

# Кузнечик

Имя входного файла: `grasshopper.in`  
Имя выходного файла: `grasshopper.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

У одного из преподавателей параллели С' в комнате живёт кузнечик, который очень любит прыгать по клетчатой одномерной доске. Длина доски —  $N$  клеток. К его сожалению он умеет прыгать только на  $1, 2, \dots, k$  клеток вперёд.

Однажды преподавателям стало интересно, сколькими способами кузнечик может допрыгать из первой клетки до последней. Помогите им ответить на этот вопрос.

## Формат входных данных

В первой и единственной строке входного файла записано два целых числа —  $N$  и  $k$  ( $1 \leq N \leq 30, 1 \leq k \leq 10$ ).

## Формат выходных данных

Выведите одно число — количество способов, которыми кузнечик может допрыгать из первой клетки до последней.

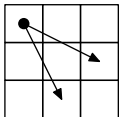
## Примеры

<code>grasshopper.in</code>	<code>grasshopper.out</code>
8 2	21

# Ход конём

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Дана прямоугольная доска  $N \times M$  ( $N$  строк и  $M$  столбцов). В левом верхнем углу находится шахматный конь, которого необходимо переместить в правый нижний угол доски. В данной задаче конь может перемещаться на две клетки вниз и одну клетку вправо или на одну клетку вниз и две клетки вправо.



Необходимо определить, сколько существует различных маршрутов, ведущих из левого верхнего в правый нижний угол.

## Формат входных данных

Входной файл содержит два натуральных числа  $N$  и  $M$  ( $1 \leq N, M \leq 50$ ).

## Формат выходных данных

В выходной файл выведите единственное число — количество способов добраться конём до правого нижнего угла доски.

## Примеры

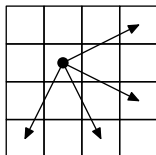
stdin	stdout
3 2	1
31 34	293930

# Ход конём – 2

Имя входного файла: `knight2.in`  
Имя выходного файла: `knight2.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Дана прямоугольная доска  $N \times M$  ( $N$  строк и  $M$  столбцов). В левом верхнем углу находится шахматный конь, которого необходимо переместить в правый нижний угол доски.

При этом конь может ходить следующим образом:



Необходимо определить, сколько существует различных маршрутов, ведущих из левого верхнего в правый нижний угол.

## Формат входных данных

Входной файл содержит два натуральных числа  $N$  и  $M$  ( $1 \leq N, M \leq 50$ ).

## Формат выходных данных

В выходной файл выведите единственное число — количество способов добраться конём до правого нижнего угла доски.

## Примеры

<code>knight2.in</code>	<code>knight2.out</code>
4 4	2
2 3	1

# ПДД в ЛКШ

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 256 мегабайта

Совсем недавно девятиклассник Коля прибыл в Летнюю Кинематографическую Школу.

Первым делом он решил посетить киностудию. В детском лагере «Олененок», где ЛКШ проводится в этом году, проложено множество асфальтированных дорожек и их пересечения образуют перекрестки. «Олененок» — необычный лагерь, и поэтому на всех дорожках и перекрестках нарисована разметка, а так же действуют правила дорожного движения, за нарушение которых полагаются штрафы.

Киностудия расположена на юго-востоке от корпуса, в котором живет Коля, поэтому школьник решил передвигаться только на восток и на юг. Ему очень хотелось побыстрее добраться до киностудии, и потому он решил не обращать внимания на правила, и переходить перекрестки как ему вздумается. Однако, как настоящий ЛКШнонок, Коля должен позаботиться о том, чтобы суммарный размер штрафов за его нарушения был минимален. Помогите ему в этом.

В вашем распоряжении карта лагеря, представляющая собой клетчатый прямоугольник  $N$  на  $M$ , в котором на пересечении  $i$ -ой строки и  $j$ -ого столбца указан размер штрафа при попадании на этот перекресток.

Корпус, в котором живет Коля находится в северо-западном углу лагеря, а киностудия — в юго-восточном. Помогите Коле добраться до места назначения, заплатив минимально возможный штраф.

## Формат входных данных

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

В последующих  $N$  строках содержатся по  $M$  чисел — карта лагеря «Олененок».

## Формат выходных данных

В первой строчке выведите одно целое число — минимальный размер штрафа, который придётся заплатить Коле. Во второй строчке выведите количество перекрестков на пути. В следующих строчках выведите координаты перекрестков, через которые школьник пройдет. Гарантируется, что штраф не превысит  $10^9$ .

## Примеры

stdin	stdout
2 2	351
163 165	3
83 105	1 1
	2 1
	2 2

# Калькулятор

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 64 мегабайта

Имеется калькулятор, который выполняет следующие операции:

- умножить число  $X$  на 2;
- умножить число  $X$  на 3;
- прибавить к числу  $X$  единицу.

