

Задача А. Покупка билетов

Имя входного файла: `tickets.in`
Имя выходного файла: `tickets.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

За билетами на премьеру нового мюзикла выстроилась очередь из N человек, каждый из которых хочет купить 1 билет. На всю очередь работала только одна касса, поэтому продажа билетов шла очень медленно, приводя «постояльцев» очереди в отчаяние. Самые сообразительные быстро заметили, что, как правило, несколько билетов в одни руки кассир продаёт быстрее, чем когда эти же билеты продаются по одному. Поэтому они предложили нескольким подряд стоящим людям отдавать деньги первому из них, чтобы он купил билеты на всех.

Однако для борьбы со спекулянтами кассир продавала не более 3-х билетов в одни руки, поэтому договориться таким образом между собой могли лишь 2 или 3 подряд стоящих человека.

Известно, что на продажу i -му человеку из очереди одного билета кассир тратит A_i секунд, на продажу двух билетов — B_i секунд, трех билетов — C_i секунд. Напишите программу, которая подсчитает минимальное время, за которое могли быть обслужены все покупатели.

Обратите внимание, что билеты на группу объединившихся людей всегда покупает первый из них. Также никто в целях ускорения не покупает лишних билетов (то есть билетов, которые никому не нужны).

Формат входных данных

Во входном файле записано сначала число N — количество покупателей в очереди ($1 \leq N \leq 5000$). Далее идет N троек натуральных чисел A_i , B_i , C_i . Каждое из этих чисел не превышает 3600. Люди в очереди нумеруются начиная от кассы.

Формат выходных данных

В выходной файл выведите одно число — минимальное время в секундах, за которое могли быть обслужены все покупатели.

Пример

<code>tickets.in</code>	<code>tickets.out</code>
5 5 10 15 2 10 15 5 5 5 20 20 1 20 1 1	12

Задача В. Калькулятор

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

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

<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. Черепашка

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

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 входных данных

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 выходных данных

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

Примеры

стандартный ввод	стандартный вывод
3 4 5 9 4 3 3 1 6 9 8 6 8 12	35
1 1 1	1

Задача D. Кролик учит геометрию.

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

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.

Пример

<code>rabbits.in</code>	<code>rabbits.out</code>
4 5 0 0 0 1 0 0 1 1 1 0 0 0 1 1 0 1 0 1 0 0	2

Задача Е. Взрывоопасные контейнеры

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

Все контейнеры в мире делятся на две категории — с тротилом и без. Только глупец поставит ящик с тротилом на другой ящик с тротилом. Поскольку вы не глупец, вы точно знаете, что тротил взрывается, особенно если на нем стоит еще один ящик с тротилом. Сколько существует способов построить башню из N ящиков так, чтобы она не взорвалась? Все ящики с тротилом одинаковы, все ящики без тротила также одинаковы.

Формат входных данных

В первой строчке находится единственное целое число N ($1 \leq N \leq 45$).

Формат выходных данных

Выведите одно число — количество способов построить невзрывающуюся башню из ящиков двух типов так, чтобы ящик с тротилом никогда не стоял на другом ящике с тротилом.

Примеры

<code>containers.in</code>	<code>containers.out</code>
1	2
2	3

Задача F. Числа фиксированной суммы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Сколько существует K -значных чисел с суммой цифр равной S ? Числа берутся в десятичной системе счисления. Ведущие нули допустимы.

Формат входных данных

K ($1 \leq K \leq 1000$) и S ($0 \leq S \leq 9 \cdot K$).

Формат выходных данных

Количество таких чисел, посчитанное по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
3 10	63

Задача G. Плавные числа

Имя входного файла: `numbers.in`
Имя выходного файла: `numbers.out`
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.

Пример

<code>numbers.in</code>	<code>numbers.out</code>
2	26

Задача Н. Лестница

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

У лестницы n ступенек, пронумерованных числами $1, 2, \dots, n$ снизу вверх. На каждой ступеньке написано число. Начиная с подножия лестницы (его можно считать ступенькой с номером 0), требуется взобраться на самый верх (ступеньку с номером n). За один шаг можно подниматься на одну или на две ступеньки. После подъёма числа, записанные на посещённых ступеньках, складываются. Нужно подняться по лестнице так, чтобы сумма этих чисел была как можно больше.

Формат входных данных

В первой строке входного файла записано целое число n ($1 \leq n \leq 100$). Во второй строке заданы целые числа a_1, a_2, \dots, a_n через пробел ($-10\,000 \leq a_i \leq 10\,000$) — это числа, записанные на ступеньках.

Формат выходных данных

В первой строке выходного файла выведите одно число — максимальную сумму, которую можно получить, поднявшись по данной лестнице.

Пример

<code>ladder.in</code>	<code>ladder.out</code>
2 1 2	3

Задача I. Рюкзак

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Найдите максимальный вес золота, который можно унести в рюкзаке вместительностью S , если есть n золотых слитков с заданными весами.

Формат входных данных

В первой строке задано два целых числа n и S ($1 \leq n \leq 300$; $1 \leq S \leq 10\,000$).

Далее следует n неотрицательных целых чисел, не превосходящих 100 000 — веса слитков.

Формат выходных данных

Выведите искомый максимальный вес.

Примеры

стандартный ввод	стандартный вывод
10 3 1 4 8	9
20 4 5 7 12 18	19

Задача J. Зайчик

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

Зайчик прыгает по прямой просеке, для удобства разделённой на n клеток. Клетки пронумерованы по порядку натуральными числами от 1 до n . Некоторые клетки заболочены: если зайчик прыгнет на такую клетку, ему несдобровать. Некоторые другие клетки просеки поросли вкусной зелёной травой: прыгнув на такую клетку, зайчик сможет отдохнуть и подкрепиться.

Зайчик начинает свой путь из клетки с номером 1 и хочет попасть в клетку с номером n , по пути ни разу не провалившись в болото и скушав как можно больше вкусной зелёной травы. Конструктивные особенности зайчика таковы, что из клетки с номером k он может прыгнуть лишь в клетки с номерами $k + 1$, $k + 3$ и $k + 5$.

Выясните, какое максимальное количество клеток с травой сможет посетить зайчик на своём пути.

Формат входных данных

В первой строке входного файла задано число n — количество клеток ($2 \leq n \leq 1000$). Вторая строка состоит из n символов; i -ый символ соответствует i -ой клетке просеки. Символ 'w' обозначает болото, символ '.' — зелёную траву, а символ '.' соответствует клетке без каких-либо особенностей. Гарантируется, что первая и последняя клетки не содержат болот и травы.

Формат выходных данных

В первой строке выходного файла выведите одно число — максимальное количество клеток с травой, которые зайчик сможет посетить на своём пути. Если зайчику не удастся оказаться в клетке с номером n , выведите «-1».

Примеры

lepus.in	lepus.out
4 ..	2
5 .w".	0

Задача К. Кузнечик

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

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.

Пример

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