

Задача А. Архивация

Имя входного файла: `rle.in`
Имя выходного файла: `rle.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Алгоритм сжатия RLE устроен по следующему принципу. Файл рассматривается как последовательность бит. Непрерывные цепочки нулей и единиц заменяются на число — длину цепочки, максимальная длина которой K ограничена (очень длинные цепочки разбиваются на несколько и кодируются несколькими числами, между которыми вставляются нули). Так же для однозначности первой цепочкой считается всегда цепочка нулей (если последовательность начинается с 1, то первым числом при сжатии пишется 0).

Формат входного файла

На первой строке дано число K ($1 \leq K \leq 1000$). На второй строке дана последовательность нулей и единиц, в конце которой находится число 2. Длина последовательности — натуральное число, не превышающее 10^6 .

Формат выходного файла

Сжатая последовательность (см. примеры).

Примеры

<code>rle.in</code>	<code>rle.out</code>
3 0 0 0 1 1 1 0 0 2	3 3 2

Задача В. В определенном порядке

Имя входного файла: `order.in`
Имя выходного файла: `order.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Во входном файле записано сначала число N , а затем N чисел. Выведите эти N чисел в следующем порядке: сначала выводятся все нечетные числа в том порядке, в каком они встречались во входном файле, а затем - все четные.

Формат входного файла

Вводится число N ($0 \leq N \leq 1000$), а затем N чисел, каждое из которых не превосходить 10000.

Формат выходного файла

Выведите N чисел в требуемом порядке.

Примеры

<code>order.in</code>	<code>order.out</code>
9 52 38 29 99 22 66 79 18 56	29 99 79 52 38 22 66 18 56

Задача С. Игра в футбол

Имя входного файла: `foot.in`
Имя выходного файла: `foot.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

На чемпионате ЛКШ по футболу теперь применяется новый способ приветствия команд. Для этого команды выстраиваются в линию. Капитаны команд встают рядом, а команды выстраиваются по сторонам от своих капитанов. Капитаны делают шаг навстречу друг другу, здороваются и отходят в концы своих команд. На некоторых матчах между капитанами может стоять судья, но на процедуру приветствия это не влияет. Напишите программу, которая выведет расположение людей на поле после приветствия капитанов.

Формат входного файла

Каждый человек, находящийся на поле, задаётся номером, написанном на его майке. Для удобства будем считать, что у судьи тоже есть номер на майке. В начале входного файла записано натуральное число N ($N \leq 200$) — число людей, находящихся на поле. Далее вводится N чисел-номеров на майках людей (номера на майках — натуральные числа ≤ 1000).

Формат выходного файла

Выведите N чисел — номеров на майках людей, находящихся на поле, после приветствия капитанов

Примеры

<code>foot.in</code>	<code>foot.out</code>
5 5 3 2 9 2	3 5 2 2 9

Задача D. Заполнение

Имя входного файла: `snake.in`
Имя выходного файла: `snake.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Заполните массив $N \times N$ элементов змейкой (см. пример).

Формат входного файла

Во входном файле записано единственное число $N \leq 500$.

Формат выходного файла

В выходной файл выведите массив из $N \times N$, заполненный "змейкой".

Примеры

snake.in	snake.out
3	1 2 3 6 5 4 7 8 9

Задача Е. Диагональки

Имя входного файла: `diagonal.in`
Имя выходного файла: `diagonal.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В квадратной таблице $N \times N$ подсчитать суммы чисел, стоящих на диагоналях.

Формат входного файла

Вводится число N ($1 \leq N \leq 100$), а затем матрица $N \times N$. Элементы матрицы - числа из диапазона *integer*.

Формат выходного файла

Вывести сначала сумму чисел на главной, а затем - на побочной диагонали.

Примеры

diagonal.in	diagonal.out
1 451	451 451
4 134 475 30 424 303 151 419 235 248 166 90 42 318 237 184 36	411 1327

Задача F. Выручка театра

Имя входного файла: `theatre.in`
Имя выходного файла: `theatre.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В театре N рядов по M мест в каждом. Даны две матрицы - в первой записаны стоимости билетов. Вторая - сообщает, какие билеты проданы, а какие - нет (1 - соответствующий билет продан, 0 - не продан).

Определите общую выручку от спектакля.

Формат входного файла

Во входном файле записано сначала число N , затем число M . После задана матрица стоимостей билетов (N строк по M чисел, каждое из чисел от 0 до 10000). Далее задана матрица проданных билетов (опять же N строк по M чисел). $N, M \leq 500$

Формат выходного файла

В выходной файл выведите общую выручку от продажи билетов.

Примеры

theatre.in	theatre.out
5 3 3 2 9 2 6 7 1 5 5 9 8 7 2 3 4 1 0 1 1 0 0 1 0 1 0 1 1 1 1 1	44

Note

В примере таблицы не разделены пустой строкой. Это не правда. В тестах между массивами ЕСТЬ пустая строка.

Задача G. Бонусный уровень

Имя входного файла: `snake.in`
Имя выходного файла: `snake.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Сережа играл в змейку и почти успевал на поезд в ЛКШ. Но он не знал, что за последним уровнем следует дополнительный "бонусный".

Он представляет собой клетчатое поле $N \times N$, в каждой клетке которого лежит яблочко. Сережина змея последовательно ползет по диагоналям данного поля, поедая фрукты. Чтобы совершать такие поползновения, ей приходится после каждой диагонали менять направление движения (подробнее см. примеры).

Формат входного файла

Во входном файле одно целое число $1 \leq N \leq 100$.

Формат выходного файла

Выведите ровно N строк по N чисел в каждой: для каждой клетки поля ее номер в последовательности прохождения поля змеей.

Примеры

snake.in	snake.out
4	1 3 4 10 2 5 9 11 6 8 12 15 7 13 14 16
5	1 3 4 10 11 2 5 9 12 19 6 8 13 18 20 7 14 17 21 24 15 16 22 23 25