

Содержание

Задача А. Сумма двух [1 sec, 256 mb]	3
Задача В. Длинный наибольший общий делитель [1 sec, 256 mb]	4
Задача С. Калькулятор [1 sec, 256 mb]	5
Задача D. Отрезки [1 sec, 256 mb]	6
Задача Е. Ancestor. Предок [1 sec, 256 mb]	7

Примеры работы с файлами и решения первой задачи.

Решение #0 sum/sum0

Решение #1 sum/sum1

Решение #2 sum/sum2

Ещё куча полезных примеров.

База: if, for, while if-for-while

Функции function

Одномерные и двумерные массивы arrays

Ввод/вывод in-out

Переопределение операторов, классы operator

Работа со строками strings

Задача А. Сумма двух [1 cec, 256 mb]

Формат входных данных

В первой строке входного файла расположены два целых числа A и B , не превосходящих 1 000 по модулю.

Формат выходных данных

Ваша программа должна выдавать в выходной файл одно число — сумму чисел A и B .

Примеры

sum.in	sum.out
2 3	5
17 -18	-1

Задача В. Длинный наибольший общий делитель [1 sec, 256 mb]

Даны два целых положительных числа. Найти их наибольший общий делитель.

Формат входных данных

Мультитест. Каждый тест задаётся двумя строками. Суммарная длина чисел до 50 000.

Формат выходных данных

Для каждого теста выведите одну строку – наибольший общий делитель.

Примеры

agcd.in	agcd.out
10	5
15	262144
10000000000000000000	1
1152921504606846976	
17	
100	

Подсказка по решению

Решение за $\mathcal{O}\left(\frac{n^2}{k^2}\right)$ зайдёт по времени. Система счисления равна 10^k .

Задача С. Калькулятор [1 cec, 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 D. Отрезки [1 sec, 256 mb]

Даны отрезки на прямой. Какое максимальное количество отрезков можно выбрать так, чтобы никакие два из них не пересекались? Отрезки считаются открытыми.

Формат входных данных

В первой строке входного файла задано целое число n ($1 \leq n \leq 100\,000$). В следующих n строках описаны отрезки; i -я из этих строк содержит два целых числа l_i и r_i через пробел — координаты начала и конца отрезка ($1 \leq l_i < r_i \leq 10^9$).

Формат выходных данных

В первой строке выходного файла выведите одно число — максимальное количество выбранных отрезков.

Примеры

segments.in	segments.out
2 1 2 3 5	2
2 1 3 2 6	1
3 2 4 3 5 1 3	2

Задача Е. Ancestor. Предок [1 sec, 256 mb]

Напишите программу, которая для двух вершин дерева определяет, является ли одна из них предком другой.

Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 100\,000$) — количество вершин в дереве. Во второй строке находится n чисел. При этом i -ое число второй строки определяет непосредственного родителя вершины с номером i . Если номер родителя равен нулю, то вершина является корнем дерева.

В третьей строке находится число m ($1 \leq m \leq 100\,000$) — количество запросов. Каждая из следующих m строк содержит два различных числа a и b ($1 \leq a, b \leq n$).

Формат выходных данных

Для каждого из m запросов выведите на отдельной строке число 1, если вершина a является одним из предков вершины b , и 0 в противном случае.

Пример

ancestor.in	ancestor.out
6	0
0 1 1 2 3 3	1
5	1
4 1	0
1 4	0
3 6	
2 6	
6 5	