

Динамическое программирование.

Вводные задачи

A. Мячик на лесенке

На вершине лесенки, содержащей N ступенек, находится мячик, который начинает прыгать по ним вниз, к основанию. Мячик может прыгнуть на следующую ступеньку, на ступеньку через одну или через 2. (То есть, если мячик лежит на 8-ой ступеньке, то он может переместиться на 5-ую, 6-ую или 7-ую.) Определить число всевозможных «маршрутов» мячика с вершины на землю.

Вводится одно число $0 < N < 31$.

Выведите одно число — количество маршрутов.

Input	Output
4	7

B. Последовательность из 0 и 1

Требуется подсчитать количество последовательностей длины N , состоящих из 0 и 1, в которых никакие две единицы не стоят рядом.

На вход программы поступает целое число N ($1 \leq N \leq 10^5$).

Выведите количество искомых последовательностей по модулю $10^9 + 7$.

Input	Output
1	2

C. Без трёх единиц

Определите количество последовательностей из нулей и единиц длины N (длина — это общее количество нулей и единиц), в которых никакие три единицы не стоят рядом.

Вводится натуральное число N , не превосходящее 40.

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

Input	Output
3	7

D. Платная лестница

Мальчик подошел к платной лестнице. Чтобы наступить на любую ступеньку, нужно заплатить указанную на ней сумму. Мальчик умеет перешагивать на следующую ступеньку, либо перепрыгивать через ступеньку. Требуется узнать, какая наименьшая сумма понадобится мальчику, чтобы добраться до верхней ступеньки.



В первой строке входного файла вводится одно натуральное число $N \leq 100$ — количество ступенек. В следующей строке вводятся N натуральных чисел, не превосходящих 100 — стоимость каждой ступеньки (снизу вверх).

Выведите одно число — наименьшую возможную стоимость прохода по лесенке.

Input	Output
3	2
1 3 1	

Одномерное динамическое программирование

Е. Радиоактивные материалы

При переработке радиоактивных материалов образуются отходы трех видов — особо опасные (тип А), неопасные (тип В) и совсем не опасные (тип С). Для их хранения используются одинаковые контейнеры. После помещения отходов в контейнеры последние укладываются вертикальной стопкой. Стопка считается взрывоопасной, если в ней подряд идет более одного контейнера типа А. Стопка считается безопасной, если она не является взрывоопасной.

Для заданного количества контейнеров N определить число безопасных стопок.

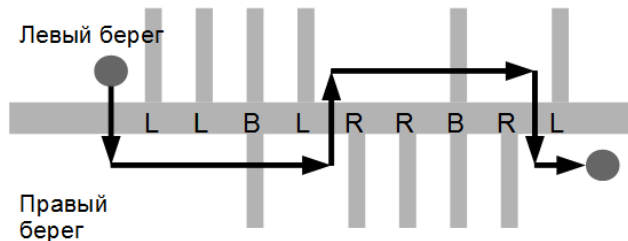
На вход программе подаётся одно число $1 \leq N \leq 10^5$.

Программа должна вывести одно число — количество безопасных вариантов формирования стопки, взятое по модулю $10^9 + 7$.

Input	Output
2	8

Ф. Поход вдоль реки

Дети пошли в поход вдоль реки. Поход начинается на левом берегу, заканчивается на правом. Дана последовательность букв 'L', 'R', 'B', означающая с какой стороны в реку впадает приток: 'L' — слева (Left), 'R' — справа (Right), 'B' — с обеих сторон (Both). Определить минимальное количество переправ, которое придётся сделать школьникам. Обратите внимание на то, что школьникам требуется пройти вдоль реки *дальше* последнего притока. То есть все притоки должны располагаться между начальной и конечной точками маршрута.



На вход программа получает строку, содержащую только символы R, L, B в произвольном порядке. Длина строки не превосходит 10^5 символов.

Выведите одно целое число — минимальное количество переправ.

Input	Output
LLBLRRBRL	5

Г. Гвоздики

В дощечке в один ряд вбиты гвоздики. Любые два гвоздика можно соединить ниточкой. Требуется соединить некоторые пары гвоздиков ниточками так, чтобы к каждому гвоздику была привязана хотя бы одна ниточка, а суммарная длина всех ниточек была минимальна.

