоящее из цифр от 0 до 9 и арифметических знаков + и -. Вася хочет добавить некоторое количество скобок в выражение, чтобы полученное значение было максимально возможным для данного выражения. Так как Вася очень много готовился, он устал и решил, что закрывающиеся скобки будут ставиться в конце выражения. Например, если выражение, которое написал Вася было $1 + 2 + 3 - 1 - 2 - 3$, тогда чтобы выражение приняло максимальное значение, скобки необходимо расставить так: $1 + 2 + 3 - (1 - 2 - 3)$.

Формат входного файла

В первой и единственной строке входного файла содержится выражение, состоящее из цифр от 0 до 9 и знаков + и -. Количество символов в исходном выражении не превышает 100000. Каждое число в выражении не превышает 100000. Никакие два арифметических знака не стоят рядом, плюс не стоит в начале выражения, и никакой знак не стоит в конце выражения.

Формат выходного файла

Выходной файл в первой строке должен содержать максимально возможное значение данного выражения. Во второй строке — исходное выражение с расставленными скобками. Если существует несколько способов расположения, выведите любой из них.

Примеры

<code>expression.in</code>	<code>expression.out</code>
<code>1+2+3-1-2-3</code>	10 <code>1+2+3-(1-2-3)</code>
<code>1+2+3-5-2+3</code>	6 <code>1+2+3-(5-(2+3))</code>

Задача J. Воскресенье 17

Имя входного файла: `sunday17.in`
Имя выходного файла: `sunday17.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Известно, что в 2009 году единственное воскресенье, которое имеет номер 17, — это 17 мая. Кроме того, зловещих пятниц 13 в 2009 году целых три.

Васе стало интересно, сколько раз день недели W с заданным номером D встречается в некотором году Y .

Вася знает, что действующий Григорианский календарь был введен папой Григорием XIII 4 октября 1582 года взамен старого Юлианского: следующим днем после четверга 4 октября стала пятница 15 октября (дней с 5 по 14 октября 1582 в григорианском календаре нет).

Месяцы: январь, март, май, июль, август, октябрь и декабрь содержат 31 день. Месяцы: апрель, июнь, сентябрь и ноябрь содержат 30 дней. Месяц февраль содержит 28 дней в обычный год и 29 дней в високосный.

Год является високосным, если выполняется одно из двух условий

- его номер делится на 4 и не делится на 100;
- его номер делится на 400.

Формат входного файла

Входной файл содержит три числа D , W и Y ($1 \leq D \leq 31$, $1 \leq W \leq 7$ и $2000 \leq Y \leq 3000$).

Формат выходного файла

В первой строке выходного файла выведите, сколько раз встречается искомый день недели с заданным номером в указанном году. В последующих строках выведите найденные даты в порядке возрастания в формате: “DD.MM.YYYY”.

Пример

<code>sunday17.in</code>	<code>sunday17.out</code>
17 7 2009	1 17.05.2009
13 5 2009	3 13.02.2009 13.03.2009 13.11.2009
1 2 2000	2 01.02.2000 01.08.2000