

Задача А. Минимум [D]

Имя входного файла: `min.in` или стандартный ввод
Имя выходного файла: `min.out` или стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Напишите программу, которая выбирает наименьшее из четырех чисел.

Формат входного файла

Четыре числа, каждое на отдельной строке. Числа не превосходят 20000.

Формат выходного файла

Выведите одно число — минимум из всех четырех чисел

Пример

<code>min.in</code> или стандартный ввод	<code>min.out</code> или стандартный вывод
4	1
3	
1	
2	

Задача В. Устный счет [D, C', C]

Имя входного файла: `count.in` или стандартный ввод
Имя выходного файла: `count.out` или стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Маленький Вася учится считать. Просто так считать он уже умеет, поэтому он придумал себе более сложную задачу. Он решил посчитать от 1 до N , соблюдая такие дополнительные правила: если число делится на 3 его нужно назвать дважды, если число делится на 7 его нужно пропустить, а если число делится и на 3 и на 7 – назвать, как и все остальные числа, ровно один раз.

Напишите программу, которая поможет Васе проверить себя

Формат входного файла

В первой строке вводится число $N \leq 1000$.

Формат выходного файла

Выведите через пробел числа от 1 до N , соблюдая Васины правила.

Пример

<code>count.in</code> или стандартный ввод	<code>count.out</code> или стандартный вывод
4	1 2 3 3 4
8	1 2 3 3 4 5 6 6 8

Задача C. Ender's Game [D]

Имя входного файла: `game.in` или стандартный ввод
Имя выходного файла: `game.out` или стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Эндер и Валентина играют в такую игру: они по очереди называют натуральные числа. Когда сумма всех названных чисел становится больше 100, тот, кто назвал последнее число, проигрывает. Дети так увлеклись названием, что забыли, что надо считать сумму. Теперь сумма давно уже больше 100 и они не знают, кто из них выиграл. К счастью, они записывали все числа, которые называли, поэтому вы сможете им помочь и узнать, кто выиграл. Начинал игру Эндер.

Формат входного файла

В отдельных строках записаны по очереди ходы Эндера и Валентины, начиная с хода Эндера. Гарантируется, что сумма всех чисел превосходит 100

Формат выходного файла

Выведите `ENDER`, если выиграл Эндер или `VALENTINE`, если выиграла Валентина.

Пример

<code>game.in</code> или стандартный ввод	<code>game.out</code> или стандартный вывод
50 50 10	VALENTINE
20 40 20 40	ENDER

Задача D. Гипотеза Коллатца [С']

Имя входного файла: collatz.in или стандартный ввод
Имя выходного файла: collatz.out или стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Рассмотрим следующую последовательность чисел. Берём любое натуральное число n . Если оно чётное, то делим его на 2, а если нечётное, то умножаем на 3 и прибавляем 1 (получаем $3n + 1$). С полученным числом выполняем те же самые действия, и так далее, пока не получим 1.

Для числа 6 мы таким образом получим последовательность

6, 3, 10, 5, 16, 8, 4, 2, 1,

содержащую 9 элементов.

Гипотеза Коллатца (одно из знаменитых, но пока недоказанных утверждений) говорит, что начиная с любого числа мы в итоге получим 1, то есть последовательность не может быть бесконечной. Мы не предлагаем вам доказывать гипотезу Коллатца, но просим написать программу, которая для данного числа определяет длину полученной последовательности.

Формат входного файла

Число $n \leq 1000$, гарантируется, что длина последовательности Коллатца для этого числа не превосходит 16000.

Формат выходного файла

Выведите одно число — длину последовательности Коллатца для данного n .

Пример

collatz.in или стандартный ввод	collatz.out или стандартный вывод
6	9

Задача Е. Автомобильные номера [С]

Имя входного файла: auto.in или стандартный ввод
Имя выходного файла: auto.out или стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В стране Лузитании решили ввести новый формат автомобильных номеров. Теперь на каждом номере будет K заглавных латинских букв и N цифр. Буквы и цифры могут быть любыми за одним исключением: если все буквы на номере одинаковые, он считается Особым Государственным Номером. В этом случае цифры на нем должны образовывать палиндром (то есть читаться одинаково слева направо и справа налево) и образовывать число, которое делится на число Q .

Лузитане переживают, хватит ли им автомобильных номеров на все автомобили. Помогите им посчитать количество разрешенных номеров.

Формат входного файла

В первой строке через пробел введены числа K , N и Q , $1 \leq K \leq 4$, $2 \leq N \leq 7$, $2 \leq Q \leq 100$.

Формат выходного файла

Выведите одно число: количество допустимых номеров.

