

Задача А. На три буквы

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 МБ

Напомним, что строка $B = b_1b_2b_3\dots b_m$, является подпоследовательностью строки $A = a_1a_2a_3\dots a_n$, если существует строго возрастающая последовательность $\{i_1, i_2, i_3, \dots, i_m\}$ индексов A , такая, что для всех $j \in \{1, \dots, m\}$, выполняется $A_{i_j} = B_j$. Например, $B = "aba"$ является подпоследовательностью строки $A = "abacaba"$. Последовательность индексов в этом случае может быть такой: $\{1, 2, 3\}$.

Пусть Вам дана строка S состоящая только из маленьких букв латинского алфавита. Ваша задача заключается в том, чтобы посчитать количество ее подпоследовательностей «abc».

Формат входного файла

В единственной строке входного файла записана строка S , длиной не более 100 000 символов.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Пример

stdin	stdout
abc	1
ab	0

Задача В. Сжатие последовательности

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 МБ

Семен изобрел новый способ сжатия данных. Правда, он пока умеет сжимать только последовательность из N единиц. Метод сжатия основывается на представлении последовательности из N единиц в виде последовательности из чисел от 1 до A , так, чтобы суммы членов обеих последовательностей совпадали (т.е. были равны N). Например последовательность 1, 1, 1, 1, 1, при $A = 3$ может быть преобразована в последовательность 1, 2, 1, 1 или 2, 3 или другие последовательности.

Ваша задача — посчитать количество способов сжать заданную последовательность.

Формат входного файла

В единственной строке входного файла записано два числа N и A ($1 \leq A \leq N \leq 1000$).

Формат выходного файла

Первая строка выходного файла должна содержать одно число — ответ на задачу.

Примеры

stdin	stdout
2 1	1
4 2	5

Последовательности, получающиеся при сжатии последовательности 1, 1, 1, 1 и $A = 2$: «1,1,1,1», «1,1,2», «1,2,1», «2,1,1», «2,2».

Задача С. Резать!

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 МБ

Держа в руках листок бумаги в клетку, у некоторых из нас возникает желание разрезать его. В данной задаче вам предлагается поддаться этому желанию!

К вам у нас есть лишь несколько просьб. Во-первых, разрезать листок можно только по линиям сетки. Во-вторых, каждый из получившихся кусочков бумаги должен состоять ровно из k клеток исходного листа.

Формат входного файла

Первая строка входного файла содержит два целых числа n и m ($1 \leq n, m \leq 100$) — соответственно, высоту и ширину листа бумаги в клетках. Вторая строка входного файла содержит число k ($1 \leq k \leq 1000$).

Формат выходного файла

Если данный листок бумаги нельзя разрезать указанным образом, в выходной файл выведите «-1».

Иначе, вам следует вывести n строк по m целых чисел в каждой — для каждой из клеток исходного листа бумаги выведите номер кусочка, в котором эта клетка оказалась. Кусочки следует нумеровать последовательными натуральными числами, начиная с единицы.

Примеры

stdin	stdout
2 2 1	1 2 4 3
2 2 2	1 1 2 2
2 2 3	-1
4 4 4	1 1 1 4 1 4 4 4 2 2 3 3 2 2 3 3

Задача D. Клеить!

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 МБ

В задаче «Резать!» базового уровня предыдущей интернет-олимпиады участникам было предложено разрезать разлинованный в квадратную клетку листок бумаги. При этом резать листок следовало только по линиям сетки, и каждый из получившихся кусочков бумаги должен был состоять ровно из k квадратных клеток исходного листа.

В данной задаче вам предлагается решить задачу жюри, а именно определить, верно ли был разрезан листок командой-участником.

Формат входного файла

Первая строка входного файла содержит два целых числа n и m ($1 \leq n, m \leq 100$) — соответственно, высоту и ширину листа бумаги в клетках. Вторая строка входного файла содержит число k ($1 \leq k \leq 1000$). Гарантируется, что число $n \cdot m$ делится на k .

Далее следует n строк по m целых чисел в каждой — для каждой из клеток исходного листа бумаги задан номер кусочка, в который клетка была определена командой-участником. Кусочки занумерованы натуральными числами начиная с единицы.

Формат выходного файла

Если листок был успешно разрезан командой, выведите в выходной файл «OK». В противном случае выведите «Wrong answer».

Примеры

stdin	stdout
4 3 6 1 1 2 1 2 2 1 2 2 1 1 2	OK
2 2 1 1 1 2 3	Wrong answer
2 2 2 1 2 2 1	Wrong answer

Задача Е. Расстояние между отрезками

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 МБ

Найдите расстояние между двумя отрезками.

Формат входного файла

В двух строках входного файла даны по четыре целых числа, не превосходящих по модулю 10 000, — координаты концов сначала первого, затем второго отрезков.

