

Содержание

09.Base [4/4]	3
Задача 9A. Число Фибоначчи [0.5 sec, 256 mb]	3
Задача 9B. Калькулятор [0.5 sec, 256 mb]	4
Задача 9C. Поле [0.5 sec (1 sec), 256 mb]	5
Задача 9D. Лестница [0.5 sec, 256 mb]	6
09.Advanced [4/7]	7
Задача 9E. Кролик [0.5 sec, 256 mb]	7
Задача 9F. Зайчик [0.5 sec, 256 mb]	8
Задача 9G. Наибольшая общая подпоследовательность [0.5 sec (1 sec), 256 mb]	9
Задача 9H. Редакционное расстояние [0.5 sec, 256 mb]	10
Задача 9I. Короля — в угол 3 [0.5 sec, 256 mb]	11
Задача 9J. НПП [0.2 sec (0.5 sec), 256 mb]	13
Задача 9K. Лабиринт знаний [0.7 sec (1.5 sec), 256 mb]	14
09.Hard [0/2]	15
Задача 9L. Новая модель телефона [0.2 sec (0.5 sec), 256 mb]	15
Задача 9M. Неприводимые многочлены [0.2 sec (0.5 sec), 256 mb]	16

Общая информация:

Вход в констест: <http://contest.yandex.ru/contest/877/>

Дедлайн на задачи 26-го ноября в 23:59.

К каждой главе есть более простые задачи (base), посложнее (advanced), и сложные (hard).

В скобках к каждой главе написано сколько любых задач из этой главы нужно сдать.

Сайт курса: <http://compscicenter.ru/courses/algorithms-1/2014-autumn/>

Семинары ведет Сергей Владимирович Копелиович,
контакты: burunduk30@gmail.com, vk.com/burunduk1

В каждом условии 2 таймлимита: для C/C++ и для Java, Python.

09.Base [4/4]

Задача 9А. Число Фибоначчи [0.5 сек, 256 mb]

Числа Фибоначчи $F_0, F_1, F_2, \dots, F_n$ определяются следующим образом: $F_0 = F_1 = 1$, а для любого $n > 1$ выполнено равенство $F_n = F_{n-1} + F_{n-2}$.

По заданному числу n выведите число Фибоначчи F_n .

Формат входных данных

В первой строке входного файла задано единственное число n ($0 \leq n \leq 91$).

Формат выходных данных

Выведите число F_n в первой строке выходного файла.

Примеры

fiblong.in	fiblong.out
1	1
2	2
3	3
4	5

Задача 9В. Калькулятор [0.5 сек, 256 mb]

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 .

Примеры

calcul.in	calcul.out
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 9С. Поле [0.5 sec (1 sec), 256 mb]

Отряду нужно пересечь прямоугольное поле размера $m \times n$ квадратов, двигаясь из левого верхнего угла в правый нижний и перемещаясь между соседними квадратами только в двух направлениях — вправо и вниз. Поле не очень ровное, но у отряда есть карта, на которой отмечена высота каждого квадрата. Опасность перехода с квадрата высоты h_1 на соседний квадрат высоты h_2 оценивается числом $|h_2 - h_1|$; опасность всех переходов в пути суммируется. Выясните, какова минимальная опасность пути из квадрата $(1, 1)$ в квадрат (m, n) .

Формат входных данных

В первой строке входного файла заданы два числа m и n через пробел ($1 \leq m, n \leq 100$). В следующих n строках записано по m чисел в каждой; i -ое число j -ой из этих строк соответствует высоте квадрата (i, j) . Все высоты — целые числа в диапазоне от 1 до 100, включительно.

Формат выходных данных

Выведите в выходной файл одно число — минимальную опасность пути из квадрата $(1, 1)$ в квадрат (m, n) .

Примеры

field.in	field.out
2 2 1 1 1 1	0
4 2 1 2 3 5 3 8 4 7	6
2 3 1 2 2 3 3 1	4

Задача 9D. Лестница [0.5 sec, 256 mb]

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.

Примеры

ladder.in	ladder.out
2 1 2	3
2 2 -1	1
3 -1 2 1	3

09.Advanced [4/7]

Задача 9Е. Кролик [0.5 sec, 256 mb]

Как и зайчик, кролик прыгает по прямой просеке, для удобства разделённой на n клеток. Клетки пронумерованы по порядку натуральными числами от 1 до n . Некоторые клетки заболочены: прыгать в них кролик боится. Некоторые другие клетки просеки поросли вкусной зелёной травой: прыгнув на такую клетку, кролик прихватит с собой пучок травы.

Кролик начинает свой путь из клетки с номером 1 и хочет попасть в клетку с номером n , по пути ни разу не провалившись в болото. Конструктивные особенности кролика таковы, что из клетки с номером k он может прыгнуть лишь в клетки с номерами $k + 2$, $k + 3$ и $k + 6$.