Пример

auto.in или стандартный ввод	auto.out или стандартный вывод
2 2 4	65078

Задача F. Рамки [C', C]

Имя входного файла: `borders.in` или стандартный ввод
Имя выходного файла: `borders.out` или стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Маленький Коля рисует мелом на асфальте. Вначале он расчертил асфальт на квадратики, так, что получилась сетка $N \times M$ клеток. Потом он начал рисовать в этой сетке “рамки”: он рисует границу прямоугольника толщиной в одну клетку. Мы записали, какие рамки нарисовал Коля. Помогите нам понять, как выглядел его рисунок.

Формат входного файла

В первой строке вводятся два числа, N, M , $N, M \leq 100$, размер поля по горизонтали и вертикали соответственно. Во второй строке одно натуральное число $k \leq 20$ — количество рамок. Далее идут k строк, описывающих рамки. В каждой строке указаны четыре числа $x_{min}, y_{min}, x_{max}, y_{max}$: координаты верхнего левого и нижнего правого угла рамки. В начале указывается координата по горизонтали, затем по вертикали. Гарантируется, что координаты не превосходят размеров сетки, гарантируется, что $x_{min} < x_{max}, y_{min} < y_{max}$

Формат выходного файла

Выведите таблицу $N \times M$ из единиц и нулей, в которой единицы стоят на месте закрасенных клеток, а нули — на месте незакрасенных

Пример

<code>borders.in</code> или стандартный ввод	<code>borders.out</code> или стандартный вывод
6 5 1 1 1 5 4	111110 100010 100010 111110 000000
6 6 2 1 1 4 4 3 3 6 6	111100 100100 101111 111101 001001 001111

Задача G. Цветная полоска [B]

Имя входного файла: `strip.in` или стандартный ввод
Имя выходного файла: `strip.out` или стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

У Пети была цветная полоска, разделенная на N клеток. Каждая клетка была раскрашена в красный, синий или зеленый цвет. Петя помнит, что полоска содержала не менее A красных, не менее B зеленых и не менее C синих клеток. Помогите ему определить, сколько существует различных раскрасок полоски из N клеток, удовлетворяющих этим условиям. Выведите ответ по модулю 10^9+7 .

Формат входного файла

В единственной строке входного файла записано четыре целых числа — N, A, B, C ($1 \leq N \leq 5000, 0 \leq A, B, C \leq N$).

Ограничения для задачи G.py: $1 \leq N \leq 1000$.

Формат выходного файла

Выведите одно число — ответ на задачу по модулю $10^9 + 7$.

Примеры

<code>strip.in</code> или стандартный ввод	<code>strip.out</code> или стандартный вывод
3 1 1 1	6
3 1 1 2	0

Задача Н. Игра с числом [В]

Имя входного файла: `number.in` или стандартный ввод
Имя выходного файла: `number.out` или стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

У Пети есть строка s , состоящая из L цифр. Он хочет сделать из этой строки строку t , которая также состоит из L цифр. У него есть n доступных операций, i -ая из которых устроена так: Петя может выбрать какую-то подстроку s длины 2, равную a_i , и заменить ее на подстроку b_i , также длины 2. Никакая строка a_i не может участвовать в двух различных операциях.

Например, пускай Пете доступны 2 операции: $13 \rightarrow 33$, $33 \rightarrow 38$. Тогда из строки 113 можно за 4 операции получить строку 388 : $113 \rightarrow 133 \rightarrow 333 \rightarrow 338 \rightarrow 388$.

По данным строкам s и t , а также описанию возможных операций определите, сможет ли Петя получить t из s , а также минимальное количество операций, которые ему для этого потребуются. Операции можно выполнять в любом порядке, каждую операцию можно применять неограниченное количество раз.

Формат входного файла

В первой строке записано число L — длина строк s и t ($1 \leq L \leq 7$).

Во второй и третьей строках записаны строки s и t . Гарантируется, что s и t состоят из цифр и длина каждой из них равна L .

В четвертой строке записано число K ($0 \leq K \leq 100$) — количество доступных операций.

В каждой из следующих K строк записаны параметры операций в формате « $a_i b_i$ », где a_i и b_i — строки из 2 цифр. Гарантируется, что ни для каких двух операций a_i не совпадают.

Ограничения для задачи Н.ру: $1 \leq L \leq 6$.

Формат выходного файла

Если невозможно получить строку t при помощи данных операций, выведите -1 . В противном случае выведите минимальное требуемое количество операций.

