

Задача А. Культ-орки на лестнице

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

В Летней Кинематографической Школе пришло время обеда, и эльф Коля поспешил в столовую. Однако для того, чтобы попасть в столовую, Коле нужно подняться по длинной лестнице, а на каждой её ступеньке в это время суток стоит по культуорку. Каждый культуорк разрешает Коле пройти по своей ступеньке только после того, как Коля запишется на мероприятие, которое тот культуорк организует. При этом никакие два культуорка не проводят одно и то же мероприятие, и все мероприятия проходят в разное время.

Коля — честный эльф, и если уж он запишется на какую игру или конкурс, то потом обязательно придёт поучаствовать. Однако Коля хочет тратить как можно меньше времени на развлечения, ведь иначе ему некогда будет дорешивать кинематографические задачки. К счастью, Коле не надо наступать на каждую ступеньку, он может перепрыгнуть через несколько. Коля хочет узнать, какое минимальное количество времени ему придётся распланировать за один проход по лестнице до столовой.

Формат входного файла

В первой строке входного файла вводятся два целых числа N и K ($1 \leq N \leq 10\,000$, $0 \leq K \leq 20$). N — количество ступенек на лестнице. K — максимальное количество ступенек, через которые Коля может перепрыгнуть за один прыжок (то есть, например, на первом шагу Коля может прыгнуть на $(K + 1)$ -ю или более низкие ступени). Во второй строке вводятся N натуральных чисел: i -е число указывает на длительность в минутах того мероприятия, которое проведёт культуорк, стоящий на i -й ступеньке. Каждое мероприятие не может длиться больше 24 часов. Ступеньки нумеруются снизу вверх, ступенькой с номером N считается весь этаж столовой.

Формат выходного файла

Выведите одно число — минимальное количество минут, которое Коле придётся распланировать.

Примеры

<code>muggers.in</code>	<code>muggers.out</code>
5 2 7 3 9 2 11	14

Задача В. Калькулятор

Имя входного файла: `calcul.in`
Имя выходного файла: `calcul.out`
Ограничение по времени: 3 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 входного файла

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

Формат выходного файла

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

<code>calcul.in</code>	<code>calcul.out</code>
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

Задача С. Черепахоконь

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

Дана клетчатая доска размером $N \times M$ в каждой клетке которой записано натуральное число. В верхнем левом углу доски сидит черепашка. Черепашка умеет делать ход конём по направлению вниз и вправо. То есть либо перемещаться на одну клетку вправо и на две вниз, либо на одну клетку вниз и на две вправо. Помогите черепашке добраться в правый нижний угол доски, собрав максимальную сумму чисел. Считается, что черепашка собирает только те числа, на которых завершает ход, а не все, по которым проползает.

Формат входного файла

В первой строке входного файла два целых числа N и M ($1 \leq N, M \leq 100$), задающие размеры доски. Далее следуют числа, записанные на доске — N строк по M положительных чисел, не превышающих 10 000, в каждой.

Формат выходного файла

Выведите одно число, равное искомой максимальной сумме, либо -1, если черепашка не может добраться до правого нижнего угла.

Примеры

knight.in	knight.out
2 3 3 2 7 1 9 5	8

Задача D. Черепашка

Имя входного файла: `turtle.in`
Имя выходного файла: `turtle.out`
Ограничение по времени: 5 секунд
Ограничение по памяти: 64 мегабайта

В левом верхнем углу прямоугольной таблицы размером $N \times M$ находится черепашка. На каждой клетке этой таблицы разлито некоторое количество кислоты. Черепашка может перемещаться вправо или вниз, при этом маршрут черепашки заканчивается в правом нижнем углу таблицы.

Каждый миллилитр кислоты наносит черепашке некоторое количество урона. Найдите наименьшее возможное значение урона, который получит черепашка после прогулки по таблице.

Формат входного файла

В первой строке входных данных записаны два натуральных числа N и M , не превосходящих 1000 — размеры таблицы. Далее идет N строк, каждая из которых содержит M натуральных чисел, разделенных пробелами — описание таблицы с указанием для каждой клетки содержания кислоты на ней (в миллилитрах). Гарантируется, что содержание кислоты в клетке не превышает 250 миллилитров.

Формат выходного файла

Программа должна вывести единственное число: минимальную возможную стоимость маршрута черепашки.

Примеры

turtle.in	turtle.out
3 4 5 9 4 3 3 1 6 9 8 6 8 12	35
1 1 1	1

Задача E. Черепашка: восстановление

Имя входного файла: `turtle-way.in`
Имя выходного файла: `turtle-way.out`
Ограничение по времени: 5 секунд
Ограничение по памяти: 64 мегабайта

Черепашка хотела бы пройти по прямоугольной таблице из левого верхнего угла в правый нижний по маршруту с наименьшими потерями. Помогите ей.

Формат входного файла

В первой строке входных данных записаны два натуральных числа N и M , не превосходящих 1000 — размеры таблицы. Далее идет N строк, каждая из которых содержит M натуральных чисел, разделённых пробелами — описание таблицы с указанием для каждой клетки содержания кислоты на ней (в миллилитрах). Гарантируется, что количество кислоты в каждой клетке не превышает 250 миллилитров.

Формат выходного файла

В первой строчке выведите одно целое число — минимальный возможный урон для черепашки. В следующих строчках выведите координаты клеток, по которым пролегает соответствующий путь. Координаты следует выводить в том порядке, в котором они встречаются в пути.

Примеры

turtle-way.in	turtle-way.out
3 4 5 9 4 3 3 1 6 9 8 6 8 12	35 1 1 2 1 2 2 3 2 3 3 3 4
1 1 1	1 1 1