Поскольку в клетке с номером n кролика ждёт его крольчиха, главное для него — оказаться там как можно быстрее, то есть сделав как можно меньше прыжков. Если маршрутов с минимальным количеством прыжков несколько, кролик предпочтёт тот из них, на котором он посетит как можно больше клеток с травой. Если и таких маршрутов несколько, кролика устроит любой из них.

По данной карте просеки найдите оптимальный маршрут для кролика.

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 через пробел два числа — минимальное количество прыжков k , за которое кролик может попасть из первой клетки просеки в последнюю, и максимальное количество клеток с травой t , которые он при этом сможет посетить. Во второй строке выведите через пробел $k + 1$ число — номера клеток, которые должен посетить кролик, в порядке их посещения.

Если кролику не удастся оказаться в клетке с номером n , выведите в выходной файл одно число -1 .

Примеры

rabbit.in	rabbit.out
4 ." ".	1 0 1 4
5 .w". .	2 1 1 3 5
9 .www.www.	-1

Задача 9F. Зайчик [0.5 сек, 256 mb]

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
9 .www.www.	-1

Задача 9G. Наибольшая общая подпоследовательность [0.5 sec (1 sec), 256 mb]

Даны две последовательности. Найдите длину их наибольшей общей подпоследовательности (подпоследовательность — это то, что можно получить из данной последовательности вычеркиванием некоторых элементов).

Формат входных данных

В первой строке входного файла записано число N — длина первой последовательности ($1 \leq N \leq 1000$). Во второй строке записаны члены первой последовательности (через пробел) — целые числа, не превосходящие 10 000 по модулю. В третьей строке записано число M — длина второй последовательности ($1 \leq M \leq 1000$). В четвертой строке записаны члены второй последовательности (через пробел) — целые числа, не превосходящие 10 000 по модулю.

Формат выходных данных

В выходной файл требуется вывести единственное целое число: длину наибольшей общей подпоследовательности, или число 0, если такой не существует.

Примеры

lcs.in	lcs.out
3 1 2 3 4 2 1 3 5	2

Задача 9H. Редакционное расстояние [0.5 sec, 256 mb]

В информатике *редакционным расстоянием* между двумя строками называется минимальное количество добавлений, удалений и замен символов, при помощи которых можно из одной строки получить другую. К примеру, редакционное расстояние между строками “ab” и “ab” равно нулю, так как строки равны между собой безо всяких изменений; расстояние между строками “short” и “ports” равно трём: в слове “short” нужно удалить из начала букву ‘s’, заменить ‘h’ на ‘p’ и добавить в конец букву ‘s’. Редакционное расстояние также называют *расстоянием Левенштейна*.

Найдите редакционное расстояние между двумя заданными строками.

Формат входных данных

В первой строчке входного файла задана одна строка, во второй — другая.

Длины строк от 1 до 100.

Формат выходных данных

В выходной файл выведите единственное число — редакционное расстояние между двумя заданными строками.

Примеры

editdist.in	editdist.out
ab ab	0
short ports	3

Задача 9I. Короля — в угол 3 [0.5 sec, 256 mb]

На каждой клетке шахматной доски размеров 8×8 записано целое неотрицательное число. Двое игроков по очереди переставляют короля, перемещая его только вправо, вверх или по диагонали вправо–вверх. Первоначально король стоит в левом нижнем углу. Игра продолжается до тех пор, пока король не окажется в правой верхней клетке доски. Игрок, переставивший короля в некоторую клетку, получает от другого игрока денежную сумму, равную числу, записанному в этой клетке. Определите стоимость игры — сумму, которая окажется в конце игры у первого игрока, если первый игрок старается её максимизировать, а второй — минимизировать.

Формат входных данных

На вход программе подается восемь строк, каждая строка содержит восемь целых неотрицательных чисел, не превосходящих 1000. В левом нижнем углу всегда записано число 0.

Формат выходных данных

Выведите единственное число — стоимость игры.

Пример

king3.in	king3.out
0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0	4

king3.in	king3.out
0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0	-3
9 9 9 9 9 9 1 9 9 9 9 9 9 1 9 2 9 9 9 9 9 9 1 9 0 9 9 9 9 9 9 9	9

Задача 9J. НПП [0.2 sec (0.5 sec), 256 mb]

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 записано одно натуральное число N ($1 \leq N \leq 1000$) — количество чисел в исходной последовательности. Далее следует N натуральных чисел, не превосходящих $2 \cdot 10^9$ — сама последовательность.

Формат выходных данных

Вывести единственное число, равное искомому количеству.

Примеры

sequence.in	sequence.out
4 3 6 5 12	3

Задача 9К. Лабиринт знаний [0.7 sec (1.5 sec), 256 mb]

Участникам сборов подарили билеты на аттракцион “Лабиринт знаний”. Лабиринт представляет собой N комнат, занумерованных от 1 до N , между некоторыми из которых есть двери. Когда человек проходит через дверь, показатель его знаний изменяется на определенную величину, фиксированную для данной двери. Вход в лабиринт находится в комнате 1, выход — в комнате N . Каждый участник сборов проходит лабиринт ровно один раз и набирает некоторое количество знаний (при входе в лабиринт этот показатель равен нулю). Ваша задача — показать наилучший результат.