Примеры

<code>number.in</code> или стандартный ввод	<code>number.out</code> или стандартный вывод
3 113 388 2 13 33 33 38	4
3 113 388 2 13 33 33 37	-1

Задача I. Крестики-крестики [B, A]

Имя входного файла: `xx.in` или стандартный ввод
Имя выходного файла: `xx.out` или стандартный вывод
Ограничение по времени: 2 секунды (для задачи I.py — 4 секунды)
Ограничение по памяти: 256 мегабайт

Петя и Вася любят играть в игру «крестики-крестики». Правила этой игры следующие: игроки по очереди делают ходы на квадратном поле $N \times N$ клеток, все клетки которого изначально свободны. На каждом ходу игрок ставит ровно один крестик в любую из свободных клеток. Если после хода игрока на поле образовался вертикальный, горизонтальный или диагональный (в любом направлении) сплошной ряд как минимум из K крестиков, этот игрок считается проигравшим.

Зачастую ребята увлекаются и пропускают тот момент, когда кто-то из них делает проигрывающий ход. Помогите им по записи партии найти первый ход, после которого на доске появился сплошной ряд хотя бы из K крестиков, либо определить, что партия еще не закончилась.

Формат входного файла

В первой строке задано два целых числа, разделенных пробелом — N , K и количество ходов в партии M ($2 \leq N \leq 1000$, $2 \leq K \leq N$, $1 \leq M \leq N \times N$).

В строках со второй по $(M + 1)$ -ую даны описания ходов. В $(i + 1)$ -ой строке дано описание i -ого хода в формате `row column`, где `row` — номер строки, а `column` — номер столбца с клеткой, куда был сделан i -ый ход.

Гарантируется, что никакие два хода не были сделаны в одну и ту же клетку.

Ограничения для задачи I.py: $2 \leq N \leq 500$.

Формат выходного файла

Если партия еще не закончена (т.е. после всех данных ходов на доске нет ни одного сплошного ряда из крестиков длины не менее K), выведите одно число 0. В противном случае выведите номер первого хода, после которого на доске появился хотя бы один такой ряд. Нумерация ходов начинается с 1.

Примеры

<code>xx.in</code> или стандартный ввод	<code>xx.out</code> или стандартный вывод
3 3 5 1 1 3 3 2 1 2 2 3 1	4
5 2 5 1 1 3 3 1 5 5 5 5 1	0

Задача J. Строка и шаблон [A]

Имя входного файла: `template.in` или стандартный ввод
Имя выходного файла: `template.out` или стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Даны две строки — текст s и шаблон t . s состоит из маленьких латинских букв, t состоит из маленьких латинских букв и знаков вопроса. Строка u *удовлетворяет* шаблону t , если они совпадают с точностью до знаков вопроса (на месте знака вопроса в u может стоять произвольная буква). Например, строка `abac` удовлетворяет шаблонам `a?a?`, `abac` и `????`, но не удовлетворяет шаблонам `b???` и `a??`.

Определите все подстроки текста s , удовлетворяющие данному шаблону t .

Формат входного файла

В первой строке записана строка s ($1 \leq |s| \leq 500\,000$).

Во второй строке записана строка t ($1 \leq |t| \leq 500\,000$). Строка t может содержать не более 10 знаков вопроса.

Формат выходного файла

В первой строке выведите одно число k — количество подстрок s , удовлетворяющих шаблону t .

Во второй строке выведите k чисел, разделенных пробелами, в возрастающем порядке — позиции начал подстрок s , удовлетворяющих шаблону t . Номера позиций нумеруются, начиная с единицы.

Примеры

<code>template.in</code> или стандартный ввод	<code>template.out</code> или стандартный вывод
<code>abacaba</code> <code>a?a</code>	3 1 3 5
<code>abacaba</code> <code>aa?</code>	0

Задача К. Добавьте ребра [А]

Имя входного файла: `edges.in` или стандартный ввод
Имя выходного файла: `edges.out` или стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Ребро неориентированного графа называется *мостом*, если после его удаления из графа количество компонент связности увеличивается.

Дан связный неориентированный граф с N вершинами и M ребрами. Определите, какое минимальное количество ребер необходимо в него добавить, чтобы после их добавления в графе не было ни одного моста.

В исходном графе могут присутствовать петли и кратные ребра. Добавлять петли и ребра, кратные какому-либо из существующих (в том числе добавленным ранее), разрешается.

Формат входного файла

В первой строке записано два целых числа, разделенных пробелом — N и M ($1 \leq N \leq 200\,000$, $0 \leq M \leq 200\,000$).

В каждой из следующих M строк записано по два целых числа, разделенных пробелом — номера вершин, являющихся концами очередного ребра.

Гарантируется, что данный граф связан.

Формат выходного файла

Выведите одно целое число — ответ на задачу.

Примеры

<code>edges.in</code> или стандартный ввод	<code>edges.out</code> или стандартный вывод
2 3 1 1 1 2 2 2	1
2 2 1 2 1 2	0
5 4 1 2 1 3 1 4 1 5	2