Определите, какое наименьшее количество операций требуется, чтобы получить из числа 1 число  $N$ .

## Формат входных данных

Во входном файле написано натуральное число  $N$ , не превосходящее  $10^6$ .

## Формат выходных данных

В первой строке выходного файла выведите минимальное количество операций. Во второй строке выведите числа, последовательно получающиеся при выполнении операций. Первое из них должно быть равно 1, а последнее  $N$ . Если решений несколько, выведите любое.

## Примеры

stdin	stdout
1	0 1
5	3 1 3 4 5
962340	17 1 3 9 27 54 55 165 495 1485 4455 8910 17820 17821 53463 160389 160390 481170 962340

# Три единицы подряд

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

По данному числу  $N$  определите количество последовательностей из нулей и единиц длины  $N$ , в которых никакие три единицы не стоят рядом.

## Формат входных данных

Во входном файле написано натуральное число  $N$ , не превосходящее 35.

## Формат выходных данных

Выведите количество искомых последовательностей. Гарантируется, что ответ не превосходит  $2^{31} - 1$ .

## Примеры

<code>stdin</code>	<code>stdout</code>
4	13

# Кролик учит геометрию.

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 64 мегабайта

Кролики — очень любопытны. Они любят изучать геометрию, бегая по грядкам. Наш кролик как раз из таких. Сегодня он решил изучить новую фигуру — квадрат.

Кролик бегает по грядке — клеточному полю  $N \times M$  клеток. В некоторых из них посеяны морковки, в некоторых нет.

Помогите Кролику найти сторону квадрата наибольшей площади, заполненного морковками полностью.

## Формат входных данных

В первой строке даны два натуральных числа  $N$  и  $M$  ( $1 \leq N, M \leq 1000$ ). Далее в  $N$  строках расположено по  $M$  чисел, разделенных пробелами (число равно 0, если в клетке нет морковки или 1, если есть).

## Формат выходных данных

Выведите одно число — сторону максимального квадрата, заполненного морковками.

## Примеры

stdin	stdout
4 5 0 0 0 1 0 0 1 1 1 0 0 0 1 1 0 1 0 1 0 0	2

# Плавные числа

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Назовём натуральное число плавным, если разность любых двух его соседних цифр не превосходит по модулю единицы. Определите количество  $N$ -значных плавных чисел. Запись числа не может начинаться с цифры 0.

## Формат входных данных

Программа получает на вход одно целое число  $N$  ( $1 \leq N \leq 20$ ).

## Формат выходных данных

Программа должна вывести одно целое число — искомое количество плавных чисел.

## Примеры

stdin	stdout
2	26



# Наибольшая последовательнократная подпоследовательность

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    1 секунда  
Ограничение по памяти:      64 мегабайта

Для заданной числовой последовательности  $a_1, a_2, \dots, a_n$  требуется найти длину максимальной последовательнократной подпоследовательности.

Для последовательнократной подпоследовательности  $a_{k_1}, a_{k_2}, \dots, a_{k_t}$  ( $k_1 < k_2 < \dots < k_t$ ) верно, что  $a_{k_i} | a_{k_j}$  при  $1 \leq i < j \leq t$  (утверждение « $a|b$ » эквивалентно « $b$  кратно  $a$ »). Подпоследовательность из одного элемента полагается последовательнократной по определению.

## Формат входных данных

В первой строке входного файла записаны  $N$  натуральных чисел ( $1 \leq N \leq 1000$ ), не превосходящих  $2 \cdot 10^9$  — последовательность.

## Формат выходных данных

Вывести единственное число, равное длине максимальной последовательнократной подпоследовательности.

## Примеры

стандартный ввод	стандартный вывод
3 6 5 12	3

# Кино

Имя входного файла: `cinema.in`  
Имя выходного файла: `cinema.out`  
Ограничение по времени: 2.5 секунды  
Ограничение по памяти: 256 мегабайта

Трое друзей летом посмотрели несколько фильмов. Для каждого школьника известно, какие фильмы и в каком порядке он посмотрел (естественно, если фильм кому-то из ребят сильно понравился, он мог его пересмотреть несколько раз). Так как друзья любят смотреть фильмы вместе, вдвоем они посмотрели максимальное возможное количество. Сколько же раз они встречались вместе?

## Формат входных данных

В первой строке входного файла заданы три числа:  $N$ ,  $M$  и  $K$  — количества фильмов, просмотренных каждым из друзей ( $1 \leq N, M, K \leq 200$ ). В следующих трех строках выписаны номера фильмов, просмотренных ими. У друзей огромные планы на будущее, поэтому фильмы нумеруются числами между 1 и  $10^9$ .

## Формат выходных данных

В первой строке выведите единственное число — максимальное кол-во просмотренных фильмов.

