

## Задача А. Шестерёнки

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Даны две сцепленные шестерёнки. У одной шестерёнки  $N$  зубцов, у другой —  $K$ . Требуется найти, какое минимальное число поворотов на один зубчик требуется сделать, чтобы шестерёнки вернулись в исходное состояние.

### Формат входного файла

В единственной строке — два числа,  $N$  и  $K$ .  $1 \leq N, K \leq 10^7$ .

### Формат выходного файла

Выведите искомое количество зубчиков. Гарантируется, что оно не более  $10^9$ .

### Примеры

<code>stdin</code>	<code>stdout</code>
2 3	6
6 21	42

## Задача В. МегаНОД

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

Найдите НОД  $N$  заданных чисел.

### Формат входного файла

Первая строка входного файла содержит натуральное число  $N$  ( $1 \leq N \leq 1000$ ) — количество чисел. Во второй строке заданы  $N$  чисел, не превышающие по модулю  $10^9$ .

### Формат выходного файла

Выведите НОД  $N$  чисел.

### Примеры

<code>megagcd.in</code>	<code>megagcd.out</code>
2	5
90 35	

## 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 стоимо-

сти 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

<code>theatre.in</code>	<code>theatre.out</code>
3 3 1 2 3 4 5 6 7 8 9	25
1 0 1 0 1 0 1 0 1	

## Задача D. 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 целые 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

### Задача Е. 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

### Задача F. 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 G. Диагонали

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