В первой строке входных данных записано число N — количество гвоздиков ($2 \leq N \leq 100$). В следующей строке заданы N неотрицательных целых чисел, не превосходящие 10000 — координаты всех гвоздиков в порядке возрастания.

Выведите единственное число — минимальную суммарную длину всех ниточек.

Input	Output
5 0 2 4 10 12	6

Н. Покупка билетов

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

подсчитывает 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).

На вход программы поступает 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.

Требуется вывести одно число — минимальное время в секундах, за которое могли быть обслужены все покупатели.

Input	Output
5 5 10 15 2 10 15 5 5 5 20 20 1 20 1 1	12
4 100 2 100 100 2 100 100 57 100 100 1 100	59

Комментарий ко второму тесту: после того, как первый покупатель взял два билета, заплатив 2, второй покупатель не должен ничего покупать, так как на него билет уже куплен.

I. Калькулятор с восстановлением ответа

Имеется калькулятор, который выполняет три операции:

1. прибавить к числу X единицу
2. умножить число X на 2
3. умножить число X на 3

Определите кратчайшую последовательность операций, необходимую для получения из числа 1 заданного числа N ($1 \leq N \leq 10^6$).

Выведите строку, состоящую из цифр 1, 2 или 3, обозначающих одну из трех указанных операций, которая получает из числа 1 число N за минимальное число операций. Если возможных минимальных решений несколько, выведите любое из них.

Input	Output
1	
5	121
562340	3333312222122213312

J. Наибольшая возрастающая подпоследовательность за $O(n^2)$

Дана последовательность чисел длины n . Определить длину наибольшей возрастающей подпоследовательности этой последовательности.

В этой задаче предполагается решение со сложностью $O(n^2)$.

В первой строке входных данных задано число N — длина последовательности ($1 \leq N \leq 1000$).

Во второй строке задается сама последовательность — целые числа, не превосходящие 10000 по модулю.

Требуется вывести длину наибольшей строго возрастающей подпоследовательности.

Input	Output
6 3 29 5 5 28 6	3

К. *Наибольшая возрастающая подпоследовательность с восстановлением ответа*

Дана последовательность чисел длины n . Определить наибольшую возрастающую подпоследовательность этой последовательности.

В этой задаче предполагается использовать решение предыдущей задачи со сложностью $O(n^2)$, но вместо длины вывести пример такой подпоследовательности. Если есть несколько подпоследовательностей максимальной длины — вывести любую.

Input	Output
6 3 29 5 5 28 6	3 5 28

L* *Две последовательности*

Дан массив целых чисел. Требуется выделить из него строго возрастающую подпоследовательность таким образом, чтобы оставшиеся элементы массива образовывали строго убывающую подпоследовательность.

В первой строке вводится натуральное число N ($2 \leq N \leq 10^5$).

Во второй строке задано N целых чисел.

В первой строке программа должна вывести число K ($1 \leq K \leq N - 1$) — длину найденной строго возрастающей подпоследовательности. Во второй строке программа должна вывести K чисел — номера элементов массива, образующих строго возрастающую подпоследовательность. Оставшаяся подпоследовательность должна быть строго убывающей.

Если это невозможно, выведите число -1 .

Input	Output
6 6 1 4 3 2 5	3 2 4 6
6 1 3 2 4 6 5	-1
3 1 1 1	-1

М. *Количество сумм*

Дано N целых чисел A_1, A_2, \dots, A_N .

Требуется найти количество различных сумм вида $k_1A_1 + k_2A_2 + \dots, k_NA_N, k_i \in \{0, 1\}$.

В первой строке вводится натуральное число N ($1 \leq N \leq 100$).

Во второй строке вводится N целых чисел через пробел. Вводимые числа неотрицательные целые и не превосходят 100.

Программа должна вывести одно целое число — ответ на вопрос задачи.

Input	Output
3 1 1 2	5
3 1 3 2	7
5 49 100 98 49 0	10

Н. *Сообщение*

В сообщении, состоящем из одних русских букв и пробелов, каждую букву заменили её порядковым номером в русском алфавите (А — 1, Б — 2, ..., Я — 33), а пробел — нулём. Требуется по заданной последовательности цифр найти количество исходных сообщений, из которых она могла получиться.

