

Задача А. Произведение матриц

Имя входного файла: `mmul.in` или `stdin`
Имя выходного файла: `mmul.out` или `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Произведением матриц A и B размера $p \times q$ и $q \times r$, соответственно, называется матрица C размера $p \times r$, элементы которой вычисляются по формуле:

$$C_{i,j} = \sum_{k=1}^q A_{i,k} \cdot B_{k,j}$$

По данным матрицам A и B найдите их произведение.

Формат входных данных

В первой строке входного файла заданы через пробел три целых числа p , q и r ($1 \leq p, q, r \leq 100$). В следующих p строках записана матрица A ; каждая из этих строк содержит q целых чисел, разделённых пробелами. Наконец, в последних q строках записана матрица B ; каждая из этих строк содержит r целых чисел, разделённых пробелами. Элементы матриц не превосходят 100 по абсолютной величине.

Формат выходных данных

В выходной файл выведите матрицу C : p строк, в каждой из которых r чисел через пробел.

Примеры

<code>mmul.in</code> или <code>stdin</code>	<code>mmul.out</code> или <code>stdout</code>
2 2 2 1 0 0 1 1 0 0 1	1 0 0 1
1 3 1 1 2 3 -1 -2 -3	-14
3 2 4 0 1 1 0 0 1 2 1 0 0 1 1 2 1	1 1 2 1 2 1 0 0 1 1 2 1

Задача В. Степень матрицы

Имя входного файла: `mpow.in` или `stdin`
Имя выходного файла: `mpow.out` или `stdout`
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Задана квадратная матрица $n \times n$. Нужно возвести ее в степень m

Формат входных данных

В первой строке задано три целых числа n , m и p ($1 \leq n \leq 100$, $0 \leq m \leq 10^{18}$, $2 \leq p \leq 10^9$).
Далее задана матрица: n строк по n целых чисел. Все числа в матрице неотрицательны и меньше p .

Формат выходных данных

Выведите матрицу: n строк по n чисел, каждое число — остаток от деления элемента на p

Примеры

<code>mpow.in</code> или <code>stdin</code>	<code>mpow.out</code> или <code>stdout</code>
3 5 239 1 0 1 1 2 0 3 2 1	120 92 56 102 84 46 21 204 120
5 10 27 1 2 3 4 5 5 4 3 2 1 11 12 13 14 15 15 14 13 12 11 1 11 1 11 1	2 5 7 10 12 16 19 26 2 9 5 9 26 3 20 19 23 18 22 17 12 22 9 19 6

Задача С. Линейные уравнения

Имя входного файла: `linear.in` или `stdin`
Имя выходного файла: `linear.out` или `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Система линейных уравнений, как всем известно, есть множество уравнений

$$\begin{aligned} a_{11}x_1 + \dots + a_{1n}x_n &= b_1 \\ &\dots \\ a_{n1}x_1 + \dots + a_{nn}x_n &= b_n \end{aligned}$$

Ваша задача — решить её.

Формат входных данных

В первой строке входного файла записано целое число n ($1 \leq n \leq 20$). В следующих n строках записано по $n + 1$ целых чисел: $a_{i1}, \dots, a_{in}, b_i$. Все эти числа не превышают 100 по абсолютному значению.

Формат выходных данных

Первая строка выходного файла должна содержать одно из следующих сообщений:

- `impossible` — решений нет
- `infinity` — бесконечно много решений
- `single` — единственное решение. В этом случае вторая строка должна содержать n чисел x_1, \dots, x_n , разделенных пробелами. Решение должно быть выведено с точностью не менее трех знаков после десятичной точки.

Примеры

<code>linear.in</code> или <code>stdin</code>	<code>linear.out</code> или <code>stdout</code>
2 1 1 1 2 2 2	<code>infinity</code>
2 1 2 0 1 2 1	<code>impossible</code>
2 1 2 1 2 1 0	<code>single</code> <code>-0.333 0.666</code>

Задача D. Головоломка

Имя входного файла: puzzle.in или stdin
Имя выходного файла: puzzle.out или stdout
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вася очень любит играть в квесты. Он любит кликать тут и там мышкой и наслаждается шутками, которых много в квестах. Но там есть еще и много разных головоломок, и Вася проводит много времени, решая их. Но однажды он столкнулся с головоломкой которую никак не мог решить. К счастью, Вася отличный программист, и он смог написать программу, которая решила головоломку и помогла ему закончить квест.