## Примеры

<code>cinema.in</code>	<code>cinema.out</code>
3 3 4 1 2 3 1 3 10 3 1 10 3	2

# Строка

Имя входного файла: `lcstr.in`  
Имя выходного файла: `lcstr.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дед Мороз составил важное сообщение, которое нужно передать другим Дедам Морозам. На всякий случай Дед Мороз записал свое сообщение два раза.

Потом он ненадолго отвлекся, а в это время один из эльфов прибежал в дом и решил пошалить. Он взял и приписал к сообщениям спереди и сзади какую-то абракадабру.

Теперь Деду Морозу надо найти наибольшую подстроку, которая могла бы быть его сообщением. Помогите ему.

Например, пусть сообщение Деда Мороза имело вид `'xxz'`. После этого эльф приписал к первому сообщению спереди букву `a`, сзади — ничего; ко второму — спереди ничего, сзади `b`. В итоге получились строки `'axxz'` и `'xxzb'`.

## Формат входных данных

Входной файл содержит две строки, состоящие из строчных латинских букв, длиной не более 5000. Гарантируется, что обе строки во входном файле непустые.

## Формат выходных данных

Выведите строку — предполагаемое сообщение.

## Примеры

<code>lcstr.in</code>	<code>lcstr.out</code>
<code>caa</code> <code>aab</code>	<code>aa</code>

# Банкомат

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

В некотором государстве в обращении находятся банкноты определенных номиналов. Национальный банк хочет, чтобы банкомат выдавал любую запрошенную сумму при помощи минимального числа банкнот, считая, что запас банкнот каждого номинала неограничен. Помогите Национальному банку решить эту задачу.

## Формат входных данных

Первая строка входных данных содержит натуральное число  $N$  не превосходящее 100 — количество номиналов банкнот в обращении. Вторая строка входных данных содержит  $N$  различных натуральных чисел  $x_1, x_2, \dots, x_N$ , не превосходящих  $10^4$  — номиналы банкнот. Третья строка содержит натуральное число  $S$ , не превосходящее  $10^4$  — сумму, которую необходимо выдать.

## Формат выходных данных

В первую строку выходного файла выведите минимальное число слагаемых (или -1, если такого представления не существует). Во вторую строку выведите это представление в любом порядке.

## Примеры

atm.in	atm.out
5 1 3 7 12 32 40	3 32 7 1

# Сокровища

Имя входного файла: `dowry.in`  
Имя выходного файла: `dowry.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дочь короля Флатландии собирается выйти за прекрасного принца. Принц хочет подарить принцессе сокровища, но он не уверен какие именно бриллианты из своей коллекции выбрать.

В коллекции принца  $n$  бриллиантов, каждый характеризуется весом  $w_i$  и стоимостью  $v_i$ . Принц хочет подарить наиболее дорогие бриллианты, однако король умен и не примет бриллиантов суммарного веса больше  $R$ . С другой стороны, принц будет считать себя жадным всю оставшуюся жизнь, если подарит бриллиантов суммарным весом меньше  $L$ .

Помогите принцу выбрать набор бриллиантов наибольшей суммарной стоимости, чтобы суммарный вес был в отрезке  $[L, R]$ .

## Формат входных данных

Первая строка содержит число  $n$  ( $1 \leq n \leq 32$ ),  $L$  и  $R$  ( $0 \leq L \leq R \leq 10^{18}$ ). Следующие  $n$  строк описывают бриллианты и содержит по два числа — вес и стоимость соответствующего бриллианта ( $1 \leq w_i, v_i \leq 10^{15}$ ).

## Формат выходных данных

Первая строка вывода должна содержать  $k$  — количество бриллиантов, которые нужно подарить принцессе. Вторая строка должна содержать номера даримых бриллиантов.

Бриллианты нумеруются от 1 до  $n$  в порядке появления во входных данных.

Если составить подарок принцессе невозможно, то выведите 0 в первой строке вывода.

## Примеры

dowry.in	dowry.out
3 6 8	1
3 10	2
7 3	
8 2	

# Получите в сумме ноль

Имя входного файла: `zerosum.in`  
Имя выходного файла: `zerosum.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 146 мегабайт

Даны целые неотрицательные числа  $a_1, a_2, \dots, a_n$ . Определите, какой знак следует поставить перед каждым из них, чтобы результат сложения данных чисел с учётом выбранных знаков равнялся нулю. Гарантируется, что способ расставить знаки таким образом существует.

## Формат входных данных

В первой строке записано целое число  $n$  ( $1 \leq n \leq 100$ ). Во второй строке через пробел записаны числа  $a_i$  ( $0 \leq a_i \leq 500$ ).

## Формат выходных данных

Выведите единственную строку из  $n$  символов.  $i$ -й символ строки должен соответствовать знаку, который требуется поставить в сумме перед  $i$ -м числом, чтобы получить в результате ноль.

## Примеры

<code>zerosum.in</code>	<code>zerosum.out</code>
4 2 3 0 1	-+--