В первой строке содержится последовательность цифр, длина которой не превосходит 100.

Вывести одно число — ответ на вопрос задачи.

Input	Output
80946	1
21705	3

О. *Количество чисел*

Вова вывел в выходной файл числа, забыв разделить их пробелами. Увидев полученный результат, Вова задумался над следующим вопросом: сколько существует различных последовательностей неотрицательных целых чисел, таких что, если выписать их без пробелов, то получится тот же результат, что и у него. Он вспомнил также, что его программа выводила не произвольные числа, а только те, что не превосходят C и не имеют ведущих нулей.

Требуется написать программу, которая позволит найти число различных последовательностей неотрицательных целых чисел, в каждой из которых любое число не превосходит C . В качестве ответа выведите последних K цифр полученного результата.

Первая строка входного файла содержит три целых числа — N, C и K ($1 \leq N \leq 50000, 1 \leq C \leq 10^8, 1 \leq K \leq 18$). Во второй строке этого файла содержится результат работы Вовиной программы, состоящий из N цифр.

В выходной файл выведите последние K цифр искомого количества последовательностей (без ведущих нулей).

Input	Output
3 11 2 111	3
19 9 1 0123456789876543210	1
1 8 3 9	0

Р. *Заполнение массива*

Требуется заполнить целочисленный массив из N элементов натуральными числами от 2 до $N+1$, используя каждое число ровно один раз, так, чтобы значение каждого элемента массива делилось бы нацело на его порядковый номер (не индекс, а именно порядковый номер: т.е. для каждого i число $A[i]$ должно делиться на $i+1$).

Напишите программу, которая для заданного N вычислит количество способов такого заполнения массива.

Вводится одно натуральное число N ($1 \leq N \leq 60000$).

Input	Output
2	1

Комментарий к тесту: массив длины 2 можно заполнить единственным способом: 3 2.

Q. *Заполнение массива - 2*

Решите предыдущую задачу для $N < 10^9$.

Input	Output
116640000	3

Р. *Валютные махинации*

Петя, изучая, как меняется курс рубля по отношению к доллару и евро, вывел закон, по которому происходят эти изменения (или думает, что вывел). По этому закону Петя рассчитал, каков будет курс рубля по отношению к доллару и евро в ближайшие N дней.

У Пети есть 100 рублей. В каждый из дней он может обменивать валюты друг на друга по текущему курсу без ограничения количества (при этом курс доллара по отношению к евро соответствует величине, которую можно получить, обменяв доллар на рубли, а потом эти рубли — на евро). Поскольку Петя будет оперировать не с наличной валютой, а со счетом в банке, то он может совершать операции обмена с любым (в том числе и нецелым) количеством единиц любой валюты.

Напишите программу, которая вычисляет, какое наибольшее количество рублей сможет получить Петя к исходу N -го дня.

Законы изменения курсов устроены так, что в течение указанного периода рублевый эквивалент той суммы, которая может оказаться у Пети, не превысит 10^8 рублей.

Первая строка входного файла содержит одно число N ($1 \leq N \leq 5000$). В каждой из следующих N строк записано по 2 числа, вычисленных по Петиним законам для соответствующего дня — сколько рублей будет стоить 1 доллар, и сколько рублей будет стоить 1 евро. Все эти значения задаются вещественными числами, не меньше 0.01 и не больше 10000.

В выходной файл выведите искомую величину с точностью не менее двух знаков после десятичной точки.

Input	Output
3 4 5 2 7 5 2	350.00
1 30.00 9999.99	100.00

S. *Бросание кубика*

Кубик, грани которого помечены цифрами от 1 до 6, бросают N раз. Определите вероятность того, что сумма выпавших чисел будет в точности равна Q .

Программа получает на вход два целых числа: N и Q ($1 \leq N \leq 500, 1 \leq Q \leq 3000$).

Программа должна вывести единственное действительное число: искомую вероятность с точностью не менее чем 10^{-6} .

Примечание: искомая вероятность равна отношению числа последовательностей длины N , составленных из чисел от 1 до 6, дающих в сумме число Q , к числу всех таких последовательностей длины N .

Input	Output
2 2	0.0277778
2 12	0.0277778