Васина головоломка представляет собой матрицу 3×3 , каждая клетка которой окрашена в черный или белый цвет. Если кликнуть на клетку, то она и ее соседи меняют свой цвет на противоположный. Цель — сделать все клетки одного цвета.

Ваша задача чуть более сложная. Пусть есть N клеток, занумерованных от 1 до N . Каждая клетка имеет множество клеток, связанных с ней. Когда игрок кликает на клетку, все клетки из множества, связанного с ней, меняют свой цвет. По данным связанным множествам и начальной раскраске выведите последовательность клеток, на которые нужно кликать, чтобы все клетки приобрели один цвет. Если есть несколько различных решений, выведите любое.

Формат входных данных

Первая строка содержит единственное целое число N ($1 \leq N \leq 200$). В $i+1$ -й строке ($1 \leq i \leq N$) находится описание множества, связанного с i -й клеткой. Каждое описание начинается с целого числа k — количество клеток в множестве, а затем идут k различных целых чисел (номера клеток). Последняя строка содержит N нулей и единиц — начальная раскраска клеток.

Формат выходных данных

Если невозможно окрасить все клетки в один цвет, выведите единственное число -1 . Иначе выведите целое число L — количество кликов, необходимых для решения головоломки, а затем L чисел — номера клеток, на которые необходимо кликать.

Примеры

puzzle.in или stdin	puzzle.out или stdout
9	9
3 1 2 4	1 2 3 4 5 6 7 8 9
4 1 2 3 5	
3 2 3 6	
4 1 4 5 7	
5 2 4 5 6 8	
4 3 5 6 9	
3 4 7 8	
4 5 7 8 9	
3 6 8 9	
1 0 1 0 1 0 1 0 1	

Задача Е. Полные квадраты

Имя входного файла: `perfect-squares.in` или `stdin`
Имя выходного файла: `perfect-squares.out` или `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Множество целых положительных чисел будем называть *полноквадратным*, если произведение его элементов является полным квадратом (равно 1, 4, 9, 16, 25, 36, ...).

Задано множество A . Определите, сколько непустых подмножеств B множества A являются полноквадратными.

Формат входных данных

В первой строке записано целое число N — количество элементов множества ($1 \leq N \leq 100$). Во второй строке записаны N попарно различных чисел a_i — элементы множества ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите одно число — количество полноквадратных подмножеств по модулю 1 000 000 007.

Примеры

<code>perfect-squares.in</code> или <code>stdin</code>	<code>perfect-squares.out</code> или <code>stdout</code>
4 49 20 500 7	3

Задача F. Гладкие числа

Имя входного файла: `smooth.in` или `stdin`
Имя выходного файла: `smooth.out` или `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Назовем число гладким, если его соседние цифры по модулю различаются не более чем на 1. Для данного N определите количество имеющих длину N .

Формат входных данных

На вход программы поступает количество цифр $N \leq 10^9$.

Формат выходных данных

Выведите ответ по модулю $10^9 + 7$.

Примеры

<code>smooth.in</code> или <code>stdin</code>	<code>smooth.out</code> или <code>stdout</code>
1	9

Задача G. Обобщенные числа фибоначчи

Имя входного файла: `fibonacci.in` или `stdin`
Имя выходного файла: `fibonacci.out` или `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Мы чуть-чуть обобщили для вас последовательность Фибоначчи, теперь:

$$f_1 = f_2 = 1$$

$$f_i = a \cdot f_{i-1} + b \cdot f_{i-2} + c \cdot 2^i + d \cdot i + e, \text{ для } i > 2$$

Дано n , найдите значение f_n , взятое по модулю $10^9 + 7$.

Формат входных данных

Неотрицательные целые числа: a, b, c, d, e, n . ($0 \leq a, b, c, d, e \leq 10^9$; $1 \leq n \leq 10^{18}$)

Формат выходных данных

Выведите f_n , взятое по модулю $10^9 + 7$.

Примеры

<code>fibonacci.in</code> или <code>stdin</code>	<code>fibonacci.out</code> или <code>stdout</code>
1 1 0 0 0 8	21
1 2 3 4 5 6	775