Формат выходного файла

В первой строке выходного файла выведите одно вещественное число — расстояние между отрезками с точностью не менее 10^{-6} .

Примеры

stdin	stdout
1 1 2 2 2 1 3 0	0.707106781

Задача F. Пожарные депо

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 МБ

В городе есть n перекрестков, некоторые из которых соединены односторонними дорогами. Всего в городе m дорог.

Мэр решил построить несколько пожарных депо. Для этого ему надо выбрать, на каких перекрестках их построить. При этом когда пожарная машина едет на пожар, она может ехать как по направлению движения по дороге, так и против этого направления. А вот при возвращении в депо срочности нет и ехать «по встречке» уже нельзя.

Помогите мэру построить минимальное число депо, чтобы при пожаре дома на любом перекрестке пожарная машина из одного из депо могла доехать до этого перекрестка (возможно против движения), а затем могла вернуться в свое депо (уже двигаясь только по направлению дорог).

Формат входного файла

В первой строке входного файла находятся два натуральных числа n и m ($1 \leq n \leq 20\,000, 1 \leq m \leq 200\,000$) — количество вершин и улиц. Далее в m строках перечислены улицы, каждая улица задаётся парой чисел — номерами начального и конечного перекрестка.

Формат выходного файла

Выведите одно число — минимальное число депо, которые необходимо построить.

Пример

stdin	stdout
6 7 1 2 2 3 3 1 1 4 1 5 4 6 6 4	2

В приведенном примере можно построить депо, например, на перекрестках 4 и 5.

Задача G. Числа

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 МБ

Задано n положительных целых чисел a_1, a_2, \dots, a_n . Необходимо выбрать из них как можно больше чисел так, чтобы сумма выбранных чисел делилась на 3.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \leq n \leq 10^5$). Вторая строка входного файла содержит n целых чисел a_1, \dots, a_n , находящихся в диапазоне от 0 до 10^9 .

Формат выходного файла

В первой строке выходного файла выведите количество выбранных чисел. Во второй строке выведите их номера во входном файле в порядке возрастания. Числа нумеруются в порядке их перечисления во входном файле натуральными числами от 1 до n .

Если ответов несколько, выведите любой.

Примеры

stdin	stdout
3 4 5 6	3 1 2 3
3 4 2 5	2 1 2

Задача Н. Дороги

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 МБ

Дорожная сеть города Нью-Флетсита устроена довольно просто. Все дороги являются отрезками единичной длины с концами в точках с целыми координатами. Этот факт — своего рода достопримечательность Нью-Флетсита.

Недавно пришедший к власти мэр считает, что он тратит слишком много времени на дорогу из дома в мэрию и обратно. Он решил построить несколько новых дорог так, чтобы этот путь был как можно короче. Естественно, новые дороги должны также являться единичными отрезками с концами в целых точках.

Вам, как главному инженеру Нью-Флетсита, поручено вычислить минимальное количество дорог, которое придется построить для осуществления плана мэра.

Формат входного файла

Первая строка входного файла содержит целое число n — количество дорог в Нью-Флетсита ($0 \leq n \leq 100$). Далее следуют n строк с четырьмя целыми числами, разделенными пробелами: x_i, y_i, x_j, y_j — координаты начала и конца соответствующей дороги ($0 \leq x_i, y_i, x_j, y_j \leq 100$). Последняя строка содержит два целых числа m_x и m_y — координаты дома мэра ($0 \leq m_x, m_y \leq 100$). Мэрия расположена в точке $(0, 0)$.

Все дороги расположены либо по горизонтали, либо по вертикали, а длина каждой из этих дорог равна единице. Движение по дорогам возможно в обе стороны.

Формат выходного файла

В выходной файл на первой строке выведите число M — количество новых дорог, которые нужно построить в Нью-Флетсита.

Примеры

stdin	stdout
1 0 0 1 0 1 1	1
5 0 0 1 0 1 0 1 1 1 1 0 1 0 1 0 2 0 2 1 2 1 2	1

Задача I. Две последовательности

Ограничение по времени: 2 секунды

Ограничение по памяти: 256 МБ

Определим последовательности a_n и b_n следующим образом: $a_1 = 2$, $a_2 = 3$, $a_3 = 4$, $a_4 = 7$, $a_n = b_{n-1} + b_{n-3}$, $n > 4$, b_n — последовательность чисел, не входящих в a_n , записанных в возрастающем порядке.

Таким образом, последовательность a_n будет выглядеть следующим образом: 2, 3, 4, 7, 13, 15, ..., а последовательность b_n — 1, 5, 6, 8, 9, 10, ...

Ваша задача состоит в том, чтобы найти a_n и b_n .

Формат входного файла

Входной файл содержит целое число n ($1 \leq n \leq 10000$).

Формат выходного файла

В первой строке выходного файла выведите a_n , во второй — b_n .

Примеры

stdin	stdout
4	7 8
10	25 16