

Задача А. 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 входных данных

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 выходных данных

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 1 0 1 0 1 0 1 0 1	25

Задача В. Кинотеатр

Имя входного файла: `cinema.in`
Имя выходного файла: `cinema.out`
Ограничение по времени: 4 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 входных данных

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

Формат выходных данных

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

Примеры

<code>cinema.in</code>	<code>cinema.out</code>
3 3	3
3 4	2

Задача С. 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 входных данных

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

Формат выходных данных

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

Примеры

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

Задача D. 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 входных данных

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

Формат выходных данных

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 Е. Бонусный уровень

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

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

Он представляет собой клетчатое поле $N \times N$, в каждой клетке которого лежит яблочко. Серезина 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 входных данных

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

Формат выходных данных

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 F. Числа Фибоначчи — 3 цифры

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Числа Фибоначчи определяются так: $\varphi_1 = \varphi_2 = 1$; $\varphi_n = \varphi_{n-1} + \varphi_{n-2}$ при $n > 2$.

Вычислите последние три цифры n -го числа Фибоначчи.

Если у него меньше трёх цифр, выведите всё число без ведущих нулей.

Формат входных данных

Одно число n ($1 \leq n \leq 1\,000\,000$).

Формат выходных данных

Выведите последние три цифры φ_n .

Примеры

стандартный ввод	стандартный вывод
3	2
972	064