

Задача А. Генераторы списков

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

На входе программа получает список целых чисел s . Ваша задача — вывести следующие списки по одному на строку:

1. Список, состоящий из квадратов элементов s .
2. Список, состоящий из остатков от деления на 11 элементов s .
3. Список, состоящий только из чётных элементов s .
4. Список, состоящий только из элементов s с нечётным количеством цифр.
5. Список, состоящий только из двузначных элементов s , записанных 2 раза подряд.
6. Список, состоящий из элементов s , стоящих на позициях, не кратных 3.

Формат входного файла

В единственной строке записаны числа, разделённые пробелами.

Формат выходного файла

В каждой из шести строк выведите соответствующий список в стандартном формате python'a.

Примеры

<code>list.in</code>	<code>list.out</code>
8 5 15 101 42 1	[64, 25, 225, 10201, 1764, 1] [8, 5, 4, 2, 9, 1] [8, 42] [8, 5, 101, 1] [1515, 4242] [5, 15, 42, 1]

Note

Для вывода каждого из списков необходимо использовать единственную строку с генератором. Например:

```
print([x for x in s])
```

Задача В. Бонусный уровень

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

Серёжа 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. Серёжина 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

<code>snake.in</code>	<code>snake.out</code>
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

Задача С. 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
3 3 1 2 3 4 5 6 7 8 9	25
1 0 1 0 1 0 1 0 1	

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 E. 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. Кинотеатр

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

Однажды, ученики B -ой средней школы города G решили съездить в кино. Администрация кинотеатра расположила их в зале размера $N \times M$, который специально был подобран так, чтобы все места были заняты школьниками. Каждому посетителю кинотеатра был выдан свой номер.

Школьники заняли свои места следующим образом: они входили в зал в порядке, в котором шли их номера, и полностью занимали сначала первый ряд, потом второй, потом третий и т.д.

11	12	13	14	15	III ряд
6	7	8	9	10	II ряд
1	2	3	4	5	I ряд

Однако классный руководитель решил, что такая рассадка плохо влияет на поведение учащихся, и пересадил их по-другому: ученики сначала занимали все первые места каждого ряда, потом все вторые места каждого ряда и т.д. (см. рисунок).

3	6	9	12	15	III ряд
2	5	8	11	14	II ряд
1	4	7	10	13	I ряд

Администрация решила выяснить, сколько учащихся не поменяют своего места после пересадки.

Формат входного файла

В первой строке входного файла заданы числа N и M ($1 \leq N, M \leq 1000$).

Формат выходного файла

В выходной файл выведите число – количество участников, которые в результате пересадки останутся сидеть на тех же местах.

Примеры

cinema.in	cinema.out
3 3	3
3 4	2