

Содержание

5.base [3/3]	3
Задача 5A. Число Фибоначчи [0.2 sec, 256 mb]	3
Задача 5B. Поле [0.2 sec, 256 mb]	4
Задача 5C. Функция [0.2 sec, 256 mb]	5
5.advanced [5/8]	6
Задача 5A. Спуск с горы [0.2 sec, 256 mb]	6
Задача 5B. Кролик [0.2 sec, 256 mb]	7
Задача 5C. Зайчик [0.2 sec, 256 mb]	8
Задача 5D. Наибольшая общая подпоследовательность [0.2 sec, 256 mb]	9
Задача 5E. НПП [0.2 sec, 256 mb]	10
Задача 5F. Короля — в угол 3 [0.2 sec, 256 mb]	11
Задача 5G. Joseph Problem [0.2 sec, 256 mb]	13
Задача 5H. Палиндром [0.2 sec, 256 mb]	14
5.hard [0/0]	15
Задача 5A. Беру? Нет, не беру! [0.2 sec, 256 mb]	15
Задача 5B. Наибольшая общая возрастающая [0.4 sec, 256 mb]	16
Задача 5C. Новая модель телефона [0.2 sec, 256 mb]	17

Общая информация:

Вход в констест: <http://contest.yandex.ru/contest/2921/>

Дедлайн на задачи: 9 дней, до 2016-10-15 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: <https://compscicenter.ru/courses/algorithms-1/2016-autumn/>

Семинары ведут Сергей Копелиович (burunduk30@gmail.com, vk.com/burunduk1) и Алексей Кладов (aleksey.kladov@gmail.com).

В каждом условии указан таймлимит для C/C++.

Таймлимит для Java примерно в 2-3 раза больше.

Таймлимит для Python примерно в 6 раз больше.

C++:

Быстрый ввод-вывод.

http://acm.math.spbu.ru/~sk1/algo/input-output/fread_write_export.cpp.html Более подробно про ввод-вывод.

http://acm.math.spbu.ru/~sk1/algo/input-output/cpp_common.html

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) переопределение стандартного аллокатора ускорит вашу программу:

<http://acm.math.spbu.ru/~sk1/algo/memory.cpp.html>

Java:

Быстрый ввод-вывод.

http://acm.math.spbu.ru/~sk1/algo/input-output/java/java_common.html

5.base [3/3]

Задача 5А. Число Фибоначчи [0.2 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 5B. Поле [0.2 сек, 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 5С. Функция [0.2 сек, 256 mb]

Вычислите функцию: $f(n) = \begin{cases} 1 & \text{если } n \leq 2 \\ f(\lfloor 6 * n/7 \rfloor) + f(\lfloor 2 * n/3 \rfloor) & \text{если } n \bmod 2 = 1 \\ f(n - 1) + f(n - 3) & \text{если } n \bmod 2 = 0 \end{cases}$

Формат входных данных

Входные данные содержат натуральное число n ($1 \leq n \leq 10^{12}$).

Формат выходных данных

Выведите значение функции по модулю 2^{32} .

Пример

function.in	function.out
7	10

5.advanced [5/8]

Задача 5А. Спуск с горы [0.2 сек, 256 mb]

В одном из горнолыжных курортов Италии проводятся соревнования по горнолыжному спуску. Каждому спортсмену предстоит скатиться с горы на лыжах. На любом этапе спуска участник получает определенное число очков. После прохождения трассы очки суммируются. Участник, набирающий наибольшее количество очков, выигрывает. Гора представляет собой треугольник, в качестве элементов которого выступают целые числа — очки за прохождение этапа. На каждом уровне спортсмену предоставляется выбор — двигаться вниз влево или вниз вправо. Начало спуска — в самой высокой точке горы, конец в одной из самых низких.

```
  1
 4 3
5 6 7
8 9 0 9
```

Требуется найти максимальное количество очков, которое может набрать спортсмен.

Формат входных данных

Во входном файле содержится целое число n — число этапов ($1 \leq n \leq 100$), далее n строк, каждая из которых характеризует свой уровень. В строке с номером i содержится ровно i целых чисел: a_1, a_2, \dots, a_i ($-100 \leq a_k \leq 100, 1 \leq k \leq i$) — количество очков в каждой из позиций.

Формат выходных данных

В результирующем файле должно находиться искомое целое число.