Формат входных данных

Первая строка входного файла содержит целые числа N ($1 \leq N \leq 2000$) — количество комнат и M ($1 \leq M \leq 10000$) — количество дверей. В каждой из следующих M строк содержится описание двери — номера комнат, из которой она ведет и в которую она ведет (через дверь в лабиринте можно ходить только в одну сторону), а также целое число, которое прибавляется к количеству знаний при прохождении через дверь (это число по модулю не превышает 10000). Двери могут вести из комнаты в нее саму, между двумя комнатами может быть более одной двери.

Формат выходных данных

В выходной файл выведите “:)” — если можно пройти лабиринт и получить неограниченно большой запас знаний, “:(” — если лабиринт пройти нельзя, и максимальное количество набранных знаний в противном случае.

Пример

maze.in	maze.out
2 2 1 2 5 1 2 -5	5

09.Hard [0/2]

Задача 9L. Новая модель телефона [0.2 sec (0.5 sec), 256 mb]

Компания Gnusmas разработала новую модель мобильного телефона. Основное достоинство этой модели — ударопрочность: её корпус сделан из особого сплава, и телефон должен выдерживать падение с большой высоты.

Компания Gnusmas арендовала n -этажное здание и наняла экспертов, чтобы те при помощи серии экспериментов выяснили, с какой высоты бросать телефон можно, а с какой — нельзя. Один эксперимент заключается в том, чтобы бросить телефон с какого-то этажа и посмотреть, сломается он от этого или нет. Известно, что любой телефон этой модели ломается, если его сбросить с x -го этажа или выше, где x — некоторое целое число от 1 до n , включительно, и не ломается, если сбросить его с более низкого этажа. Задача экспертов заключается в том, чтобы узнать число x и передать его рекламному отделу компании.

Задача осложняется тем, что экспертам предоставлено всего k образцов новой модели телефона. Каждый телефон можно бросать сколько угодно раз, пока он не сломается; после этого использовать его для экспериментов больше не удастся.

Подумав, эксперты решили действовать так, чтобы минимизировать максимально возможное количество экспериментов, которое может потребоваться произвести. Чему равно это количество?

Формат входных данных

В первой строке входного файла записаны через пробел два целых числа n и k — количество этажей в здании и количество предоставленных телефонов ($1 \leq n \leq 100\,000$, $0 \leq k < n$).

Формат выходных данных

В выходной файл выведите единственное число — минимальное количество экспериментов, которое потребуется совершить, чтобы узнать число x и использовать не более k телефонов. Если решить задачу невозможно, выведите вместо этого -1 .

Примеры

newphone.in	newphone.out
4 2	2
4 1	3

В первом примере сначала следует бросить телефон со второго этажа. Если он сломается, то второй бросок следует сделать с первого этажа. В случае поломки станет известно, что $x = 1$. Иначе мы узнаем, что $x = 2$.

Если же при броске со второго этажа телефон не сломался, бросим телефон с третьего этажа. При поломке будет ясно, что $x = 3$. Иначе из условия $1 \leq x \leq 4$ следует, что $x = 4$.

Всего будет сделано два эксперимента. В них будет использовано не более чем два телефона.

Во втором примере следует сначала бросить единственный данный нам телефон с первого этажа, если он не сломается, то со второго, а если опять не сломается, то с третьего. При первой же поломке мы узнаем точное значение x . Если после трёх бросков телефон так и не сломался, то $x = 4$.

Задача 9М. Неприводимые многочлены [0.2 sec (0.5 sec), 256 mb]

Найти количество неприводимых унитарных $f \in \mathbb{Z}_p[x] : \deg f = n$ ($p \in P$).

Напомним, \mathbb{Z}_p — это поле вычетов по модулю p . $\mathbb{Z}_p[x]$ — множество всех многочленов над полем \mathbb{Z}_p , то есть многочленов, коэффициенты которых являются остатками по модулю p .

Ненулевой многочлен называется *унитарным*, если его старший коэффициент равен 1.

Многочлен $f \in \mathbb{Z}_p[x]$ называется *неприводимым*, если не существует таких многочленов $g, h \in \mathbb{Z}_p[x] : \deg g > 0, \deg h > 0$ и $f(x) = g(x)h(x)$.

Формат входных данных

В единственной строке входного файла два числа: простое p и натуральное n .
 $1 \leq p, n \leq 10^9; 1 \leq p^n < 10^{18}$.

Формат выходных данных

Выведите единственное число — количество неприводимых унитарных многочленов степени n над полем остатков по модулю p .

Примеры

irreduce.in	irreduce.out
2 2	1
3 4	18

Замечание

В первом примере единственный такой многочлен — $x^2 + x + 1$.