Пример

slalom.in	slalom.out
4	20
1	
4 3	
5 6 7	
8 9 0 9	

Задача 5B. Кролик [0.2 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 5С. Зайчик [0.2 sec, 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 5D. Наибольшая общая подпоследовательность [0.2 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	2
1 2 3	
4	
2 1 3 5	

Задача 5E. НПП [0.2 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 5F. Короля — в угол 3 [0.2 сек, 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	4
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	
0 0 0 0 0 0 0 0	

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

Задача 5G. Joseph Problem [0.2 sec, 256 mb]

N мальчиков стоят по кругу. Они начинают считать себя по часовой стрелке, счет ведется с единицы. Как только количество посчитанных достигает p , последний посчитанный (p -й) мальчик покидает круг, а процесс счета начинается со следующего за ним мальчика и вновь ведется с единицы.

Последний оставшийся в кругу выигрывает.

Можете ли вы посчитать, номер выигравшего мальчика в исходном кругу? (мальчики нумеруются числами от 1 до N по часовой стрелке, начиная с того самого мальчика, с которого начинался счет).

Формат входных данных

Во входном файле два целых числа — N и P ($1 \leq N, P \leq 10^6$).

Формат выходных данных

Выведите номер выигравшего мальчика.

Пример

	joseph.in		joseph.out
	3 4		2

Задача 5Н. Палиндром [0.2 sec, 256 mb]

Палиндромом называется строка, которая читается одинаково как слева направо, так и справа налево. Требуется найти самый длинный палиндром P , получающийся из данной строки S удалением любого (возможно, нулевого) количества символов.

Формат входных данных

Входной файл содержит строку S , состоящую из строчных латинских букв (a–z). Длина S не превышает 1 000 символов.

Формат выходных данных

Выходной файл должен содержать искомый палиндром. Если таких палиндромов несколько, выведите любой из них.

Примеры

palindr.in	palindr.out
anna	anna
perevorot	ror

5.hard [0/0]

Задача 5А. Беру? Нет, не беру! [0.2 sec, 256 mb]

Саша и Петя играют в весёлую игру. Петя рисует на листе бумаги таблицу $m \times n$ и заполняет её целыми числами. После этого Саша ставит свою фишку на клетку $(1, 1)$ и может каждым своим ходом двигать её вправо или вниз. Кроме того, если Сашина фишка находится на числе большем, чем все числа, которые он брал до этого, Саша может сказать «беру», и тогда Петя записывает на бумажку то число, на котором стоит Сашина фишка (если Саша не брал ранее ни одного числа, то он может сказать «беру» на любом числе). Игра заканчивается, когда Сашина фишка больше не может ходить, и количество очков, которые набрал Саша, зависит от того, сколько чисел он взял. Помогите Саше «взять» как можно больше чисел!

Формат входных данных

Входной файл состоит из одного или нескольких наборов входных данных (не более 5). Набор входных данных начинается строкой, в которой записаны числа m и n ($1 \leq m, n \leq 100$). Далее следуют n строк по m чисел в каждой — таблица, которую нарисовал Петя. Все числа не превосходят по модулю 10000.

Файл завершается набором с $m = n = 0$.

Формат выходных данных

Для каждого из наборов входных данных выведите в первой строке максимальное количество чисел M , которые может взять Саша. Во второй строке выведите один из возможных вариантов Сашиних действий — строку из символов R, D и T, где R означает ход вправо ($x \leftarrow x + 1$), D обозначает ход вниз ($y \leftarrow y + 1$) и T обозначает взятие числа, на котором в данный момент стоит фишка.

Разделяйте вывод для разных наборов входных данных одной пустой строкой.

Пример

take.in	take.out
2 2	1
1 1	TRD
1 1	
2 3	3
1 2	TRTDDT
1 1	
1 3	
0 0	

Задача 5B. Наибольшая общая возрастающая [0.4 sec, 256 mb]

Даны две последовательности чисел — a и b . Нужно найти наибольшую общую возрастающую подпоследовательность. Более формально: такие $1 \leq i_1 < i_2 < \dots < i_k \leq a.n$ и $1 \leq j_1 < j_2 < \dots < j_k \leq b.n$, что $\forall t : a_{i_t} = b_{j_t}, a_{i_t} < a_{i_{t+1}}$ и k максимально.

Формат входных данных

На первой строке целые числа n и m от 1 до 3 000 — длины последовательностей. Вторая строка содержит n целых чисел, задающих первую последовательность. Третья строка содержит m целых чисел, задающих вторую последовательность. Все элементы последовательностей — целые неотрицательные числа, не превосходящие 10^9 .

Формат выходных данных

Выведите одно целое число — длину наибольшей общей возрастающей подпоследовательности.

Система оценки

Подзадача 1 (50 баллов) $n \leq 400$.

Подзадача 2 (50 баллов) $n \leq 3\,000$.

Пример

lcis.in	lcis.out
6 5 1 2 1 2 1 3 2 1 3 2 1	2

Задача 5С. Новая модель телефона [0.2 сек, 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

Замечание

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 $x = 4$.